ML

An Educational Revolution

Group 16

May 2024

Afonso Cadete | 20211519

Bruna Faria | 20211529

Catarina Oliveira | 20211616

Joana Rosa | 20211516

Martim Serra | 20211543

NOVA Information Management School

Instituto Superior de Estatística e Gestão de Informação

Universidade Nova de Lisboa

ABSTRACT

The project aims to explore the intersection of education and machine learning to develop a predictive model to forecast student performance based on first-year data, in addition to information from their university application. To achieve this goal, several preprocessing techniques were applied, especially in the categorical variables, as they had a large number of unique values; hence, reclassification and binarization were used to mitigate high-dimensionality problems. One-hot encoding was then applied to the reclassified features. These steps ensured correct interpretation of the results of the models, avoiding misconceptions of ordinality, and facilitating an effective analysis. In the modelling phase, a total of 14 scikit-learn classification models were tuned using Grid/Bayes search algorithms to determine the best performing model. Finally, two chosen models were achieved, a Multi-Layer Perceptron (MLP) and a Logistic Regression (LR). The latter model provided a better understanding of the 24 selected features, but with a comparatively lower performance than the best performing model - the MLP. The performance of the models was assessed using k-fold (where k = 10) cross-validation, with the MLP achieving an average weighted F1-score of 0.767 across the validation folds, while the LR had an average weighted F1-score of 0.761. With the information acquired during this project some policies and strategies were also recommended according to the relevance of the selected features to predict academic success. Although conducted as part of a Kaggle competition where access to the real target values of the test set was restricted, this report highlights the robustness of the MLP model and its potential applicability in real-world scenarios.

KEYWORDS

Machine Learning; Predictive Modelling; Educational Data Mining; Multi-Layer Perceptron; Logistic Regression

INTRODUCTION

Portuguese universities have long been suffering difficulties with chronic underfunding, which has led to high dropout rates and low student satisfaction. Therefore, over time more and more Portuguese high-school students have been on the lookout to acquire quality higher education overseas.

However, admission processes at foreign universities, such as Ivy League institutions are highly competitive and tend to have strict entrance standards that do not only depend on the student's grades. Thus, the motivation behind this project lies on the clear need for the automatization of this selection process in order to be able to find the most promising candidates.

Previous studies in this area of educational data mining revealed that it is an effective tool to explore more in depth the relationships between educational data and academic success [2]. Furthermore, learning analytics have also proven to be useful for improved decision making in admission processes to determine which students might be underperforming, or falling behind their colleagues, and in need of personalized learning plans, by ranking students by their level of risk [2].

In summary, the main goal of this project was to develop a machine-learning model that would be able to accurately foresee if a student would or not be successful at an Ivy League university based on the information from their application along with data from their freshman year. Moreover, the project was also part of a Kaggle competition where the team did not have access to the real value of the target of the test set; however, this characteristic will not be delved into this report.

DATA EXPLORATION

Description of the Data

In order to develop the project, one dataset regarding students' demographic and academic data was analysed. This dataset counted with 6072 students, divided into 4744 rows on the training set, 1328 on the test set, and 37 variables that are presented on table 8 of the appendix. The target variable was considered to be success and was represented by whether the student gave up, was holding on (if it is enrolled after the normal duration of the course) or succeeded.

Exploratory Data Analysis

Since this dataset counted on such a great number of variables, it was important to check whether all of them presented meaningful information or if there were duplicated rows, multiple missing values in the same row or column that could limit compatibility with algorithms. 1645 Rows were found to be duplicated on the training set and were consequently dropped.

The decision was made to remove the variable *Observations*, since it was fully composed of missing values. The next column with the highest number of missing values was *Mother's qualification*, with less than 3% of observations having missing data, which was deemed manageable. As for the rows, 3 were found to be entirely filled with missing values except for the target variable and so were excluded from the analysis. The next row with the biggest amount of missing values counted with only 7 values out of 36 (already excluding *Observations*), which was found to be admissible. Plus, variables that had only two unique values were converted into Boolean datatype to be treated as categorical furtherly in the analysis.

Inconsistencies

Another important thing to consider were inconsistencies. One that was found was that 14 observations had "evening attendance" added on the *Course* variable but the morning shift participation was considered to be "True". To solve this inconsistency, *Morning shift participation* was considered as a missing value and imputed later. To better understand the following inconsistencies, it is crucial to define how some of the variables were interpreted, namely *N units credited*, *N units taken*, *N units approved*, *N units scored* and *unscored*, which are described on table1.

Variable	Interpretation
N units taken	Units that the student was enrolled in. Include both the approved and the credit units.
N units approved	Must include the ones that were credited since no one can be credited for an unapproved unit.
N units credited	Units approved on another course that were found to be equivalent on the current course.
N scored units	Total evaluations in the semester.
N unscored units	Units that weren't scored/success or failure wasn't evaluated.

Table 1 - Dataset Variable Explanation

After interpreting each of the variables, it was confirmed that if there were 0 units taken, then all the other units-related columns also had a 0 value. In addition, one checked that there were not more units credited or approved than taken in the same semester. When the number of units approved was 0, it was rightfully verified that the average grade in the same period was also 0. It was found that 7 students had more units credited than approved on the same period, which didn't make sense since one wouldn't be credited for an unapproved unit. Therefore, the decision was made to consider both values as missing values. Another found inconsistency was that when the number of units taken was bigger than 0, scored and unscored units couldn't both be 0, since both variables contradicted each other. So, in the 263 cases where this happened, *N units taken* was considered to be 0, in order to maintain consistency of the pattern mentioned in the beginning of the paragraph.

