Exploiting Academic Records for Predicting Student Drop Out: a case study in Brazilian higher education

Allan Sales, Leandro Balby, Adalberto Cajueiro

Universidade Federal de Campina Grande, Brazil
allan.melo@ccc.ufcg.edu.br
{lbmarinho,adalberto}@computacao.ufcg.edu.br

Abstract. Students' drop out is a major concern of the Brazilian higher education institutions as it may cause waste of resources and decrease graduation rates. The early detection of students with high probability of dropping out, as well as understanding the underlying causes, are crucial for defining more effective actions toward preventing this problem. In this article, we cast the drop out detection problem as a classification problem. We use a large sample of academic records of students across 76 courses from a public university in Brazil in order to derive and select informative features for the employed classifiers. We create two classification models that either consider the specific course to which the target student is formally enrolled or not, respectively. We contrast both models and show that we have better results by not distinguishing the students by course.

Categories and Subject Descriptors: H.2.8 [**Database Management**]: Database Applications; I.2.6 [**Artificial Intelligence**]: Learning

Keywords: Classification, Drop out, Education, Educational Data Mining, Higher education institutions, Learning analytics, Machine Learning

1. INTRODUCTION

With the creation of several public policies towards expanding the access to Brazilian higher education, the number of enrollments has notably increased in recent years. In 2013, for example, more than 7 million enrollments were registered and this number is continually growing up [INEP 2013b]. However, it is estimated that only 62.4% of these enrollments succeeds in getting an undergraduate degree [INEP 2010], which suggests a high rate of drop out students.

The student drop out problem occurs widely in several levels of formal education around the world. The most common reasons associated with this problem are poor grades, bad teaching or badly structured subjects, getting a job before or during the studies, lack of employment perspective, family-related issues and lack of aptitude for the course [Gaioso 2005; Barroso and Falcão 2004; Adachi 2009; Andriola et al. 2006]. Many studies have pointed out that drop out is more often in the beginning of the courses[1], due to some of the aforementioned reasons [Dekker et al. 2009; Pal 2012]. Considering the data set we used in our experiments (see Section 4 for more details) comprising 76 higher education courses in the Federal University of Campina Grande (UFCG) - Brazil[2], we observed that the number of drop outs per semester (see the left hand side of Figure 1) occur mainly in the beginning of the course and may cause a big cost - in R$ - to the university [INEP 2013a]. Although the number of drop outs abruptly decreases across the semesters, the cost to the university decreases more softly (or

[1]By course we mean an academic discipline, which is equivalent to an academic major in the US
[2]http://www.ufcg.edu.br

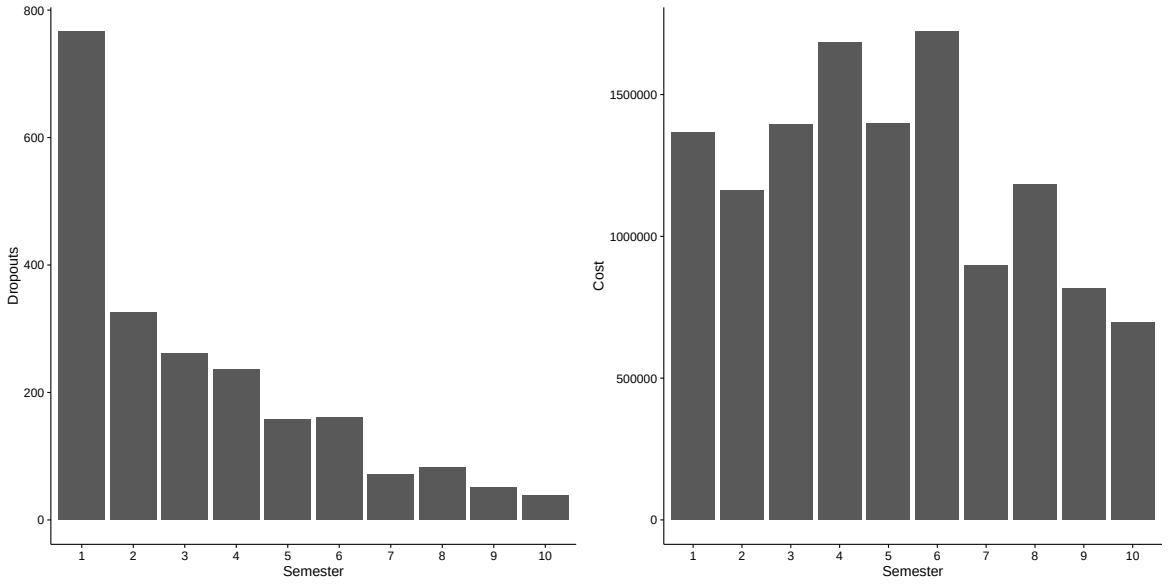

Fig. 1. 1) Number of drop outs per semester enrolled, and 2) cost of drop outs per semester enrolled at UFCG.

even increase), showing that the later the drop out occurs more expensive it becomes, turning the drop out problem an issue independent of when it occurs. The right hand side of Figure 1 shows the cost of drop outs per semester for UFCG's students from the first to the tenth semester of enrollment in 2013.

The student drop out problem might impact:

—The society, which will lack more qualified professionals;
—The students, which will take longer to obtain their degrees (assuming that the drop out was due to a course change);
—The university, that invests in infrastructure - i.e. computers, rooms, teachers - that will be underused due to the lack of students. It is estimated that in 2013, each student had an annual cost of R$21.383,00 [INEP 2013a] to the Brazilian universities per year.

In this article we formulate the student drop out prediction problem as a supervised learning problem using features extracted from academic records of students. To this end, we employ classification models that categorize these students into two different classes: 'drop out' or 'continue'. I.e., we want to predict which students will not continue in the university after the end of a given semester. We create two classification models that either consider the course/semester to which the target student is formally enrolled or not, respectively. The idea is to investigate whether it pays off to create a specialized model for each course/semester pair instead of just creating one model per semester. We evaluated many different learning configurations with a variety of techniques, such as feature selection and class balancing, and discovered that not considering the course leads to better results than otherwise.

