Aprendendo a Identificar Estudantes em Risco em Educação a Distância Usando Contagem de Interações

Learning to Identify At-Risk Students in Distance Education Using Interaction Counts

Douglas Detoni 1, 2
Cristian Cechinel 3, 4
Daniela Brauner 3, 5
Ricardo M. Araujo 1, 6

Data de submissão: 15/02/2016, Data de aceite: 20/06/2016

Resumo: Evasão de estudantes é um dos principais problemas em cursos de educação a distância. Um dos desafios é desenvolver métodos para prever o comportamento de estudantes, de forma que professores e tutores possam identificar estudantes em risco de evasão tão cedo quanto possível e prover assistência antes que a evasão ocorra ou o estudante reprove. Modelos de Aprendizado de Máquina tem sido utilizado para prever e classificar estudantes nesses cenários. No entanto, enquanto estes modelos mostram resultados promissores em alguns casos, usualmente utilizam atributos que torna difícil a transferência para outros cursos e plataformas. Neste artigo, provemos uma metodologia para classificar estudantes utilizando apenas contagem de interações de cada estudante. Avaliamos esta metodologia utilizando um conjunto de dados de dois cursos baseados na plataforma Moodle. Executamos experimentos que consistem em treinar e avaliar três modelos de aprendizado de máquina (Máquina de Vetor de Suporte, Bayes Ingênuo e Adaboost com árvores de decisão) em diferentes cenários. Provemos evidências que padrões contidos nas interações podem prover informações úteis para classificar estudantes em risco. Esta classificação permite a personalização de atividades apresentadas a estes estudantes (de forma automática ou através de tutores) como forma de tentar evitar a evasão.

Palavras-chave: predição de evasão, ambiente virtual de aprendizagem, aprendizado de máquina, analíticas de aprendizagem, mineração de dados educacionais

1Centro de Desenvolvimento Tecnológico, Universidade Federal de Pelotas - Pelotas, RS, Brasil
2{ddetoni@inf.ufpel.edu.br}
3Faculdade de Educação, Universidade Federal de Pelotas - Pelotas, RS, Brasil
4{contato@cristiancechinel.pro.br}
5{daniela.brauner@ufrgs.br}
6{ricardo@inf.ufpel.edu.br}
Abstract: Student dropout is one of the main problems faced by distance learning courses. One of the major challenges for researchers is to develop methods to predict the behavior of students so that teachers and tutors are able to identify at-risk students as early as possible and provide assistance before they drop out or fail in their courses. Machine Learning models have been used to predict or classify students in these settings. However, while these models have shown promising results in several settings, they usually attain these results using attributes that are not immediately transferable to other courses or platforms. In this paper, we provide a methodology to classify students using only interaction counts from each student. We evaluate this methodology on a data set from two majors based on the Moodle platform. We run experiments consisting of training and evaluating three machine learning models (Support Vector Machines, Naive Bayes and Adaboost decision trees) under different scenarios. We provide evidences that patterns from interaction counts can provide useful information for classifying at-risk students. This classification allows the customization of the activities presented to at-risk students (automatically or through tutors) as an attempt to avoid students drop out.

Keywords: dropout prediction, virtual learning environment, machine learning, learning analytics, educational data mining

1 Introduction

Student dropout is one of the main problems faced by distance learning courses. A recent report of Distance Learning in Brazil published by the Brazilian Association for Distance Learning (ABED) has shown that at the same time the number of enrollments significantly increased in the country in the last years, dropout rates also increased. According to ABED, the average dropout rates in the regulated full distance courses of Brazilian institutions who participated in the survey was 19.06%. The main dropout reasons observed in the study were the lack of time to study and to attend the courses, the accumulation of work activities, and the lack of adaptation to the methodology.

In this context, one of the major challenges for researchers is to develop methods to predict the behavior of students so that teachers, tutors and other stakeholders are able to identify at-risk students (i.e. students risking dropping out of the course) as early as possible and provide assistance before they drop out or fail in their courses. Other challenges in the field also involve predicting students performance and automatically classifying them into different groups according to their profiles so that personalized activities can be delivered to them.

