UNIVERSIDADE FEDERAL DO CEARÁ

CENTRO DE CIÊNCIAS

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

MESTRADO EM CIÊNCIA DA COMPUTAÇÃO

ARTUR MESQUITA BARBOSA

EXPLORING LEARNING ANALYTICS APPROACHES TO MINIMIZE UNDERGRADUATE EVASION

FORTALEZA

2017

ARTUR MESQUITA BARBOSA

EXPLORING LEARNING ANALYTICS APPROACHES TO MINIMIZE UNDERGRADUATE EVASION

Dissertação apresentada ao Curso de Mestrado em Ciência da Computação do Programa de Pós-Graduação em Ciência da Computação do Centro de Ciências da Universidade Federal do Ceará, como requisito parcial à obtenção do título de mestre em Ciência da Computação. Área de Concentração: Computação Gráfica

Orientadora: Emanuele Marques do Santos

Coorientador: João Paulo Pordeus Gomes

FORTALEZA

2017

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca Universitária
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

M543e Mesquita Barbosa, Artur.
Exploring learning analytics approaches to minimize undergraduate evasion / Artur Mesquita Barbosa. – 2017.
59 f. : il. color.

Dissertação (mestrado) – Universidade Federal do Ceará, Centro de Ciências, Programa de Pós-Graduação em Ciência da Computação, Fortaleza, 2017.
Orientação: Profa. Dra. Emanuele Marques dos Santos.
Coorientação: Prof. Dr. João Paulo Pordeus Gomes.

1. Learning Analytics. 2. College Evasion. 3. Machine Learning. 4. Data Visualization. I. Título.
CDD 005

ARTUR MESQUITA BARBOSA

EXPLORING LEARNING ANALYTICS APPROACHES TO MINIMIZE UNDERGRADUATE EVASION

Dissertação apresentada ao Curso de Mestrado em Ciência da Computação do Programa de Pós-Graduação em Ciência da Computação do Centro de Ciências da Universidade Federal do Ceará, como requisito parcial à obtenção do título de mestre em Ciência da Computação. Área de Concentração: Computação Gráfica

Aprovada em: 20/12/2017

BANCA EXAMINADORA

Emanuele Marques do Santos (Orientadora)
Universidade Federal do Ceará (UFC)

João Paulo Pordeus Gomes (Coorientador)
Universidade Federal do Ceará (UFC)

Prof. Dr. Creto Augusto Vidal
Universidade Federal do Ceará (UFC)

Prof. Dr. Luciano de Andrade Barbosa
Universidade Federal de Pernambuco (UFPE)

To everyone who loved and supported me.

ACKNOWLEDGMENTS

I cannot let this big moment in my life pass by without recognizing the support of family, friends and Professors.

First of all, I would like to thank my parents, Barbosa and Sheyla, and my sister Beatriz, for showing the great love they have for me and a huge support in a very difficult moment during this process;

All my family members, especially Ana Adail, Carlos Eduardo Barbosa, Geovana, Gabriela, José Carlos, Keyla, Luiza, Shirley and Rosy, who were great supporters for all my achievements in life;

My advisor, Professor Emanuele, for giving me this opportunity, for teaching me her great knowledge, essential for this process, and for her patient, supportive and friendly chat during our meetings;

My co-advisor, Professor João Paulo, for his support on this research;

All my fellow friends who were colleagues in college, especially Alexandre Sombra, André Luis, André Bastos, César Goersch, Dener Miranda, Diego Parente, Eduardo Rodrigues, Felipe Zschornack, Lucas Brock, Nilo Araujo and Pedro Igo Sousa, for always being there to laugh and to give support;

My friends who were colleagues in high school, especially Beatriz Arruda, Gabriel Pinheiro, Igor Brasil and Luísa Nakayama, for their company and great moments;

AIESEC, for teaching me some great values in life, and especially to Andreza Cardoso, Aurília Rodrigues, Débora Oliveira, João Pedro Viana, Letícia Feitosa, Manoel Freitas and Nathânia Ramos. It was a pleasure to meet all of them and have their company in the last semester. Also thanks for the opportunity in Sweden;

Finally, the Computer Science department and Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) for the financial support of this research.

“You miss 100% of the shots you don’t take.”

(Michael Scott)

RESUMO

Um dos maiores desafios enfrentados pelos educadores é a redução da evasão universitária em suas instituições. O objetivo principal das abordagens de *Learning Analytics* neste tópico costuma ser a classificação binária de estudantes em propensos a evadirem-se ou não. No entanto, isto não é suficiente para os educadores realizarem intervenções personalizadas para reduzir a taxa de evasão. Além disso, apesar da estrutura do currículo acadêmico influenciar a performance do estudante, ainda existem poucos trabalhos sobre análise curricular na literatura. Assim, esta dissertação propõe duas abordagens para minimizar a evasão no curso de Computação na Universidade Federal do Ceará (UFC) através da análise de dados de 892 estudantes. Inicialmente, é apresentada uma análise aprofundada dos dados obtidos para melhor compreendê-los e encontrar padrões. Então, é proposta uma estratégia de predição baseada no paradigma da classificação com opção de rejeição, na qual os estudantes são classificados nas duas classes descritas anteriormente, além de poderem ser rejeitados aqueles que têm alta probabilidade de serem classificados erroneamente. Estes últimos são provavelmente aqueles que precisarão passar por uma intervenção personalizada. Por fim, é proposta uma técnica de aprendizagem automática para avaliar a estrutura de um currículo acadêmico através da construção de um modelo linear que descreve a relação entre as disciplinas do curso, baseado nas informações de performance dos estudantes. Os resultados são exibidos numa ferramenta de visualização amigável para o usuário, que permite contrastar e comparar a estrutura atual com a proposta pelo modelo.

Palavras-chave: Learning Analytics. Evasão Escolar. Aprendizado de Máquina. Visualização de Dados.

ABSTRACT

One of the most difficult challenges that educators face today is reducing the high student dropout rates in their institutions. Usually, the primary goal of Learning Analytics approaches in this topic is to produce a binary classification of students that are prone to drop out or not. However, this is not enough for educators to initiate a personalized intervention to reduce the evasion's rate. Also, the structure of the curriculum plays a prominent role in the students' performance, and despite this fact, works that analyze curricula's structures are scarce in the literature. This dissertation proposes two approaches to minimize the evasion in the Computer Science program at the Federal University of Ceará (UFC) by analyzing data from 892 students. At first, an in-depth analysis of the acquired data to find patterns and get insights is presented. Then, we propose a prediction strategy based on the classification with reject option paradigm, in which students are classified into the two classes described above and may also reject the patterns with a high probability of being misclassified. These are probably the ones who should be subjected to an intervention. Finally, we also propose a data mining technique that evaluates a curriculum's structure by building a linear model describing the relationship between courses based on the students performance information. The results are visualized in a user-friendly tool, which allows for contrast and comparison between the actual structure and the modeled one.

Keywords: Learning Analytics. College Evasion. Machine Learning. Data Visualization.

LIST OF FIGURES

LIST OF TABLES

TABLE OF CONTENTS

1 INTRODUCTION

In recent years, the amount of educational data has been increasing and becoming more accessible, following the development of modern technologies (devices and software) available to people in general. In the meantime, this availability of educational data gave the researcher and educators the possibility to do more accurate analyses, and consequently, to get insights to improve education in many different areas.

Learning Analytics has appeared in the academic community to support educators in making better decisions and turning education more personalized to the students (DIETZ-UHLER; HURN, 2013). The development of this subject area could be explained by four factors (FERGUSON, 2012): the growth of big data field; the rise of online learning; political concerns on improving educational quality and; benefits for many people and educational institutions.

On the First International Conference on Learning Analytics & Knowledge (LAK11), Learning Analytics was defined as "the measurement, collection, analysis and reporting of data about learners and their contexts, for purposes of understanding and optimising learning and the environments in which it occurs"(LAK, 2011). Greller and Drachsler (2012) support that it has two primary objectives: prediction, which is the use of Machine Learning techniques to automatically support educators in their decisions; reflection, which is getting insights on data that could promote an evaluation on student performance or teaching method.

One of the most difficult challenges that educators and leaders in higher education face today is reducing the high student dropout rates in their institutions. The Brazilian Ministry of Education (MEC) defines evasion as “the departure of a student from his or her undergraduate program of origin, without graduating" (MEC, 1995, pp. 15). In this country, the student dropout causes a loss of public resources in federal universities and produces a lack of skilled workers, which impacts negatively on its development (FILHO *et al.*, 2007).

Recently, there has been a rise in the number of approaches that apply Learning Analytics techniques trying to solve this problem (ANURADHA; VELMURUGAN, 2015; KANTORSKI *et al.*, 2016; BADR *et al.*, 2014; BRITO *et al.*, 2014). These approaches use academic and socioeconomic data collected from students to early diagnose the students prone to evasion and then take appropriate measures for preventing it.

The goal of these approaches is to divide the students into two groups: those who are at risk of evading the program and those who will probably graduate. Unfortunately, this information is not enough for educators to make interventions on the students at risk. First of all,

there are a few reasons for the evasion, such as personal and social issues (APARECIDA *et al.*, 2011), which are out of the institution's scope. Finally, the number of students considered at risk may be too large for the university's limited human resources, and so that there should be a way to select the students most likely to drop out, but with more chances to stay in the program with some counseling.

In the meantime, we know that the structure of the curriculum plays a big role in the development of students' knowledge and their performance (ANURADHA; VELMURUGAN, 2015). Despite this fact, research using Learning Analytics to analyze curricula, pointing out strengths and weaknesses, is not very common in the literature. Additionally, there is a significant demand for tools to assist educators in using data and evidence to build better curricula.

The Computer Science program at the Federal University of Ceará (UFC) achieved one of the highest scores in the 2014 edition of the ENADE – an exam that evaluates the quality of Brazilian undergraduate programs (INEP, 2017). Despite the program's excellence, its average dropout rate between 2005 and 2015 was about 45%, considering students who were admitted by either entrance exam or National High School Exam (ENEM) scores, and around 49% if we take account of all the students who already left the program. The entrance exam devised by UFC was applied from UFC's foundation until 2010; and the ENEM, a national exam that evaluates high school graduates, is conceived by MEC and has been used as the entrance exam at UFC since 2011.

For many years, professors and students have been aware of this fact and have been discussing it in workshops every semester. Many students complain about the difficulties of the courses, remember their colleagues that evaded and even confess they are also thinking about quitting the program. The undergraduate program director was very concerned about this situation and was interested in strategies to minimize this problem.

If we take a look at any undergraduate program, it is not very difficult to see that there are students in a favorable situation and on course for graduation, and there are those in a critical condition, getting terrible grades or not attending classes and prone to drop out. However, there is a group of students that are struggling to graduate: they are attending classes, committed to the program and passing in most of the courses, but they are not getting good grades. We believe that this group will be probably misclassified when provided as an input for an algorithm that classifies the students into dropout prone or non-dropout prone groups. Also, we believe that these are the ones who will most benefit from the educators' intervention and they should be

prioritized.

Besides evasion, another problem we need to take a closer look is retention, which occurs when a student is obliged to stay in the program longer than the time required. About 80% of the graduate students retained themselves, which is a high rate (see Section 3.3.4) and may indicate that the structure of the curriculum could be playing a part in this behavior. Notice that this delay could be discouraging for the students, and it can be a trigger to dropout.

1.1 Objectives

Regarding the situation of the high evasion rate on the Computer Science program at UFC and the motivations explained above, the goals of this dissertation are:

- Provide an in-depth analysis of the students' data, giving an overview of their profiles and some insights about their behavior that could justify the evasion, besides supporting the hypotheses elaborated in this work;
- Identify and prioritize the students in their first year for whom there is no certainty if they will drop out or not. This detection will be made automatically using the Machine Learning technique Classification with Reject Option. This approach classifies between two classes, but reject the patterns for which their class is uncertain;
- Compare the structure of the curriculum with a model built from the grades. In this analysis, we will investigate hidden relations between courses and contrast the structures. The model is based on students academic background and using Machine Learning linear techniques, comparing which one has the best performance: the Synthetic Control Method (SCM) or the Linear Regression with positive coefficients method. Then, the comparison is made using statistical methods. A user-friendly visualization tool was also developed to display the results.

The results shown are applied only to the Computer Science program at UFC, but the process described in this dissertation is general enough and can be used by other undergraduate program directors to generate a more in-depth analysis of their students.

1.2 Structure

This dissertation is organized as follows: in Chapter 2, we present the related work. In Chapter 3, the collected students dataset is described and analyzed. Then, in Chapter 4, a

method for identifying and prioritizing the students at risk of dropping out is proposed. In Chapter 5, we propose the method for evaluating the official curriculum. Finally, the conclusion and future work are presented in Chapter 6.

2 RELATED WORK

This chapter is organized into two sections. In the first section, we discuss works about evasion and retention prediction, and in the second section, we discuss the related work on curricula evaluation and visualization.

2.1 Retention and Evasion Prediction

The Learning Analytics literature is plenty of works identifying the students at risk of evading or retaining using white box algorithms that perform a binary classification between the risky and nonrisky students (ROBERTO; ADEODATO, 2012; BALANIUK *et al.*, 2011; TAMHANE *et al.*, 2014; MÁRQUEZ-VERA *et al.*, 2013; LAKKARAJU *et al.*, 2015; COSTA *et al.*, 2015; BRITO *et al.*, 2014; MARIA *et al.*, 2016; KANTORSKI *et al.*, 2016; PASCOAL *et al.*, 2016). Such methods are advantageous when we want to know which attributes lead to these phenomena. However, most of them follow the idea of just dividing the students between the two classes, which may not be helpful to the context of undergraduate program, such as Computer Science at UFC.

For example, there is a work that used Bayesian Network to predict students' dropout at SENAI (MARIA *et al.*, 2016). A network was modeled with the supervision of two faculty members to select the most important attributes and to calculate the chances of a student evading the school. Then, the algorithm is applied to do a binary classification into candidate to drop out or not. The results were displayed on a web page for each student containing the academic information, the probability of evading and the student's situation compared to others. There is no ranking of the most at risk, which makes it difficult to prioritize the ones for intervention. This difficulty can be observed in an example on the paper: a student had the probability of 52% to stay in the program and so he or she has been classified as non-dropout prone (see Figure 1). However, we believe he or she could be a suitable candidate to be counseled by educators because the certainty about his classification is very low.

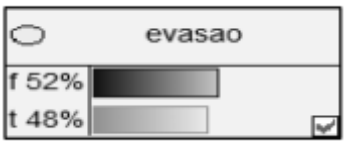

Figure 1 – The example of a student classified as non-dropout prone as illustrated in Maria *et al.* (2016).