Distributions of the variables were studied both before and after the procedures done so far and weren't found to impact visualizations. Nevertheless, the most interesting insights are described in the following paragraphs.

Target Variable

The distribution of the target variable consisted of the majority of the students having succeeded, followed by the ones who gave up and finally the ones holding on.

Categorical Variables

For categorical variables, the elected visualization tool was stacked bar charts (which also consider the distribution of the target variable).

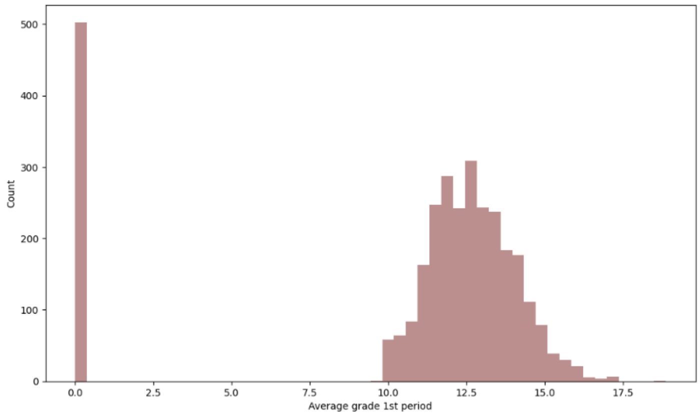
The variable *registered* was excluded since it only had 1 unique value, which means that it would not impact the target variable at all.

In general, what is considered as standard was not only the majority but also followed the pattern mentioned on the target variable. One might consider the standard university student as someone young (around 20s), single, with the previous qualification being the 12th grade, Portuguese (since this study occurred in Portuguese universities), assigned to the morning shift, not having special needs, debts, or fees to regularize and most do not hold a scholarship as well. Data proved this stereotype to be accurate, since most of the students followed these conditions while also following the *target* variable general pattern (e.g. students were mostly single and their value in *target* variable was 'succeeded'). The student parents' most common *qualifications* were 12th, 4th, 9th, 6th, and bachelor's degree. Both the mother's and father's most common *occupations* were unskilled work, administrative staff, engineer, private security, scientist, and superior-level professional. However, fathers differed greatly in areas, such as: engineering, armed forces and assembly works, where the numbers increased significantly when compared to the mothers. For both the *qualification* and *occupation*, the *target* variable followed the same general trend. In terms of *courses*, nursing had the highest enrolment, while Biofuel Production Technologies had the lowest. In fact, nursing students also presented the biggest rate of "success" out of all courses. As health-related courses tend to be attended mostly by females, there were more females in this study than males, and accordingly to what was stated before they presented a higher rate of success than males (who more often tended to give up). Holding a scholarship (*external funding*), even in minority, seemed to also have a positive effect on success rates. On the other side, having *debts* or not *regularized fees* seemed to negatively impact success rates, since these students tended to give up more often. Some (binary) variables didn't seem to affect success rates such as *morning shift*, *displacement, international* and *special needs*, due to the fact that the distribution of the target was the same for both values.

Numerical Variables

The numerical variables' distribution was accessed through histograms and bar charts. Most variables presented a right-skewed distribution, meaning smaller values were generally more common such as in *age* (where students tend to be young), *application order* and *mode*, or *number of units taken/approved/credited*. For *social popularity* the distribution was fairly even among the values. *Previous qualifications* and *entry score* both presented normal distributions. As for *average grades in both 1st* and *2nd periods* their distributions were also rather normal. Despite this, there was still a large number of students with an average grade of 0 (figure 1). These might be students who gave up on that period and such results could later affect the use of linear models, since these can be quite sensitive to extreme values.

Figure 1 – Distribution of Average grade 1st period



Correlations

One of the found correlations was between *application mode* and *age* - when the *application mode* was 0 (meaning the student entered via special contingent) the *age* distribution was broader with the median being closer to 30 than 20, which makes sense considering that this includes students over 23 years old and diploma holder contingencies. On the rest of *application mode* categories (1st, 2nd, and 3rd phases), *age* was quite centred between 18 to 23, given the fact that this is the most common way the majority of the students enrols in university. Another positive correlation that was noticed was between *previous qualification score* and *entry score*. As one might guess, a good student in high school will likely continue to be a good student at university.

Some other variables that showed positive correlations were the ones about units, especially when they held the same goal but differed on 1st or 2nd period. *N units taken* were also positively correlated with *N units approved*, and the latter presented a positive correlation with *Average grade* as well.

METHODOLOGY

In an initial stage of the project, there was a heavy stage of data exploration – that contained verification of duplicates, inconsistencies, missing values, and distributions – and which was already mentioned in greater detail in the above section.

Following the same pattern of thought as before, in the next few paragraphs all the steps applied in the project and the reasoning behind the taken decision will be explained. In addition, please take into consideration that the steps mentioned in the subsequent sub-sections were first applied to the train data and then replicated for the test sets.

Feature Engineering

Data Cleaning and Normalization

Regarding the preprocessing of data, some minor alterations had to be made to ensure the consistency of the data. The first change that was applied was the transformation of both semester's *average grades* to a scale from 0 to 200, since previously there were some values in another range.