One of the main advantages of distance learning courses is the large amount of data generated by the interactions in the Virtual Learning Environments (VLE), which opens up new
possibilities to study and understand these interactions [6, 12]. For that, Machine Learning algorithms have been used to build successful classifiers using diverse students’ attributes. However, previous works typically use very specific attributes that often are not generalizable to other courses or environments. In this context, we propose to exploit solely students’ interaction counts over time in order to predict at-risk students. We hope our approach provides a more generalizable framework that is independent of platform or particularities of how courses are being conducted.

Our main goal in this paper is to understand the usefulness of interaction counts to identify at-risk students using trained classifiers. Our specific goals in this paper are (i) to understand the performance differences between some commonly-used models and training algorithms and (ii) to understand the role of different sources of training data.

By training three classifiers on data set extracted from two different courses, we show that the approach of only using interaction counts can be useful to classify at-risk students. We show that trained models can generalise to different students and even different course and that the tested classifiers perform well, albeit with different trade-offs.

This paper is organized as follows. Section 2 summarizes related work. Section 3 introduces the methodology used, including the dataset and classifiers. The results of the experiments are presented on Section 4. Finally, Section 5 concludes the paper and points out future research venues.

2 Related Work

The technology of distance education allows one to measure engagement of students [3] by looking into system logs and evaluating the intensity of students’ interactions in the different activities available inside virtual classrooms. The notion of interaction with Learning Management Systems (LMS) is mentioned in several works related to distance education; and learning effectiveness and efficiency has often been associated to different measures of students’ interactions inside LMS [10] that are normally highly correlated to their success in the courses. For instance, [13] observed that students who had the highest rates of access to the contents of a distance course also received the highest grades, and [4] found that the total number of clicks made by students is strongly correlated with the student’s final grade in a course. It is also already established in the literature of Learning Analytics and Educational Data Mining that students’ success in distance courses is directly correlated to their engagement inside LMS and several works were already developed to early predict academic performance and at-risk students.

These plethora of researches differ in many forms, from the automated techniques used to train and test the models for prediction (neural networks, logistic regression, dis-
Aprendendo a Identificar Estudantes em Risco em Educação a Distância Usando Contagem de Interacções

criminant analysis, decision trees, support vector machines), to the source (different LMS, academic systems, exams, registration forms) and amount of data, and the combination of attributes used in these models (students interactions, gender, age, ethnicity, working experience, educational level, home computer availability). A more extensive review of the related literature can be found in [15], but here we mention a few recent works to briefly illustrate.

In [10], the results of three different machine learning techniques (feed-forward neural networks, support vector machines, and probabilistic ensemble simplified fuzzy ARTMAP) were combined to predict student dropout on two introductory e-learning courses of the University of Athens. For that, the authors used time-invariant attributes (such as gender, residency, working experience, educational level) and time-varying attributes that depict the student progress along the courses (multiple choice test grades, project grades, project submission dates, section activities, number of posts in the course forums) and were able to achieve 85% of overall student classification rate at the beginning of the two courses. Another example is the work of [7] who tested and trained models for detecting at-risk students by using Weka implementation [5] of four different techniques (logistic regression, support vector machines using sequential minimal optimization, J48 decision trees, and Naive Bayes). The authors included in the models a series of attributes, such as demographic information about the students (age, gender), information about their previous and current performances (verbal and math SAT scores, partial grades) and current academic situation (full-time or part-time student, freshman or senior), and information extracted from the students interactions with the LMS (number of sessions opened, number of discussion forum threads read and contributed by them, number of assessments submitted). The models were able to detect at-risk students with high Recalls (higher than 80%) and low False Positive rates (less than 17%). In this study, Naive Bayes models presented the best overall results.

At last, one can also mention the work of [11] who identified a total of 13 variables simply correlated to students grade inside a virtual course and selected three of them to generate a multiple linear regression model able to correctly predict at-risk students with 73.7% of overall precision. The authors highlighted that not all variables inside the LMS are useful predictors, and that the predictive utility of many variables is dependent upon course design and pedagogical goals.