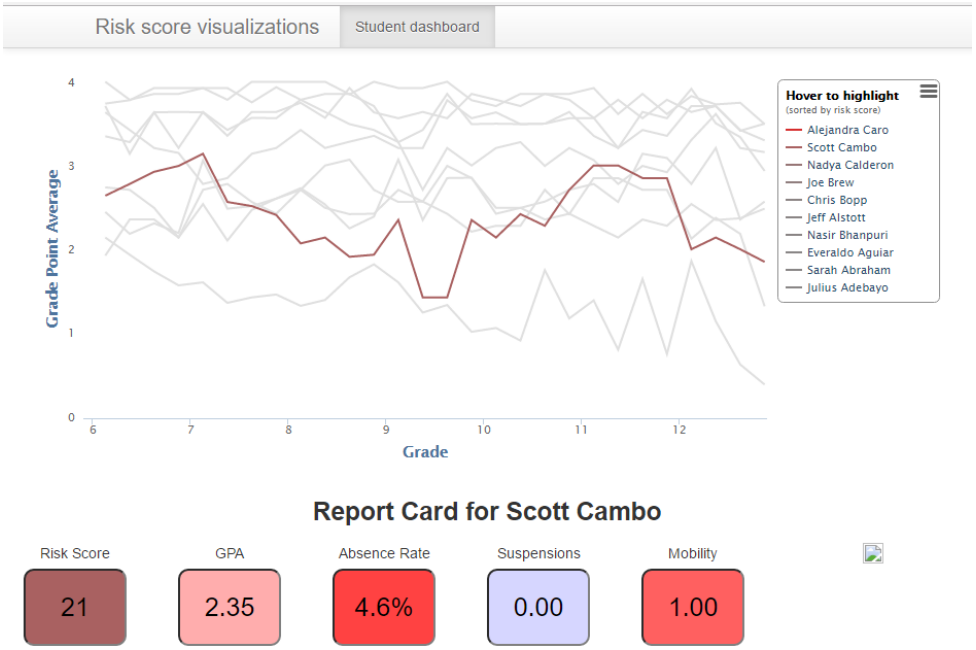

Figure 2 – The web visualization tool used to show the ranking of the students most at risk of retention (LAKKARAJU *et al.*, 2015).

On the other hand, some works tried to solve the problem stated above. In the United States, for instance, a framework was created to predict the risk of retention in High School (LAKKARAJU *et al.*, 2015). The researchers tested it on a dataset of 200,000 students from two U.S. school districts, containing their grade averages, absence rates, social and demographic information, etc. Because these districts had limited resources, initially the students had to be ranked, according to some measure of risk such that students at the top of the list are verifiable at higher risk. Once educators have such ranked list available, they can just choose the top k students from it and assist them. The ranking was displayed on a web visualization tool (see Figure 2). The approach used, as a measure of risk, the confidence estimates provided by the following algorithms: Random Forest, Adaboost, Logistic Regression, SVM and Decision Tree (TAN *et al.*, 2005). The experiments showed that Random Forest had the best results. Besides building a ranking, this approach also identified the students at risk of retention.

Also, it is important to detect an occurrence of a student at risk of dropping out as soon as possible, so educators have enough time to do appropriate interventions. The evasion rate in the first year of college is about two to three times higher than the following years, not only in Brazil but all over the world (FILHO *et al.*, 2007).

At the Federal University of Pernambuco (UFPE), researchers tried to identify the students from 1998 to 2008 at risk of retention in six undergraduate programs (ROBERTO;

ADEODATO, 2012) using Induction Rules to discover how soon retention could be detected so that the educators could intervene and avoid it. Also, this technique indicated the most influential attributes of this behavior. According to the authors, this could save up to 5 million dollars of the university's resources. Moreover, the prediction resulted in an area under the ROC curve of 0.84, which is a good result.

The work of A. Tamhane *et al.* (2014) tried to identify the students at risk of failing two American exams: Criterion-Referenced Competency Tests (CRCT) and Iowa Test of Basic Skills (ITBS). It used Logistic Regression to separate the students, achieving a ROC curve of 0.924. Also, they studied how soon they could detect a student with a poor performance, testing the ROC-curve value obtained when using all the features of the data or only their grades. They showed that the results were close to each other and a good value can be acquired for students at the 5th-grade students at least. In Bayer *et al.* (2012), the goal was to predict dropout in an undergraduate program, including social data in the attributes. They achieved good results as of the 4th-semester.

Notice that our approach divides the students into two groups like most of the previous works, but differs by rejecting those with a high probability of being misclassified. That allows educators to prioritize students with high probability of graduating when subjected to personalized intervention activities and, furthermore, follow them up even with limited resources.

2.2 Curricula Analysis and Visualization

Most of the works related to this topic focus on recommending an academic path to students. This leads to the constrained satisfaction problem, in order to provide an optimal curriculum model respecting the prerequisites limitations. Concerning to this fact, the work of Wu (2005) proposed a curriculum modeler to solve this problem. It divided this process into two phases. First, it built an initial model conditioned to constraints like each course prerequisite. Second, it provided a software application to promote the interaction between the modeler and the student, allowing the modeler to request some constraints (E.g.: number of courses to be taken each semester). Our approach differs by comparing the official curriculum with the modeled one, which is built using students' transcripts.

Regarding the use of historical academic data to build the model, some papers proposed to find the students' frequent paths by mining them (PECHENIZKIY *et al.*, 2012; WANG; ZAÏANE, 2015). Wortman and Rheingans (2007) developed a visualization tool to show

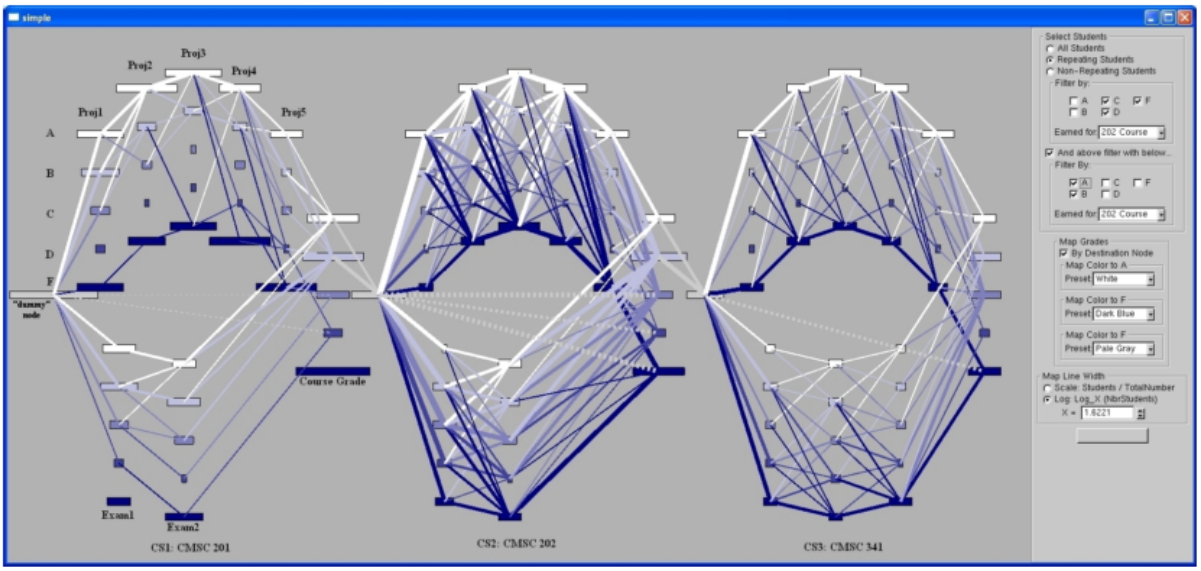

Figure 3 – Visualization tool proposed by Wortman (2007)

the paths in the first three courses in a Computer Science program. Each course is placed side by side, creating, for each one, columns of 5 nodes for each activity and each exam, plus an extra central column on the right and an extra central node on the left (see Figure 3). Each node in a column represents a grade, where the higher it is placed, the higher the grade is. Edges were created for each student, connecting each node representing an exam to the other one, as well as for the activities. The last one of both type of columns are connected to a centralized column that represents the final grade. That indicates the performance of a student while he or she is enrolled in the course. Finally, an edge is created returning to the central node of the course if the student failed it, or connecting to the other course in the right if the student passed. The visualization allows educators to detect patterns in students behavior while taking these courses. Nevertheless, this approach did not make use of any machine learning technique to predict patterns, as we did in our approach.

Another work provided recommendations to the students' career in their program (CAMPAGNI *et al.*, 2015). The researchers tried to find the frequent paths, like the papers cited before, and then clustered the students based on their performance to propose an ideal career. However, they did not provide any user-friendly visualization tool to show the results, which could benefit the educators who do not have much knowledge of Data Mining or Machine Learning.

When we look at the papers that focus on visualizing the curriculum, most of them are just a tool for analysis and visualization and do not propose any changes on the curriculum or a model that could improve the students' academic trajectory. Also, most of them are desktop-based, which constrains the accessibility for every user to the tool. Our approach is web-based,

allowing any person with a computer, a browser and an internet connection to access it.

Figure 4 – Visualizing academic trajectory patterns (GAMA; GONCALVES, 2014; JORDÃO *et al.*, 2014)

One example is a visualization that displayed the curriculum on a web page, with nodes representing courses and edges connecting courses to demonstrate a student path, to visualize academic trajectory patterns (GAMA; GONCALVES, 2014). Each node was represented as a circle, divided proportionally in two parts: the amount of student who failed in this discipline (colored red) and the amount of those who passed (colored green). Also, the higher is the number of students following the same path, the thicker is the edge. In this paper, the goal was to compare two types of edges: a simple one and a Bezier cubic curve, showing that the last one is better. But in a subsequent paper (JORDÃO *et al.*, 2014), an interaction was added so that the user could click on a course and the tool would focus on that course and on those that are related.

Another approach by J. Géryk (2015) contrasts with previous works by visualizing students behavior differently. Clusters are created, dividing the students into groups based on their particular field of study. Each student is plotted on a two-dimension chart, where the y-axis represents the average number of credits and the x-axis represents the average grade. This visualization is animated, meaning that each plot will move according to time, semester by semester. If students drop out the program, their plot goes red and fall down the x-axis. Below this visualization, we have a parallel coordinates diagram where each column represents an

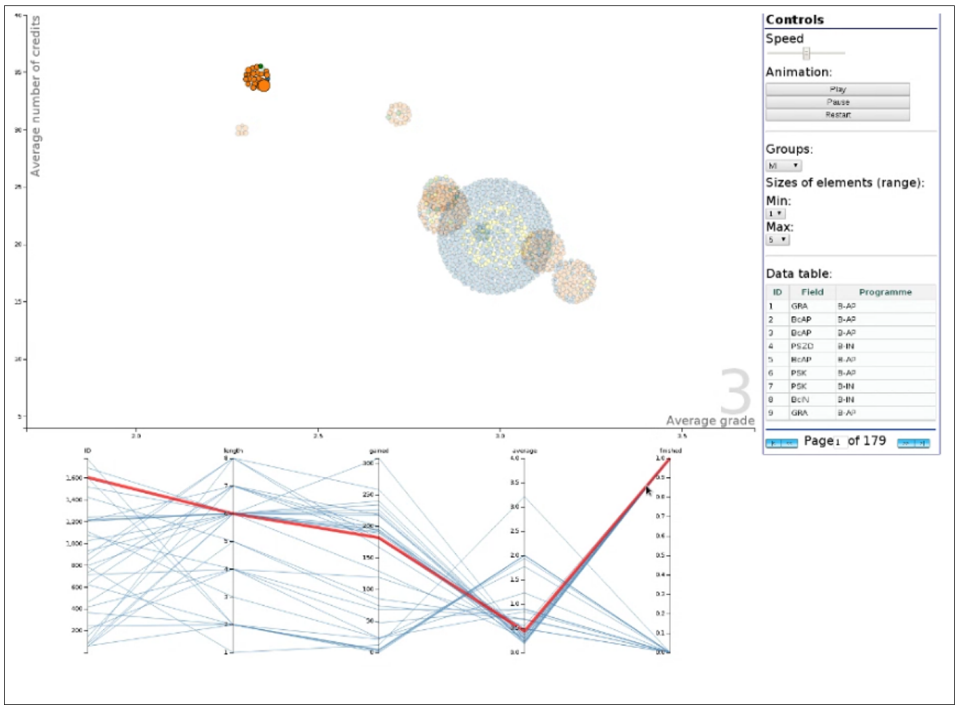

Figure 5 – Visualization tool proposed by Géryk (GÉRYK, 2015)

attribute, allowing the user to see patterns. The researcher observed that this tool was useful for big datasets and called more the attention of the spectator to get insights about the students' behavior. In the meantime, he compared the performance on finding some patterns against line plots and scatter plots.

3 DATA ANALYSIS

Before applying any machine learning procedure to the data, it is essential to clean up and understand the data to build the models. In this chapter, we will describe the academic rules for enrolled undergraduate students, such as the criteria to be approved, how the GPA is computed etc. Then, we will present the tools used to collect the data and the results of an in-depth analysis of the students' profiles.

3.1 Background information about UFC rules

At UFC, Professors are free to decide their way to evaluate the students in a course. Despite this liberty, they are required to follow some rules when grading students.

The grade is defined as numerical, from 0 to 10. There are three possibilities for a student's grade G:

1. if $G \geq 7$, the student is directly approved. We will refer this grade to be an A grade.
2. if $4 \leq G < 7$, the student must take a final exam E_f, and a new grade G_f is computed as

$$G_f = \frac{G + E_f}{2}. \tag{3.1}$$

   If $G_f \geq 5$, the student is approved and G_f is referred as a B grade. Otherwise, the student failed.
3. Otherwise, the student failed.

Also, a student can also fail when their attendance rate in a course is less than 75%, causing the final grade to be 0. Finally, students are allowed to withdraw from courses, up to a certain limit per semester.

The GPA varies from 0 to 10,000 and it is calculated using the following equation (PROGRAD, 2017a):

$$GPA = \left(1 - \frac{0.5 \cdot T}{C}\right) \cdot \left(\frac{\sum_i P_i \cdot C_i \cdot N_i}{\sum_i P_i \cdot C_i}\right), \tag{3.2}$$

where:

- T is the sum of all withdrawn courses' workloads;
- C is the sum of all enrolled courses' workloads;
- N_i is the final grade of the course i;
- C_i is the workload of the course i;

- P_i is the period when the student got enrolled on course i, which is calculated using the following equation: $P_i = \min\{6, \text{period in which the student got enrolled}\}$.

Also, when a student fails by attendance rate the same course more than twice or more than four different courses, they are terminated from the program. Finally, each program has its maximum period of years for students to graduate, or else they will be administratively withdrawn from the program (PROGRAD, 2017b).

It is also important to emphasize that the student must have a good GPA to be eligible for scholarships and be approved on international exchange programs. In other words, having a good performance at university guarantees the students' participation in activities inside the institution that will develop their career. Otherwise, they will be excluded from these activities, which in turn could discourage them from being engaged with the program.

