Predict students' dropout and academic success

Brigida Santarpia, Luca Barbato, Luigi Emanuele Zippo

Corso di Algoritmi e Applicazioni per la Data Science
A.A 2022-23

1 Introduction

In this paper, data from Portalegre Polytechnic University (IPP), Portugal, are analyzed and used to build machine learning classification models. In a first step, data identification is performed, followed by a careful description and exploration of the data in order to best understand and frame them. Next, data analytics activities are presented, with a special focus on the data visualization phase. The main objective is to provide a system to identify, at a very early stage, students with potential difficulties in their academic journey, so that student support strategies can be put in place.
The dataset includes information known at the time of student enrolment: educational background, demographic data and socio-economic factors. The problem is formulated as a classification task in three categories, where there is a strong bias towards one of the classes. Classification models are trained and evaluated, both with standard machine learning algorithms and with boosting algorithms. Our results show that the boosting algorithms respond better to the specific classification task than standard methods. However, even these boosting algorithms fail to correctly identify most cases in one of the minority classes.

2 Explorative Data Analysis

In this section, we present the dataset, the methods used to deal with the unbalanced nature of the data and the methodology used to construct and

evaluate the classification models.

2.1 Dataset Description

In this study we use institutional data (acquired from different disjointed databases) on students enrolled in degree courses at the Polytechnic University of Portalegre, Portugal. The data refer to the records of students enrolled between the academic years 2008/09-2018/2019 and from different degree courses, such as agronomy, design, pedagogy, nursing, journalism, management, social services and technology.

As just noted, these are structured data and contain variables relating to demographic factors (age of enrolment, gender, marital status, nationality, address code, special needs) socio-economic factors (student-worker, parents' qualifications, parents' occupations, parents' employment status, scholarship, student debt) and student's educational background (admission grade, years in high school, order of choice by course of enrolment, type of course in high school).

The features in the dataset are mainly categorical but, at the time of publication, this has already undergone a label-encoding process, which transforms categorical variables into integer variables without adding new features (unlike one-hot encoding). Likewise, any anomalies and missing data have already been dealt with, as the dataset has no missing data, duplicates and outliers.

Each record has been classified as 'Dropout', 'Enrolled' and 'Graduate', which refers to the student's status at the end of the normal term.

- "Graduate" means that the student has graduated within the terms of the course of study;
- "Enrolled" means that the student, at the end of the normal course term, is still enrolled;
- "Dropout" means that the student has abandoned the course of study.

The distribution of records among the three categories is unbalanced, with two minority classes, 'Dropout' and 'Enrolled'. 'Dropout' accounts for 28% of the total records and 'Enrolled' accounts for 16% of the total records, while the majority class, 'Graduate', accounts for 56% of the records. The classes we most want to correctly identify are the minority classes, as students

in these classes are the ones who could benefit from planned support and educational guidance.

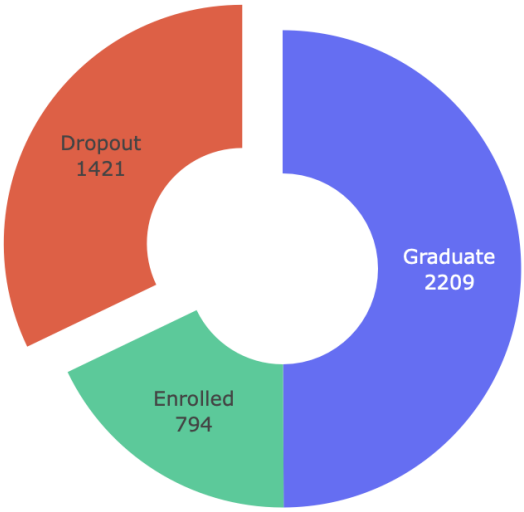

Figure 1: Pie chart with distribution of records according to 'Target'.

2.2 Data Visualization

At this stage we use various techniques to represent the data in a graphical form and highlight interesting relationships between them

In the following histograms, we represent on the x-axis the age of the students at the time of enrolment and on the y-axis the respective number of occurrences.

