ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

SECOND CYCLE DEGREE IN COMPUTER SCIENCE

AI Course Project

Group name : SmashAI

Student: **Gabriele SPINA**
Badge number: **0001038853**
E-mail: **gabriele.spina2@studio.unibo.it**

ACADEMIC YEAR 2021 – 2022

Contents

1 Introduction

Every year there is plenty of students enrolling to several university faculties. Some of them will get good results, others give up their studies for various reasons. This second group of people, even if it is intuitively less numerous than the first one, deserves attention. That is because University dropout is the worst event that could happen in the life of a student; it also negatively affects the University itself, mainly from an economic point of view, but not only. It would be helpful to understand the main reason for their choices, for instance on the one hand to improve the services offered by the University itself where it seems there may be gaps, on the other hand to or even to intervene, in some way, in the University career of a student who is presumed to be about to drop out. Some ideas could be:*(i)* to organize specific intensive courses for students who are having particular problems *(ii)* to intervene from a psychological point of view to help these students. About the second point, it is not to be overlooked that someone may drop out of University for personal problems, whether they regard anxiety and similar or private or others.

Ideally, every University would like to prevent enrolled students from abandoning their courses of study before graduating. This is the reason why it would be interesting to find some kind of solution to the problem. An idea is to have some intelligent technique that is able to prevent, in terms of predictions in time, such dropouts; having a technique like that would be the key to intervene before the problem became more concrete. This report discusses a first possible solution to the problem of University dropouts. In particular, through the use of artificial intelligence techniques. This can be done with the courtesy of the University of Bologna, who provided the the data concerning the students enrolled at the University during the years $2016 - 2018$. The data in question have been analyzed and fed into machine learning models to be able to get a first idea about the students whose careers seem to be taking a bad turn somehow. The idea would be that by possessing this information there would be the possibility of intervene in time on the careers of these students (e.g., organizing specific intensive courses) in order to be able to prevent, at least some of them, from dropping out of University.

This work illustrates the first steps in the construction of such an intelligent system. The features (i.e., the various information related to each student) are analyzed to retrieve the ones that are relevant to understand the causes of the dropouts. Furthermore, the choice of the method used is justified, even from a theoretical point of view, not only regarding the kind of data used, but also considering various alternatives, among those that have proved suitable, in order to be able to compare them and opt for the one that proves to be one of the most suitable choices. Finally, the results are illustrated and discussed, observing how the problem can actually be

Feature	Possible values
Gender	1,2
Age range	1,2,3
High school ID	1 to 10
High school grade	60 to 100
Additional learning requirements	1,2,3
University course scope	1 to 16
Mean of exams grade	18 to 30
Total of CFUs	3 to 120
Dropout	0,1

Table 1: List of most interesting features

tackled with the proposed technologies, obtaining results that encourage a possible continuation of the work and a possible real application.

2 Proposed Method

2.1 Data Preparation

The data are organized in many Excel sheets (`.xlsx`), which are pseudo-anonymized data describing 67280 students enrolled in different courses (both bachelor's, master's, and unique cycle's degrees) during the years 2016 − 2018. The data are composed of some features that can be considered static since that is the information about the students at the moment of subscription at the University; then there are the data relating to the student's actual University career, i.e., dynamic, which report information about the exams taken during the student career and information regarding the additional learning requirements. In Table 1 are listed the features judged as most informative for the prediction of the dropout: on the right the feature names, on the left the possible values that the feature could assume.