3.2 Collecting data

In 1999, the Computer Science undergraduate program had its curriculum reformulated. At that time, it was created the 2000.1 curriculum with 31 mandatory courses and a number of other optional courses, requesting the student to complete 3280 hours of course workload to graduate. This curriculum was active until 2015 when there was another reformulation, and a new curriculum was established in 2016.

Regarding this fact, we decided to collect undergraduate data from students enrolled in the 2000.1 curriculum, from 2005 to 2015, in which comprises the students who already graduated in Computer Science and the last freshmen of this curriculum. The dataset was obtained in collaboration with the computer science department and with the director of Undergraduate Studies in Computer Science at UFC.

The first step consisted of downloading all the students' transcripts. Each transcript offers some personal data such as name, registration number, father's and mother's name, date of birth, id number and address; the academic data, such as the program, the active curriculum, the status, the admission method, the student's current period, record of all enrolled courses and grades, and finally, if applicable, the period when the student finished or dropped out of the university, followed by the reason for that.

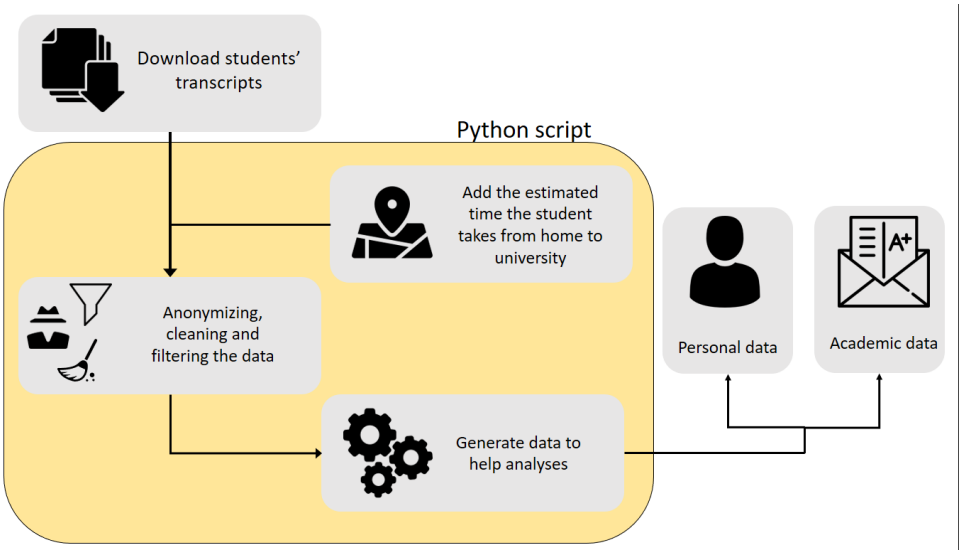

Figure 6 – Script's scheme to fetch Computer Science students' data. The transcripts in PDF format are downloaded from the University system by the program director. Then, the script is run to extract the data from the documents. Before anonymizing, it computes the time to arrive at the university by public transportation from each student's house, computed by Google Maps®, and includes it in the dataset. Then, the data is anonymized, cleaned and filtered. Additionally, it generates data like enrollment duration, age at the admission, gender etc, that could be helpful to the analyst in further analyses. Finally, it generates the datasets with students' data and the transcripts with their grades. Icons made by Freepik and EpicCoders from <www.flaticon.com>.

Some of this information is very sensitive. To respect privacy policies, a script was developed to extract, collect, anonymize, clean, filter and format the data as well as helping in further analyses. The script was delivered to the Computer Science director to run in a controlled environment, avoiding sensitive data leakage. The scheme in Figure 6 gives an overview of this process.

Table 1 – Description of the data attributes collected from each Computer Science student at UFC using the script.

Attribute	**Type**	**Description**
ID	int	Random number assigned to each student
Time	int	Time in seconds from the students neighborhood to the campus (calculated by Google Maps®).
Gender	string	Gender of the student: 1 for Male, 2 for Female.
Mobility	int	The student participated in international academic mobility: 1 for yes, 0 for no.
Age	int	Student's age at admission.
Enrollment duration	float	Enrollment duration of the student from admission to current time or to his last semester at University.
Admission Method	string	Admission method of the student: ENEM, entrance exam, transferred from another institution etc.
Departure Method	string	Departure method of the student from University: Graduation, program withdrawal, program change etc
Readmission	int	1 if the student was admitted again (by the entrance exam or by ENEM), 0 otherwise. (It is possible for an active or an evaded student to be readmitted in the same program and in this case all his or her failed courses will be erased from the transcript).
GPA	float	Partial GPA of the first and second semester (GPA is the weighted arithmetic average of the grades, where the weights are the courses' credits) - see Section 3.1.
Student status	string	Student's status at the university: 1 for graduated, 2 for dropped out, 3 for active.
Student workload	int	Sum of student's workload from completed courses.

Table 2 – Description of the data attributes collected from each student's record of all enrolled courses and grades.

Attribute	Type	Description
Period	float	Period the course was taken.
Code	string	Course code on University's system.
Name	string	Course's name.
Workload	float	Course's workload in hours per semester.
Credits	float	Course's workload divided by 16, which is the number of weeks in a semester, resulting in the Course's workload per week.
Class code	string	Class in which the student got enrolled for the course he took.
Attendance rate	float	Attendance rate in the course.
Grade	float	Final grade in the course.
Status	string	Final result in the course: passed, failed, transferred the credits or withdrawn.

The script produces two outputs: a dataset anonymized with students' profile data and a folder with datasets for each student with their enrolled courses and grades. The first dataset contains the attributes detailed in Table 1 and the datasets in the folder contain the attributes detailed in Table 2. Notice that the provided script can be used by any program director to download the students' transcripts and easily convert them into dataset suitable for analyses. The directors can generate their own analyses, get insights and make decisions that could improve their program.

The data used in this dissertation was collected at the end of 2016, including information from 892 students who were enrolled in the program between 2005 and 2015. Details will be shown in the next section.

3.3 Data overview and insights

Machine Learning techniques depend on the background data to build a model that best approaches reality. The choice of the attributes to compose the dataset is an important aspect to guarantee this proximity.

The analyst in charge of this process should know the context of the problem to elaborate the hypothesis properly. Additionally, the supervision of an expert in this subject should be very helpful to accomplish good results for this study.

As we mentioned in the introduction, the Computer Science program's average dropout rate between 2005 and 2015 was about 49%, considering all the students who already left the program.

Figures 7 and 8 show a first overview of the dropout problem in the Computer Science program. The division into the two groups, as shown in Figure 7, is almost equal, indicating a well-balanced dataset and reducing the probability of a biased classification by the

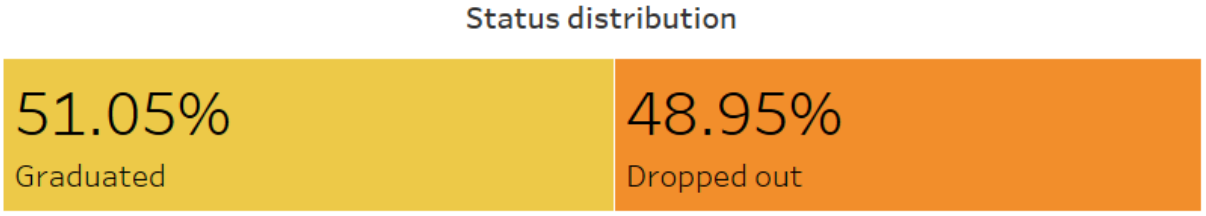

Figure 7 – Status distribution among the students who lest the program.

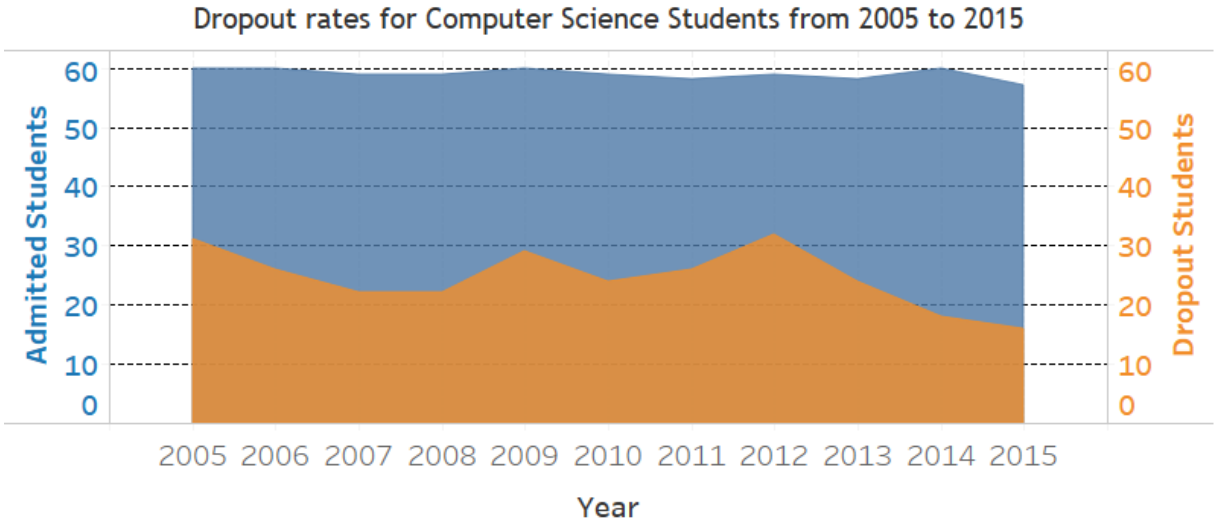

Figure 8 – Dropout rate by year of Computer Science students at UFC who were admitted by entrance examination or by ENEM. The blue area indicates the number of admitted students and the orange area indicates the number of those that eventually dropped out. Notice that the years 2013, 2014 and 2015 have incomplete data, and the dropout rate could be even higher.

Machine Learning techniques. Also, looking at each year's dropout, it is observable that, despite having a few fluctuations, it depicts an upward trend starting in 2009 up to 2012. The class of 2012 was considered the worst with relation to evaded students, showing that more than 50% of the students dropped out. We cannot see the same trend after 2012 because most of the students were still active. Also, we observed in our data, for those admitted by ENEM, the same behavior stated by Filho *et al.* (2007), in which the evasion is two or three times higher in the first year than the following years (see Figure 9). That is why is so important to detect the risk of evasion as early as possible so educators can do appropriate interventions quickly.

In a first moment, we observe some previous works and their attribute choices. Despite a few variations, most of the works agreed that grades and attendance rate are the most influential attributes to predict dropout and students' performance (MOSELEY; MEAD, 2008; BAYER *et al.*, 2012; MARQUEZ-VERA *et al.*, 2011; LAKKARAJU *et al.*, 2015; BADR *et al.*, 2014). A student with very low grades and poor attendance in class would not be able to keep up with the program and will probably evade, so there is no surprise regarding this fact.

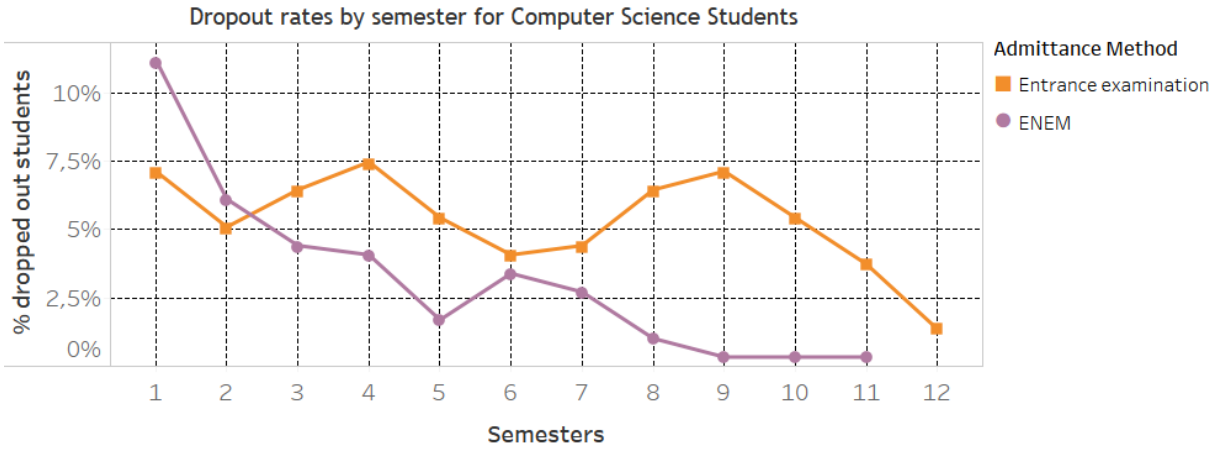

Figure 9 – Dropout rate by semester of Computer Science students at UFC who were admitted by entrance examination (orange line) or by ENEM (purple line).

Furthermore, reviewing the goals set in Section 1.1, we need only these attributes to fulfill the last objective because detecting relationships between courses depends only on the students' academic performance, and the second objective depends on many other factors (APARECIDA *et al.*, 2011).

In the following sections, we evaluate the attributes according with their usefulness and also describe the Computer Science students' profile.

3.3.1 Gender

Gender is an interesting attribute to be evaluated in the Computer Science program. It is well-known by students, professors, and staff that the majority of the Computer Science community is formed by males. Despite the reduced population of females, the distribution of dropped-out and graduated students is very close and their average grades are higher than men's, as we can see in Figure 10.

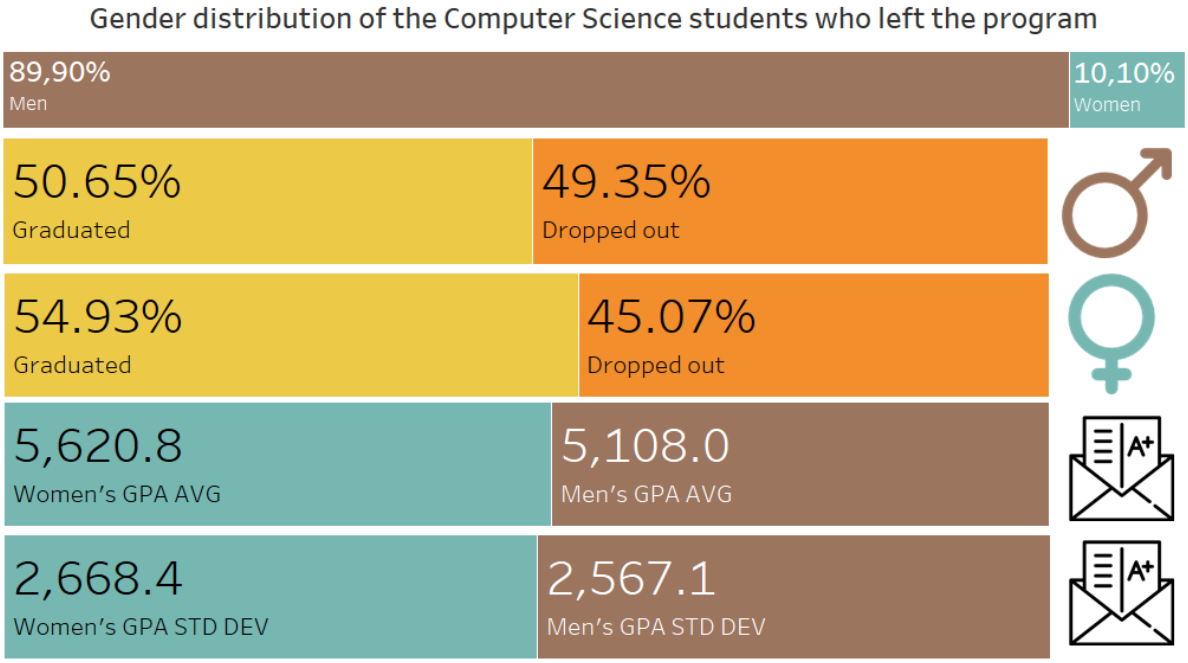

Figure 10 – Gender distribution in the Computer Science program. In the first row, we see the distribution of men and women. The second and the third row show the proportion between dropped-out and graduated male and female, respectively. Finally, the last two rows show the average GPA and its standard deviation by gender. Icons made by Freepik from <www.flaticon.com>.