Our approach is inserted in the field known as Educational Data Mining (EDM) [Romero and Ventura 2013] which has been a powerful tool to help educational institutions to devise better corrective and preventive actions such as improving the allocation of resources and staff or advising students identified as potential drop outs. Some works have appeared recently proposing to apply machine learning to detect students' drop out (see Section 2). We extend these works with the main following contributions:

—We use a large and comprehensive data set of students' academic records from 76 different courses of a public Brazilian university;
—We propose new discriminative features that are not found in the reviewed literature;
—We propose two different perspectives to approach the problem, i.e., either creating one classifier per semester or one classifier per course/semester;
—We conduct a feature selection analysis, using different feature selection algorithms, in order to discover which features have the highest impact in the classifiers' performances on both perspectives described above;
—We evaluate several configurations of classification algorithms considering only the semester or course/semester pairs and choose the best setups to contrast these two perspectives.

2. RELATED WORK

In our previous work [Sales et al. 2015], we proposed a classification model to detect students with high probability of dropping out in the first semester and showed that good results can be achieved with only a small number of features. In the current article, we extend that work by proposing classification models that predict drop outs for any given course and semester. We also count with new data, such as the number of credits necessary to complete each course, which enabled us to extract new informative features. The main differences between this article and the previous one are: i) the inclusion of a thorough feature selection analysis, ii) the investigation of classification models considering only the semester or the course/semester of interest and iii) the investigation of class imbalance techniques such as the One-Sided Selection [Kubat et al. 1997] algorithm combined with random undersample/oversample.

There are several works that address the students' drop out problem, each one approaching a different perspective of the problem. Below we briefly describe the ones that are most related to ours.

Zender et al. [2014] assume that the lack of adaptation of students in their first academic semester may be amongst the main reasons for drop out. For example, freshmen may have problems adapting to a new environment, a new teaching methodology and even creating new social ties. The proposed solution is to create a pervasive game that students play during their first few weeks in the university. It was found that the game created made it easier for students to adapt to the university environment.

Moretti et al. [2014] investigate the retention of students in Computer Science courses and how to communicate the content of the course subjects more clearly. To achieve this goal, they investigated the following questions: (i) what programming language for introductory subjects helps the student to understand more clearly the instructions, (ii) how much weight must be given to the home activities, tests, quizzes, projects and extra activities for the student understand more easily the content, and (iii) if the students are more interested in subjects with publicly available online syllabus. They concluded that interpreted languages and an even weighting for projects and exams correlate with higher instructional clarity ratings.

Yadin [2011] investigates the drop out problem in an introductory programming subject of a Computer Science course. The author proposes a number of actions (e.g., use of procedural languages like Python) to make the discipline most effective as it relates to clearer instructions to the student. It is stated that such measures helped to reduce the failing students up to 77%.

Note that while the aforementioned works propose to first take actions toward addressing drop out and then measuring their impact, we first identify which students are in eminence of drop out for then communicating, eventually, the decision makers of the university.

Márquez-Vera et al. [2013] investigate students failure at high school in a city of Mexico. They use several popular classification algorithms and propose a genetic algorithm approach that considers cost-sensitive learning and class imbalance techniques. We consider the drop out problem in Brazilian

public higher education which is a related but different problem in comparison to drop out in high school.

Mustafa et al. [2012] exploit whether registration data of students (e.g., financial support, age, gender and disabilities) in the courses of Computer Science and Engineering at the University of Chittagong, are good features for predicting drop out. The authors use decision trees classifiers and conclude that the most important features to predict drop out are financial support, age and gender. It is further stated that the accuracy of the decision trees were about 38.10%. Thus, while financial support, age and gender have some impact on the prediction performance, using them alone is by no means sufficient for accurately predicting drop out.

Pal [2012] proposes to predict drop out before the students start their first academic year. It means that the student is already enrolled in the university, but has not yet started to take classes. To accomplish that, the author test four classification algorithms using socioeconomic and pre-university (e.g., student grades in high school) data. The models vary the accuracy rate from 67.7% to 85.7%. He concludes that the performance of students in high school is the most discriminative feature in the classification model. Our model differs from this approach in the sense that we consider students coursing any semester in the university and, furthermore, we do not have access to socioeconomic or pre-university information about the students thus relying solely upon students' academic records.

Dekker et al. [2009] investigate the drop out detection problem in Electrical Engineering courses after or before the first academic semester. To accomplish this goal they used students' data during their first academic semester and pre-university data as input to eight classification algorithms. They used cost-sensitive learning for handling class imbalance and evaluate the algorithms before and after this treatment. They measure accuracy, true positive, true negative, false positive and false negative rates. The conclusion of their work is that the pre-university data was not effective and that the grades in linear algebra and calculus subjects were important for predicting the progress of students in the rest of the course. We follow a similar approach, but we consider students in any course/semester including 76 different courses.

Balaniuk et al. [2011] address the drop out prediction problem using data from 11,495 students in three courses (Journalism, Law and Psychology) of a higher education institute in Brasilia, Brazil. Three classification algorithms were used to classify students into "drop out" and "graduate", and as input for training the models, they used both socioeconomic information and academic information of the students. They concluded that it is possible to identify students with high risk of dropping out with an accuracy of 80.6%.

Manhães et al. [2014] propose a similar approach as Balaniuk et al. [2011] with the key difference that, similarly to us, only features extracted from academic records are used. They used five classification algorithms and data from six courses of the Federal University of Rio de Janeiro, Brazil: Civil Engineering, Mechanical Engineering, Production Engineering, Law, Physics and Pharmacy. Their approach yielded an accuracy of at least 87% for each course. Our work is very similar to this in terms of the approach used, but we consider the number of semesters enrolled as an important factor to make the classification and also consider more courses in our evaluation.

3. PROBLEM FORMULATION

As mentioned in previous sections, we formulate the student's drop out problem as a binary classification problem. Binary classification typically considers a set of m-dimensional feature vectors $X \in \mathbb{R}^m$, a set of classes $Y = \{+, -\}$ (in our case 'drop out' and 'continue'), and a training set of the form $D^{train} = \{(\vec{x}_1,y_1),\ldots,(\vec{x}_n,y_n)\}$ where $\vec{x} \in X$ is a vector of attributes and $y_i \in Y$ represents the class which $\vec{x}_i$ belongs to. The idea is to find a classification function $\hat{y} : X \rightarrow Y$ that minimizes the error in the test set $D^{test} = \{(\vec{x}_1,y_1),\ldots,(\vec{x}_p,y_p)\}$, that is unavailable during training, i.e., $D^{test} \cap D^{train} =$

$\emptyset$. More formally, the goal is to minimize:

$$err(\hat{y}; D^{test}) = \frac{1}{|D^{test}|} \sum_{(\vec{x},y) \in D^{test}} l(y, \hat{y}(\vec{x})) \quad (1)$$

where $l : Y \times Y \rightarrow \mathbb{R}$ is a loss function measuring, for any test instance $(\vec{x}, y) \in D^{test}$, the misfit between the true y and the predicted value $\hat{y}(\vec{x})$. We instantiate classification for our problem as described in the following subsections.