Next, the *Course* variable suffered three distinct changes. Firstly, the phrase "evening attendance" was removed from all observations of this column, since it only represents redundant information considering the existence of the variable *Morning Shift Participation*. Secondly, one of the course names was changed to "Echinculture" instead of the original "Equinculture", since the package that translated the course names from Portuguese to English used that form of writing. Finally, the values for the *Course* feature were capitalized, to comply with the aesthetic already present in the dataset.

Feature Creation

In the context of the problem at hand, the team believed it made sense to create some new features that would capture useful information, based on the available data, and improve the final predictions.

Hence, the two features created concerned comparisons of the student's average grade with the average grade of the courses. The first one – *course application mode entry score difference* – represents the difference between the entry score of the student and the mean of the course entry score depending on the application mode. The second and third ones – *Course average grade 1st period difference; Course average grade 2nd period difference* - is slightly similar; however, it differs on the fact that it is the difference between the average grade and the mean of the course average grades for both periods (1st and 2nd).

Categorical Features Reclassification & Binarization

The original dataset of this project had a lot of categorical variables, each with many unique values. Therefore, it was crucial for the team to focus on how to deal with this type of feature – reduction of unique values of categorical variables.

On this note, and after the analysis of the distributions of variables (which was explained in depth in the Data Exploration section), such as *Marital Status* and *Nationality,* it was clear and as expected that there were dominant categories (Single, Portuguese). In this sense, the binarization of these variables was implemented (True: dominant categories; False: otherwise).

Afterwards, the *Course* feature was transformed into *Course Area* column through two main steps: the first was web scraping of the possible university areas in Portugal (taken from the official DGES website); while the second consisted of the mapping of the original list of 15 courses to the 10 corresponding areas. Although this change does not represent a huge difference in absolute terms, it is still a one-third category values reduction, which already has some impact on a computational level. In addition, since the DGES website was in Portuguese it was necessary to translate the information to English, and the Python library used was connected to Google Translator API (*googletrans*).

Finally for the *Previous*, *Mother's and Father's qualification* variables, with respectively, 10, 17 and 17 categories, two different approaches were implemented. The first one relied on the categorical reclassification of these variables into 4 ordinal values (0: no education; 1: basic education; 2: intermediate education; 3: advanced education). Meanwhile the second one, was a numerical classification where the number of years of studies of each student were returned – this considering that the pupils had a normal academic career, with no failures and also completed higher education courses in the average time stipulated (0: no education; 3-12: 3rd to 12th grade; 13: incomplete Bachelor's degree; 15: Bachelor's degree; 16: Post-graduate degree; 17: Master's degree; 21: PhD).

Additionally, a new Boolean variable for technological courses was created from *Previous Qualification* (True if student's parents had a technological course, False otherwise).

Encoding

The next phase of the project, consisted of the encoding of the remaining object variables (since all the other features had already been transformed into Boolean types) – in this case, the *Course* feature with one-hot encoder. Though, several other options of encoding were initially considered to deal with the categorical variables, the reasoning behind the use or non-use of each of these techniques is briefly explained in table 2.

Encoders	Used?	Observations
Label	No	It does not have a straightforward manner to deal with unseen data.
Ordinal	No	Applying it a-priori of data partition would generate data leakage. However, cross-validation was demanded in this problem, so its code application would be much more complex.
Frequency	No	Suffers from the same problem of the Ordinal Encoding.
One-Hot	Yes	Using it with the initial column would end in the curse of dimensionality problem (more than 150 columns) but after the categorical features binarization/reclassification it became the most appropriate technique to be used.
Target	No	Since the project had a multiclass problem, there was no benchmark value for unseen categories that didn't bias the model as 0.5 for binary cases. Furthermore, more complex approaches available in Category Encoders library were excluded for the same reason (e.g.: M-estimate, James-Stein).

Table 2 - Encoders Explanation

Scalers

Feature scaling is a crucial step when it comes to preparing data for machine-learning models, aiming to ensure that features are on the same scale and thus preventing certain variables from dominating due to their larger magnitudes. (table 3)

Scalers	Used?	Observations
Standard	No	Given some features don't follow an approximately normal distribution.
MinMax	Yes	Although it doesn't deal well with outliers, it is very recommended when the features have pre-established boundaries.
Robust	Yes	Very good to deal with outliers.
Power (Transformer)	Yes	It aims to reshape the data into a distribution that resembles a Gaussian distribution. The method used was the Yeo-Johnson since there were negative values in some features.

Table 3 - Scalers Explanation

Feature Selection

Finding the most relevant information from a wide range of available features is crucial to potentially improve the performance of the models and produce better forecasts. Therefore, taking this into consideration, feature selection techniques were applied to the dataset's features.

Since throughout the whole project cross-validation was implemented, the beginning of this stage was of no exception and a stratified k-fold was employed (k=10), to then be able to build a more generalized model, given that several seeds produce more trustworthy results than just one. Next, for each subset, the data was scaled, and the missing values were imputed with a K-Nearest Neighbours imputer to avoid the problem of data leakage. The exact same process was also applied in the modelling phase.

As for the selection of categorical variables, a Chi-Square test for independence was applied. Most of the previous categorical variables were transformed into Boolean values; hence, they can be utilized as numeric. For that reason, the methods used for numeric features – Recursive Feature Elimination, Sequential Feature Selection, Extra Trees Classifier, and Lasso Regression – were also applied to them. Additionally, consider that all the feature selection procedures were given the same level of importance.