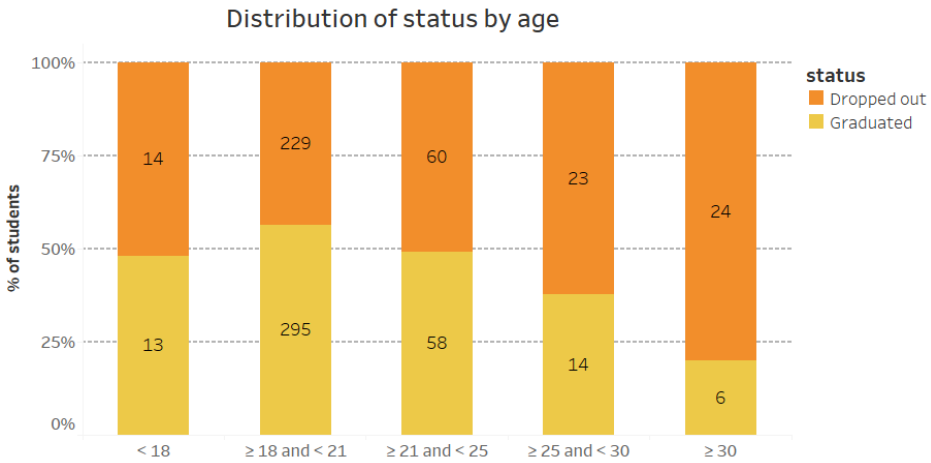

Figure 11 – Distribution of status by age at admission, considering the students who left the program.

3.3.2 Age

Most of the students are admitted when they are 18, usually the time they finish high school. We can observe by choosing a student who graduated at random, the probability of he or she to be younger than 25 is high. The majority of older people does not graduate, as we can see in Figure 11. This attribute can considerably influence the experiments' results.

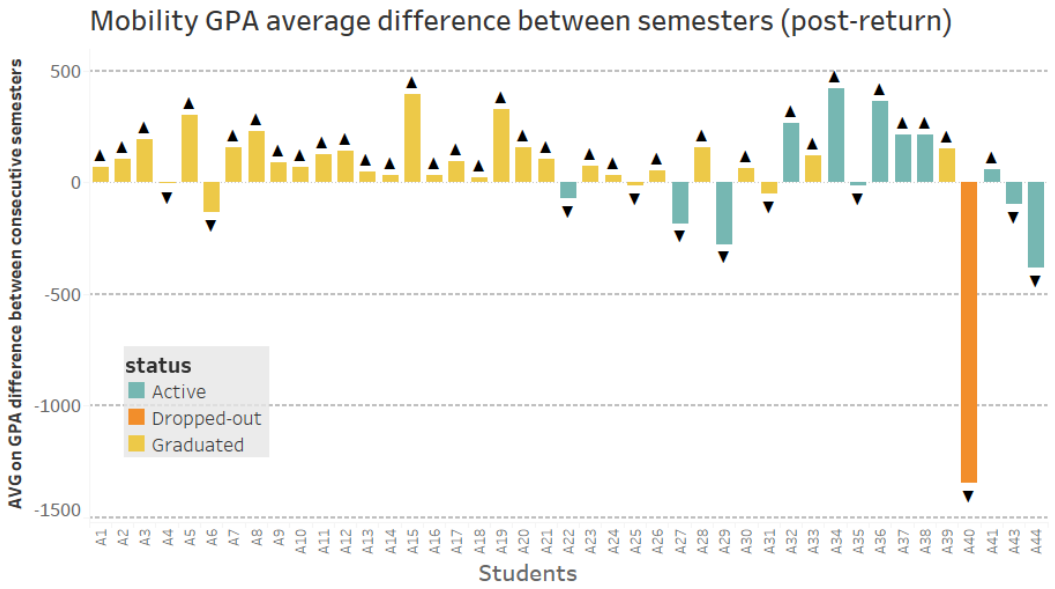

Figure 12 – Average of the difference of GPA between two consecutive semesters for each student who participated in an exchange program, after coming back to Brazil. One student who dropped out is represented by the orange bar; the active students are shown in blue and; the students who graduated are shown in yellow. For those who increased their grades, a mark of a triangle pointing up is exhibited above the bar, and for those who decreased their grades, a mark of a triangle pointing down is exhibited below the bar. Most of the students increased their grades, contradicting the perspective that they lose their interest in the program after coming back.

3.3.3 Mobility

In 2011, the Brazilian Government launched a program called "Ciência Sem Fronteiras"(Science without borders), offering 100 thousand scholarships to undergraduate students to travel and study in universities abroad, and they could transfer the credits when they return to Brazil. Besides this program, UFC also has a bilateral agreement with two French universities to send and receive students, paid by Brazilian Government, in a program called Brafitec. By analyzing students' transcripts, we detected that 46 students participated in a kind of international mobility, and 3 of them were currently abroad when the data was collected.

The director believed that, despite the strict constraints to be eligible for these programs, students were not performing well in the program after coming back. The data showed that, actually, most of them increased their grades (see Figure 12). Furthermore, more than 90% of these students graduated and the rest was distributed this way: two students were traveling, one student is active, and another dropped out. Additionally, the average GPA of these students is 7,269, which is high for the Computer Science program. This attribute is a great indicator that the student would be classified as non-dropout prone by the classifier algorithms.

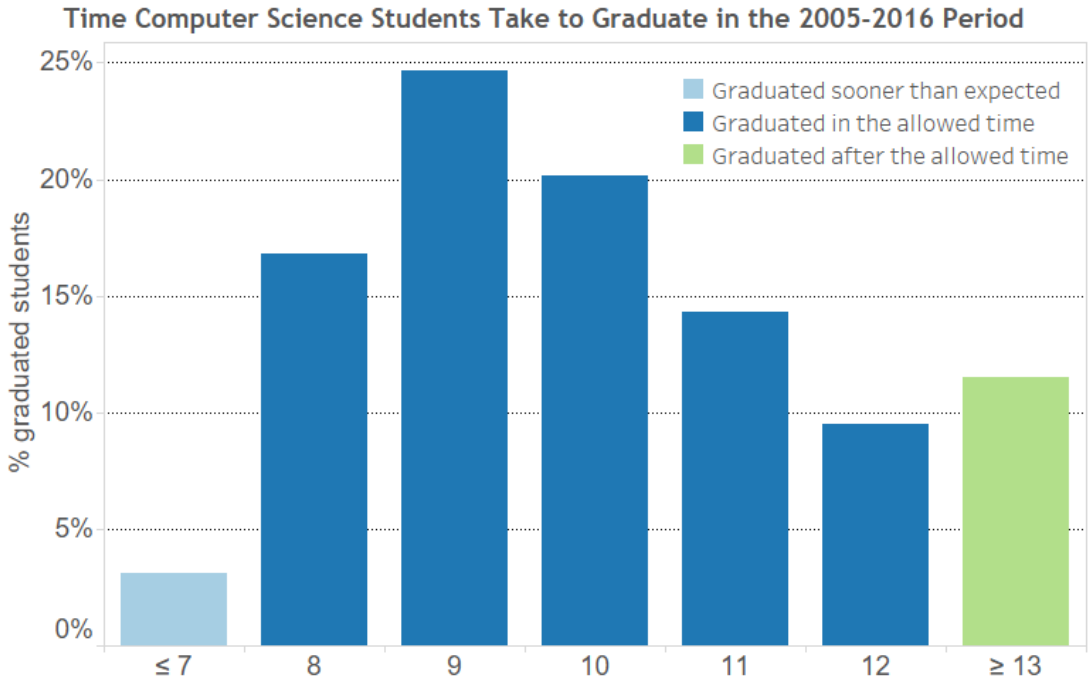

Figure 13 – Distribution of graduation time in semesters for Computer Science students. Most of the students take more time than the standard time suggested to graduate (8 semesters).

3.3.4 Enrollment duration

The 2000.1 curriculum, in the way it was composed (see Section 3.2), was designed to be completed in 8 semesters. When we look at Figure 13 exhibiting the time to graduate detected in the data, we can see that there is a large concentration in 9 to 12 semesters. Also, more than 10% are finishing the course after the allowed time. This result provided us with the motivation to investigate whether the structure of the curriculum could be influencing the retention rate.

3.3.5 Readmission

An interesting case has been observed: some of these students were readmitted by retaking the entrance exam or using their ENEM scores so that the failed courses were erased from university's record and they were able to transfer the credits from the courses they passed. This is a strategy to gain more time to stay in the program, erasing their bad records and earning opportunities such as getting a scholarship, which would not be available to the students with a certain number of failed courses.

This creates the hypothesis that after being readmitted, these students would graduate

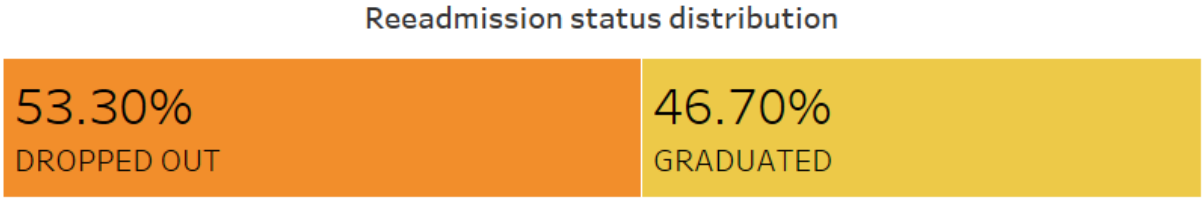

Figure 14 – Distribution of status for readmitted students, considering the students who left the program. Contradicting the hypothesis, many students are dropping out again.

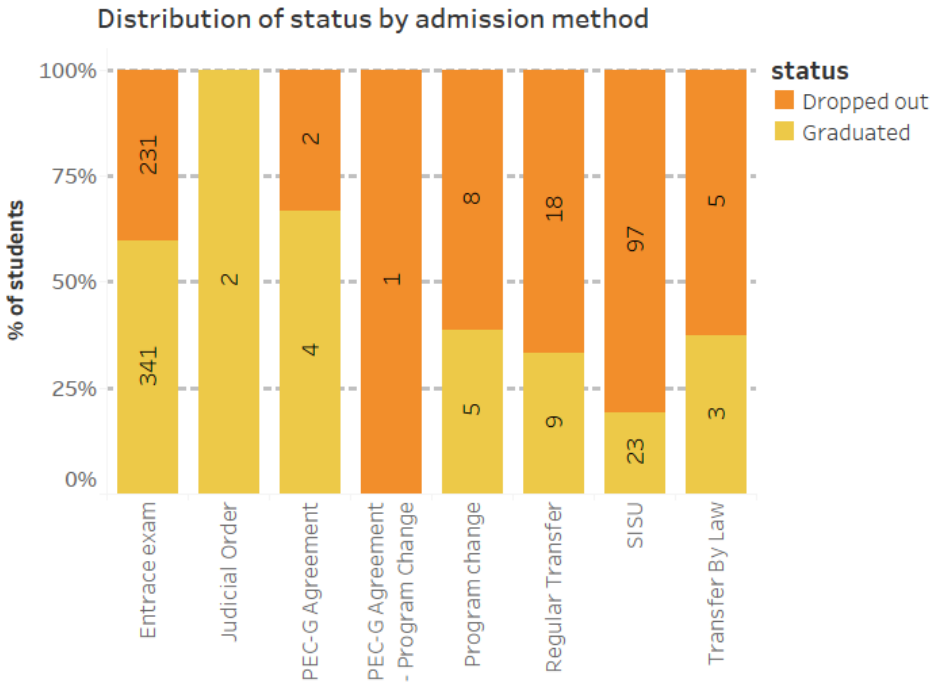

Figure 15 – Distribution of status by admission method, considering the students who leave the program.

because of the facilities obtained, as explained above, but instead, the number of dropped out students is still high for them. Figure 14 shows that about 53% of the readmitted students have dropped out.

3.3.6 Admission Method

In Figure 15, we can observe a great difference between the status distribution, when considering the type of admission. This could be a great indicator to predict the evasion. Despite this fact, one must be aware of some tricky cases, like the distribution for SISU: less than 25% graduated, which is low. However, this method has started to be used in 2011, and most of the students admitted in this way are still active.

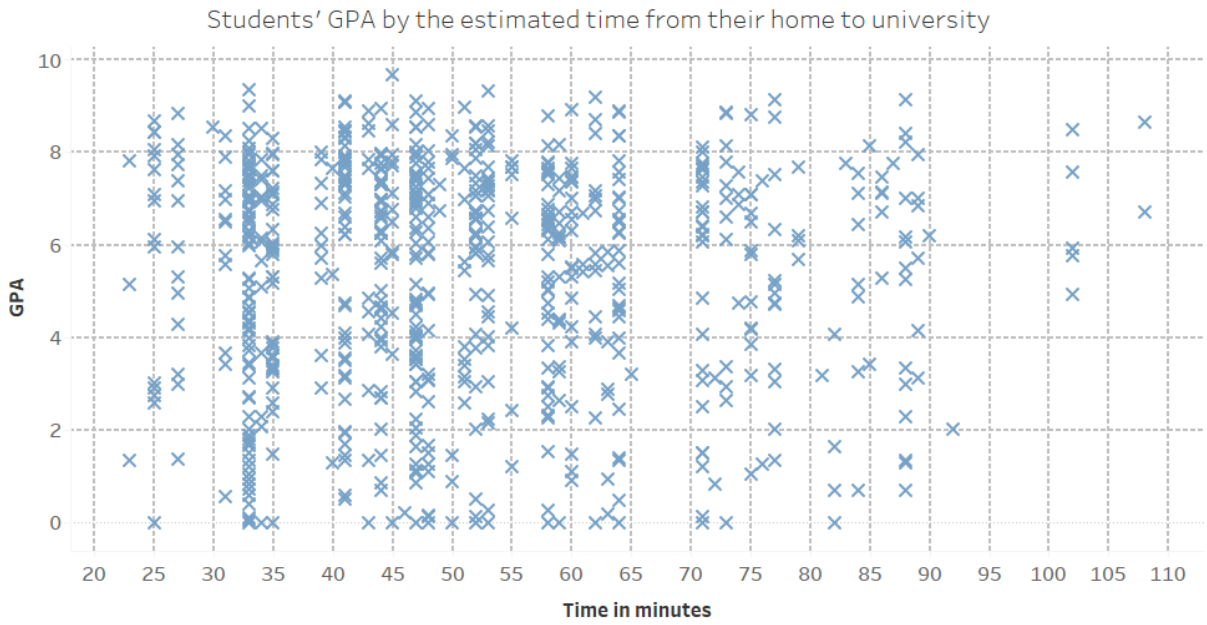

Figure 16 – Distribution of Students' GPA by the estimated time they take from home to university using public transportation.