This last statement is critical to understand the huge variation among the existing solutions and the difficulties to generalize them to different educational scenarios. As stated by [2], there are different kinds of interactions that take place in an online virtual environment and “there is not yet solid framework available when it comes to decide which specific data must be analyzed”. The approach of the present study differs from previous ones as we are only using the counting of interactions inside the LMS to test and train models for predicting at-risk students. The main idea here is to evaluate a data-driven approach more independent of platform or particularities of how courses are being conducted, that could be further applied
to other distance learning scenarios.

3 Methodology

In order to tackle our goals, we use a data set extracted from the institutional Moodle platform of the Federal University of Pelotas in Brazil, to train classifiers under different conditions and scenarios. Our basic workflow consists of: (i) extract the relevant features (interaction counts) from the data set; (ii) train different classifiers on the extracted data, using different scenarios and conditions and (iii) evaluate and contextualize the results. In what follows, we detail each step.

3.1 Data set

The experiments presented in this paper are based on logs of the execution of two majors taking place in a distance education setting. These logs register every interaction of students, teachers and teacher assistants throughout the execution of each course composing the majors. Interaction types vary across courses and majors and the most common found in our data set are: log in, forum view, course view, resource view, assignment view, forum add post, chat talk, chat view.

The data set is composed of two majors, named here A and B. A major in its totality is composed of several periods, each with activities spread throughout 7 weeks. Each period is further divided into a few courses that students take simultaneously and non-optionally. The majority of activities take place on line using the Moodle platform, but a few, such as evaluations, are taken in a physical location. Each course is executed by a teacher and several teaching assistants. For a student to pass to the next period she must pass all courses in that period. If she fails on a single course, then she fails the whole period.

The complete data set is divided in four subsets (cases), each containing information from a single period of a single major. This is depicted in Table 1, where cases are denoted by major and period – e.g. A1 denotes the first period of major A. The second period of a major is composed of the same students from the first period, but now on a different set of courses. Students between majors are completely different.

3.2 Feature Extraction

For each listed case, we extract one example per student. An example is a tuple \( < \vec{a}_s, l_s > \) where \( l_s \) is the label indicating whether student \( s \) passed or failed the period and \( \vec{a}_s \) is the attribute vector for the same student. Attributes are divided into two groups, \( \vec{c}_s \) and \( \vec{m}_s \), such that \( \vec{c}_s = w_{1,s}, w_{2,s}, \ldots, w_{n,s} \) contains interaction counts for all \( n \) weeks for
Tabela 1. Example sets used for training and testing.

<table>
<thead>
<tr>
<th>Case</th>
<th>Major</th>
<th>Period</th>
<th># Students</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1</td>
<td>A</td>
<td>1st</td>
<td>133</td>
</tr>
<tr>
<td>A2</td>
<td>A</td>
<td>2nd</td>
<td>101</td>
</tr>
<tr>
<td>B1</td>
<td>B</td>
<td>1st</td>
<td>183</td>
</tr>
<tr>
<td>B2</td>
<td>B</td>
<td>2nd</td>
<td>161</td>
</tr>
</tbody>
</table>

student \( s \) and \( \vec{m}_s = m_{1,s}, m_{2,s}, \ldots, m_{n,s} \) is the standardized count defined as:

\[
m_{i,s} = \frac{w_{i,s} - \mu_i}{\sigma_i}
\]

where \( w_{i,s} \) is the interaction count for week \( i \) and student \( s \), \( \mu_i \) and \( \sigma_i \) are, respectively, the mean and standard deviation of all interaction counts for week \( i \) in the subset. We count all interactions from students, without any consideration on interaction type.

In this paper, we set \( n = 3 \), about half the available weeks, as we are interested in verifying the possibility of an early detection of at-risk students – i.e. classify students after three full weeks since the beginning of a period. We however consider different values of \( n \) in Section 4.6. We removed from the data set trivial cases consisting of students that drop out of the course (hence leading to zero interactions in several weeks). One major challenge in this data set is that there are not usually many failed students – it is far more common for students to drop out at the beginning – providing few positive examples to work with.