3.1 Classification per Semester

In this scenario we want to predict whether a given target student from a given semester will drop out the course. Our feature set is then defined as $X = \{X_t : t \in T\}$ where $T \subseteq \mathbb{N}$ is the set of semesters a student may have coursed so far (e.g. $t = 5$ if the student has coursed five semesters since his entrance). We now have a training set for each specific semester, i.e., $D_t^{train} = \{(\vec{x}_1, y_1)^t, \ldots, (\vec{x}_n, y_n)^t\}$. Thus, for each $t \in T$ we will have a classifier $\hat{y}^t : X^t \rightarrow Y$ that will predict whether a student in semester $t \in T$ will drop out or not.

3.2 Classification per Course/Semester

Similarly to the classification per semester, but here we have a separate training set and classifier for each (course,semester) pair. In this case, the feature set is defined as $X = \{X_t^c : t \in T, c \in C\}$ where C is the set of courses available in the university. Now, for each $c \in C$ and $t \in T$ we will build a classifier $\hat{y}_c^t : X_t^c \rightarrow Y$ that will predict whether a student from course $c \in C$ in semester $t \in T$ will drop out or not.

4. DATA PREPARATION

The dataset used in our experiments was kindly provided by the administration of UFCG. The data set consists of academic records of UFCG students from 2002 to 2014 across 76 different courses. This represents 12.5 years of data (or 25 semesters) with around 32,342 students enrolled during this period, from which 12,560 have dropped out in some point of his course, resulting in 61.16% of courses completed. Table I enumerates and describes the data fields available in this data set.

4.1 Data Preprocessing

Before analysing which features should be used as input in our models, it was necessary to prepare the data described at Table I in order to deal with the following situations:

Table I. Data fields description.

Column	Description
Enrollment id	Unique identifier of the student
Course id	Unique identifier of the course
Semester id	Identifier of the semester (e.g. 2014.1 means the first semester of 2014)
Entry semester	Semester when the student started the course
Last semester	Last semester the student was enrolled before drop out
Subject id	Identifier of the subject
Credits	Weight of the subject in the course (based on the number of class hours)
Grade	Grade of the student in the subject in the [1,10] range
Situation	Situation of the student in the subject (approved, failed by grade, by attendance or enrollment cancel)
Subject type	'Required' if the subject is mandatory in the course, 'Optional' otherwise.
Drop out code	A code that identifies the type of drop out (e.g. drop out by abandonment and by conclusion)
Course credits	Total of credits needed to complete the course (based on the number of class hours)

—Drop out code. There are several codes used in UFCG to characterize the drop out of a student in a course, e.g., drop out by abandonment, by university transfer or by death, and every code had to be mapped to 'drop out' or 'continue'. To label the training instances, we have checked, for each student in semester t, if he was still enrolled in semester $t+1$ and, if so, the instance received class label 0 (or 1 otherwise). For example, a student that dropped out his course in the third semester will receive code 0 in the first and second semester and code 1 in the third. Students who concluded their courses received label 0 in all semesters.

—Semester calculation. We calculated the current semester of every student using the semester id and the entry semester by just counting the number of semesters that has passed since his entry semester until the semester id. We called this attribute N.SEMESTERS.ENROLLED.

—Course re-entrance. In most of Brazilian public universities (including UFCG), it is possible for a student to re-enter in the course he is already enrolled through the Brazilian High School National Exam known as ENEM. This results in a new enrollment id where his academic records will contain only the subjects in which he was approved while using the old id. We handle this situation by identifying these students and creating a new student id aggregating the records spread over all the possible past enrollment ids associated to him. This will eliminate the so called fake freshman and the fake drop out cases.

4.2 Feature Selection

We now turn to select the most important features for drop out prediction. For doing that, we first considered all features introduced by Manhães et al. [2014] and some others created by us, such as:

—N.SEMESTERS.ENROLLED: captures how long, in semesters, the student is enrolled in a course;
—COURSE.COMPLETED: indicates how distant the student is from getting his degree;
—N.CREDITS.TOTAL: indicates the amount of effort a student will have to employ this semester;
—N.CREDITS.DIF: indicates how much the student is deviating from his class.

The idea to create N.CREDITS.DIF came from the observation that it is usual for students' course plans (i.e. the subjects selection for each semester), in general, to be similar, thus if a student gets too far from the course plans of his peers it might be a sign of drop out.

It is worth mentioning that we have chosen to use the features of Manhães et al. [2014] because her research is one of the most recent in the area and also one of the most similar to ours. Table II shows the complete list of features. The features we introduced are highlighted with an asterisk.

We then applied, for each distinct semester, the statistical test of Mann and Whitney [1947] on the features considering two samples on the training set: the features associated with instances of class 0 and 1 respectively. If the test results indicate that the features in the different samples come from the same population, we can conclude that these features are not discriminative and should be discarded. To complete our features set, we also added the STATUS.SEM feature.[3] Table III shows the semesters in which each feature is considered discriminative. In this research, we worked with students enrolled until the tenth semester due to the small number of drop outs in later semesters.

5. EXPERIMENTAL SETUP

We considered two basic assumptions to create the classification models investigated in this article. The first one assumes that the reasons or features that explain drop out are the same (or not vary much) regardless the course a student is enrolled in (we called this approach Global Model). The second one assumes that these reasons are dependent on the specific course under consideration and

[3]It was not included in the Wilcoxon-Mann-Whitney test since it is not a numeric feature.

Table II. Feature set and description.