3.3.7 Time to arrive at the university

Students usually complain about the time they take from home to UFC. For those who live far away from the university, they have to wake up too early to arrive in time for the first class at 8 A.M. Since this program has courses all day and demands a lot of homework, some students must reduce their sleep time and they criticize this fact, arguing that this affects their performance.

Regarding this situation, one may suppose these students have lower grades than the students who live near the university and that could be discouraging to stay in the program. But this premise is shown to be false in Figure 16: there is no correlation between the time to arrive at the university (based on the neighborhood the student lives) and the student's GPA. Figure 17 shows where the students live and the average GPA in each neighborhood.

Despite most of the students being concentrated in neighborhoods with higher Human Development Index (HDI) like Aldeota, Meireles and Centro (PMF, 2017), where they have more bus line options and take less time to arrive at the university even by car, compared to downtown districts, they have lower GPA average than others in neighborhoods much far away from the campus.

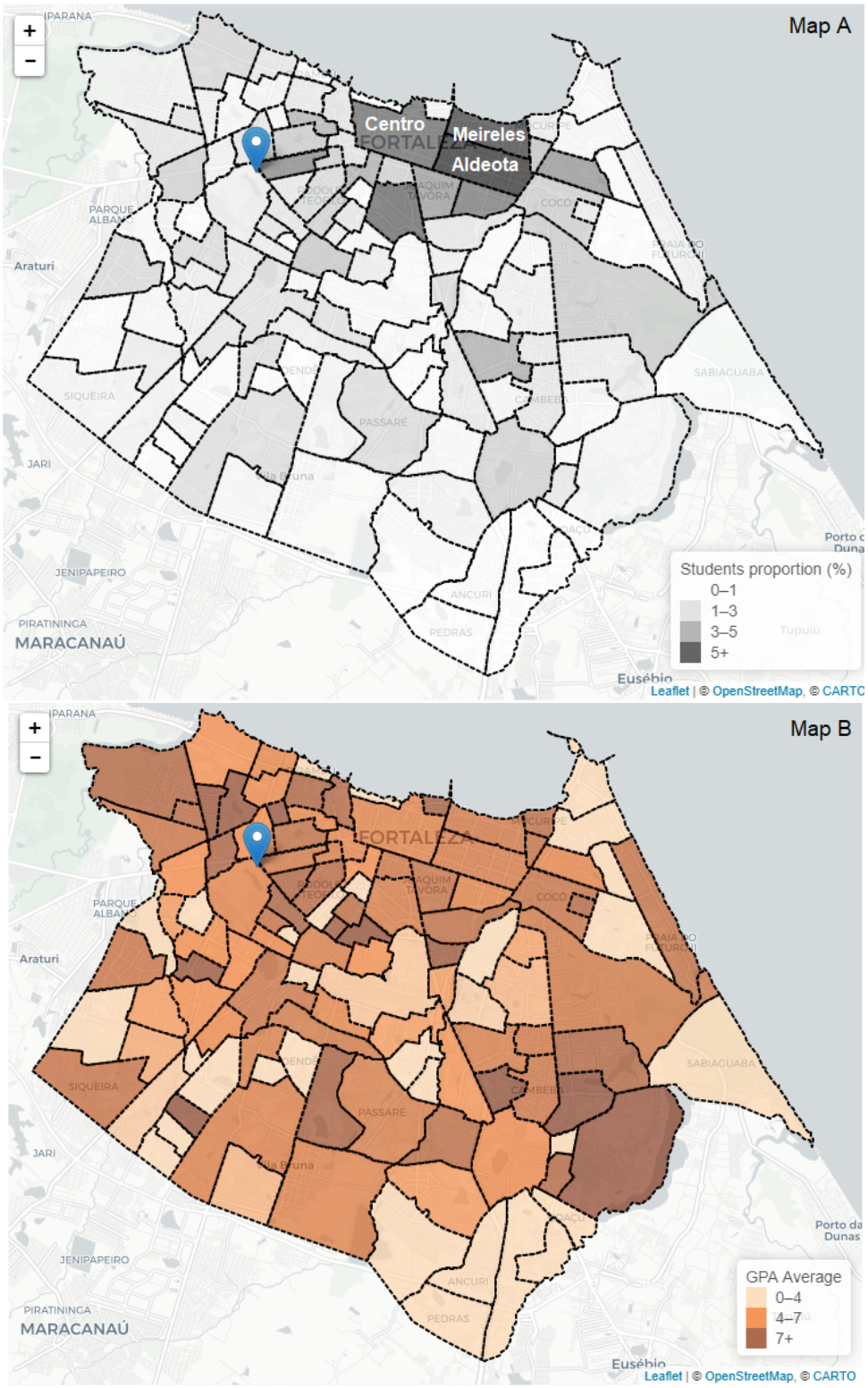

Figure 17 – Maps of the city of Fortaleza - Ceará, Brazil, divided by neighborhoods with the students distribution (Map A) and the GPA average of the students who live there (Map B). The higher is the saturation, the greater is the value in both maps. The Computer Science building is represented by the blue pin.

4 IDENTIFYING AND PRIORITIZING THE DROPOUT-PRONE STUDENTS

In this chapter, we propose a data mining technique that classifies students into two groups: the dropout prone and non-dropout prone. In addition, the approach rejects those with great probability of misclassification by the algorithm. Since the choice between these two classes is uncertain for the rejected students, they may probably succeed if subjected to an intervention process and remain in the program until their graduation.

This chapter is organized as follows: in Section 4.1, the theoretical foundations are explained. Then, the methodology used in this experiment is presented in Section 4.2. Finally, the results are shown in Section 4.3.

4.1 Theoretical Foundation

4.1.1 Classification with reject option

Classification with reject option comprises a set of techniques to improve classification results in decision support systems (CHOW, 1970). Roughly speaking, it consists in avoiding to classify an unseen instance **x**, if the decision is considered not sufficiently reliable so that the rejected instance can then be handled by a different classifier, or manually by a human.

As mentioned before, in possession of a "complex" dataset, every classifier is bound to misclassify some data samples. Depending on the costs of the errors, misclassification can degrade the classifier's performance. Therefore, techniques in which the classifier can abstain from providing a decision by delegating it to a human expert (or to another classifier) are very appealing. In the following, we limit the discussion of reject option strategies to the binary classification problem. For that, we assume that the problem involves only two classes, i.e., when $N = 2$, henceforth referred to as $\{\mathscr{C}_1, \mathscr{C}_2\}$. However, the classifier must be able to output a third class, the reject one $\{\mathscr{C}_1, \mathscr{C}_{Reject}, \mathscr{C}_2\}$).

The implementation of reject option strategies requires finding a trade-off between the achievable reduction of the cost due to classification errors, and the cost of handling rejections (which are application-dependent). Thus, Chow (CHOW, 1970) proposed to design classifiers by minimizing the empirical risk, defined as

$$\widehat{R} = E + \alpha R \tag{4.1}$$

where

- R is the ratio of rejected patterns among the samples used in validation phase;
- E is the ratio of misclassified patterns among the samples used in validation phase;
- α is the rejection cost (whose value must be specified in advance by the user).

It is worth highlighting that a low α leads to a classifier that rejects many patterns, thus decreasing its error rate. A high value for α, on the other hand, leads to a classifier that rejects few patterns which, in turn, increases its error rate.

The design of classifiers with reject option can be systematized in three different approaches for the binary problem:

1. **Method 1**: It involves the design of a single, standard binary classifier. If the classifier provides some approximation to the a posteriori class probabilities, $P(C_k|\mathbf{x})$, $k = 1,2$, then a pattern is rejected if the largest value among the posterior probabilities is lower than a given threshold. For this method, the classifier is trained as usual and the rejection region is determined *after* the training phase, heuristically or based on the optimization of some post-training criterion that weighs the trade-off between the costs of misclassification and rejection.
2. **Method 2**: It requires the design of two, *independent*, classifiers. A first classifier is trained to output $\mathscr{C}_1$ only when the probability of $\mathscr{C}_1$ is high and a second classifier trained to output $\mathscr{C}_2$ only when the probability of $\mathscr{C}_2$ is high. When both classifiers agree on the decision, the corresponding class is output. Otherwise, in case of disagreement, the reject class is the chosen one. The intuition behind this approach is that if both classifiers have high levels of confidence in their decisions then the aggregated decision should be correct in case of agreement. In case of disagreement, the aggregated decision is prone to be unreliable and hence the rejection would be preferable (SOUSA *et al.*, 2009).
3. **Method 3**: It involves the design of a single classifier with embedded reject option; that is, the classifier is trained following optimality criteria that automatically take into account the costs of misclassification and rejection in their loss functions, leading to the design of algorithms specifically built for this kind of problem (FUMERA; ROLI, 2002; SOUSA *et al.*, 2009).

We will see in Section 4.2.2 that we used Method 2 to build our classifier with reject option.

4.1.2 Measuring classifiers results

There are four important measures to evaluate binary classifier decisions:

- *True Positive (TP)*: Number of positive patterns classified as such;
- *False Positive (FP)*: Number of negative patterns classified as positive;
- *True Negative (TN)*: Number of negative patterns classified as such;
- *False Negative (FN)*: Number of positive patterns classified as negative.

Regarding these concepts, a Confusion Matrix is a quick way to visualize them and give a verdict about the binary classifier performance. It is defined as follows: the columns represent the real label for the pattern and the lines represent the classifier label. It means that the main diagonal contains the number of examples correctly classified and the antidiagonal contains the misclassified ones. This is illustrated in Table 3.

Table 3 – Confusion Matrix definition.

	POSITIVE	NEGATIVE
POSITIVE	TP	FP
NEGATIVE	FN	TN

4.2 Methodology

4.2.1 Dataset Description

The data was filtered to get information about the students' behavior on the required courses of the first year, such as grades, attendance rate and final result (passed or failed). Also, we collected the academic and social information obtained in data extraction process (see Section 3.2) and calculated the average grade of the first and second semester. All these attributes are described in detail in Table 4.

The evasion rate in the first year is two to three times higher than the following years according to Silva Filho *et al.* (2007), and we confirm this information in our data for those admitted by ENEM, as described in Chapter 3. Also, it is important to detect the risk of evasion as early as possible, so educators can apply appropriate interventions quickly. These are the reasons why we choose to work only with the students' information for this period.

Finally, we tested the algorithm with 32 students admitted in 2015, so that the

Table 4 – Description of the data attributes collected from each Computer Science student at UFC.

Attribute	Type	Description
Time	int	Time in seconds from the neighborhood to the campus (calculated by Google Maps®).
Gender	int	Gender of the student: 1 for Male, 2 for Female.
Mobility	int	The student participated in international academic mobility: 1 for yes, 0 for no.
Age	int	Student's age at the admission.
Admission Method	int	Admission method of the student: ENEM, entrance exam, transferred from another institution etc.
Readmission	int	1 if the student was admitted again (by the entrance exam or by ENEM), 0 otherwise. (It's possible for an active or an evaded student to be readmitted in the same program and in this case all his or her failed records will be erased from the transcript).
Attendance rate	float	Attendance rate in first semester.
GPA	float	Partial GPA of the first and second semester (GPA is the weighted arithmetic average of the grades, where the weights are the courses' credits).
Course Status	int	Status of the mandatory courses of the first year: 0 for fail, 1 for passed.
Student status	int	Student's status at the university: 1 for graduated, 2 for dropped out, 3 for active.

program director could validate our method with a very recent example.

4.2.2 Classifier Design

As we mentioned before, we used **Method 2**, described in Section 4.1.1 to generate classifiers with reject option. When choosing such an approach, we considered two main aspects: classifier adaptation and classification performance. The first criterion is related to the easiness of using any classifier as the base classifier of our method. In this sense, **Method 3** is the worst choice since the formulation of the classifier has to be modified. In both **Method 1** and **Method 2** any classifier could be used. Considering the performance of these approaches, previous works like (MESQUITA *et al.*, 2016) and (OLIVEIRA *et al.*, 2016) showed that **Method 2** led to the best performances.

We selected the Feedforward Neural Network with Random Weights (FNNRW, (SCHMIDT *et al.*, 1992)) as the base learner. The FNNRW is a Perceptron based neural network that randomly assigns the weights of its hidden layer. After that, the weights of the output layer are estimated using Ordinary Least Squares. The method is widely used because of its simple formulation and remarkable performance in various applications.

The adaptation of the standard FNNRW to **Method 2** is achieved by using a weighted cost function, where the weights are chosen according to the class of each training example. With this modification, two FNNRWs can be trained and each of them is biased to one of the classes. This characteristic is essential to the design of a reject option classifier according to **Method 2**. The adaptation used in this work was proposed by (MESQUITA *et al.*, 2016).

4.3 Experiments and results

As a first step, we conducted an experiment where several popular machine learning methods are used in the dropout prediction task. Note that in this initial test, no reject option is included in any method. The goal of the test is to verify if dropout-prone students can be identified by using the features presented in Section 4.2.1, along with classification algorithms.

In this experiment, the dataset was randomly split into training ($2/3$ of the dataset) and testing ($1/3$ of the dataset). The whole test was repeated ten times and the average accuracies are shown in Table 5.

Table 5 – Performance of several classifiers in dropout prediction.

Classifier	**Accuracy (%)**
FNNRW	86.37
MLP	86.43
SVM	88.32
Naive Bayes	85.82
K-NN	84.12

The results presented in Table 5 show that any classifier can correctly identify dropout prone and non-dropout prone students in at least 84% of the cases. It shows that the FNNRW is achieving good results, closer to other algorithms. Beside its simplicity, as we see in Section 4.2.2, these factors validate its choice to be the base classifier with reject option.