In order to obtain the features as described in the tables, the data has been manipulated to get the ranges indicated. Some features have been also discarded, so they are not listed in the table; that is because they are not relevant to predict the dropout problem (e.g., the campus where courses are taken, the geographical origin of the student, and similar). Here it is provided a highlight of all of the features, giving a little explanation about how the original data has been modified in some cases. The *Gender* feature can assume binary value, that is male or female. The age of the students in the dataset was reported as birth date, so it has been modified to obtain the age at the moment of subscription, then split in 3 possible values, the ones representing the feature *Age range*; i.e., 1 for students with less than 22 year, 2 for the ones between 22 and 25 year, 3 for the remaining ones. The *High school ID* represents the different schools from where the students come; *High school grade* stands for the mark of graduation in high school. The feature *Additional learning requirements* was originally represented by 2 different features, one indicating the fact that a student got the requirements and the other indicating if they were successfully passed. That was clearly a bad representation for a learning model, so it has been filtered to obtain 3 possible values; i.e., 1 for students who has no need to do the additional requirements exams, 2 for the ones that have to do the exams, but failed them, 3 for the ones that passed. The *University course scope* represents the various possible field of the courses; the choice of this feature, despite the courses id, has been made because it is more concise (there are fewer alternatives). The *Mean of exams grade* is the mean of all the exams performed by the student, and the *Total of CFUs* is the sum of all the credits related to those exams. At last there is the *Dropout*, the features that will be the class to predict.

2.2 Problems Definition

Various formulations of the problem have been made to establish the features that carry more information about the reason why the dropout happens, and find the best combination. It has been observed that the major percentage of students abandoning the career are from bachelor's degrees, following up the ones from unique cycle degrees, and last the master's ones. More in specific a percentage of 11.2% students drop out from the bachelor, 9.03% from the unique cycle, and 3.79% from the master. Therefore, one could expect that predictions in the case of bachelor's degrees can be more effective, as the data is too unbalanced regarding the dropouts and the learning models can benefit from a major number of examples where the dropout happens. So four different formulation of the problem of the prediction of dropout has been analyzed:

1. All the type of degrees
2. Only bachelor's degrees
3. Only unique cycle's degrees
4. Only master's degrees (in that case the division made for the feature *Age range* has been modified as the students subscribing to master's degrees are older)

Moreover, to understand which ones are the features that bring more benefit to the prediction, so, it can be said, the main reasons behind the dropouts, in a first analysis it has been selected only the static ones (that is, all except the *Mean of exams grade* and the *Total of CFUs*), without the *Additional learning requirements*. The *Additional learning requirements* has been excluded because one would expect that the students that mustn't pass the exams related got fewer possibility to drop out, as the ones that must pass the exams but didn't get higher. Therefore, it is interesting to observe if effectively that information improves the quality of the predictions. Then, also the dynamic data has been added and the results from the various dataset obtained were compared; this analysis has been made only on problem 1 and the best formulation is used for all the problems.

Moreover, in [1] it is highlighted how the major number of dropouts happens in the first year of the student career, so they referred only to the first year. Therefore, also in this work the experiments has been carried out with only the data coming from the first year of the University, because it is the most critical phase in the career of the students, so the best moment to make an eventual intervention to prevent dropout.

2.3 Models Adopted

The problem analyzed fell in the supervised learning field, as the overcome of every instance is explicated in the data. The classification algorithms used to make the predictions of the dropouts has been chosen, with respect to other ones, even for choices made in [1] but also because most of the features are categorical and some algorithm (as decision trees) is proved to be more adaptable with this kind of data. It has been considered the K-Nearest Neighbors (`KNN`) [2], Decision Tree (`DT`) and Random Forest (`RF`) [3].

`KNN` is a simple machine learning algorithm used in a variety of applications, it uses the concept of proximity to classify a data point. The base is that similar points usually are found near to each other, k is the number of near points to consider for each class. `DT` is a tree-like structure in which the nodes represents a condition/test on a feature and the branch the outcome of the test, the leaves are the class predicted. `RF` is nothing more than a collection of decision trees classifiers combined randomly. Some little test on the parameters of the algorithms (e.g., k for `KNN`, number of trees for `RF`) has been made by doing some tests interactively (directly on terminal), showing that the standard configurations are a proper choice, even considering the computational costs when the parameters become higher.

3 Experimental Results

The experiments have been performed using the Python programming language (version `3.9.5`) and the `scikit-learn` framework [4] (version `1.1.1`). The phases of train and test were run on a Linux Workstation equipped with i7 4-Core 4 Ghz processor and 16 GB of memory.