In general, the feature selection process has a more analytical side, which can generate subjectivity and work overload for the teams. As a means of simplifying this process, in this project a more mathematical approach was chosen – through the count of how many times each of the columns was selected by the methods (i.e. for a feature to be selected it should have been accepted by, at least, three of the techniques). Since the Boolean features were also accessed by the Chi-Square test of independence, those which were considered important by it and two other methods were also selected. Due to the fact that there were 10 folds in the cross-validation, 10 distinct feature combinations were obtained, which were then evaluated, by a logistic regression model, in subsets considering comparisons between the features frequency from 10 down to 6 (10: only features that were selected 10 times; 6: features that were selected 6, 7, 8, 9, and 10 times).

As mentioned above, for a variable to be selected it had to be accepted by at least 3 of the feature selection methods, since this was the threshold that prioritized performance. From now on this approach is mentioned as “more than 2”, for simplicity reasons. However, for future and more complex work purposes and to assess the trade-off between performance and complexity the team also experimented with a threshold of 4 (the variable had to be accepted by all the used techniques) – from now on mentioned as “more than 3”. To mimic a real-life scenario, data was duplicated multiple times as an attempt to achieve 154 800 observations which is the number of students that entered university in Portugal, in 2023.

Resampling

According to Elor and Averbuch-Elor (2022), when using classifiers such as MLP, SVM, AdaBoost among others, augmenting the data with SMOTE-like techniques might be useful to improve performance. Therefore, oversampling with SMOTE was applied to the data and performance was tested.

Additionally, to balance the data undersampling was also considered. However, due to the lack of training data compared to the test data (due to the fact that the ratio had already reached the limit of what was recommended) this was not possible to be experimented with.

Model Selection

Similarly to the other phases, in the modelling step several approaches, namely Scikit-Learn library models were tested. In total, fourteen different models were experimented, ten of them without predetermined estimators - Gaussian Naïve Bayes, Logistic Regression, Decision-Tree Classifier, K-Nearest Neighbours, Multi-Layer Perceptron (MLP), Support Vector Machine, Linear Discriminant Analysis, Stochastic Gradient Descent, Random Forest, and Gradient Boosting Classifier - and four of them with predetermined estimators - Stacking, Voting, Bagging, and AdaBoost Classifiers.

As a first step, to find the best hyperparameters for subsequent usage in the models, Grid Search was ran for the ten models without predetermined estimators. Nevertheless, since the development of machine-learning applications is governed by an iterative process, this approach was protracted for the slight changes in the models with a bigger search space. In addition, it ended up leading us to much the same conclusions, and therefore at a later stage, Bayes Search was applied for every model except Gaussian Naïve Bayes, Logistic Regression and Linear Discriminant Analysis. Both the Grid and Bayes Search were applied with cross-validation and formatted to consider as the final output the model with best performance, that did not exceed a difference of 5% between train and validation sets.

As a result, the models that performed the best were combined with the most adequate scalers for each of them to then develop the optimal ten models; consequently, they were used as base estimators to build the four ensemble methods. For Stacking and Voting Classifiers, Grid Search and two groups of estimators were used: one group with the ten base models as estimators, and the other with just the best four out of those. As a result, the best model for each ensemble method was extracted. In terms of the Bagging and AdaBoost Classifiers, after careful testing it was determined that the running time and the complexity associated with these models did not justify their performance results. Instead, a simple application with the ten algorithms as estimators was applied, and the best model for each one of the two ensemble methods was found.

In the end, the final fourteen best models were compared between each other, and the best among them was chosen through performance, overfitting, and execution time. Due to interpretability needs and in the context of this problem, two final models were considered, one which focused primarily on performance, and another with more focus on interpretability.

Model Evaluation

Once the model selection process is over, the performance of the models was evaluated. In this stage, for simplicity in the evaluation of the models, a data partition seed resulting from the holdout method - which was within the standards of the cross-validation results - was used. The team's focus relied on achieving the most interpretable model to fully comprehend the obtained predictions. Furthermore, in this step, the metrics of evaluation – F1-score, precision, and recall – of the different categories were obtained to better understand the results. After this analysis, a confusion matrix was generated to evaluate if the difficulties in classifying the classes were consistent on both training and validation sets; thus, having one more guarantee about overfitting control. Additionally, through the confusion matrix and with the help of a two-dimensional UMAP (dimensionality reduction technique), an error analysis was carried out to understand which were the misclassified observations.

RESULTS

Even though several experiments were performed, presenting every result in intricate detail would be risking overwhelming readers with too much information. Therefore, the baseline for the comparisons in this section is the data preparation pipeline that produced the best model.

As mentioned before, feature selection was achieved by performing a stratified k-folds (k=10) with the previously stated methods. The results of the most representative (most similar with the final decision) iteration are presented on table 8. The summarized results of all iterations are presented on table 4; note that the features which had a frequency of 8 or more iterations ended up being the selected ones.