To verify the impact of the reject option, we conducted tests varying the rejection cost α. For each α value we chose the classifier that minimizes the risk of misclassification $\widehat{R}$. By doing so, our algorithm is designed to classify a student only when the probability and incorrect classification is minimized.

The performance of FNNRW with reject option was assessed with the Accuracy Rejection curve (A-R). The A-R curve presents the accuracy for each rejection rate and the results of our experiment are shown in Figure 18.

According to the A-R curve, we can notice that the accuracy of our method increases with the number of rejections. Such an accuracy improvement is expected since increasing the rejection rates turns the algorithm to be more conservative, only classifying students when the degree of certainty is very high.

Another interesting point regarding our approach is that the rejected students are

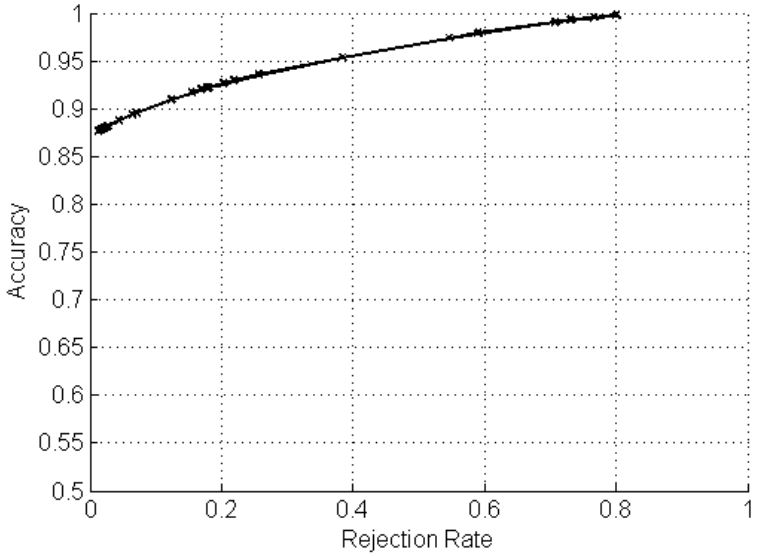

Figure 18 – Accuracy Rejection curve for the FNNRW with reject option in the dropout prediction problem.

the ones that are difficult to predict whether they are going to leave the program or not. In the current setting, one can suppose that students classified as dropout prone are the ones that will almost certainly leave the program. Non-dropout prone students are probably doing well and may conclude their studies, and the reject students comprise a risky class, where the effort of advisors and educators can possibly lead to good results. To validate this hypothesis, we used a trained classifier with a reject option of 20% in a dataset comprising students that started the course in 2015. A subset of 32 students was used in our experiments. Under the described conditions, the classifier achieved an accuracy of 71%. Detailed results are presented in the confusion matrix in Table 6.

Table 6 – Confusion Matrix of the 32 students admitted in 2015 classification.

	ACTIVE	DROPOUT
ACTIVE	9	0
DROPOUT	8	11

By analysing the confusion matrix, we can see that no student that dropped out of the program was assigned as a non-dropout one. On the other hand, 8 students that are currently active were classified as dropout prone. Although this was assigned as a wrong classification, a more detailed analysis revealed that these students are currently in a very difficult situation. Some features of these 8 students are shown in Table 7.

Table 7 – Misclassified students' performance in their first year. Each row represents a student, and each column represents a course. A cell is colored orange if the student failed that course, otherwise it is colored white.

Student	Calc I	LinAlgeb	DiscMath	Prog I	DigiCirc	Calc II	Physics I	DataStrct	Prog II	Data Trsf	Attendance (%)	GPA
A											68.87	1207.3
B											93.58	2749.1
C											83.07	5394.1
D											62.05	373.4
E											62.27	0[1]
F											87.57	4332.3
G											82.51	4986.3
H											86.7	4139.7

Table 8 – Rejected students' performance in their first year. Each row represents a student and each column represents a course. A cell is colored orange if the student failed that course, otherwise it is colored white.

Student	Calc I	LinAlgeb	DiscMath	Prog I	DigiCirc	Calc II	Physics I	DataStrct	Prog II	Data Trsf	Attendance (%)	GPA
X											84.51	5691.6
Y											95.79	2743.4[2]
Z											89.16	5000.8
W											93.12	6524.0

As we can see, all students failed at least 3 courses. Also, notice that some of them have an attendance rate less than 75% and a GPA less than 5000, which indicates that they are getting bad grades in courses and missing too many classes. We can suppose, considering this situation, although these students are still active, they have a high dropout probability.

In contrast with this result, we can verify the same features for the 4 students that were rejected by our method. The data in Table 8 show that, in most cases, the student failed in less courses and they are attending classes, more than the ones misclassified as dropped out.

It is observable that the rejected ones are struggling to graduate. They are attending classes and being approved in courses, but they are not getting good grades. This is an essential information for the educator to intervene and help those students, since their efforts do not reflect in good grades, which can be discouraging to continue in the program. An intervention can help them to understand their difficulties and to make an academic career plan to be more successful in courses and eventually graduate.

4.4 Summary

In this chapter, a new approach to minimize the problem of evading students was presented, classifying them in three groups: the ones who will certainly get their diploma, the

[1] Transferred credits are not used when computing the GPA.

[2] This student transferred the credits of a few courses, so his GPA got lower.

ones who will certainly drop out of college and the ones whom we are not sure about their future. In a first moment, the experiments used several binary classifiers, achieving accuracy higher than 80% for all cases. Then, we applied the FNNRW classifier to divide the students into the two groups and rejected those students that could belong to either group. We validated our approach with the Accuracy Rejection curve. Also, we applied our approach on a sample of 32 students from the 2015 class, and performed a deep analysis using their academic data to verify the hypothesis that the rejected students are the ones who will most benefit from the educators' intervention.

5 ANALYSING THE CURRICULUM'S STRUCTURE

Another hypothesis we wanted to verify is whether the structure of the curriculum could be inferred from the data. In this chapter, we describe how we verified this hypothesis. The Synthetic Control Method (SCM), which has been successfully applied in previous studies in social science and economics (HINRICHS, 2012; ABADIE *et al.*, 2010), is used to build the model. It detects relationships between two courses, in which one will be set as prerequisite of another if a significant dependency has been found and if it must be done before the target course, according to the official curriculum. And so, a new curriculum is modeled and proposed. Then, the results are exhibited in a web user-friendly visualization tool.

In Section 5.1, the methodology used in this experiment is presented, including the collected dataset and how we chose the methods for building the model. The experiments and results to design a curriculum model are shown in Section 5.2. The visualization tool is presented in Section 5.3. Finally, the curriculum is evaluated in Section 5.4, comparing the Computer Science course curriculum established in 2000 with the model.

5.1 Methodology

5.1.1 Dataset Description

In order to graduate, the students in this curriculum are required to take 31 mandatory courses (they must also take a certain number of elective courses, but these were not considered in this work). The provided dataset contained information about the mandatory courses taken by each student. The data was filtered to get information about the students when the courses were taken, such as the final grade, attendance rate, semester and partial GPA. These attributes are described in Table 9.

Notice that for the students still active or the students that dropped out, we have the information only for the courses taken until then. Besides, there were a few cases when the student was able to take a course without fulfilling its prerequisites (this is allowed in some specific situations by the program). When this occurred, we left that course out of the analysis.

Table 9 – Description of the attributes collected for each mandatory course taken by the Computer Science students in the dataset.

Attribute	Type	Description
Course ID	int	A unique number identifying the course.
Grade	float	Final grade for the course.
Attendance	float	Percentage of attendance (0 to 100) for the course.
Semester	float	Semester when the course was taken.
GPA	float	Partial GPA of the most recent semester before the course was taken, 0 if it was taken in the first semester (GPA is the weighted grade point average, where the weights are the courses' credits).

5.1.2 Synthetic Control Method

The Synthetic Control Method (SCM) was first introduced in the seminal paper (ABADIE; GARDEAZABAL, 2003). In order to try to estimate the impact of terrorism in the Basque Country's economy in the 70's, the authors used regions in Spain to find a counterfactual group to represent a synthetic Basque Country, where the terrorism has never happened and, in this way, they would measure the differences between the real Basque Country and its synthetic counterpart. To do so, the work finds a convex combination of the Spanish regions that best represent the data describing the Basque Country in the pre-terrorism period.

This approach could also be used to infer the impact of a specific measure in a undergraduate course, like changing its instructor, or changing the methods used in the classroom. In our work, however, we are interested in the convex combination produced by the method, from which we can extract some information about the relations between different courses in the curriculum.

The SCM method works as follows. We first assume we have a matrix $Y_{n\times m}$, where each row represents a student, and each column represents the mandatory courses, and so $y_{i,j}$, for $1 \leq i \leq n$ and $1 \leq j \leq m$, represents the grade obtained by the student i on course j. For now, we assume there is no missing data. Also, let $X_{k\times m}$ be a predictor matrix, which contains k different measures regarding each of the courses. The choice of which measures to use is at the researcher's discretion. In this paper, we selected the following measures as predictors: average grade of students with final grade above 7 (hereby named as A grades, or B otherwise) in that course, average grade of B graded students, average grades overall, average attendance of A graded students, average attendance of B graded students, average attendance overall, average GPA from A graded students at the moment they enroll in that course, average GPA from B graded students, average number of A graded students in that course, average number of B graded students.

Our goal is to find a combination of course grades that best represent a certain target course. Consider that the last column in the X and Y matrices, denoted by X_1 and Y_1, contain the data related to our target unit (treatment unit, in Abadie and Gardeazabal's original work). The rest of the matrices, X_0 and Y_0, comprehend data from the rest of the courses, hereby named control units.

Let $W_{m-1\times 1}$ be a vector where $0 \leq w_i \leq 1$ and $\sum_1^{m-1} w_i = 1$. Our purpose is thereby to find such a vector, where the synthetic group, Y_0W, best represents the target unit output Y_1. To measure this representation, we use a $V_{k\times k}$ diagonal matrix in which each element $v_{i,i}$ accounts for the relative importance of the i-th predictor when comparing the difference between the synthetic group and the target unit. We arrive at an iterative optimization loop of the form:

$$V = \arg\min_V(||Y_1 - Y_0W(V)||) \tag{5.1}$$

$$W(V) = \arg\min_W((X_1 - X_0W)^T V(X_1 - X_0W)) \tag{5.2}$$

To obtain the desired output, we apply this method considering one course as the target unit at a time. In addition, we only use courses as control units for a certain target if they occur before that target in the curriculum; i. e. for a target unit in the M-th semester, $M > 1$, only the courses in semester $M-1$ or earlier are used as controls.

5.1.3 Measuring SCM's Results

As detailed in 4.1.2 section on Chapter 4, there are four important concepts for measuring a binary classifier, since it is possible to achieve a high accuracy if it is strongly biased to one class.

Based on them, there are two more measures:

- *Precision:* The number of positive labeled as such by the number of patterns labeled positive;
- *Recall:* The number of positive labeled as such by the number of positives patterns;

And to combine them, we use the *F-measure* (or *F1-score*), which is the weighted harmonic mean of the precision and the recall. It is defined below:

$$F = \frac{2TP}{2TP + FN + FP} \tag{5.3}$$

Where 0 is the worst value for it and 1 the best.

Table 10 – Average Error and F1 Score for SCM and Linear Regression.

	Synthetic Control Method	Linear Constrained Regression
Average Error	0.13	0.13
F1 Score	0.27	0.20

The use of F-measure is explained in the next section.

5.2 Curriculum model design

To build the curriculum, two experiments were performed to predict students' grade on courses. The first has made using SCM and the second using linear regression with positive weights, so that we can verify the performance of SCM. The coefficients of the linear model reveal the courses that are used to predict the grade. We decided to use constrained linear regression (positive weights) because negative relations between grades of different courses do not seem to make sense.

Notice that, in this dissertation, we are interested only in find the relationships between courses, so evaluate the accuracy on predicting the grade is out of our scope and further details can be found in the work of Barbosa *et al.* (2017). Regarding this fact, the coefficients obtained in experiments were used to define how the courses are relate each other. For that, we consider the prerequisite prediction task as a binary classification problem. In this setup, each possible required course is classified as being part of the prerequisites or not being part of the prerequisites. For both SCM and linear regression, we considered a course as a prerequisite if its associated coefficient is greater than 0.1. By using this framework, we can assess the performance of the methods using the standard F1 score. In Table 10, we present the average results for the error and the F1 score metrics.

The results in Table 10 show that SCM achieved a F1 score greater than the linear regression's. That performance indicates that SCM's predicted dependency between courses is closer to the official curriculum. It is also important to notice that both methods had low F1 scores (F1 scores range from 0 to 1). This fact is expected because we considered a simplified scenario in which we wish to infer the dependency of courses only using the performance of the students, and there are also many other factors that influence on this dependency, such as the contents of the course, methodology and instructor. Although we used this simplified assumption, several interesting insights were generated by our method. In the next section, we describe our

developed visualization tool and then, in Section 5.4, we explain our insights and how we used this tool to find them.

5.3 Visualization tool

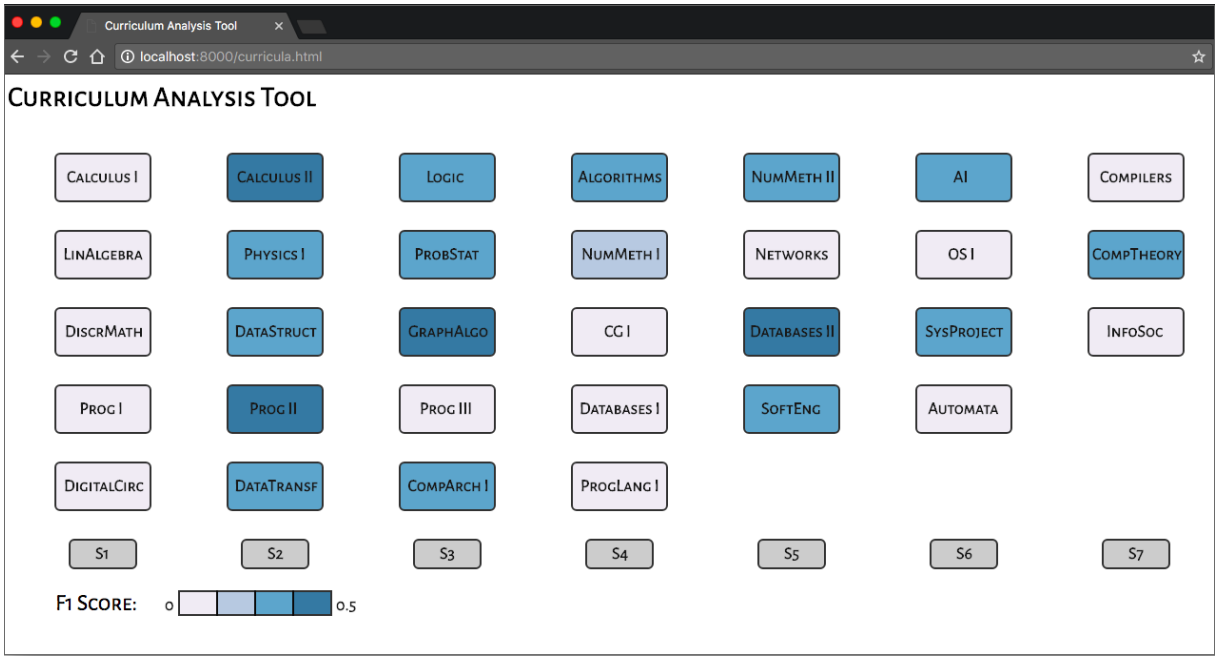

Figure 19 – Visualization tool's overview. The curriculum's mandatory courses are displayed in chronological order. Each course is colored in shades of blue (the saturation increases according to its F1 score).