3.3 Classifiers

We consider three classifiers commonly used in the literature: Support Vector Machines (SVM), Naive Bayes and Decision Trees (using Adaboost). We used commonly used parameters for the models and training algorithms. SVM was trained using the SMO algorithm [14], set with \( C = 10 \) and a polynomial kernel with exponent \( e = 2 \). Adaboost used one-level decision trees (decision stumps) and 10 boost iterations. Naive Bayes used a priori probabilities set to equal probabilities between classes.

Except where noted, before training the models we balance the classes using sampling with replacement [9], so that the models are not excessively biased towards the majority class. We found that doing so improves performance considerably on average, in particular for SVM. The test sets are never resampled.
Classification of Students at Risk in Distance Education Using Interaction Counting

Table 2. False Positives and False Negatives for each classifier trained and tested in different periods of the same majors.

<table>
<thead>
<tr>
<th></th>
<th>SVM</th>
<th>adaboost</th>
<th>bayes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case</td>
<td>FP</td>
<td>FN</td>
<td>FP</td>
</tr>
<tr>
<td>A1</td>
<td>0.05</td>
<td>0.47</td>
<td>0.05</td>
</tr>
<tr>
<td>A2</td>
<td>0.05</td>
<td>0.67</td>
<td>0.06</td>
</tr>
<tr>
<td>B1</td>
<td>0.01</td>
<td>0.78</td>
<td>0.03</td>
</tr>
<tr>
<td>B2</td>
<td>0.05</td>
<td>1.00</td>
<td>0.03</td>
</tr>
<tr>
<td>μ</td>
<td>0.04</td>
<td>0.74</td>
<td>0.04</td>
</tr>
<tr>
<td>σ</td>
<td>0.02</td>
<td>0.19</td>
<td>0.01</td>
</tr>
</tbody>
</table>

4 Results

4.1 Same Period

In this experiment, we are interested in whether patterns generated by a group of students within the same period of the same course are generalizable for different students. This case is of limited interest in a real-world scenario, as both groups would be taking the course simultaneously, but it provides a basic verification of if the methodology is applicable - i.e. if each student has an unique pattern, then it may not be possible to use this information to classify different students.

For each course and each period, we performed a 10-fold cross-validation \[8\] where the examples are randomly partitioned in 10 sets, the models are trained over 9 of them, tested on the remaining one and the process is repeated each time with a different test set. In this case we do not use resampling to perform class balancing, since we must guarantee that students appearing in the test set never appear in the training sets. We report on the averages over the 10 runs.

Table 2 shows measured False Positives (FP) and False Negatives (FN) for each algorithm. Adaboost and NaiveSVM provide the lowest FP rates, while Naive Bayes provides the lowest FN rate. Naive Bayes also shows the best balance between FP and FN rates, as further evidenced by Figure 1 where the average ROC Area Under the Curve (AUC) for each classifier is also shown.

4.2 Different Periods

A more useful approach in a real-world scenario is to train the model using all students from one period and then applying the trained model to the other one. The same students
(except those that failed the previous period) would be present in both periods, however different courses would be involved with potentially different dynamics.

In this experiment we use one period of one major for training while the other period from the same major is used for testing – for each test case $t_{i,m}$ (major $m$ and period $i$) we trained the models using the cases $t_{i_2 \neq i,m}$ (the other period of the same major).

Table 3 shows the results of the four resulting combinations. It is now possible to balance the classes in the training set, as the test set is completely disjoint, and we do so. We can observe that the number of FP when using SVM and Adaboost increase substantially when compared to the previous experiment, while FN rates are better. The exception is Naive Bayes, that manages to keep the same FP rate at the cost of a considerable increase in FN rate. In all cases standard deviations are higher, showing that results are less consistent across different cases. Figure 2 further shows that Naive Bayes again has the best overall performance.

These results show that classifying the same students in different courses from different periods is more difficult than classifying different students in the same courses and period. We take this as evidence that courses’ dynamics play an important role in the patterns generated by students - i.e. different students in the same course behave more similarly than the same student in different courses.

4.3 Different majors

Another possible approach is to train the model using data from a different major to use it in another. This potentially allows for more examples to be available for training, which is certainly true for our data set, but students and courses are completely different between
Tabela 3. False Positives (FP) and False Negatives (FN) for each classifier trained and tested in different periods of the same majors.