We first observe that the age ranges from 17 to 70 years with a normal distribution with a maximum around 20 years. We can see this from the KDE line (Kernel Density Estimate) which approximates the distribution of the data. We wanted to analyse the distribution of records according to age not only over the whole dataset, but also between "Dropout", "Enrolled", "Graduate". We then compared the distribution densities in Figure 3.

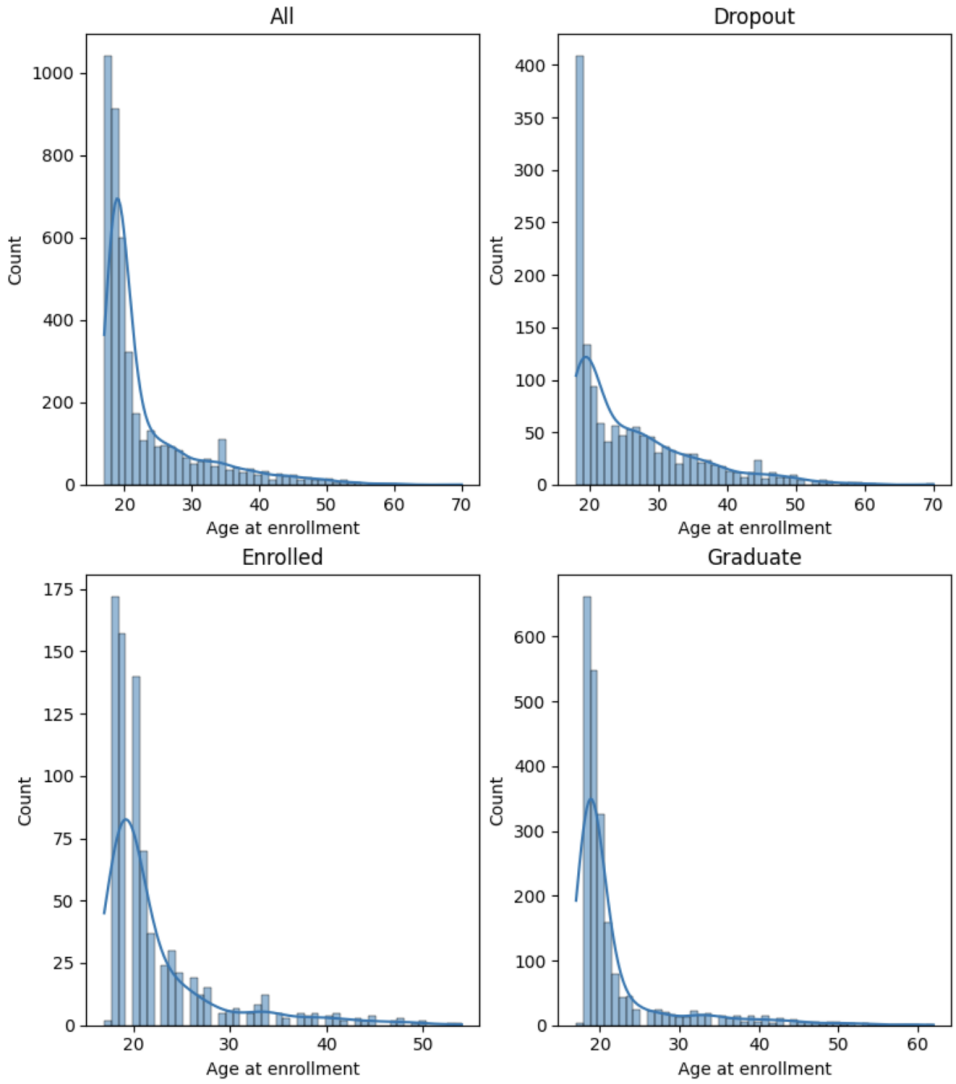

Figure 2: Histograms of age at enrolment

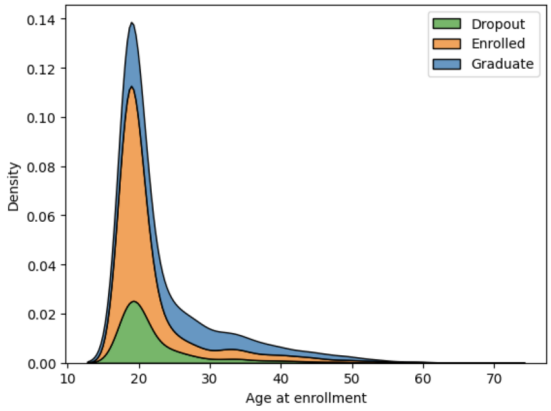

Figure 3: Density of age distribution as a function of 'Target'.

As we can see from Figure 3, the maxima of the distribution density accumulate around the age of 20 in all three cases

Next, we represented the number of students according to gender, noting that between the 'Dropout' and 'Enrolled' students there are no particular differences, while among the 'Graduate' students there is a clear disproportion: female students are about three times as numerous as male students.