Sometimes evaluating the program's curriculum by only looking at numbers can be very difficult for educators who do not have the necessary knowledge in statistics and machine learning. To account for that, we developed a user-friendly interactive web visualization tool that provides all the information educators need to perform an intuitive and in-depth analysis.

The visualization tool is composed of three main views: the overview, the course influence view and the influence value view. When the analysts load the tool on the browser, the first view they see is the overview (see Figure 19). In this view, the curriculum's mandatory courses are displayed in chronological order, each semester is represented by a column and each course is depicted by a box labeled with its code or a nickname provided by the analyst (the full name of the course is shown as a tooltip when passing the mouse over it). Also, each course is colored in shades of blue, and the darker the shade is, the larger is the course's F1 score. The scale of these values is shown in the legend at the bottom-left of the page. The analyst can also click on the legend's boxes to create a threshold for this value and filter only the courses that

satisfy the selected threshold. The main purpose of this view is to show the differences between the official curriculum's structure and the structured acquired from the data at a single glance and without displaying too much detail (MUNZNER, 2014, Chapter 5). In the first versions of this tool, we also considered showing the courses' prerequisites by connecting them with edges but we soon decided not doing so to avoid clutter.

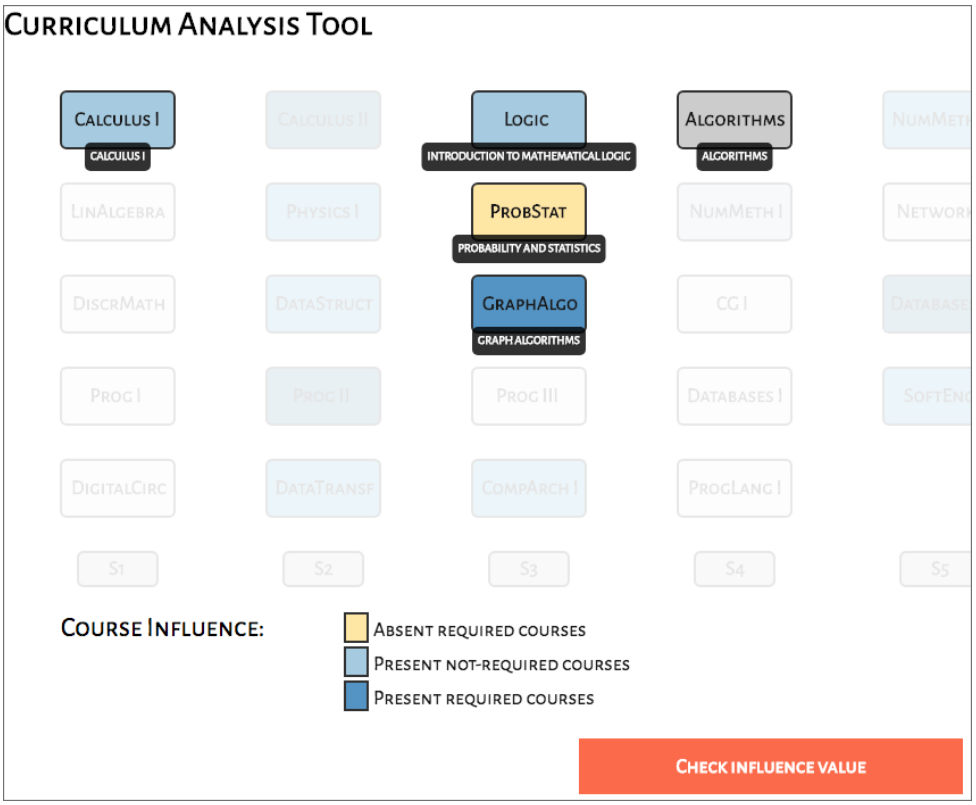

Figure 20 – Course influence view. The courses that influence the selected *Algorithms* shown in gray are displayed in shades of blue (all the other courses are faded out). The prerequisite the model considered to influence *Algorithms* is shown in dark blue, and the real prerequisite considered not to influence it shown in yellow.

After looking at the overview, the analyst can investigate each course more deeply using the course influence view (see Figure 20). To open this view, the user clicks on the course's box, the view focuses only on the analysis results of that course. In this view, the selected course is shown in gray, its required courses that were classified as influent are colored in a dark blue, its required courses that were not classified as influent are colored in yellow, and the courses that are not required but were classified as influent are colored as a light blue. If the analysts want to go even deeper and see how much influence each course receives, they can also click on the "Check influence value"button, and bring up the course's influence value view (see Figure 21). The visualization changes, fading out the yellow-colored boxes and changing the colormap to a shade

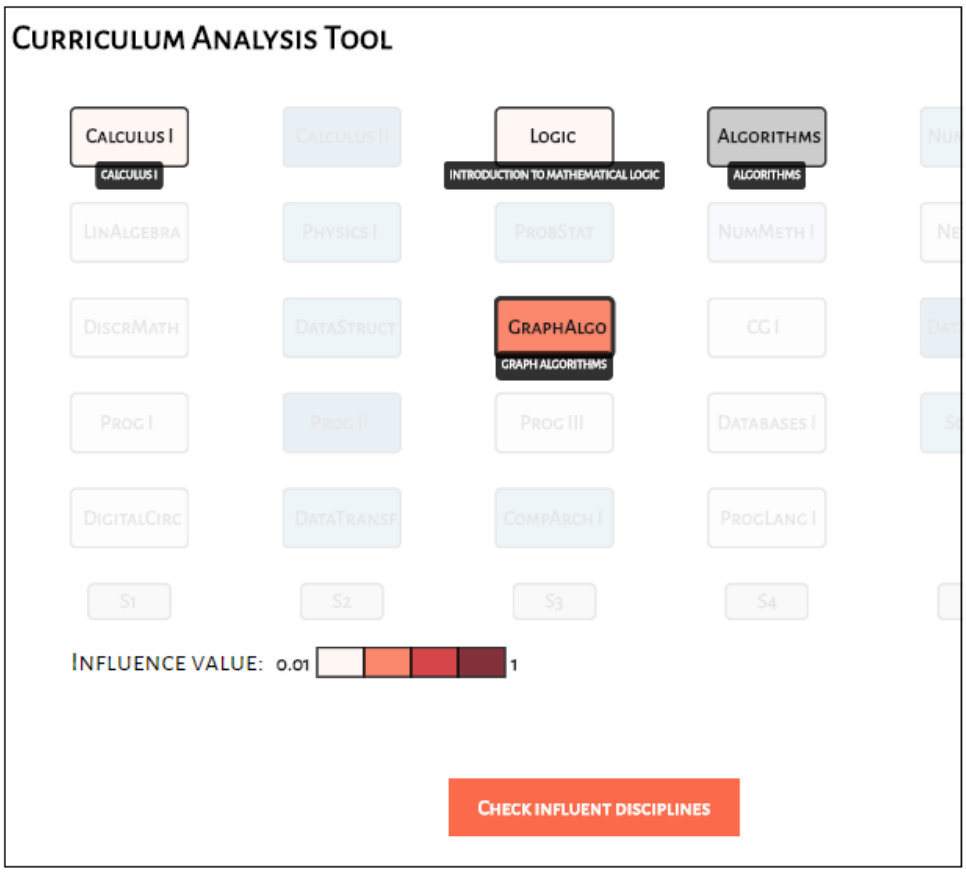

Figure 21 – Influence value view for *Algorithms*. The courses that influence the selected *Algorithms* shown in gray are displayed in shades of red (all the other courses are faded out), in which the saturation increases according to the influence rate.

of red now encoding the influence rate. In this colormap, the saturation increases according to the influence rate. Similarly as the legend boxes in the overview, the legend boxes in this view also allow filtering courses based on a threshold value (see Figure 22). The user can still perceive the official required courses because they have a thicker border around their boxes (see *Graph Algorithms* in Figure 21). By clicking anywhere on the background, the users are brought back to the overview.

This tool allows the users to see how different the official curriculum is from the model built from the data, exploring different levels of detail. In the next section, we describe how we used this tool to gain some insights about the SCM results.

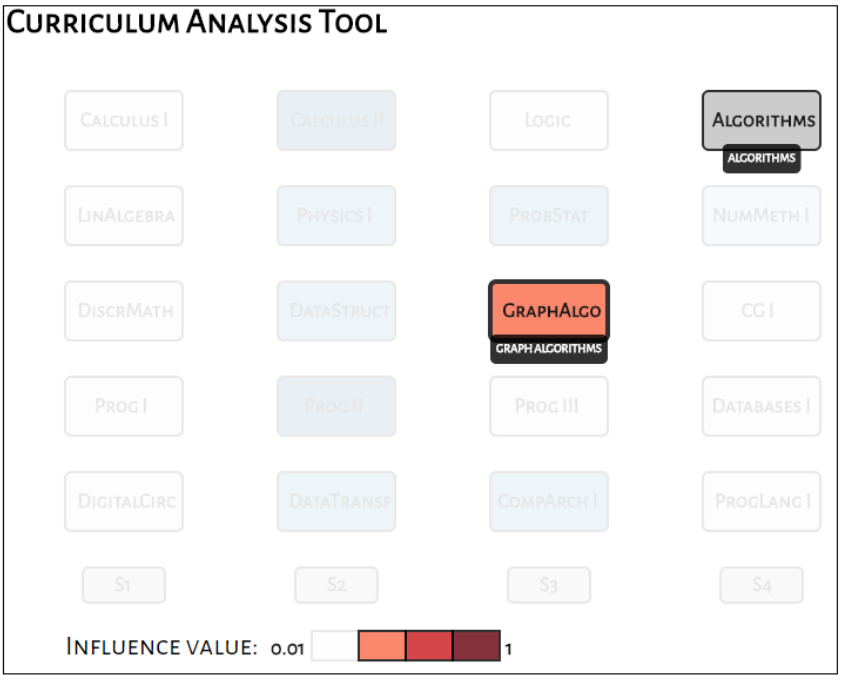

Figure 22 – Influence value view for *Algorithms* with threshold applied. The courses that influence the selected *Algorithms* shown in gray and respect the threshold set are displayed in shades of red (all the other courses are faded out), in which the saturation increases according to the influence rate. The threshold can be set by clicking in one of the legend boxes. Comparing with Figure 21, we can see that all boxes colored with the same color of the legend's first box were faded out.

5.4 Discussion

One of the main questions we wanted to answer about the analysis was why we obtained a low value for the F1 score. By using the visualization tool, starting at the overview (see Figure 19), we can see that, in fact, the official curriculum did not match well with the model because of the high number of light and medium blue boxes. A plausible explanation for this is that our method uses information only about the grades of the students, and this is not enough to infer the relations between courses. In fact, other aspects such as methodologies, professors, the amount of theory versus practice are not being considered in our work and may have a significant impact on the results. However, there were some interesting findings. We can see in Figure 19 that only a small number of courses have the darkest shades of blue, which means that few courses in the official curriculum matched well with the model. The course *Databases II* in semester 5, has only *Databases I* as a prerequisite in the official curriculum. If we see the influence value view of *Databases II*, as shown in Figure 23, we can conclude that although other courses were considered to influence *Databases II*, the model assigned a higher value to *Databases I*. And the same behavior was detected for the courses *Calculus II*,

Programming II and *Graph Algorithms*, which are equally colored.

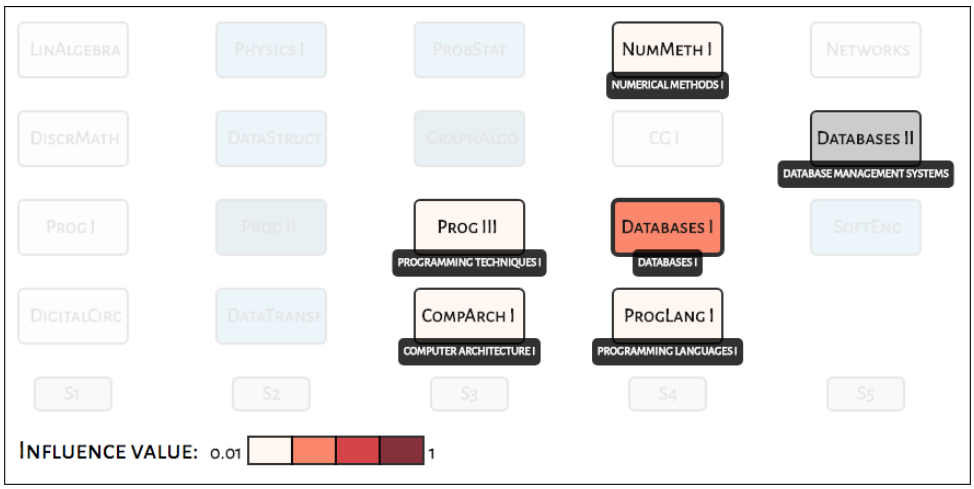

Figure 23 – Influence value view for *Databases II*, in which the saturation increases according to the influence rate. In the model. *Databases II* is more influenced by *Databases I*, being consistent with the official curriculum.

Going back to the overview in Figure 19, we notice that most of the courses fit an intermediary value for the F1 score. In general, this is when the model does not include some of the prerequisites present in the official curriculum. In the case of *Algorithms*, for example, the model did not include the prerequisite *ProbStat* (see Figure 20). However, this case was very interesting because, in a recent update of the curriculum in 2016, this prerequisite was removed from the list of *Algorithms*' prerequisites and so, in the end, the model agreed with reality.

Another interesting case is when the model considered other courses to influence a certain course, outside the list of its prerequisites, also causing a low F1 Score. Take for example, *AI* (Artificial Intelligence) in Figure 24. In this case, the model included *DiscrMath* and *DataStruct* in the courses that influence *AI*, but that are not direct prerequisites. However, by inspecting other courses' prerequisites we will discover that they are indeed indirect prerequisites and the student must have attended those courses before taking *AI*.