<table>
<thead>
<tr>
<th>Case</th>
<th>SVM FP</th>
<th>SVM FN</th>
<th>adaboost FP</th>
<th>adaboost FN</th>
<th>bayes FP</th>
<th>bayes FN</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1</td>
<td>0.46</td>
<td>0.26</td>
<td>0.17</td>
<td>0.16</td>
<td>0.18</td>
<td>0.10</td>
</tr>
<tr>
<td>A2</td>
<td>0.10</td>
<td>0.33</td>
<td>0.07</td>
<td>0.42</td>
<td>0.18</td>
<td>0.25</td>
</tr>
<tr>
<td>B1</td>
<td>0.11</td>
<td>0.83</td>
<td>0.05</td>
<td>0.83</td>
<td>0.08</td>
<td>0.67</td>
</tr>
<tr>
<td>B2</td>
<td>0.29</td>
<td>0.40</td>
<td>0.57</td>
<td>0.00</td>
<td>0.46</td>
<td>0.00</td>
</tr>
<tr>
<td>μ</td>
<td>0.24</td>
<td>0.49</td>
<td>0.22</td>
<td>0.38</td>
<td>0.22</td>
<td>0.28</td>
</tr>
<tr>
<td>σ</td>
<td>0.15</td>
<td>0.22</td>
<td>0.21</td>
<td>0.32</td>
<td>0.14</td>
<td>0.25</td>
</tr>
</tbody>
</table>

Figura 2. Average False Positives, False Negatives and ROC Area for each classifier trained and tested in different periods of the same majors.

training and testing. Our count-only approach is most useful in this scenario, as different majors are likely to use different features from the platform or even different platforms entirely.

To test this scenario, for each test case $t_{i,m}$ we trained models using each period of the other major: $t_{i,m_2 \neq m}$. We report for each training set the average results of using the two training sets (Table 4). We can observe that false positives are slightly lower and false negatives slightly higher, but overall results are largely similar to the previous scenario. Classifiers’ performances are now closer to each other (Figure 3), although Naive Bayes still provides the best overall performance and balance between FP and FN rates. These results are evidence that students’ patterns are as useful across majors as they are within the same major.
Aprendendo a Identificar Estudantes em Risco em Educação a Distância Usando Contagem de Interações

Tabela 4. Resultados de treinamento em exemplos de um período de um major e teste em outro.

<table>
<thead>
<tr>
<th>Case</th>
<th>SVM FP</th>
<th>SVM FN</th>
<th>adaboost FP</th>
<th>adaboost FN</th>
<th>bayes FP</th>
<th>bayes FN</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1</td>
<td>0.32</td>
<td>0.34</td>
<td>0.16</td>
<td>0.53</td>
<td>0.10</td>
<td>0.45</td>
</tr>
<tr>
<td>A2</td>
<td>0.22</td>
<td>0.42</td>
<td>0.11</td>
<td>0.67</td>
<td>0.13</td>
<td>0.54</td>
</tr>
<tr>
<td>B1</td>
<td>0.18</td>
<td>0.44</td>
<td>0.13</td>
<td>0.33</td>
<td>0.27</td>
<td>0.25</td>
</tr>
<tr>
<td>B2</td>
<td>0.22</td>
<td>0.20</td>
<td>0.30</td>
<td>0.20</td>
<td>0.43</td>
<td>0.10</td>
</tr>
<tr>
<td>μ</td>
<td>0.23</td>
<td>0.35</td>
<td>0.18</td>
<td>0.41</td>
<td>0.24</td>
<td>0.32</td>
</tr>
<tr>
<td>σ</td>
<td>0.17</td>
<td>0.15</td>
<td>0.14</td>
<td>0.31</td>
<td>0.16</td>
<td>0.34</td>
</tr>
</tbody>
</table>

Figura 3. Médias de Falsos Positivos, Falsos Negativos e Área ROC para cada classificador treinado e testado em diferentes majors.