Type/Value	Feature	Description
Id code (string)	Student id	Student's identifier
Id code (string)	Course id	Course's identifier
{1 to n} (numeric)	Semester id	Current semester's identifier
{0 or 1} (string)	drop out code	Target variable
{0 to n} (numeric)	N.(APPR, FAIL.GRADE, ABFL, FAIL, CANCELLED)	Respectively, number of subjects approved, failed by grade, failed by attendance, failed (by grade and attendance) and enrollment cancelling in the semester
{0 to 10} (numeric)	MEAN.APPR	Average grade of the approved subjects in the semester
{0 or 1} (numeric)	STATUS.SEM	The semester status (Defined with 0 if the student failed all subjects, 1 otherwise)
{0 to 10} (numeric)	SEM.MEAN	Mean of all subjects grades the student is enrolled in the semester
{0 to 10} (numeric)	GPA	Harmonic mean composed by the Grade and Credits in the semester
{0 to 10} (numeric)	N.SEMESTERS.ENROLLED*	Number of semesters the student coursed so far
{0 to 1} (numeric)	COURSE.COMPLETED*	Number of credits concluded divided by total number of credits to conclude the course
{0 to n} (numeric)	N.SUBJ.(REQUIRED, OPTIONAL, TOTAL)	Number of required, optional and total subjects the student is enrolled in semester
{0 to n} (numeric)	N.CREDITS.(REQUIRED, OPTIONAL, TOTAL, APPR)*	Number of credits of required subjects, optional subjects, total subjects and approved subjects in semester
{0 to 1} (numeric)	PROP.N.(ABFL, FAIL, APPR, BLOCKED)	Proportion. Value of the N.ABFL, N.FAIL, N.APPR and N.BLOCKED divided by the total of subjects in semester
{0 to n} (numeric)	N.CREDITS.DIF*	Difference between the number of credits the student is enrolled and the mode of his class (students of same semester entrance of same course)
{0 to n} (numeric)	N.CREDITS.(REQUIRED, OPTIONAL, APPR).DIF*	Difference between the number of credits of required, optional and approved subjects the student is enrolled and the mode of his class (students of same semester entrance of same course)

thus may positively impact the performance of the classifier if taken into consideration (called Specific Model henceforth). For both assumptions, we propose and evaluate several models and then compared the best ones from each case to see which assumption more faithfully describe the data.

As for the classification algorithm, we decided to use Random Forest, which is a very strong state-of-the-art method, having been applied with success in many different domains [Breiman 2001].

5.1 Model Setup

In order to get the best model setup we created different Random Forest models considering the following factors: feature selection, class imbalance techniques and the number of trees in the forest, as described in Table IV. The reasons for considering these factors are explained as follows: 1) feature selection is important since we want to find out which attributes are more important to predict drop

Table III. Discriminative features by semester.

Attributes	Semesters Enrolled
N.SUBJ.REQUIRED	1, 2, 3, 4, 5, 6, 7, 8, 10
N.SUBJ.OPTIONAL	1, 2, 3, 6, 7, 8, 9, 10
N.SUBJ	1, 2, 3, 4, 5, 6, 7, 8, 9
N.CREDITS.REQUIRED	2, 3, 4, 5, 6, 7, 8, 10
N.CREDITS.OPTIONAL	1, 2, 3, 6, 7, 8, 9, 10
N.CREDITS	1, 2, 3, 4, 5, 6, 7, 8, 9
COURSE.COMPLETED	1, 2, 3, 4, 5, 6, 7, 8, 9, 10
N.APPR	1, 2, 3, 4, 5, 6, 7, 8, 9, 10
N.FAIL.GRADE	3, 4, 5, 6, 7, 8, 9, 10
N.ABFL	1, 2, 3, 4, 5, 6, 7, 8, 9, 10
N.BLOCKED	1, 2, 3, 4, 5, 6, 7, 8, 9, 10
MEAN.APPR	1, 2, 3, 4, 5, 6, 7, 8, 9, 10
PROP.N.ABFL	1, 2, 3, 4, 5, 6, 7, 8, 9, 10
PROP.N.APPR	1, 2, 3, 4, 5, 6, 7, 8, 9, 10
PROP.N.FAIL.GRADE	3, 4, 5, 6, 7, 8, 9, 10
PROP.N.BLOCKED	1, 2, 3, 4, 5, 6, 7, 8, 9, 10
N.SUBJ.DIF	1, 2, 3, 4, 5, 6, 7, 8, 9
N.CREDITS.DIF	1, 2, 3, 4, 5, 6, 7, 8, 9
N.CREDITS.REQUIRED.DIF	1, 2, 3, 4, 5, 6, 7, 8, 9
N.CREDITS.OPTIONAL.DIF	1, 2, 3, 4, 5
SEM.MEAN	1, 2, 3, 4, 5, 6, 7, 8, 9, 10
GPA	1, 2, 3, 4, 5, 6, 7, 8, 9, 10
N.CREDITS.APPR	1, 2, 3, 4, 5, 6, 7, 8, 9, 10
N.CREDITS.APPR.DIF	1, 2, 3, 4, 5, 6, 7, 8, 9, 10

outs; 2) class imbalance techniques are important in scenarios where the number of instances of the majority class is much greater than the number of instances of the target class (what is true in our case); 3) in general, the more trees are included in the random forest, the better the results. Thus, we created different model configurations labelled like: feature selection-class imbalance-number of trees in the forest. I.e. a model configuration described as FALSE-FALSE-10 is a model created without making any feature selection or class balancing and with 10 trees in the random forest. Following the same logic, a model configuration described as gain.ratio-Oversample-100 is a model created using feature selection (gain ratio algorithm), balancing the classes based on One-Sided Selection (OSS) and random oversample and with 100 trees in the random forest.

We run the experiments using the R statistical tool[4] with the following packages: *FSelector*[5] (for feature selection), *unbalanced*[6] (for class imbalance), and *randomForest*[7] (for running the random forest classifiers).

The experiment is split into two phases: 1) validation phase and 2) test phase. The validation phase consists of choosing the best model configuration by using the data from 2002.1 to 2012.2 for training and 2013.1 for validation. The test phase consists of retraining the model - using data from 2002.1 to 2013.1 - based on the best configuration found in the validation phase and testing with data from 2013.2. Thus, we expect to simulate the scenario where we have students enrolled in 2013.2 and we want to predict which ones are at risk of dropping out so that the university administration can act before it happens.

We ran each model configuration 20 times to calculate a confidence interval. We do that because random forests and the class imbalance techniques make some random choices that may slightly change the results from execution to execution. For each configuration, the F-measure was calculated

[4] https://www.r-project.org/
[5] https://cran.r-project.org/web/packages/FSelector/index.html
[6] https://cran.r-project.org/web/packages/unbalanced/index.html
[7] https://cran.r-project.org/web/packages/randomForest/index.html

Table IV. Factors and possible values.

Factor	Values			
Mutual Information	Information Gain	Gain Ratio	Symmetrical Uncertainty	False
Class Balancing	Undersample (OSS + Undersample)	Oversample (OSS + Oversample)	False	-
Numer of Trees	1	10	50	100

based on the mean of every semester's F-measure - in classification per semesters' scenario - or every course/semester pair - in classification per semesters/course's scenario.