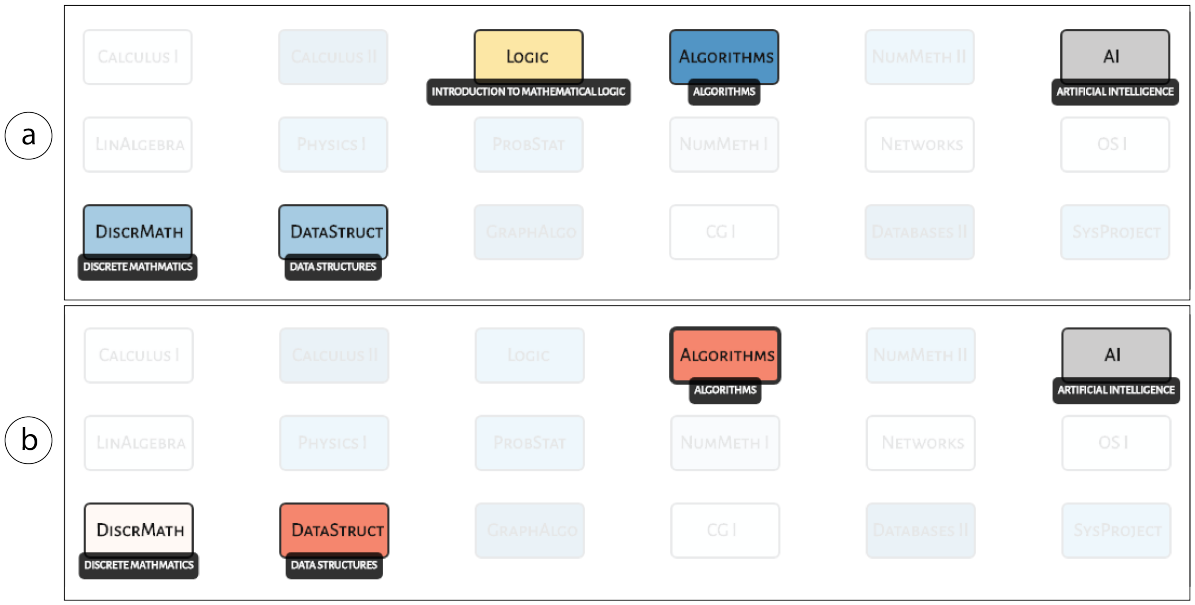

Figure 24 – Inspecting the courses influencing *AI*: a) Course influence view and b) Influence value view for *AI*.

5.5 Summary

In this chapter, we wanted to verify whether the structure of the curriculum could be inferred from the data. Our approach was based on the Synthetic Control Method (SCM), which builds a linear model describing the relation between courses based only on student performance information. We compared the proposed model with a linear regression model with positive coefficients, achieving better results. We also developed a web tool to visualize the results. This tool allows for contrast and comparison between the official curriculum and the model built based on the data and can be used by educators to evaluate the current structure of the curriculum, find hidden relations between courses and build hypotheses about the model.

6 CONCLUSION AND FUTURE WORK

The evasion is a challenge present in many universities around the world. Today, the amount of educational data available and improvements on computers capacity allowed researches to develop a computational approach to minimize this problem. Learning Analytics has been growing in importance by making possible a personalized education to students, using Data Mining, Machine Learning and Data Visualization techniques, and has been playing an fundamental role in attacking the dropout problem.

In this dissertation, we showed that evasion also affects the Computer Science program at UFC. Almost half of the students from 2005 to 2015 abandoned it, most of them in the first year, which leads the director to review hypotheses about the reasons and build strategies.

In Chapter 3, the data was analyzed regarding the context of the program and the students. The dataset collected has academic and demographic anonymous information about the students, which has been observed to be very useful on giving an overview about the possible causes of dropout. Then, we explored Learning Analytics techniques in order to minimize evasion, regarding some known issues detected by the program director. First, in Chapter 4, it was proposed a new approach to the problem of evading students, classifying in three groups: the ones who will certainly get their diploma, the ones who will certain drop out of college and the ones whom we are not sure about their future. This approach gives the possibility to educators to follow more closely the students who are trying to finish the course, but for some personal or any other problem, they are not getting good grades, and, in turn, can be a trigger to drop out.

The experiments were done with several binary classifiers, which classified the students into two groups and rejected those that are not clear in which group they should belong to. They achieved an accuracy value greater than 80%, which is excellent. Then, we validated our approach with the Accuracy Rejection curve. Also, the hypothesis about the rejected students to be the ones who might need counseling was verified by applying our approach on a sample of 32 students from the 2015 class, followed by a deep analysis using their academic data. This is a different approach from all the previous techniques discussed in Chapter 2. We believe that it is very useful for programs with limited human resources, providing a way to prioritize the students that need intervention.

Finally, regarding the curriculum problem that was probably a cause for the great number of students retained (see Figure 13), its structure was evaluated by building a model based on students' academic performance data. Two linear models were built, describing the

relationship between courses: the Linear Regression with positive weights and the Synthetic Control Method. The experiments showed that SCM obtained a F1-score higher than the linear regression, indicating more proximity with the reality, despite obtaining the same value for the average error.

Besides providing a good approach, it is important to show the results in a appropriate way to educators. An interactive visualization tool was developed to compare the actual curriculum with the model. Also, it allowed to quickly see which courses influence the result of others according to students' performance. This visualization allowed to promote an interesting discussion about the Computer Science curriculum and to observe that some changes made in the 2016.1 curriculum (recall Section 3.2) were also detected by the model.

As future work, it is possible to acquire other datasets at UFC to see if the same behavior observed with the Computer Science students is also present in other programs. Furthermore, the department could plan on devising a more thorough study to follow the rejected students for a year and see if we obtain positive results. Also, we would like to use the data to build customized curricula, suggesting the best order for taking the courses, respecting the curriculum's constraints, and that would improve students' performance. This would require devising a more complex visualization tool. We are also considering ways to incorporate other information about the courses and how to encode this information to try to improve our F1 scores.

REFERENCES

ABADIE, A.; DIAMOND, A.; HAINMUELLER, J. Synthetic control methods for comparative case studies: Estimating the effect of california's tobacco control program. **Journal of the American Statistical Association**, v. 105, n. 490, p. 493–505, 2010.

ABADIE, A.; GARDEAZABAL, J. The economic costs of conflict: A case study of the basque country. **American Economic Review 93**, Poder Executivo, 2003. 113-132.

ANURADHA, C.; VELMURUGAN, T. A data mining based survey on student performance evaluation system. **5th IEEE International Conference on Computational Intelligence and Computing Research, IEEE ICCIC 2014**, n. December 2014, p. 43–47, 2015.

APARECIDA, C.; BAGGI, S.; LOPES, D. A. Evasão e Avaliação Institucional: Uma Discussão Bibliográfica. 2011.

BADR, A.; DIN, E.; ELARABY, I. S. Data Mining: A prediction for Student's Performance Using Classification Method. **World Journal of Computer Application and Technology**, v. 2, n. 2, p. 43–47, 2014.

BALANIUK, R. *et al.* Predicting evasion candidates in higher education institutions. In: **Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)**. [S.l.]: Springer Berlin Heidelberg, 2011. v. 6918 LNCS, p. 143–151.

BARBOSA, A. *et al.* Using Learning Analytics and Visualization Techniques to Evaluate the Structure of Higher Education Curricula. In: **Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação - SBIE)**. [S.l.: s.n.], 2017. v. 28, n. 1, p. 1297.

BAYER, J. *et al.* Predicting drop-out from social behaviour of students. **Proceedings of the 5th International Conference on Educational Data Mining**, n. Dm, p. 103–109, 2012.

BRITO, D. M. de *et al.* Predição de desempenho de alunos do primeiro período baseado nas notas de ingresso utilizando métodos de aprendizagem de máquina. **Anais do Simpósio Brasileiro de Informática na Educação**, v. 25, n. 1, p. 882–890, 2014.

CAMPAGNI, R. *et al.* Data mining models for student careers. **Expert Systems with Applications**, Elsevier Ltd, v. 42, n. 13, p. 5508–5521, 2015.

CHOW, C. On optimum recognition error and reject tradeoff. **IEEE Transactions on Information Theory**, v. 16, n. 1, p. 41–46, 1970.

COSTA, F. *et al.* Predição de sucesso de estudantes cotistas utilizando algoritmos de classificação. **Anais do Simposio Brasileiro de Informática na Educação**, v. 26, n. Sbie, p. 997, 2015.

DIETZ-UHLER, B.; HURN, J. E. Using Learning Analytics to Predict (and Improve) Student Success: A Faculty Perspective. v. 12, n. 1, 2013.

FERGUSON, R. Learning analytics: drivers, developments and challenges. **International Journal of Technology Enhanced Learning**, Inderscience Publishers, v. 4, n. 5-6, p. 304–317, 2012.

FILHO, R. L. L. e. S. *et al.* A evasão no ensino superior brasileiro. **Cadernos de Pesquisa**, v. 37, n. 132, p. 641–659, 2007.

FUMERA, G.; ROLI, F. Support vector machines with embedded reject option. In: **Proceedings of the 1st International Workshop on Pattern Recognition with Support Vector Machines (SVM'2002)**. [S.l.]: Springer, 2002. p. 68–82.

GAMA, S.; GONCALVES, D. Visualizing large quantities of educational datamining information. In: **Proceedings of the International Conference on Information Visualisation**. [S.l.]: IEEE, 2014. p. 102–107.

GÉRYK, J. Using visual analytics tool for improving data comprehension. **International Educational Data Mining Society**, ERIC, 2015.

GRELLER, W.; DRACHSLER, H. Translating Learning into Numbers : A Generic Framework for Learning Analytics Author contact details :. **Educational Technology & Society**, v. 15, n. 3, p. 42 – 57, 2012.

HINRICHS, P. The effects of affirmative action bans on college enrollment, educational attainment, and the demographic composition of universities. **The Review of Economics and Statistics**, v. 94, n. 3, p. 712–722, 2012.

INEP. **Conceito ENADE 2014**. 2017. Access date: 21 dez. 2017. Disponível em: <http://portal.inep.gov.br/conceito-enade>.

JORDÃO, V.; GAMA, S.; GONÇALVES, D. EduVis: Visualizing educational information. **8th Nordic Conference on Human-Computer Interaction, NordiCHI 2014**, p. 1011–1014, 2014.

KANTORSKI, G. *et al.* Predição da Evasão em Cursos de Graduação em Instituições Públicas. **Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação - SBIE)**, v. 27, n. 1, p. 906, 2016.

LAK. **First International Conference on Learning Analytics and Knowledge Call For Papers**. 2011.

LAKKARAJU, H. *et al.* A Machine Learning Framework to Identify Students at Risk of Adverse Academic Outcomes. **KDD '15 Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining**, p. 1909–1918, 2015.

MARIA, W. *et al.* Rede Bayesiana para previsão de Evasão Escolar. **Anais dos Workshops do Congresso Brasileiro de Informática na Educação**, v. 5, n. 1, p. 920, 2016.

MÁRQUEZ-VERA, C. *et al.* Predicting student failure at school using genetic programming and different data mining approaches with high dimensional and imbalanced data. **Applied Intelligence**, 2013.

MARQUEZ-VERA, C.; ROMERO, C.; VENTURA, S. Predicting School Failure Using Data Mining. **Proceedings of the 4th International Conference on Educational Data Mining**, n. December, 2011.

MEC. **Comissão especial de estudos sobre a evasão nas universidades públicas brasileiras**. Brasília, DF: ANDIFES/ABRUEM, SESu, MEC, 1995.

MESQUITA, D. P.; ROCHA, L. S.; GOMES, J. P. P.; NETO, A. R. R. Classification with reject option for software defect prediction. **Applied Soft Computing**, v. 49, p. 1085 – 1093, 2016.

MOSELEY, L. G.; MEAD, D. M. Predicting who will drop out of nursing courses: A machine learning exercise. **Nurse Education Today**, v. 28, n. 4, p. 469–475, 2008.

MUNZNER, T. **Visualization Analysis & Design**. 1. ed. Boca Raton, FL: CRC Press - Taylor & Francis Group, 2014. (A K Peters Visualization Series).

OLIVEIRA, A. C. d. *et al.* Efficient minimal learning machines with reject option. In: **2016 5th Brazilian Conference on Intelligent Systems (BRACIS)**. [S.l.: s.n.], 2016. p. 397–402.

PASCOAL, T. *et al.* Evasão de estudantes universitários: diagnóstico a partir de dados acadêmicos e socieconômicos. **Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação - SBIE)**, v. 27, n. 1, p. 926, 2016.

PECHENIZKIY, M.; TRCKA, N.; De Bra, P.; TOLEDO, P. CurriM: Curriculum Mining. **Proceedings of the 5th International Conference on Educational Data Mining**, n. i, p. 1–4, 2012.

PMF. **Desenvolvimento humano, por bairro, em Fortaleza**. 2017. Access date: 30 nov. 2017. Disponível em: <http://en.calameo.com/read/0032553521353dc27b3d9>.

PROGRAD. **IRA – Índice de Rendimento Acadêmico**. 2017. Access date: 28 nov. 2017. Disponível em: <http://www.prograd.ufc.br/perguntas-frequentes/perguntas-frequentes-ira/>.

PROGRAD. **Vida Acadêmica**. 2017. Access date: 28 nov. 2017. Disponível em: <http://www.prograd.ufc.br/perguntas-frequentes/vida-academica/>.

ROBERTO, H.; ADEODATO, P. J. L. A data mining approach for preventing undergraduate students retention. **IEEE World Congress on Computational Intelligence**, p. 1–8, 2012.

SCHMIDT, W. F.; KRAAIJVELD, M. A.; DUIN, R. P. W. Feedforward neural networks with random weights. In: **Proceedings., 11th IAPR International Conference on Pattern Recognition. Vol.II. Conference B: Pattern Recognition Methodology and Systems**. [S.l.: s.n.], 1992. p. 1–4.

SOUSA, R.; MORA, B.; CARDOSO, J. S. An ordinal data method for the classification with reject option. In: **Proceedings of the International Conference on Machine Learning and Applications (ICMLA'09)**. [S.l.: s.n.], 2009. p. 746–750.

TAMHANE, A. *et al.* Predicting student risks through longitudinal analysis. **Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '14**, p. 1544–1552, 2014.

TAN, P.-N.; STEINBACH, M.; KUMAR, V. **Introduction to Data Mining, (First Edition)**. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2005. ISBN 0321321367.

WANG, R.; ZAÏANE, O. R. Discovering Process in Curriculum Data to Provide Recommendation. In: **Proceedings of the 8th International Conference on Educational Data Mining**. [S.l.: s.n.], 2015. p. 580–581.

WORTMAN, D.; RHEINGANS, P. Visualizing trends in student performance across computer science courses. **ACM SIGCSE Bulletin**, v. 39, n. 1, p. 430, 2007.

WU, K. **Modeling an academic curriculum plan as a mixed-initiative constraint satisfaction problem**. Tese (Doutorado) — School of Computing Science-Simon Fraser University, 2005.