4.4 Using More Data

Em seções anteriores, apenas dados de um treinamento de um período e major foram usados. Isso permitiu uma melhor comparação entre diferentes cenários. Agora, voltamos nossa atenção para usar todo o possível dados para treinamento, de forma que possamos obter mais insight sobre o efeito de disponibilizar mais dados disponíveis aos algoritmos de aprendizado.

Comparamos o treinamento do período de um major, e teste em cada período do outro major - isto é, usando todos os dados disponíveis de um major para classificação em outro major. Tabela 5 mostra os resultados. Podemos observar que todos os classificadores beneficiam do aumento de exemplos, particularmente os exemplos positivos adicionais, levando a uma redução significativa nos falsos negativos.

Para Naive Bayes e SVM, isso reduz o desempenho em termos de taxa de falsos positivos; entretanto, Adaboost é capaz de reduzir tanto Rates e agora exibe um AUC muito próximo de Naive.

Rev. Inform. Teor. Apl. (Online) • Porto Alegre • v. 23 • n. 2 • p. 124-140 • novembro/2016
Tabela 5. Results of training on examples from all periods of one major and testing on another.

<table>
<thead>
<tr>
<th>Case</th>
<th>SVM</th>
<th>adaboost</th>
<th>bayes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FP</td>
<td>FN</td>
<td>FP</td>
</tr>
<tr>
<td>A1</td>
<td>0.83</td>
<td>0.26</td>
<td>0.02</td>
</tr>
<tr>
<td>A2</td>
<td>0.44</td>
<td>0.17</td>
<td>0.15</td>
</tr>
<tr>
<td>B1</td>
<td>0.14</td>
<td>0.28</td>
<td>0.25</td>
</tr>
<tr>
<td>B2</td>
<td>0.40</td>
<td>0.00</td>
<td>0.15</td>
</tr>
<tr>
<td>μ</td>
<td>0.43</td>
<td>0.18</td>
<td>0.15</td>
</tr>
<tr>
<td>σ</td>
<td>0.25</td>
<td>0.11</td>
<td>0.08</td>
</tr>
</tbody>
</table>

Figura 4. Average False Positives, False Negatives and ROC Area for each classifier trained and tested in different majors.

Bayes’ (Figure 4). Indeed, each can be seen as providing a different trade-off between FP and FN, with Naive Bayes providing fewer FN and Adaboost fewer FP.

A final scenario further extends the previous one by adding one period of the same major that is being tested - i.e. for each test set, we use all remaining periods from all majors to compose a training set. This addition mostly affects the SVM, which has its FP rates greatly reduced, but at expense of a small increase in FN rates. Adaboost and Naive Bayes show almost no changes.

4.5 Experiment Comparison

In order to better compare performance across all executed experiments, Figure 6 shows a comparison of the different metrics for each experiment, using the Naive Bayes classifier (which provided the best overall results). We can observe that classifying using
Aprendendo a Identificar Estudantes em Risco em Educação a Distância Usando Contagem de Interações

**Tabela 6.** Resultados de separação de um caso único e treinamento em todos os outros casos.

<table>
<thead>
<tr>
<th>Caso</th>
<th>SVM</th>
<th>adaboost</th>
<th>bayes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FP</td>
<td>FN</td>
<td>FP</td>
</tr>
<tr>
<td>A1</td>
<td>0.18</td>
<td>0.10</td>
<td>0.05</td>
</tr>
<tr>
<td>A2</td>
<td>0.04</td>
<td>0.50</td>
<td>0.17</td>
</tr>
<tr>
<td>B1</td>
<td>0.06</td>
<td>0.61</td>
<td>0.15</td>
</tr>
<tr>
<td>B2</td>
<td>0.54</td>
<td>0.00</td>
<td>0.34</td>
</tr>
<tr>
<td>μ</td>
<td>0.21</td>
<td>0.31</td>
<td>0.18</td>
</tr>
<tr>
<td>σ</td>
<td>0.20</td>
<td>0.26</td>
<td>0.10</td>
</tr>
</tbody>
</table>

**Figura 5.** Média de False Positivos, False Negativos eÁrea do ROC para cada caso quando separando um caso único e treinando em todos os outros casos.

data de um major diferente, quando comparado com usar dados de um período diferente do mesmo major, produz resultados de AUC semelhantes, mas com taxas maiores de FP e FN.