After applying the filters to obtain the features as described in the previous Section, each dataset must be re-sampled in the right way to do the learning statistically correct. That means that, first of all, the intersection between the train and the test dataset must be empty. It is also necessary, then, that the two classes are balanced, that is, the number of instances classified as 0 are the same number of those classified as 1. To adjust the class distribution of the dataset, it has been used a simple technique of random undersampling for all the experiments; since the dataset is too unbalanced, using oversampling is not the best choice. The undersampling consists in removing samples from the majority class; but it is very likely to discard useful or important samples. The split in train and test sets was made in a proportion of, respectively, 80% and 20%. The operations of balancing and splitting the datasets were carried out with 10 different random seeds (not for the feature analysis part) with the aim of ensuring that the choice of the instances of the resulting datasets is not affected a lot, mainly, by the undersample but also by the split, that is, not only the ”worst” instances are considered.

To determine the effectiveness of the learning, there are several evaluation metrics that could be observed. The choice of the metrics felt on accuracy (ACC), specificity (SPEC), and sensitivity (SENS). The ACC is the ratio of number of correct predictions to the total number of input samples, i.e., $Accuracy = \frac{TruePositive+TrueNegative}{TotalSample}$. The SPEC is the true negative rate, the ratio of the true negatives to the total number of instances with negative class, i.e., $Specificity = \frac{TrueNegative}{TrueNegative+FalsePositive}$. The SENS is the true positive rate, the ratio of true positives to the total number of instances with positive class, i.e., $Sensitivity = \frac{TruePositive}{TruePositive+FalseNegative}$.

3.1 Features Analysis

In Table 2 are illustrated the results from the different combinations of features; with the standard parameters, $k = 5$ for KNN and $nTrees = 100$ for RF. Taking into account only the static features the KNN and the DT give more or less the same results for all the metrics and higher than RF; instead for SENS where the RF gives a best result. Adding the *Additional learning requirements* the results are slightly better confirming what already said in Subsection 2.2, however still not enough to consider the predictions accurate. Regarding the last combination, i.e., when the dynamic features are added, there is the real gain for all the algorithms analyzed, mainly RF seems to get the best results. Anyway, the dynamic features are not available at the moment of subscription, so to get good results one should wait for a student to start taking exams.

Features selected	Learning model	Accuracy	Sensitivity	Specificity
Static	KNN	0.57	0.55	0.57
	DT	0.56	0.57	0.56
	RF	0.52	0.65	0.51
Static + ALR	KNN	0.56	0.59	0.56
	DT	0.58	0.63	0.57
	RF	0.57	0.65	0.56
All	KNN	0.80	0.87	0.79
	DT	0.78	0.75	0.78
	RF	0.82	0.88	0.82

Table 2: Results of the features analysis (ALR stands for *Additional learning requirements*)

3.2 Experiments on Different Degrees

Now the results for the 4 problems defined in Subsection 2.2 are exposed, taking into account only the best features configuration. In Tables 3, 4, 6, 5 are shown the results, in order, for problem $1, 2, 3, 4$ carried out with different random seeds, and the mean values. Differently from what was expected there are no major differences between the various kind of degrees. Anyway, a little increase on SENS is obtained in case of bachelor and unique cycle's degrees; it is an important result as this metric deals with the positive class, i.e., the dropout. Regarding the methodology, the RF exhibits the best results almost for all the formulations.

4 Discussion and Conclusions

We have argued throughout this work some several ways to try to predict the dropout problem. Different machine learning techniques have been applied to different kind of degree courses, i.e., bachelor, master, and unique cycle. The feature analysis part in Subsection 3.1 showed how the prediction is too much difficult at the moment of subscription due to the lack of information about the students' careers. Considering also the exams taken by a student during the first year of University, the results are instead quite good. The best results are obtained with bachelor's and unique cycle degrees, as expected, but the differences between the two remaining problem formulations (all the degrees and only master's one) are few.