5.2 Global Model

This corresponds to the scenario described in Section 3.1 where the idea is to have a separate classifier per semester.

5.2.1 *Mutual Information.* In order to investigate whether the importance of the features vary across the semesters, we computed the information gain, gain ratio and the symmetrical uncertainty of each feature of Table III. We created a model for each mutual information algorithm using the 3 most important features assigned by them. Table V shows the most important features by semester for each feature selection algorithm.

In Table V it is possible to observe that the importance of the features indeed varies depending on the feature selection algorithm used. Other interesting conclusions that can be draw are listed below:

—COURSE.COMPLETED and N.CREDITS.APPR are considered important despite the semester;
—The importance of N.CREDITS.APPR increases across the semesters;
—STATUS.SEM seems to be a good feature to be considered in a model for the first semester;

Table V. Feature importance per semester.

Information Gain										
Attributes	1	2	3	4	5	6	7	8	9	10
N.APPR	0.244									
N.CREDITS.APPR	0.324	0.212	0.197	0.239	0.258	0.266	0.285	0.276	0.317	0.376
COURSE.COMPLETED	0.985	0.752	0.580	0.575	0.467	0.476	0.392	0.341	0.434	0.205
N.CREDITS.REQUIRED		0.171				0.207	0.242	0.265		0.238
N.CREDITS.TOTAL			0.183	0.176	0.221				0.257	

Gain Ratio										
Attributes	1	2	3	4	5	6	7	8	9	10
COURSE.COMPLETED	0.199	0.160	0.139	0.134	0.126	0.123	0.112	0.110	0.126	
PROP.N.ABFL	0.182									
STATUS.SEM	0.392	0.171	0.141	0.102	0.095	0.100				
N.SUBJ.OPTIONAL		0.125								
N.CREDITS.OPTIONAL			0.115							0.118
N.CREDITS.APPR				0.077	0.089	0.094	0.099	0.106	0.116	0.142
N.CREDITS.REQUIRED							0.088			0.123
N.CREDITS.TOTAL								0.101		
N.SUBJ.TOTAL									0.097	

Symmetrical Uncertainty										
Attributes	1	2	3	4	5	6	7	8	9	10
N.CREDITS.APPR	0.139	0.091	0.088	0.102	0.115	0.119	0.127	0.131	0.146	0.177
COURSE.COMPLETED	0.300	0.238	0.201	0.195	0.175	0.174	0.153	0.145	0.172	0.120
STATUS.SEM	0.145									
N.CREDITS.OPTIONAL		0.090	0.096							
N.CREDITS.TOTAL				0.083	0.102			0.123	0.119	
N.CREDITS.REQUIRED						0.095	0.111			0.134

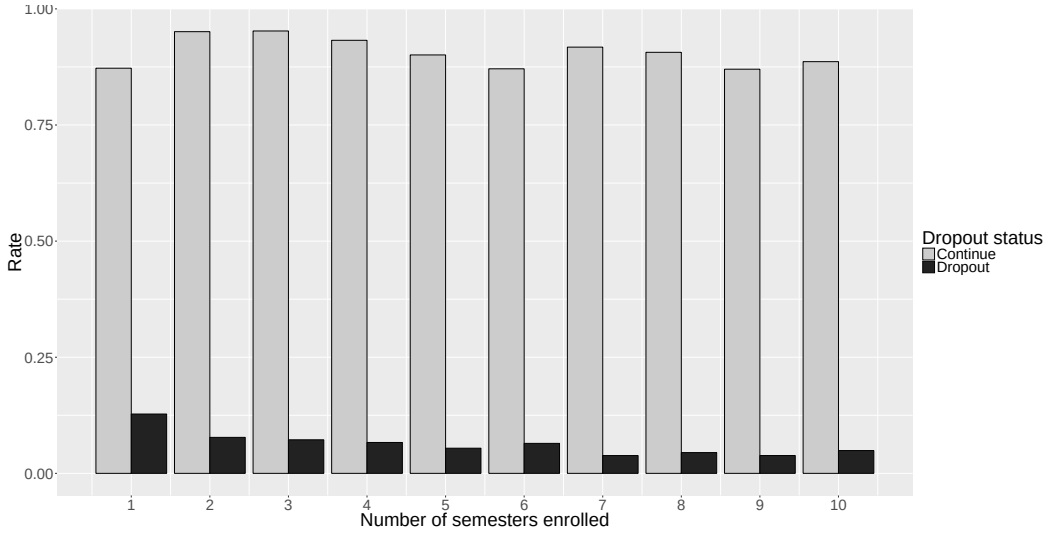

Fig. 2. Drop outs per semesters enrolled.

5.2.2 *Class Imbalance.* As depicted in Figure 2, the percentage of drop outs in each semester is much lower than the percentage of students who continue in their respective courses. This represents a problem known in the classification literature as class imbalance, a scenario where the classification may be biased towards classifying all test instances with the majority class [He and Garcia 2009].

To handle this problem, we applied the OSS algorithm [Kubat et al. 1997] - which is an undersampling method resulting from the application of Tomek Link [Tomek 1976] and Condensed Nearest Neighbor [Hart 1968] - followed by random undersample or random oversample method. The idea is, in a first moment, to use the OSS to remove noise and borderline instances from our training set and then, in a second moment, balance the proportion of instances in each class using random undersample or random oversample.

5.3 Specific model by Course/Semester

This corresponds to the scenario described at Section 3.2 where the idea is to have a separate classification model for each course/semester pair. In the subsequent subsections we will describe in detail this process.

5.3.1 *Mutual Information.* In Table VI we present the information gain, gain ratio and symmetrical uncertainty for the Computer Science and Pharmacy courses and we can draw the following observations about them:

—Feature importance varies across the semesters, feature selection algorithm used and course investigated;
—Some semesters of a course present just a few, or zero, drop outs. These course/semester pairs may not need a classifier.

We did this for all the courses, but presented the results only for these two courses due to space constraints. The results are shown only for Computer Science and Pharmacy, but the same conclusions hold for other courses.

For each mutual information algorithm, we create a classifier for each course/semester pair using the three most important features.

Table VI. Feature importance per semesters in Pharmacy and Computer Science.