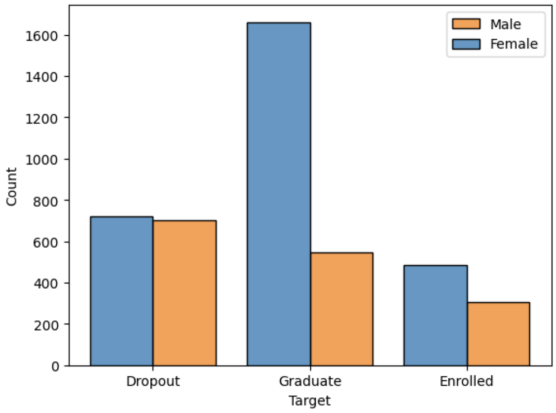

Figure 4: Barplot with gender distinction

Using the same type of graph, we instead analysed the number of students with debts. We have noticed that the number of students with debts is significantly lower than the non-debtor ones, moreover is particularly concentrated among "Enrolled" and "Dropout" students. Actually, among 'Graduate'

students we have about 4 % debtors, while among 'Enrolled' and 'Dropout' students the debtors are 11 % and 22 % respectively.

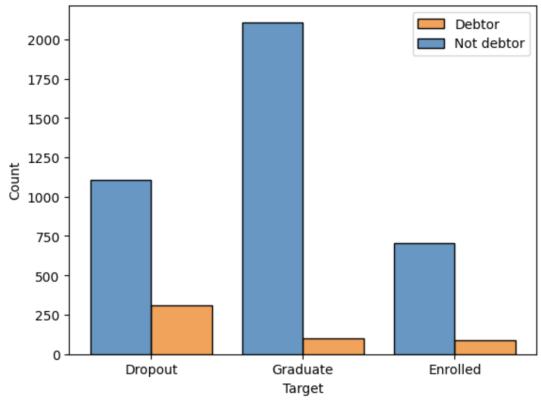

Figure 5: Barplot with debtors distinction

Next, we represented the university admission grade as a function of the grade of the previous qualification. A linear trend is evident but no particular distribution of scholarship students and drop-outs can be detected.