The results can be considered satisfying as an accuracy of the 80% is reached in almost all the cases where the exams are considered, the same is for specificity. While with regard to sensitivity, mainly for bachelor's and unique cycle degrees, the results show a percentage of more than the 90%; that is the most interesting result as this evaluation metric deals with the prediction of just the dropouts.

Random Seed	Learning Model	Accuracy	Sensitivity	Specificity
0	KNN	0.80	0.85	0.80
	DT	0.81	0.70	0.82
	RF	0.83	0.84	0.83
1	KNN	0.79	0.90	0.79
	DT	0.78	0.82	0.78
	RF	0.81	0.91	0.81
2	KNN	0.80	0.84	0.80
	DT	0.78	0.78	0.78
	RF	0.83	0.85	0.83
3	KNN	0.80	0.81	0.80
	DT	0.76	0.74	0.76
	RF	0.83	0.87	0.83
4	KNN	0.81	0.84	0.81
	DT	0.78	0.78	0.78
	RF	0.83	0.84	0.83
5	KNN	0.80	0.89	0.80
	DT	0.77	0.86	0.77
	RF	0.81	0.86	0.81
6	KNN	0.78	0.84	0.78
	DT	0.77	0.77	0.77
	RF	0.81	0.89	0.81
7	KNN	0.79	0.85	0.78
	DT	0.76	0.78	0.77
	RF	0.80	0.85	0.80
8	KNN	0.81	0.87	0.81
	DT	0.76	0.83	0.76
	RF	0.83	0.87	0.83
9	KNN	0.78	0.89	0.78
	DT	0.78	0.80	0.78
	RF	0.80	0.88	0.80
Mean	KNN	0.80	0.86	0.80
	DT	0.78	0.79	0.78
	RF	0.82	0.87	0.82

Table 3: Results for problem 1, that is, all degrees

Random Seed	Learning Model	Accuracy	Sensitivity	Specificity
0	KNN	0.78	0.89	0.78
	DT	0.75	0.92	0.75
	RF	0.81	0.94	0.81
1	KNN	0.79	0.89	0.78
	DT	0.72	0.88	0.71
	RF	0.81	0.93	0.81
2	KNN	0.79	0.86	0.79
	DT	0.78	0.91	0.78
	RF	0.81	0.94	0.80
3	KNN	0.79	0.85	0.79
	DT	0.78	0.88	0.78
	RF	0.81	0.93	0.80
4	KNN	0.78	0.86	0.78
	DT	0.76	0.93	0.75
	RF	0.82	0.94	0.82
5	KNN	0.80	0.89	0.79
	DT	0.76	0.92	0.75
	RF	0.81	0.98	0.80
6	KNN	0.80	0.88	0.80
	DT	0.74	0.88	0.74
	RF	0.82	0.93	0.81
7	KNN	0.78	0.83	0.78
	DT	0.75	0.87	0.75
	RF	0.80	0.91	0.79
8	KNN	0.81	0.89	0.80
	DT	0.75	0.90	0.75
	RF	0.82	0.93	0.82
9	KNN	0.81	0.89	0.80
	DT	0.76	0.92	0.75
	RF	0.82	0.96	0.81
Mean	KNN	0.79	0.87	0.79
	DT	0.76	0.90	0.75
	RF	0.81	0.94	0.81