Computer Science - Gain Ratio										
Features	1	2	3	4	5	6	7	8	9	10
N.ABFL	0.325									
PROP.N.ABFL	0.304	0.282	0.182						0.219	
STATUS.SEM	0.309		0.166	0.179	0.157	0.197				
N.BLOCKED		0.381				0.202				
PROP.N.BLOCKED		0.410				0.216				
COURSE.COMPLETED			0.161							
N.CREDITS.OPTIONAL.DIF				0.248						
PROP.N.FAIL				0.215				0.213	0.206	
MEAN.APPR					0.157					
N.CREDITS.DIF					0.217					
N.CREDITS.APPR.DIF							0.139		0.264	0.230
N.SUBJ.DIF							0.087			
SEM.MEAN								0.201		
PROP.N.APPR								0.234		

Computer Science - Information Gain										
Attributes	1	2	3	4	5	6	7	8	9	10
COURSE.COMPLETED	0.353	0.319	0.221			0.087				
PROP.N.APPR	0.180	0.154	0.167	0.082				0.047		
PROP.N.FAIL	0.182	0.132	0.148						0.046	
N.APPR				0.090				0.043		
N.CREDITS.TOTAL				0.067						
GPA					0.069					
N.CREDITS.APPR					0.065	0.087				
N.CREDITS.APPR.DIF					0.078		0.052		0.093	0.095
PROP.N.BLOCKED						0.081				
N.SUBJ.DIF							0.055			
SEM.MEAN								0.049		
PROP.N.ABFL									0.061	

Pharmacy - Information Gain										
Attributes	1	2	3	4	5	6	7	8	9	10
N.CREDITS.TOTAL	0.498	0.156	0.171			0.152			0.364	
N.CREDITS.APPR	0.426	0.226	0.172	0.200	0.201	0.202	0.205	0.180		
N.SUBJ.REQUIRED	0.499									
COURSE.COMPLETED		0.226	0.172	0.200	0.201	0.202	0.205	0.180		
N.CREDITS.REQUIRED							0.140	0.108		
N.APPR									0.350	
N.SUBJ.TOTAL									0.349	

5.3.2 *Class Imbalance.* As already mentioned, since the number of drop outs in the scope of a course semester is much lower then all the courses of that semester, we had too many course/semester pairs with few, or even zero, drop outs. This high level of imbalance may severely bias the classifier even using class imbalance techniques. Thus, for this scenario, we created a classification model only when at least 10 drop outs occurred. Figure 3 shows the absolute numbers (and percentages) of drop outs across the semesters for the courses of Pharmacy and Computer Science.

Similarly to the Global model, we applied the One-Sided Selection algorithm followed by random undersampling/oversample.

6. EVALUATION

In this section we first present the results of the two approaches presented in the previous section considering different learning configurations. After that, we chose the best setup of each approach and compare their performances on the test set.

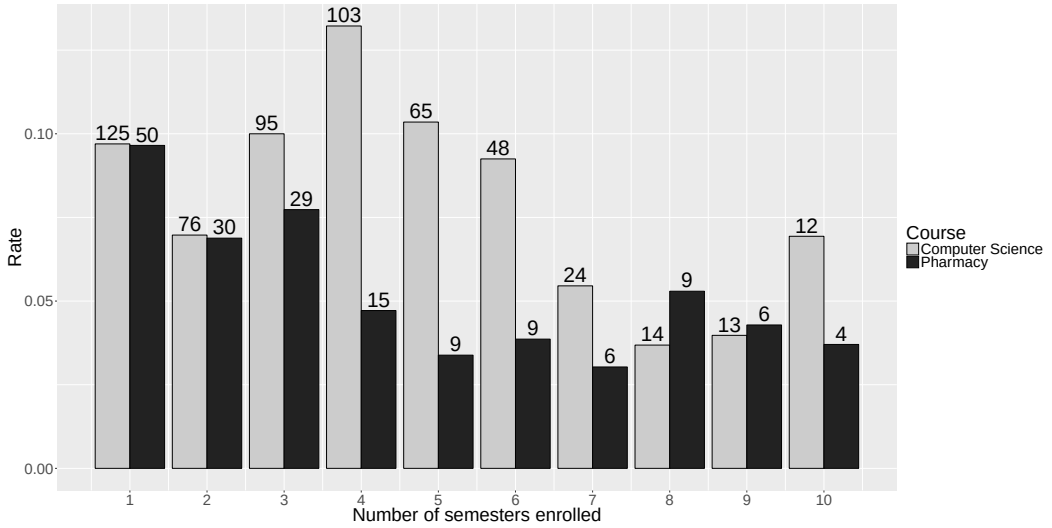

Fig. 3. Drop outs by semester enrolled.

6.1 Global Model Selection

Figure 4 depicts the results considering the F-measure's mean of all semesters for each configuration - the color of the bar indicates the feature selection and balancing algorithms used and the x-axis indicates the number of trees of the random forest. To get a better visualization, we are only showing the configuration factors that yielded the best results.

The best results are achieved when the gain ratio algorithm is used in any configuration. The main difference between feature selection using the gain ratio and the other two algorithms is the importance given to the STATUS.SEM feature, which makes a lot of sense if we consider that a student failing all subjects in a semester is a potential candidate for dropping out. It is also possible to notice that

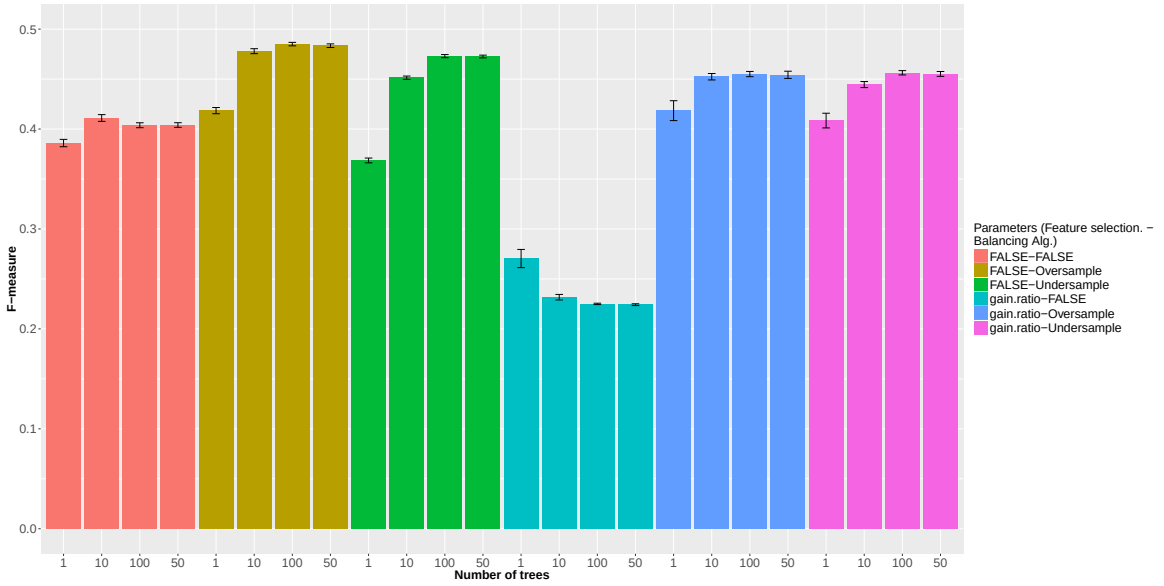

Fig. 4. F-Measures of Global Model's configurations.

