TÉCNICO LISBOA

Profiling Users in Gamification

Sofia Mendes Gonçalves

Thesis to obtain the Master of Science Degree in

Information Systems and Computer Engineering

Supervisor: Professor Cláudia Martins Antunes

Examination Committee

Chairperson: Professor João António Madeiras Pereira
Supervisor: Professor Cláudia Martins Antunes
Member of the Committee: Professor Rui Miguel Carrasqueiro Henriques

November 2019

Acknowledgments

I would like to thank my advisor, Professor Cláudia Antunes, for her guidance, support and knowledge transmitted throughout the development of this thesis. Her encouragement, patience and trust were crucial in several moments of self-doubt.

To my parents, for the continuous support on my decisions and encouragement throughout my academic life. To my brother and sister, who are always available to cheer me up.

Abstract

The implementation of gamification in an educational context had a significant impact not only on the motivation but also on the learning performance of students. Gamified environments revealed different behaviors from different students in the same conditions, making it interesting to define student profiles.

Educational Data Mining concerns the development of methods for exploring data that come from educational environments. It aims to better understand the student's learning process and to identify their learning settings to improve educational outcomes.

This thesis proposes the exploration of the knowledge discovery process to build a predictive model to early detect students' learning profiles on data collected from a gamified course. Given the sparsity of the data, a prime consolidation in a Data Warehouse was performed to ease the data mining process. In the profiling phase, we compared the performance of four learning algorithms on two differently labeled datasets, over several phases throughout the semester.

The results showed that the characteristics of the datasets and the selected hyperparameters consisted of essential factors to varying the performance of the classifiers.

Our approach ensures the possibility of predicting the students' learning profiles by five weeks into the semester.

Keywords

Gamification; Gamified Course; Data Warehouse; Profiling; Student Modeling

Resumo

A implementação da gamificação num contexto académico teve um impacto significativo, não só na motivação, como no desempenho da aprendizagem dos alunos. Ambientes gamificados revelaram diferentes comportamentos da parte de diferentes alunos nas mesmas condições, tornando a definição de perfis de alunos um assunto de interesse.

Educational Data Mining diz respeito ao desenvolvimento de métodos para explorar dados provenientes de ambientes educativos. O objetivo é compreender melhor o processo de aprendizagem do aluno e identificar os seus métodos de aprendizagem de forma a melhorar os resultados a nível académico.

Esta tese propõe a exploração do processo de descoberta de conhecimento de forma a construir um modelo preditivo para detectar os perfis dos alunos o mais perto do início do semestre possível, a partir de dados extraídos de uma unidade curricular gamificada. Dada a natureza esparsa dos dados, foi efectuada uma consolidação num armazém de dados, de forma a facilitar o processo de *data mining*. Na fase de detecção de perfis, comparámos o desempenho de quatro algoritmos de aprendizagem em dois conjuntos de dados rotulados de forma diferente em várias fases do semestre.

Os resultados mostraram que as características dos conjuntos de dados e dos hiperparâmetros selecionados consistem em factores de maior impacto do desempenho dos classificadores.

A nossa abordagem garante a possibilidade de prever o perfil de aprendizagem dos alunos ao fim de cinco semanas, após o início do semestre.

Palavras Chave

Gamificação; Unidade Curricular Gamificada; Armazém de Dados; Detecção de Perfis; Criação de Modelos

Contents

List of Figures

List of Tables

Acronyms and Abbreviations

AIC	Akaike Information Criterion
CBE	Computer-Based Educational Systems
COS	Cosine Similarity
DW	Data Warehouse
EDM	Educational Data Mining
ETL	Extraction-Transformation-Loading
IEDMS	International Educational Data Mining Society
IST	Instituto Superior Técnico
LDA	Linear Discriminant Analysis
LR	Logistic Regression
LSVM	Linear Support Vector Machines
MAD	Mean Absolute Difference
MAE	Mean Absolute Error
MCP	Multimedia Content Production
MOOCs	Massive Open Online Courses
MSc	Master of Science
NB	Naïve Bayes
PC	Pearson's Correlation Coefficient
RFE	Recursive Feature Elimination

RF	Random Forests
RMSD	Root Mean Squared Difference
SMOTE	Synthetic Minority Oversampling Technique
SVM	Support Vector Machines
XP	Experience Points

1

Introduction

Over the years, the integration of game elements in contexts other than games has shown promising results in terms of improvement of people's engagement in a specific activity [1] [2]. This use of game design elements in non-game contexts is called *gamification* [3].

Gamification emerged and started to be applied in several areas, such as marketing, productivity, health, finance, education, news, media, among others. In education, it plays an important part in increasing student engagement and improving the learning experience, since today's students have grown in more interactive environments and consider traditional learning demotivating [4]. However, different students have different learning styles, and it is important to explore the pedagogical effects of gamification in terms of impact in each student [5].