Table 4: Results for problem 2, that is, only bachelor's degrees

Random Seed	Learning Model	Accuracy	Sensitivity	Specificity
0	KNN	0.78	0.89	0.78
	DT	0.75	0.92	0.74
	RF	0.81	0.93	0.81
1	KNN	0.79	0.89	0.78
	DT	0.72	0.88	0.71
	RF	0.81	0.94	0.81
2	KNN	0.79	0.86	0.79
	DT	0.78	0.91	0.77
	RF	0.80	0.94	0.80
3	KNN	0.79	0.85	0.79
	DT	0.78	0.89	0.77
	RF	0.81	0.94	0.80
4	KNN	0.78	0.86	0.78
	DT	0.75	0.93	0.75
	RF	0.82	0.93	0.81
5	KNN	0.80	0.89	0.79
	DT	0.75	0.91	0.75
	RF	0.81	0.98	0.80
6	KNN	0.80	0.88	0.80
	DT	0.75	0.90	0.74
	RF	0.82	0.93	0.82
7	KNN	0.78	0.83	0.78
	DT	0.74	0.86	0.73
	RF	0.81	0.91	0.80
8	KNN	0.81	0.89	0.80
	DT	0.77	0.91	0.77
	RF	0.91	0.93	0.81
9	KNN	0.81	0.89	0.80
	DT	0.78	0.94	0.77
	RF	0.81	0.96	0.81
Mean	KNN	0.79	0.87	0.79
	DT	0.76	0.91	0.75
	RF	0.82	0.94	0.81

Table 5: Results for problem 3, that is, only unique cycle's degrees

Random Seed	Learning Model	Accuracy	Sensitivity	Specificity
0	KNN	0.80	0.89	0.80
	DT	0.70	0.52	0.81
	RF	0.78	0.88	0.78
1	KNN	0.82	0.89	0.82
	DT	0.78	0.92	0.78
	RF	0.77	0.93	0.77
2	KNN	0.85	0.82	0.85
	DT	0.77	0.77	0.77
	RF	0.81	0.84	0.81
3	KNN	0.82	0.87	0.82
	DT	0.76	0.89	0.76
	RF	0.78	0.88	0.78
4	KNN	0.85	0.80	0.85
	DT	0.61	0.89	0.60
	RF	0.82	0.85	0.82
5	KNN	0.84	0.83	0.84
	DT	0.85	0.59	0.86
	RF	0.84	0.84	0.84
6	KNN	0.85	0.80	0.85
	DT	0.81	0.74	0.81
	RF	0.82	0.87	0.82
7	KNN	0.83	0.76	0.83
	DT	0.61	0.83	0.60
	RF	0.79	0.78	0.79
8	KNN	0.80	0.87	0.79
	DT	0.65	0.77	0.64
	RF	0.80	0.85	0.80
9	KNN	0.82	0.85	0.81
	DT	0.66	0.85	0.66
	RF	0.80	0.86	0.80
Mean	KNN	0.83	0.84	0.83
	DT	0.72	0.78	0.73
	RF	0.80	0.86	0.80

Table 6: Results for problem 4, that is, only master's degrees

The main limitation is the fact that at the moment of subscription is not too easy to understand if a student will drop out or not, you may just have an idea. Anyway, integrating the data during the students' careers with the results of the exams can bring more and more improvements to prediction and therefore provide a good support tool to intervene in cases at risk.

In the future, possible improvements could be obtained in a first attempt by banally using new data coming from the following academic years, as the dataset in exam is not so big. An interesting work was already performed in [1] by dividing the problem between the various studies courses, observing that not all the courses got the same number of dropouts and consequently in some of them the prediction can be more effective. Moreover, it could be interesting to observe different targets instead of the dropout, for example, transfers to other Universities or steps to other courses. Maybe work in that direction could also bring some information about the reasons for the dropout as the transfers and the steps can also be considered harmful events in some cases. A possible explanation behind that statement could be that a dropout doesn't necessarily mean a failure in the academic life of a student, it can also be a change in the direction of a different University course that is more suitable for that student.

References

[1] Del Bonifro, F., Gabbrielli, M., Lisanti, G., & Zingaro, S. P. (2020, July). Student dropout prediction. In International Conference on Artificial Intelligence in Education (pp. 129-140). Springer, Cham.

[2] Fix, E., & Hodges, J. L. (1989). Discriminatory analysis. Nonparametric discrimination: Consistency properties. International Statistical Review/Revue Internationale de Statistique, 57(3), 238-247.

[3] Breiman, L.: Random forests. Mach. Learn. 45(1), 5–32 (2001). https://doi.org/10.1023/A:1010933404324

[4] Pedregosa, F., et al.: Scikit-learn: machine learning in python. J. Mach. Learn.Res. 12(Oct), 2825–2830 (2011)