Frequency	Features
10	Debtor, Morning shift participation, Average grade 1st period, N units approved 1st period, N units approved 2nd period, Age at enrollment, External Funding, Gender_Male, Regularized Fees, N units taken 2nd period, Nationality, Father's qualification, Course average grade 1st period difference, Course area_Teacher training/trainers and education sciences, Course area_Engineering and related techniques, Course area_Health, Course area_Social services, N scored units 2nd period
9	Course average grade 2nd period difference, Entry score, Social Popularity, Marital status
8	Course area_Business sciences, N units taken 1st period
7	Average grade 2nd period, N scored units 1st period
6	Mother's occupation, N unscored units 2nd period
5	International, Father's occupation, Course area_Veterinary sciences
4	Course area_Personal services, Course application mode entry score difference, Application mode
3	Mother's qualification
2	Course area_Art, N unscored units 1st period, Application order, Course area_Information and journalism
1	N units credited 1st period, Technological course
0	N units credited 2nd period, Special needs, Displaced, Previous qualification score, Previous qualification

Table 4 – Features' Importance

Two approaches to deal with the *Qualification* variables (parents' and the individuals') were tried on. However, since these variables had a low impact on the performance (according to feature selection) it was decided to proceed with the one that gathered the best result which was the second one as explained on the methodology.

Having the optimal selection of features, several models were experimented on, and their respective results can be seen on table 5. The scaling method was individually tested and chosen for each model according to the best performance achieved.

	Time	Train	Validation
Stacking (with TOP4) RS	7.128+/-0.12	0.797+/-0.01	0.768+/-0.02
Multi-Layer Perceptron RS	**0.395+/-0.0**	**0.782+/-0.0**	**0.767+/-0.02**
Voting (with TOP4) RS	1.223+/-0.47	0.787+/-0.0	0.767+/-0.02
Bagging (with MLP) RS	0.814+/-0.34	0.784+/-0.0	0.764+/-0.02
Logistic Regression RS	**0.472+/-0.16**	**0.763+/-0.0**	**0.761+/-0.02**
Gradient Boosting RS	0.776+/-0.0	0.818+/-0.0	0.761+/-0.02
Support Vector Machine RS	0.156+/-0.01	0.767+/-0.0	0.76+/-0.02
AdaBoost (with LogR) RS	0.8+/-0.02	0.754+/-0.0	0.749+/-0.02
Stochastic Gradient Descent RS	0.073+/-0.0	0.752+/-0.0	0.747+/-0.01
Linear Discriminant Analysis MS	0.013+/-0.0	0.751+/-0.0	0.745+/-0.02
Decision Tree MS	0.011+/-0.0	0.799+/-0.01	0.733+/-0.02
Random Forest PT	0.109+/-0.0	0.78+/-0.0	0.732+/-0.02
K-Neighbours PT	0.002+/-0.0	0.781+/-0.0	0.731+/-0.02
Naive Bayes PT	0.003+/-0.0	0.679+/-0.0	0.677+/-0.03

Table 5: Weighted F1-scores for each classifier. RS: Robust Scaler, MS: MinMax Scaler, PT: Power Transformer

Taking into consideration execution time, overfitting and performance, the chosen model was the MLP. As for the model which showed increased interpretability associated with better performance, the one that stood out was the Multinomial Logistic Regression.

Some other experiments were made using two different feature selections approaches: “more than 2” (24 features) and “more than 3” (11 features). As expected, the one that used more variables took slightly more time to complete (around 0.045 seconds); however, its performance was also better (0.741 for more than 3 vs 0.767 for more than 2). As for the attempt done using 154 800 observations the difference in time was way more outstanding, totalling to around 1 minute between more than 2 or more than 3 approaches.

Another experiment regarding feature selection was based on the number of times the variable was selected on the iterations of the stratified k-folds. Based on performance, the ideal number of times the feature should appear to be selected was considered to be 8.

Another comparison performed was with an oversampled dataset. The results of this dataset, using MLP, did not show any advantage compared with the unbalanced dataset, having even worst performance and more time of execution. To reach a clearer vision of how well the best model predicted each class, a classification report was presented and is visible on table 6, this report shows results using the logistic regression model. However, MLP results were quite similar. According to this table, the class “Succeeded” was the one with the highest F1-score, meaning that it is the one that is better predicted by the model. By observing precision and recall, one concludes that recall has a higher value which means that the model is better at predicting students that succeeded than the ones who didn’t (more false positives – see confusion matrix, figure 2). As for the class “holding on”, the opposite happened: besides being the one with the lowest F1-score, it also has a better precision than recall, meaning the model predicted better the students who are not holding on than the ones who are (more false negatives). Finally for the class “gave up”, precision and recall were fairly similar.

	Precision	Recall	F1-score	Support
Gave up	0.80	0.75	0.77	696
Holding on	0.54	0.37	0.44	389
Succeeded	0.81	0.93	0.86	1082

	Precision	Recall	F1-score	Support
Gave up	0.79	0.80	0.79	298
Holding on	0.51	0.32	0.39	167
Succeeded	0.82	0.93	0.87	464

Table 6 - Classification Report for training set (left) and validation set (right)

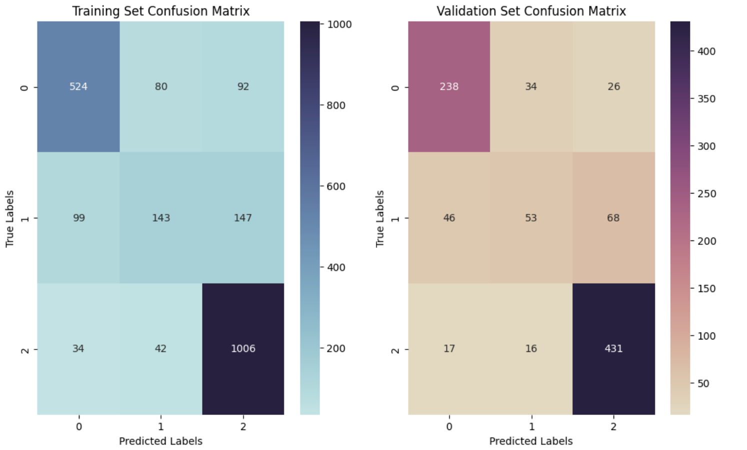


Figure 2 – Confusion Matrix of Logistic Regression

Regarding the results of the best interpretable model – the logistic regression – it is important to notice which are the features with the largest coefficients - most relevant to predict the academic success of the students. The results regarding these coefficients are presented on table 7.