Educational Data Mining (EDM) is a discipline concerned with the development of methods to explore data that comes from educational settings in order to better understand the students. Therefore, through the application of data mining techniques, it is possible to group students according to their characteristics, to predict student's performance in terms of behavior, score, and mark, and to create profiles by developing cognitive models of students considering their motivation, learning styles and learning behaviors [6].

The objective of this work is to explore the knowledge discovery process to do an early detection of student's learning profiles, based on their interaction with a gamified course, and further study how to adapt the learning experience to its individual needs to improve the learning experience. This will assist educators on identifying with which type of students they are dealing with in order to take adequate measures to help them obtain better results.

In the last decade, several studies tried to explain how the behavior of the student affects its performance, and to predict that performance through simple formulas [7] or tabular tools [8], disregarding the temporal nature of the data. Others explored the sequential nature of data as an important factor in student profiling [9] [10], the anticipation of results based on the exploration of temporal precedences [11], and few attempted to use sequence classifiers [12] and clusters [13] with promising results.

My proposal is to create a predictive model of student behavior by building classifiers that take into account the temporality and sequentiality of the data for anticipating and improving student profiles.

The rest of the thesis is organized as follows: Chapter 2 introduces the concept of gamification and describes its applications in several areas, with emphasis on the educational context; Chapter 3 presents insights to related work on the area of Educational Data Mining; after this, in Chapter 4 we propose a model for a Data Warehouse to consolidate data in a more suitable way for further student profiling; and in chapter 5 we present the results of applying our approach in order to early predict the students' profiles; finally, chapter 6 concludes this work and proposes some potential expansions for future work.

2

Gamification

Contents

The term *gamification* originated online back in the year 2008 but it was only in the second half of 2010 that it saw widespread adoption [1]. Although the term is relatively new, the idea of gamification is not. Games and simulations have been used by the military for hundreds of years which made them pioneers in the use of these tools to explore its applications [3]. Also, there are several books dated from the 1960's that explore how games affect life and psychology and even movies from the 1980's that approach this theme [2]. A *game* is an interactive activity characterized by a rule-based system which continually provides challenges to one or several players. They are portrayed by a fictional context in the form of a story, graphics, and music [14] and encourage players to compete towards goals and involve them in an active learning process in order to fulfill those goals and master the game mechanics [14] [15].

As defined by Deterding et al. (2011), *gamification* consists of the use of game design elements in non-game contexts [3]. Some of the most common elements are the ones proposed by Reeves and Read [16] as the "Ten Ingredients of Great Games":

Self-representation with Avatars

By definition, an *avatar* consists of a figure or icon that represents the player in the game. The opportunity to represent himself and exert control over that representation affects the psychology of using technology. It allows the player to become part of the scene, speak and interact with other players, making him an element of the narrative instead of being just a recipient of a story written by someone else, leading to more engagement within the game.

Three-dimensional Environments

Visually rich three-dimensional environments are built for the previously mentioned avatars to live, increasing the interestingness of the game by allowing the player to navigate in a game space with similar properties to the real world. The environment is created using 3D graphical models rendered onto a two-dimensional screen in order to simulate depth, making the virtual place as close to the real world as possible.

Narrative Context

It is essential in a game to have a good back-story because narratives guide the player's actions, as well as organize the character roles within the game. A good narrative allows the player always to have access to the information about his character or his team and helps them to act according to the story. It has important psychological advantages because stories make people more engaged in the game.

Feedback

Feedback in games is given by progress bars, status gauges and also big or small colored numbers, all displayed in a dashboard, in a relatively easy way to understand if the player is doing things right or wrong. Usually, numbers refer to the player's health, the strength of an attack, the amount of currency accumulated so far or even to the time left for a mission. The player's awareness of his progress given

[1] https://trends.google.pt

by this quantitative feedback is also important to increase his engagement.

Reputations, Ranks, and Levels

Reputation is important to inform not only the place of a player in a game hierarchy but also to show other players a specific player's competencies, talents, masteries, and special experiences. Being aware of the reputation of a player can help us to make choices and understand if it is relevant or not to interact with him and engage in a social scene. One of the main objectives of a game is to augment reputation or increase levels and share them with fellow players and friends. Ranks and levels are also responsible for transmitting feedback to the player and making him aware of his progress within the game.

Marketplaces and economies

Currency systems are also an essential feature in games since they support the existence of an economic scoring which might include revenue, profit, salary, and savings. In multiplayer games, currency systems allow players to perform trades amongst each other and help them understand the value of their belongings, creating an economic behavior similar to the one presented in real life.

Competition Under Rules that are Explicit and Enforced

The primary goal of a game is to win. However, in single or multiplayer games, there is some variance between players' competitive urges, so it is important to implement rules. Rules are responsible for games to work and being aware of what is possible to do in the game is part of the fun of game exploration. Having a set of well known and enforced rules establish a sense of fairness in gameplay.