As expectativas, todos os experimentos não fornecem classificações perfeitas, mas todos os resultados são melhores do que se esperaria com escolha aleatória.

4.6 **Número de Semanas**

Em este trabalho, optamos por realizar classificações dos estudantes após três semanas, por meio da metade de um período. Em alguns casos, pode ser desejável realizar esta classificação mais cedo ou mais tarde em um curso. Nesta seção, experimentamos com mudanças no número de semanas disponíveis ao classificador.

Para realizar o experimento e permitir um grande conjunto de treinamento, focamos no caso em que usamos um único período como conjunto de teste e treinamos os classificadores usando todos os períodos restantes. Relatamos...
on the average over all test sets. We set a maximum of 5 weeks due to unavailability of data beyond this number of weeks for a few courses. Figure 6, 8 and 9 plot the measurements of FP, FN and AUC, respectively, for each classifier and number of weeks. The first thing to notice is that SVM is highly volatile, changing abruptly FP and FN rates. This is in accord with the previous results, where SVM consistently underperformed the other classifiers with the chosen settings.

As shown in Figure 7, FP rates for both Naive Bayes and Adaboost are monotonically decreasing over the tested range. However, as seen in Figure 8, FN rates show mixed results, with Naive Bayes FN rates remaining constant across the range. AUC (Figure 9) reflects these differences, peaking around four weeks and then slightly decaying. One likely reason for the observed decaying in performance with more weeks is the very different dynamics of each course in the data set, with some ending earlier than others, adding to the noise in the data. Nonetheless, it is possible to observe that performance largely and quickly saturates when more weeks are made available, suggesting that limiting the number of weeks available, as we have done in previous sections, does not strongly impact performance.

These results, especially when considering AUC, indicate that using Naive Bayes and allowing four weeks worth of data would bring the best overall results. It may be noted again, however, that Adaboost and Naive Bayes are largely complementary – Adaboost provides lower False Positives, while Naive Bayes consistently provides lower False Negatives.

5 Conclusions

Our results show that using only interaction counts can provide useful information for classifying students as possible drop outs. While the absolute performance, as measured here
Aprendendo a Identificar Estudantes em Risco em Educação a Distância Usando Contagem de Interações

Figura 7. False Positives using different number of weeks.

Figura 8. False Negatives using different number of weeks.

by false positives and false negatives, are hardly perfect, they are much better than random choice and could be useful in a multi-level classification system.

When considering the different sources of training data for the models, we have found that the source of training data has an impact in the classifiers’ ability to properly generalize to new data. On average, training over data from the same students that will be classified, but in a different period, yields only slightly better results than training on a completely different set of students. When a larger set of examples was used, adding examples from the students being classified yielded almost no change in the observed metrics. This is evidence that patterns from interaction counts are useful beyond the scope they are generated, a fundamental issue if, as we had argued, interaction counts are to be useful across different educational settings.

The main contribution of this work was to provide a methodology and initial validation experiments on the task of at-risk student classification in distance education majors using...
Aprendendo a Identificar Estudantes em Risco em Educação a Distância Usando Contagem de Interações

![ROC Area Under the Curve (AUC) using different number of weeks.](image)

**Figura 9.** ROC Area Under the Curve (AUC) using different number of weeks.

only interaction counts. We are now working on greatly expanding the data set to include a wider and more diverse set of cases, including direct comparisons with previous techniques over the same data set. Another future venue of research comes from the complementarity observed between Adaboost and Naive Bayes, which indicates that the use of an ensemble may be beneficial.

The results showed the usage potential of interaction count data. So, as future work, would be interesting to use other kind of logs to validate the proposed technique in other contexts and platforms. Thereby, the solution would be considered general and platform independent.

**Contribuição dos autores:**
- Douglas Detoni: Coleta e limpeza de dados, aplicação da metodologia, geração dos resultados.
- Cristian Cechinel: Construção da metodologia, avaliação dos resultados.
- Ricardo M. Araujo: Construção da metodologia, avaliação dos resultados.
- Daniela Brauner: Revisão da metodologia e resultados.

**Referências**