the configuration described as FALSE-Oversample-100 - which means: no feature selection, balanced using OSS and random oversample and 100 decision trees in random forest - had the highest value of F-measure with a slight difference to the model FALSE-Oversample-50 and is the configuration chosen to represent the Global Model.

Some other conclusions worth pointing out are:

—The best results are achieved when we do not make feature selection. It is also notable that balancing the training set has had the expected effect and contributed into improve the results;

—F-measure's rate gets higher as the number of trees in random forest are increased, but forests with 50 and 100 trees usually have no statistical difference.

6.2 Specific Model Selection

We also followed the same approach used for the Global model here and the results are depicted in Figure 5. Notice that the best models, with approximately equal performance, are: FALSE-Oversample-100 and FALSE-Oversample-50. Since the configuration FALSE-Oversample-100 was chosen as the best for the Global Model, we chose this configuration as the best for the Specific Model.

Additionally, some other conclusions can be made based on Figure 5 as follows:

—All specific model configurations that used feature selection have a lower value than the similar configurations of the Global Model. A possible reason is that the number of instances extracted from each course/semester pairs is much less than considering only the semester as the Global model does;

—Balancing the training set did not have the expected effect of rising F-measure's value. This effect can be noticed independently of using feature selection and might also be the result of the fewer number of instances.

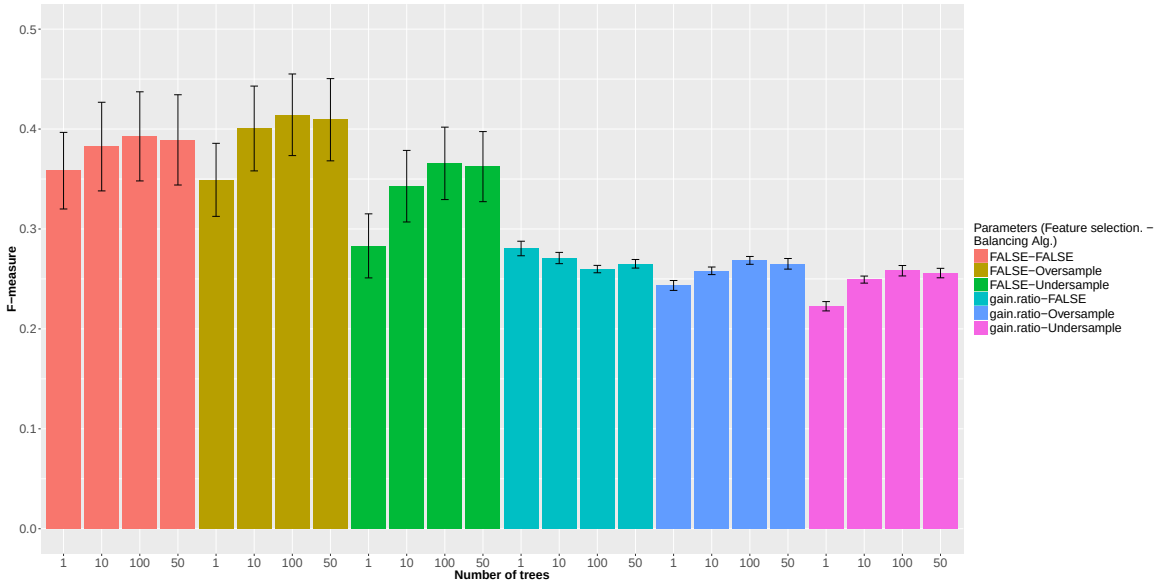

Fig. 5. F-Measures of Specific Model's configurations.

Table VII. Classification results per semesters.

Global Model					
Nº of semesters	Accuracy	F-Measure	Recall	Precision	Kappa
1	0.951	0.881	0.866	0.897	0.850
2	0.946	0.707	0.635	0.797	0.678
3	0.939	0.595	0.561	0.632	0.562
4	0.913	0.468	0.336	0.768	0.428
5	0.934	0.514	0.388	0.760	0.483
6	0.927	0.373	0.281	0.553	0.339
7	0.956	0.269	0.212	0.368	0.248
8	0.954	0.333	0.225	0.636	0.315
9	0.955	0.157	0.085	1.000	0.151
10	0.962	0.157	0.172	0.555	0.249
Mean	**0.944**	**0.456**	**0.376**	**0.697**	**0.430**

Specific Model					
Nº of semesters	Accuracy	F-Measure	Recall	Precision	Kappa
1	0.932	0.800	0.810	0.824	0.759
2	0.926	0.570	0.579	0.624	0.531
3	0.892	0.408	0.454	0.485	0.369
4	0.824	0.273	0.233	0.409	0.196
5	0.900	0.473	0.510	0.607	0.433
6	0.860	0.239	0.237	0.312	0.188
7	0.845	0.100	0.100	0.100	0.018
8	0.886	0.264	0.306	0.291	0.215
9	0.943	0.000	0.000	0.000	-0.016
10	0.968	0.000	0.000	0.000	0.000
Mean	0.898	0.313	0.323	0.365	0.269

6.3 Global Model vs Specific Model

To compare both Specific and Global models, we used the precision, recall, F-measure, accuracy and the kappa metrics. Table VII shows the results for the configurations that achieved the best results in each case. From the results we can observe that:

—The Global Model usually achieves superior results, for all used metrics;
—All metrics, except accuracy, tend to decrease as the semesters are increasing for both models. This is a consequence of the fact that the higher the semester, less drop outs are observed. Thus the problem gets harder for higher semesters;
—Cohen's Kappa indicates, especially for the Global Model, that the results were not achieved by chance. In fact, only the Specific Model, starting by the sixth semester, behaves like a random classifier. The closer to 0 is the value of Kappa, the more likely to a random model the classifier behaves;

7. CONCLUSIONS AND FUTURE WORK

In this article we cast the students' drop out problem as a classification problem. We applied this study in a public Brazilian federal university and evaluated several configurations of a state-of-art classifier, approaching the drop out problem through two perspectives: 1) building one classification model per semester, and 2) building one classification model per course/semester. We used feature selection and class imbalance techniques for each perspective and selected the best configuration for comparing them.