	Gave up	Holding on	Succeeded
N units approved 2nd period	-1.663436	-0.712897	2.376333
Regularized Fees	-1.547987	0.238528	1.309459
N units approved 1st period	-0.383272	-0.775076	1.158348
External Funding	-0.204672	-0.324209	0.528881
Course area_Social services	-0.452900	-0.012488	0.465388
Course average grade 2nd period difference	-0.305932	0.017645	0.288287
Morning shift participation	-0.261108	0.081484	0.179624
Average grade 1st period	-0.020777	-0.088310	0.109087
Entry score	-0.011550	-0.011335	0.022884
Social Popularity	0.119013	-0.130295	0.011281
Course area_Health	-0.065472	0.065504	-0.000033
Gender_Male	0.179546	-0.122654	-0.056893
Course average grade 1st period difference	-0.042978	0.108661	-0.065683
Marital status	0.375024	-0.288263	-0.086761
Age at enrollment	0.274762	-0.155799	-0.118963
Father's qualification	0.021630	0.152938	-0.174569
Course area_Business sciences	0.055973	0.221932	-0.277906
N scored units 2nd period	-0.050883	0.329520	-0.278637
Nationality	0.830442	-0.402266	-0.428176
Debtor	0.357741	0.121935	-0.479676
N units taken 1st period	0.296860	0.225507	-0.522367
N units taken 2nd period	0.564724	0.383851	-0.948575
Course area_Teacher training/trainers and education sciences	0.603494	0.365428	-0.968922
Course area_Engineering and related techniques	0.826576	0.437137	-1.263713

Table 7 – Logistic Regression's feature coefficients

DISCUSSION

In line with what was previously mentioned the team reached the conclusion that having two final models for two distinct goals – performance or complexity – was what made the most sense in the context of the problem.

In terms of performance there were three main models with similar values in the validation set – MLP, Stacking and Voting Classifiers. In the end, the MLP was the chosen one, since in comparison with the others it showed less overfitting and is much simpler; therefore, taking less time to run. Nevertheless, when it comes to better interpretability, a logistic regression revealed itself to be the model with the best performance. It's also important to have in mind that multinomial logistic regressions still hold some assumptions. That said, the applied multinomial logistic regression counted on ridge regularization to try to mitigate multicollinearity [15] and achieved a low standard deviation (0.02), meaning the coefficients are relatively stable. In future research, it would be interesting to build a pipeline to deal with data focusing more on respecting the assumptions of a logistic regression, since in this project the focus was to implement different models and therefore the data pipeline ended up being broader.

Shifting the focus to the complexity of the chosen models, an important comparison to make is about the tradeoff between a model that takes 24 columns and another with only 11 features. In this case, the model with the higher number of columns also shows an improved performance by 2.5 percent in the validation set and just an additional 3 hundredths of a second per iteration when compared to the other model. In order to further explore the possible scalability of the more complex model, a real context was simulated with much more observations, which in the end proved to only add a minute to the running time. Hence, it is proven that, for this purpose, the performance and complexity tradeoff is worthy.

By looking at the 24 variables and their respective coefficients in the logistic regression, presented on table 7, some insights might be retrieved. The variables that have shown to have more weight on the prediction were *N units approved on the 2nd period* followed by *regularized fees*. These two variables show biggest coefficients on the class "succeeded" meaning that, when the number of units approved is bigger, or the fees are regularized (value 1), the probability (log-odds) of belonging to the class "succeeded" is higher. On the opposite side, the probability of belonging to the class "give up" is very low (high negative number). It's also interesting to notice that regarding the units approved, the ones on the 2nd semester seem to have more impact on the target variable than the ones approved on the 1st semester. It also seemed that students on engineering courses are less likely to succeed since the coefficient is much lower for this class, compared to the "gave up" one. Some other relevant features are *external funding*, students that have scholarships seem to be more likely to succeed, and *n units taken* on 2nd and 1st period (again, 2nd being more relevant than 1st) in the more units taken the less likely is the student to be succeeded.

After careful examination of all the decisions made throughout the project, some limitations in this work were identified. Among them, the possible binning of the numeric variable *Average Grades* to avoid the problem of having too many observations where the students had 0 as their evaluation, since this can badly affect some linear-based models. Following the same line of thought, it would be interesting to try the same approach without the class *"holding on"* since it revealed to be a mixture of the other two classes of the target. Furthermore, the optimization of the K-Nearest Neighbours Imputer through the tuning of its parameters instead of using default values or testing other methods could be relevant for further research. Another possible suggestion to improve the project would be to give different weights/importance to each of the feature selection techniques rather than simply calculating the frequency of their results. Finally, experimenting with models outside of the scikit-learn library and correcting the mistakes found during the error analysis might make a difference to improve the final performance of the models.