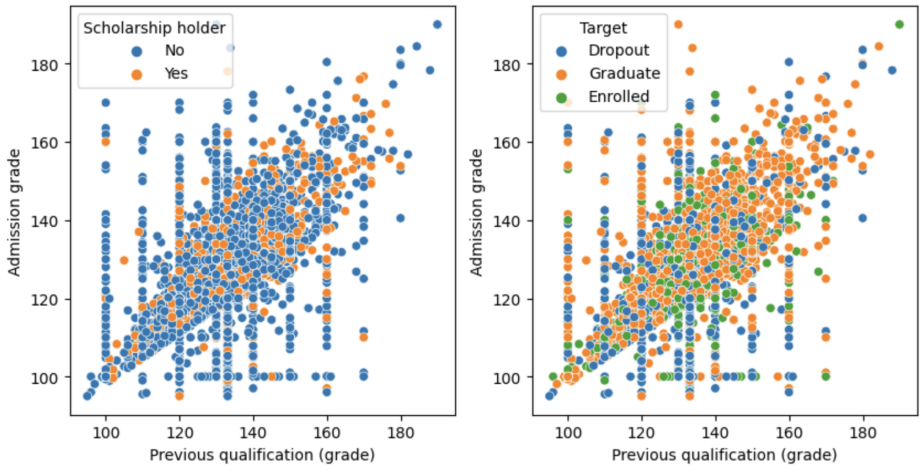

Figure 6: Scatterplot with 'Scholarship holder' and 'Target'

In the process of data visualisation, we also helped ourselves with PCA (Principal Component Analysis): this uses linear algebra tools to identify the

directions that maximise variance. By projecting along the first two principal components, it is possible to represent the data in a Cartesian plane. These new components are linearly dependent on the previous features but do not always have real meaning

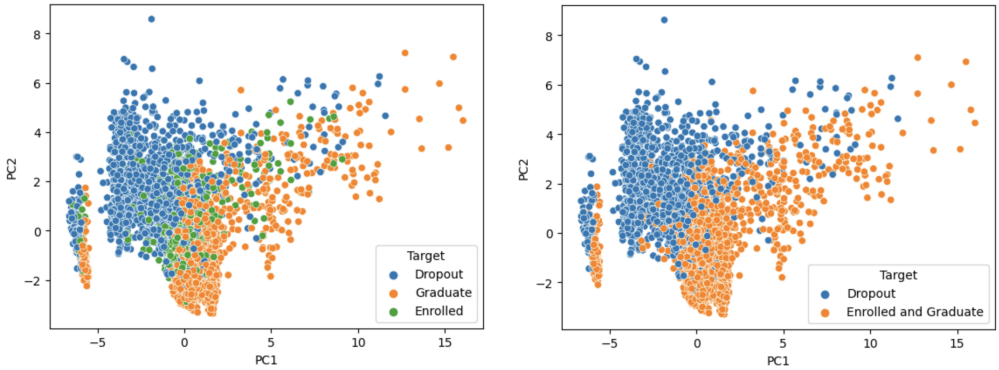

Figure 7: PCA with multiclass and binary 'Target'

We can distinguish two main clusters of 'Dropout' and 'Graduate' students, while 'Enrolled' students are not clearly separated from the classes just mentioned. So, considering the purpose of our analysis, we thought it was appropriate to reformulate the problem on the basis of this result. We decided to switch from a multi-class classification problem to a binary classification problem by focusing our attention on the "Dropout" students. Consequently, for the remainder of the analysis, we combined the "Graduate" and "Enrolled" classes.

3 Model

3.1 Feature Selection

In order to get better prediction and to improve training times we implemented a feature selection process. Having mostly categorical features, we chose to use the χ^2 rather than the Pearson correlation. The χ^2 is used in statistics to test the independence of two events; in our case we aim to select the features that are strongly dependent on the variable 'Target'. By calculating the various scores, we have produced the following graph.

