A Machine Learning Predictive System to Identify Students in Risk of Dropping Out of College

Gabriel Silva[1], Marcelo Ladeira[1]

University of Brasília, Brazil
mladeira@unb.br

Abstract. The University of Brasília (UnB) suffers from undergraduate student drop out, which implies negative academic, economic and social consequences. UnB's approach to the problem consist of separating it's students in two groups: those that are in risk of dropping out and those that are not, and counsel the students in the former group. This paper describes the development of a predictive system capable of indicating the risk of a student dropping out. This way, UnB could act before it became too late and also act according to the risk presented by the student. For the development of the predictive system, data of students from computer science related courses that entered and left UnB from 2000 to 2016 were used. The data do not contain student identification. Machine learning (ML) algorithms were used to induce models that had their performance analyzed. The ML algorithms applied were ANN, linear regressor, Naive Bayes, random forests and SVR. Machine Learning models got, in general, good performance. The best performance came from the models induced using linear regression. Results obtained indicate potential in using machine learning to predict the risk of students from computer science related courses dropping out of university. The methodology used can be applied for other courses from UnB or other universities.

Categories and Subject Descriptors: I.2.6 [**Artificial Intelligence**]: Learning

Keywords: machine learning, data mining, student evasion, UnB

1. INTRODUCTION

Student dropping out of Brazilians universities are a significant problem, with academic, social and economic consequences. University of Brasilia (UnB) is not an exception, being significantly affected by the problem [1].

This paper describes a possible approach for the problem: the development of a predictive analysis system that estimates the risk of a given student dropping out of college. In case of success, the system would allow the university to take measures with flexibility (according to the risk presented by a student) and in advance (before it became too late).

The system outputs, for every student, a triple (v_1, v_2, v_3), with positive values that sum 1. These values indicate, respectively, the chance of the given student graduate, evade or migrate from the course he or she is in.

The rest of this article is organized in the following structure: in the next section, the methods used are thoroughly explained; after that, the good results obtained are presented and discussed; finally, conclusion is made and ideas for future work are exposed.

[1] UnB had a money lost estimated in R$ 95.6 millions, according to the Brazilian press Correio Brasiliense [cor]

2. METHODS

In this section, the methodology used is discussed. First, the data obtained is described, along with the train and test division. After that, feature selection is explained. Next, the division by semester is motivated. Afterwards, the machine learning algorithms used are listed, with their parameters configuration. Finally, the performance metric used is explained.

2.1 Data, Data Division and Train and Test Division

The data used are from undergraduate students that entered and left UnB courses from 2000 to 2016. To simplify the analysis, only courses in areas related to computer science were considered. Therefore, the courses considered were: Computer Science (bachelor degree), Computing (licentiate degree), Computer Engineering, Mechatronics Engineering, Network Engineering and Software Engineering.

In exploratory data analysis, it was possible to observe that the features varied significantly with the course of the student being considered. Another useful information obtained was that the proportion of students capable of graduating depends on the age the students enter university: students that enter older in university are less likely to graduate. These observations led to the decision of partitioning the data in four distinct databases:

—Senior Students: all students with more than 30 years.

—Young Students from FT: contain all students that entered in UnB with 30 years or less and course Mechatronics Engineering or Network Engineering. This courses have the distinction of being associated with FT (Faculty of Technology).

—Young Students from Computing: contain all students that entered in UnB with 30 years or less and course Computing. Computing is a licentiate degree in UnB, meaning that the students from the course are prepared to be teachers of Computer Science. Another peculiarity is the fact that it is the only night course from the ones we are considering.

—Young Students from Computer Science: contain all students that entered in UnB with 30 years or less and course Computer Science (bachelor degree), Computer Engineering or Software Engineering.

In order to induce models with machine learning algorithms, data were partitioned in a training set and a test set. As indicated in [da Silva and Adeodato 2012], a realistic division of the data for the problem we are dealing with should be an "Out of Sample" division, in which the training set is composed by the firsts instances and the test set from the last ones. This way, we get a realistic scenario in which the models are induced to be applied in a future time. With this in mind, the training set was composed of students that entered in university from 2000 to 2009 while the test set was composed of students that entered in university from 2010 to 2016.