CONCLUSION

In this project, extensive experiments, with multiple models allied to different feature selections and a thorough data preprocessing, were evaluated in order to develop a machine-learning model that might be helpful to accurately predict the academic success of students. Two of the models stood out among others, MLP and Logistic Regression, each of them presenting its benefits and considerations. MLP was the one with the highest weighted F1-score (0.767) among all and is the definite choice if one is looking to achieve optimal performance. Nevertheless, its interpretability is questionable, and it might be hard to understand how much each variable weighs on the prediction. For this matter, then the Logistic Regression seems to be the best choice, since among the most interpretable models, it was the one which delivered the best performance (0.761). Besides choosing the best model for each goal, it is also quite important to choose the features included in them. Thus, the combination of the 24 features previously mentioned was crucial to achieve a reliable validation score. Therefore, this project was successful in its ability to predict the academic success of students using machine-learning techniques including models in the scikit-learn library.

Some policies and strategic procedures might also be inferred through the information extracted from this project. For instance, having debts or fees not regularized was associated with less successful students while having a scholarship did seem to positively influence success. An interesting policy to increase the success of the students would be to offer more scholarships, especially if they were related to high grades and a fair number of approved units. Moreover, since courses such as engineering were related with more withdrawals, it would be noteworthy to check what type of teaching methodologies were being applied on equally demanding courses, such as health related ones where students are more likely to succeed and adopt some of their strategies. Regarding the number of *units taken*, it can be said that the more taken units the more likely the student is to give up. To fight this, it could be helpful to be enrolled in less units per semester, by discarding less important subjects and focusing on the most essential ones to the course; therefore, crediting them more heavily and deepening the knowledge provided to the student.

For future research it would be interesting to use more state-of-the-art classifiers such as XGBoost and Catboost, using the present project as a benchmark for performance. Another interesting research would be finding if this pipeline could help predicting other related questions, such as if the student continued or not for a master's degree, or if he started to work on the semester after ending the course.

REFERENCES

[1] Yağcı, M. (2022). *Educational data mining: prediction of students' academic performance using machine learning algorithms*. Smart Learning Environments. https://doi.org/10.1186/s40561-022-00192-z

[2] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot,

M., Duchesnay, E., Louppe, G. (2012). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research.* pp. 2825-2830.

[3] Baby D., Devaraj S., Hemanth, J., Anishin Raj M. (2021) Leukocyte classification based on feature selection using extra trees classifier: A transfer learning approach. *Turkish Journal of Electrical Engineering and Computer Sciences* 29 2742-2757

[4] Lanjewar, M.G., Parab, J., Shaikh, A.Y., Sequeira, M. *(2022).* CNN with machine learning approaches using ExtraTreesClassifier and MRMR feature selection techniques to detect liver diseases on cloud. *Cluster Comput* doi.org/10.1007/s10586-022-03752-7

[5] L. (2023, October 2). *Portugal com o maior número de sempre de alunos do ensino superior*. PÚBLICO. https://www.publico.pt/2023/10/02/sociedade/noticia/portugal-maior-numero-alunos-ensino-superior-2065301

[6] Amorim, L., Cavalcanti,G., Cruz, R. (2023). The choice of scaling technique matters for classification performance. *Applied Soft Computing Volume 133*

[7] Elor, Y., Averbuch-Elor, H. (2022). To SMOTE, or not to SMOTE?. https://arxiv.org/pdf/2201.08528

[8] *1.13. Feature selection*. (n.d.). Scikit-learn. https://scikit-learn.org/stable/modules/feature_selection.html#recursive-feature-elimination

[9] *1.11. Ensembles: Gradient boosting, random forests, bagging, voting, stacking*. (n.d.). Scikit-learn. https://scikit-learn.org/stable/modules/ensemble.html

[10] *UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction — umap 0.5 documentation*. (n.d.). https://umap-learn.readthedocs.io/en/latest/

[11] *1.1.3. Linear Models*. Lasso (n.d.). Scikit-learn. https://scikit-learn.org/stable/modules/linear_model.html#lasso

[12] .13. Feature selection. (n.d.). Scikit-learn. https://scikit-learn.org/stable/modules/feature_selection.html#sequential-feature-selection

[13] 1.2 Linear and Quadratic Discriminant Analysis. (n.d.). Scikit-learn. https://scikit-learn.org/stable/modules/lda_qda.html

[14] 1.5 Stochastic Gradient Descent. (n.d.). Scikit-learn. https://scikit-learn.org/stable/modules/sgd.html#classification

[15] Gao, S., Shen, J. (2007). Asymptotic properties of a double penalized maximum likelihood estimator in logistic regression. *Statistics & Probability Letters*, 77(9), 925–930. https://doi.org/10.1016/j.spl.2007.01.004