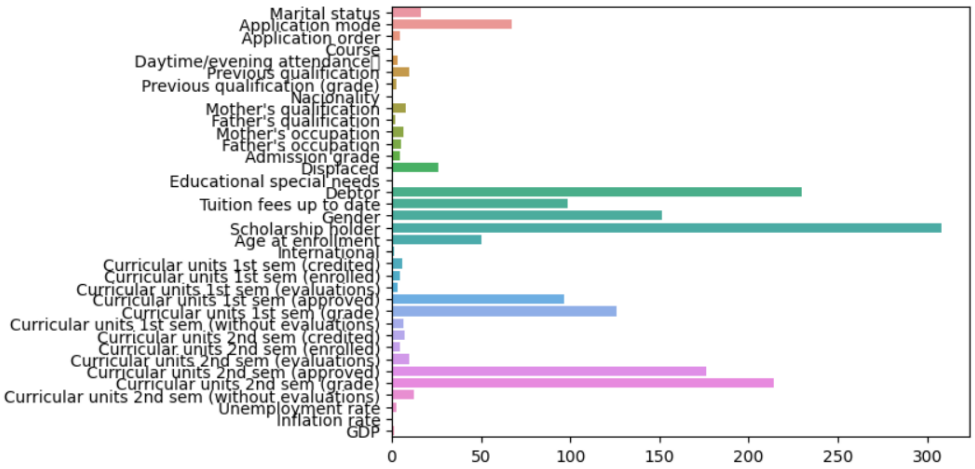

Figure 8: χ^2 scores for each feature

Therefore the most useful and selected feature are:

'Application mode'	'Debtor'
'Tuition fees up to date'	'Gender'
'Scholarship holder'	'Age at enrollment'
'Curricular units 1st sem (approved)'	'Curricular units 1st sem (grade)'
'Curricular units 2nd sem (approved)'	'Curricular units 2nd sem (grade)'

3.2 Selection of the Best Model

First, we split the dataset into a test set and a training set (20% and 80% of the original dataset, respectively) respecting the class distribution of the 'Target'. We then scaled the data with the StandardScaler, which subtracts the mean from each value and divides by the standard deviation, as many Machine Learning algorithms perform better on data with a normal distribution. We have now tested four different Supervised Learning models in order to evaluate which one performs best on the dataset:

- **Decision Tree**: is an algorithm constructed by "asking a series of questions" about a feature in the dataset and each time an answer is received the process is iterated until a label is assigned to each record. At each node, the decision tree searches through the features for the

value to split on that results in the greatest reduction in Gini Impurity; the series of questions and their possible answers can be organised in the form of a decision tree, a hierarchical structure made up of oriented nodes and edges

- **K-Nearest Neighbours**: assigns labels by calculating the K nearest points in a given metric under the assumption that similar objects are close to each other
- **Random Forest**: is a union of Decision Trees that assign labels indepedently; then, by majority vote, the results are combined by choosing the label with the most votes for each record
- **XGBoost**: trains a series of Decision Trees sequentially, i.e. each model 'learns' from the mistakes made by the previous model

We tested the four models both with and without Cross Validation: this technique consists of dividing the training set into k disjoint subsets, where $k-1$ are used to train the models while the last one is used to test it. We decided to set the value at $k = 10$. To evaluate the performance of the models, we chose the metric 'F1':

$$F1 = 2\frac{precision * recall}{precision + recall}$$

This is particularly useful in our case given the uneven distribution of the 'Target'. It combines both precision and accuracy of the model: maximising the F1 metric therefore means minimising false negatives; since a false negative would lead to a waste of economics, and of resources in general, which could be useful to the academic career of other at-risk students, we considered the F1 metric the most suitable for our purpose. Below are the results obtained:

	Without CV	With CV
Decision Tree	68.45 %	67.77 %
Random Forest	75.41%	74.85%
K-Nearest Neighbour	71.66%	73.26 %
XGBoost	75.58%	73.26%

As can be seen from the results, the optimal model is XGBoost. The performance of Random Forest is particularly close to that of the chosen model,

but even if it had been a little higher, we would have preferred XGBoost as it is much less computationally expensive.

3.3 Hyperparameters Tuning

To improve the performance of the chosen model, we used the Grid Search technique, which tests the model over and over again by varying the hyperparameters in order to evaluate which combination is optimal.

Parameters	**Range of Optimality**
'min_child_weight':	[0.5, 1, 1.5, 2, 5]
'gamma':	[1, 1.5, 1.7, 1.75, 1.8, 2, 3]
'subsample':	[0.7, 0.8, 0.9, 0.95, 1.0]
'colsample_bytree':	[0.7, 0.8, 0.9, 1.0]
'max_depth':	[2, 3, 4, 5]
'learning_rate':	[0.1, 0.01]

We chose these ranges experimentally: starting with the standard parameters of XGBoost, we launched several GridSearches, thickening the chosen parameters in the neighbourhoods of the best parameters.