2.2 Features and Feature Selection

After initial research (see [da Silva and Adeodato 2012] or [Kinnunen and Malmi 2006]) on which features should be included for the models to train, the following personal features were selected initially:

—Age when student started the course

—Course

—Entered via quota or not

—Race

—Sex

—Type of secondary school (Private or Public)

—Way In

In addition to personal features, academic features were, obviously, also considered:

—Amount of Credits Done

—Average Grades in a Semester

—Pass rate on the hardest subject of a semester

—Indicator of Condition: UnB classifies students as "in condition" or not "in condition" and this classification (see [man]) is related to the risk of students dropping out. This boolean variable indicates if the student is "in condition" or not.

—Pass Rate, Fail Rate and Drop Rate: indicate, respectively, the percentage of the subjects a given student passes, fails and drops.

—Position in relation to fellow students: for a given student S, this feature indicates, from all students of the same year, semester and course of S, how many have higher grades than S.

—Rate of Academic Improvement: reason between grades of a student in the current semester by the grades of the same student in the previous one.

Feature selection was done afterwards. The features race and type of school were eliminated from further analysis, because of having a significant amount of missing values (more than 40%). To avoid redundancy, a Kendall test (see [Noether 1981]) was applied to check if any two of the attributes had very strong correlation. Because the test indicated that the fail rate and the pass rate were significantly related, a decision was made to consider only the pass rate (this strong correlation was verified for all 4 databases).

Moreover, to check if all the features were really necessary, decision trees were employed: features that did not appear as part of a decision tree were not considered in further analysis. Results indicated that, for young students of computing the feature course was irrelevant (which makes sense, since all students from this group have the same courses). For senior students, the features course, quota and drop rate were considered irrelevant. For the other 2 databases, no feature was considered irrelevant.

2.3 The Necessary Semester Division

The predictive system, for the business problem here considered, should be capable of calculating the drop out risk for students in the beginning of the course and for students in the end of the course. In relation to that, students academic features change from semester to semester.

This indicates that a necessary semester division must be carried out. Thus, the models are induced and evaluated separately semester by semester. Therefore, initially, the models are induced on the train dataset containing features from the 1^o semester of the students and are evaluated on the test dataset containing features from the 1^o semester of the students. Next, this procedure is repeated for the 2^o semester and so on.

2.4 Machine Learning Algorithms

The machine learning algorithms used in this research were: ANN, linear regressor, Naive Bayes, random forest, SVR. For more information on this algorithms, we recommend the excellent books [Kelleher et al. 2015] and [Abu-Mostafa et al. 2012]. To establish a baseline, the ZeroR model was considered (this model simply always picks the most common class as the predicted one). The Python programming language was used, combined with the `scikit-learn` library [Pedregosa et al. 2011] (v. 0.18.1).

Table I. ANN's configuration, according to the database

Database	**Hidden Layer Size**	**Learning Rate**
Senior Students	24	0.7
Young Students from FT	100	0.001
Young Students from Computing	36	1.0
Young Students from Computer Science	36	0.001

Table II. SVR's configuration, according to the database

Database	**Kernel Type**	**Penalty Parameter**
Senior Students	linear	1.0
Young Students from FT	linear	1.0
Young Students from Computing	linear	1.0
Young Students from Computer Science	RBF	1.0

Table III. Naive Bayes configuration, according to the database

Database	**Feature Distribution**
Senior Students	Gaussian
Young Students from FT	Gaussian
Young Students from Computing	Bernoulli
Young Students from Computer Science	Multinomial

Preliminary studies were made to determine the best configurations for the ANN, the Naive Bayes and the SVR parameters. For the other machine learning algorithms, their default configurations were used. The results varied according to the databases, and are shown on Tables I, II, III.

2.5 Performance Measures

As is standard in machine learning, the models were induced on the train dataset and had their performance measured in a test dataset, with unseen data. This process was done for each one of the semester studied and works as explained next.