APPENDIX

	RFE	Sequential	Tree-based	Lasso	DECISION
Course area_Art	1	0	0	1	0 (-)
Course area_Business sciences	1	0	0	1	1 (+)
Course area_Engineering and related techniques	1	1	0	1	1 (+)
Course area_Health	1	0	0	1	1 (+)
Course area_Information and journalism	1	0	0	0	0 (+)
Course area_Personal services	1	0	0	0	0 (+)
Course area_Social services	1	1	0	1	1 (+)
Course area_Teacher training/trainers and education sciences	1	1	0	1	1 (+)
Course area_Veterinary sciences	1	0	0	1	0 (-)
Marital status	1	0	0	1	1
Application mode	1	1	1	0	1*
Application order	1	1	1	0	1*
Morning shift participation	1	0	0	1	1
Previous qualification	1	1	0	0	0
Previous qualification score	1	0	1	0	0
Nationality	1	1	0	1	1
Mother's qualification	1	0	1	0	0
Father's qualification	1	1	1	1	1
Mother's occupation	1	0	0	1	0
Father's occupation	1	1	0	1	1*
Entry score	1	1	1	0	1
Displaced	1	0	0	0	0
Special needs	1	0	0	0	0
Debtor	1	1	1	1	1
Regularized Fees	1	1	1	1	1
Gender_Male	1	1	1	1	1
External Funding	1	1	1	1	1
Age at enrollment	1	1	1	1	1
International	1	1	0	0	0
N units credited 1st period	1	0	0	1	0
N units taken 1st period	1	1	1	0	1
N scored units 1st period	0	0	1	1	0
N units approved 1st period	1	1	1	1	1
Average grade 1st period	1	0	1	1	1
N unscored units 1st period	1	0	0	1	0
N units credited 2nd period	1	0	0	1	0
N units taken 2nd period	1	1	1	0	1
N scored units 2nd period	1	1	1	1	1
N units approved 2nd period	1	1	1	1	1
Average grade 2nd period	1	0	1	0	0
N unscored units 2nd period	1	0	0	1	0
Social Popularity	1	0	1	1	1
Course application mode entry score difference	1	1	1	0	1*
Course average grade 1st period difference	1	0	1	1	1
Course average grade 2nd period difference	1	1	1	1	1
Technological course	1	0	0	1	0
TOTAL	45	23	23	31	28

*Table 8: Results of the most representative iteration of the stratified k-folds. (-): categorical variable not selected by chi-square, (+): categorical variable selected by chi-square; * variable was selected on this specific iteration but didn't meet the requirements defined to be selected in the overall group of features used.*

ANNEX

BACKGROUND

- ***One-Hot Encoder:*** used to represent categorical variables as numeric values. This technique creates a binary vector where each of the categories has a unique index and only one of the values in the vector can have a value of 1 (present). It can be relevant above other encoding methods since it ensures that the model takes care of each category independently with no order between them.
- ***Extra-Trees Classifier:*** similarly to random forests it selects a random subset of candidate variables. However, for this model the threshold is randomly obtained for each feature and the one which performs the best is the chosen to be the splitting rule. There is a chance that it can slightly increase the bias, but it reduces the variance of the model, increasing accuracy and decreasing the execution time. [2,3,4]
- ***UMAP:*** short for Uniform Manifold Approximation and Projection, it is a dimension reduction technique that can visualize high-dimensional data. Unlike traditional dimension reduction techniques, UMAP is based on a graph theory that attempts to keep the local structure of the data while reducing its dimensionality. [10]
- ***Lasso Regression:*** stands for Least Absolute Shrinkage and Selection Operator and it is a type of Linear Regression that is able to minimize the sum of squared errors between observed and predicted values and also add a penalty that shrinks the regression coefficients towards 0; hence, removing them from the mode. It is useful as a feature selection method that can deal with multicollinearity in large datasets. [11]
- ***Recursive Feature Elimination****:* method for feature selection that recursively eliminates the least important features until the desired number of variables is selected. It works by repeatedly fitting the model and then removing least significant columns; therefore, improving model efficiency and its interpretability. [8]
- ***Forward - Sequential Feature Selection:*** technique that iteratively finds the best new feature to add to the set of selected variables to include in the model. This selection process ends whenever the desired number of features is achieved. [12]
- ***Stacking Classifier:*** this is an ensemble method that combines multiple classifiers by training a meta-classifier on their predictions. It tends to have a higher predictive performance by taking advantage of the different strengths of each model. [9]
- ***Bagging Classifier:*** ensemble method that trains multiple instances of a base classifier on random subsets of the training data and then combines their predictions, by averaging or voting, to reduce the overfitting and the variance of the model. It might be advantageous for unstable models.[9]
- ***Voting Classifier:*** using a majority vote (hard vote) or the average projected probability (soft vote), it is an ensemble method that combines conceptually distinct machine learning classifiers to predict class labels. It is useful to counterbalance the weaknesses of each of a set of similarly performing models. [9]
- ***Gradient Boosting:*** ensemble method that uses a classifier algorithm (in the context of this project) and develops an additive model step-by-step. It enables the optimization of any differentiable loss function. [9]

- ***AdaBoost:*** the fundamental idea behind this ensemble method is to fit a sequence of weak learners (*models that are only slightly better than random guessing*) on recurrently modified versions of the data. The final prediction is then generated by adding all their predictions together using a weighted majority vote (or sum). [9]
- ***Linear Discriminant Analysis:*** a traditional classifier with a linear decision region, produced by fitting class conditional densities to the data and the use of Bayes' rule. This classifier has closed-form solutions that can be easily computed, is inherently multiclass, have proven to work well in practice, and have no hyperparameters to tunning. [13]
- ***Stochastic Gradient Descent Classifier:*** linear classification algorithm that iteratively updates the model parameters, in a direction that minimizes the error, by looking at one training example at a time; therefore, making it efficient for large datasets. [14]

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação

Universidade Nova de Lisboa