Parameters	**Best Values**
'colsample_bytree'	0.9
'gamma'	1.7
'learning_rate'	0.1
'max_depth'	3
'min_child_weight'	0.5
'subsample'	0.8

3.4 Fitting

To prevent our model from overfitting or underfitting, we used several techniques. First of all, by already eliminating features that were less correlated with the 'Target', besides gaining an advantage from a computational point of view, we significantly reduced the risk of overfitting. Then, with parameter tuning, we made the model of the right complexity, avoiding overly simple models that would have led to underfitting. Finally, to check how effective

these techniques were, we ran the model again, with the optimal parameters, on both the train set and the test set. Then, again using the F1 metric, we compared the errors in the two runs.

Accuracy on Training Set	78.58%
Accuracy on Test Set	77.29%

It can be seen that the two errors are very similar, indicating that the model is neither too simple nor too complex.

4 Conclusion

4.1 Model Interpretation

To interpret the model, we calculated the *feature_importances_* and obtained the following results:

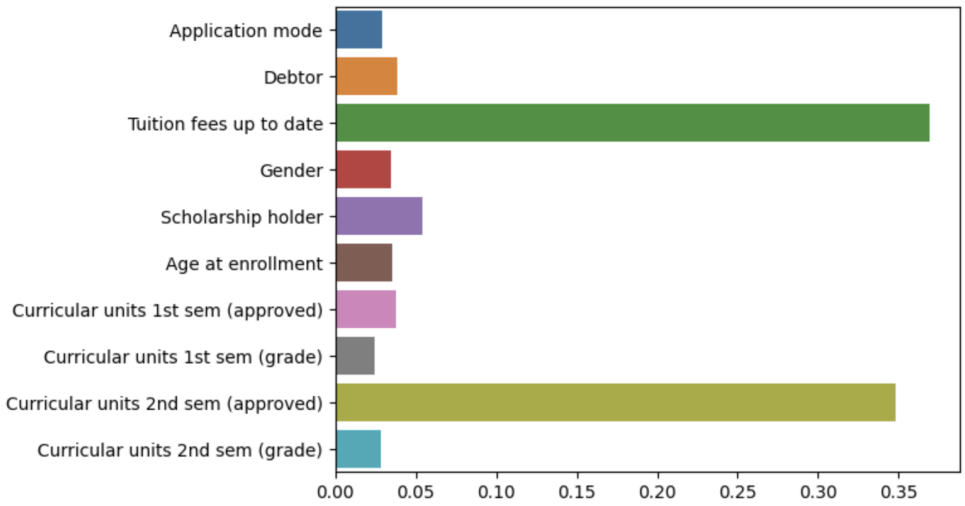

Figure 9: Feature importance for prediction

As we can see, the feature that most influences the XGBoost result is 'Tuition fees up to date', immediately followed by 'Curricular units 2nd sem(approved)'. Thus, we can see how both economic and academic factors

influence the student's career. This is not the case for most social factors, like parents' qualification and occupation, marital status, displaced status and nationality. In fact, among the ten features we selected, we find 5 features related to the student's academic career, 3 related to individual economic status and lastly age and gender. It is interesting to note that the overall economic conditions (unemployment rate, GDP of the country, inflation rate) do not significantly influence academic dropout as opposed to the personal one (tuition fees, scholarship, debts). These results are very encouraging for our task since we do not use data concerning performance after graduation or dropout (for example job held or income received), so we can predict at an early stage whether a student is at risk or not.

4.2 Future implementations

Lastly, the model can be improved in order to achieve better performance, for example, with better search for optimal hyperparameters combined with *data augmentation* techniques to balance the classes. Another way to proceed could also be the search for meaningful features using other feature selection techniques or implementing other models such as Deep Learning ones.