Each machine learning model generates, for each student in each semester, a triple that indicates the assessed possibility of a student graduating, evading or migrating. The highest value of the triple is used as the model prediction. This prediction is then compared to what really happened to the student.

The metric used to evaluate the performance of the models was the F-measure. A given model has, for a given database, values of F-measure calculated, one for each semester. To summarize the performance of the model for a database, the mean of the F-measures was calculated. The models were compared to the ZeroR classifier.

3. RESULTS AND DISCUSSION

Table IV. F-measure Mean per Model, for Young Students from FT

ML Model	**F-measure**
ANN	0.77
Linear Regressor	0.80
Naive Bayes	0.56
Random Forest	0.74
SVR	0.76
ZeroR	0.64

Table V. F-measure Mean per Model, for Young Students from Computing

ML Model	**F-measure**
ANN	0.85
Linear Regressor	0.87
Naive Bayes	0.76
Random Forest	0.86
SVR	0.82
ZeroR	0.70

Table VI. F-measure Mean per Model, for Young Students from Computer Science

ML Model	**F-measure**
ANN	0.68
Regressor Linear	0.77
Naive Bayes	0.65
Random Forest	0.76
SVR	0.70
ZeroR	0.60

Table VII. F-measure Mean per Model, for Senior Students

ML Model	**F-measure**
ANN	0.62
Linear Regressor	0.75
Naive Bayes	0.28
Random Forest	0.71
SVR	0.80
ZeroR	0.61

The mean of the F-measures was calculated for each model and database. The results are shown on Tables IV, V, VI and VII.

This results show that, in general, the ML models have better performance than the ZeroR (Naive Bayes being the exception). The bad performance of the Naive Bayes algorithm may be due to the fact that the features are not conditionally independent, a hypothesis necessary for the use of the algorithm. Lastly, the good result obtained by the linear regressor should be detached. That learning model obtained the best results in three of the four databases, with a mean value of F-measure close to 0.8. This goes in accordance with the theory of machine learning that assures that, generally, linear models are not likely to overfit and are good initial alternatives to the problem [Abu-Mostafa et al. 2012].

4. CONCLUSION

The linear regressor algorithm induced models with good performance in assessing the risk of a student graduating, dropping out or migrating. This result show the viability of using machine learning for predicting the risk of students dropping out of college. The methodology used can be applied for other undergraduate courses from UnB or other universities.

A natural sequence for this research would be the implementation of dropping out related actions in UnB based on the risk predicted by the system here described. This actions should be evaluated based on criteria such as drop out reduction obtained and acceptance by university members. Another possible sequence would be testing the system for other courses of UnB or another university.

REFERENCES

`http://www.correiobraziliense.com.br/app/noticia/cidades/2015/10/10/interna_cidadesdf,501999/evasoes-na-universidade-de-brasilia-causam-prejuizo-de-r-95-mi.shtml`. Acessed in July 7, 2015.

`http://www.unb2.unb.br/administracao/decanatos/deg/downloads/index/guiacalouro.pdf`. Acessed in July 7, 2017.

Abu-Mostafa, Y. S., Magdon-Ismail, M., and Lin, H.-T. *Learning from data*. Vol. 4. AMLBook New York, NY, USA:, 2012.

da Silva, H. R. B. and Adeodato, P. J. L. A data mining approach for preventing undergraduate students retention. In *Neural Networks (IJCNN), The 2012 International Joint Conference on.* IEEE, pp. 1–8, 2012.

Kelleher, J. D., Mac Namee, B., and D'Arcy, A. *Fundamentals of machine learning for predictive data analytics: algorithms, worked examples, and case studies.* MIT Press, 2015.

Kinnunen, P. and Malmi, L. Why students drop out cs1 course? In *Proceedings of the second international workshop on Computing education research.* ACM, pp. 97–108, 2006.

Noether, G. E. Why kendall tau. *Teaching Statistics* 3 (2): 41–43, 1981.

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., and Duchesnay, E. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* vol. 12, pp. 2825–2830, 2011.