From this work, we can draw the following important conclusions:

—Features extracted from academic records alone carry a strong signal about drop out occurrence;

—The percentage of course completed by the student in the semester is an important factor to predict drop out, which leads us to the hypothesis that the student considers the effort and the time spent in the course before dropping out;

—Feature importance varies according to the target course and semester, and mainly in initial semesters a small number of them is sufficient for achieving good results;

—The global model achieves better results than a Specific model for any semester, which can be seen as a positive discovery since instead of building 760 models - 76 courses times 10 semesters - we only need to build 10 models - a classifier for each semester.

—Even using balancing techniques, some semesters have a poor recall and precision rate, which indicate that our attributes are not sufficient to predict drop outs and some others, as well as alternative approaches, might have to be considered.

As future work, we intend to extend this approach to consider socioeconomic data of the students. Our hypothesis is that the set of important factors on students' drop out identification in later semesters is mostly social, thus more difficult to be mapped in the data currently available to us. We also intend to deploy this model in the Academic Management System of UFCG in order to help administrators, professors and students identify and prevent drop out.

REFERENCES

Adachi, A. A. C. T. *Evasão e Evadidos nos Cursos de Graduação da Universidade Federal de Minas Gerais*. M.S. thesis, Programa de Pós-Graduação em Educação. Universidade Federal de Minas Gerais. Belo Horizonte, 2009.

Andriola, W. B., Andriola, C. G., and Moura, C. P. Opiniões de docentes e de coordenadores acerca do fenômeno da evasão discente dos cursos de graduação da universidade federal do ceará (UFC). *Ensaio: Avaliação e Políticas Públicas em Educação* 14 (52): 365–382, 2006.

Balaniuk, R., do Prado, H. A., da Veiga Guadagnin, R., Ferneda, E., and Cobbe, P. R. Predicting evasion candidates in higher education institutions. In *Proc. of the First International Conference on Model and Data Engineering*. Springer-Verlag, pp. 143–151, 2011.

Barroso, M. F. and Falcão, E. B. Evasão universitária: O caso do Instituto de Física da UFRJ. *IX Encontro Nacional de Pesquisa em Ensino de Física* vol. 9, pp. 1–14, 2004.

Breiman, L. Random forests. *Machine Learning* 45 (1): 5–32, 2001.

Dekker, G., Pechenizkiy, M., and Vleeshouwers, J. Predicting students drop out: A case study. In *Proc. of the 2nd Int. Conference on Educational Data Mining*. pp. 41–50, 2009.

Gaioso, N. P. d. L. *A evasão discente na Educação Superior no Brasil: na perspectiva de alunos e dirigentes*. M.S. thesis, Universidade Católica de Brasília, Brasília-DF, 2005.

Hart, P. E. The condensed nearest neighbor rule (Corresp.). *IEEE Transactions on Information Theory* 14 (3): 515–516, 1968.

He, H. and Garcia, E. A. Learning from imbalanced data. *Transactions on Knowledge and Data Engineering* 21 (9): 1263–1284, 2009.

INEP. Ensino superior mantém tendência de crescimento e diversificação. `http://portal.inep.gov.br/visualizar/-/asset_publisher/6AhJ/content/ensino-superior-mantem-tendencia-de-crescimento-e-diversificacao`, 2010.

INEP. Acesso e permanência no ensino superior. `http://portal.mec.gov.br/index.php?option=com_docman&view=download&alias=17199-cne-forum-educacao-superior-2015-apresentacao-10-jose-soares&Itemid=30192`, 2013a.

INEP. Censo da educação superior 2013. `http://download.inep.gov.br/educacao_superior/censo_superior/apresentacao/2014/coletiva_censo_superior_2013.pdf`, 2013b.

Kubat, M., Matwin, S., et al. Addressing the curse of imbalanced training sets: one-sided selection. In *Proc. of the Fourteenth Int. Conference on Machine Learning*. Vol. 97. pp. 179–186, 1997.

Manhães, L. M. B., da Cruz, S. M. S., and Zimbrão, G. Evaluating performance and dropouts of undergraduates using educational data mining. In *Proc. of the Twenty-Ninth Symposium on Applied Computing*, 2014.

Mann, H. B. and Whitney, D. R. On a test of whether one of two random variables is stochastically larger than the other. *Annals of Mathematical Statistics* 18 (1): 50–60, 1947.

Moretti, A., Gonzalez-Brenes, J., McKnight, K., and Salleb-Aouissi, A. Mining student ratings and course contents for computer science curriculum decisions, 2014.

Mustafa, M. N., Chowdhury, L., and Kamal, M. S. Students dropout prediction for intelligent system from tertiary level in developing country. In *IEEE Int. Conference on Informatics, Electronics & Vision*. pp. 113–118, 2012.

Márquez-Vera, C., Cano, A., Romero, C., and Ventura, S. Predicting student failure at school using genetic programming and different data mining approaches with high dimensional and imbalanced data. *Applied Intelligence* 38 (3): 315–330, 2013.

Pal, S. Mining educational data to reduce dropout rates of engineering students. *Int. Journal of Information Engineering and Electronic Business* 4 (2): 1–7, 2012.

Romero, C. and Ventura, S. Data mining in education. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 3 (1): 12–27, 2013.

Sales, A., Balby, L., and Cajueiro, A. Predicting student dropout: A case study in brazilian higher education. In *Proc. of the 3rd Symposium on Knowledge Discovery, Mining and Learning*, 2015.

Tomek, I. Two Modifications of CNN. *IEEE Transactions on Systems, Man, and Cybernetics* SMC-6 (11): 769–772, 1976.

Yadin, A. Reducing the dropout rate in an introductory programming course. *ACM Inroads* 2 (4): 71–76, 2011.

Zender, R., Metzler, R., and Lucke, U. Freshup—a pervasive educational game for freshmen. *Pervasive and Mobile Computing* 14 (C): 47–56, 2014.