Teams

A few years back, multiplayer games were not as popular as they are nowadays. Whether played at a computer, console or phone, multiplayer games had become more attractive than single player games when LAN (local area network) parties became popular because players were able to connect their devices allowing them to compete. Nowadays, broadband is accessible and affordable for many people and allow players to play with and against each other over the Internet. Because of this, social relationships are formed in the game and can be as engaging as those in real life. Players interact with each other, reveal personalities and share experiences while collaborating to reach team goals.

Parallel Communication Systems that can be Easily Configured

Although visuals in games are important, it is the written and spoken communication that enables a significant part of the social engagement. In many of the existing multiplayer games, players can communicate with each other, either by voice or text, and these platforms usually offer an easily configurable communication experience that may adapt to the player's style.

Time pressure

Being on the clock increases the fun of the game and challenges the players to do things in the time allotted resulting in the sense of accomplishment. In multiplayer games, it encourages players to team up and collaborate in order to achieve a goal that was under uncertain winning conditions.

2.1 Applications

Although these ten elements are proposed as relevant to create great games, they should not be understood as strictly required to be implemented all together as a recipe. For example, avatars are very common in roleplaying games, but when we are playing card games, they are not necessarily needed. Game elements shall be treated as a set of building blocks that can be combined according to the type of game that the creators are aiming at [3].

Gamified applications consist of a system that merely uses several elements from games with an intention different from creating an entertainment game in order to increase engagement, the joy of use, and to improve the user experience [3].

Gamification may be applied to almost every aspect of life, such as marketing, productivity, health, finance, education, news, media, among many others. Many companies are starting to gamify their services since it supports user engagement [3], creates close relationships between the platform and the users, also increasing the platform popularity, and enhances positive patterns in service use [14]. Moreover, in order to understand if gamification is, in fact, useful, there is also an increasing amount of investments being made by a remarkably large number of firms into gamification-related efforts [17].

2.1.1 Gamification in Marketing

Gamification has been applied in Marketing very widely. There are several applications in the most varied areas that use gamification for marketing and customer retention purposes. Some of these applications implement badges and offer bonuses as the users achieve specific objectives. However, neither points nor badges in any way constitute a game. This light form of gamification is called pointification. An example of pointification is TV Time [2], an app to track television series and shows, gives badges to their users when they perform actions such as voting on their favorite character, commenting on a show or watching a specific number of episodes of a show in a row. Pointification can also be used as a tool to encourage a desirable website usage behavior, which is the case of Stack Overflow, a programming question-and-answer website, where users receive points for answering usefully to questions put by other users, and when achieving a specific point threshold, they can earn additional privileges, such as being able to monitor the site and help to keep it on track [18].

2.1.2 Gamification in Health

Many workout applications use pointification to encourage users to exercise and take care of their overall health. Users are rewarded with points for performing specific activities during their workouts and increase level according to the collected points. These sort of applications usually have achievements

[2] https://www.tvtime.com

that give badges for accomplishing fitness milestones. Although applications tend to implement pointification in order to achieve a goal, it may also work the other way around. Take as an example Pokémon Go, a location-based game which subtly promotes physical exercise as it enforces users to move around the real world to collect creatures. In this case, it is valid to call it gamification as it invoves a gameful experience that supports the user's overall value creation. Studies reveal that the app was responsible for an increase of more than 25% of the number of steps taken, compared to the user prior activity [19].

2.1.3 Gamification in Education

In the education system, students are ranked in a class according to a grade-point average (GPA) which may be comparable to a high score in a game. Honorable mentions and scholarships can be compared to earning badges or bonuses [20]. This is why games and their usage in education have been studied for years. In 2009, Quest to Learn was set up by the New York City Department of Education with funding from the Bill and Melinda Gates Foundation and from the MacArthur Foundation, becoming the first school ever centered around game-based learning. The main goal was to create a space where students could learn to collaborate, communicate and solve real-world problems, in a more engaging and relevant way, taking into account that games make users aware if they are succeeding or failing and, in case of failure, allow them to try again and motivate them to succeed [21]. In 2011, Microsoft released Ribbon Hero 2, a game that works as an add-on for the Office service that challenges people to explore the platform and helps them use it effectively [22].

2.2 Psychological Impact of Gamification

Gamification has become a trending topic since it has proven that when implemented, it increases user activity and retention [1] [3], engages people, motivates action, promotes learning and solves problems [23].

Hamari el al. (2014) refers to the use of game mechanics as a means to implement motivational affordances [17]. Motivational affordance results in psychological outcomes which cause further behavioral outcomes: since the user is enjoying the service as a "gameful" experience, he is intrinsically motivated and is having fun, and this leads to a positive result, such as being prepared for a task or being able to work together as a team. These positive results, however, may depend on the motivation of the users and on the nature of the gamified system and may also occur only for a short time since the user may be thrilled because of the novelty effect. In cases where gamification is removed when the user is engaged, it usually has a negative impact since he gets invaded by a sense of loss because he can no longer track his performance and progress by checking his earnings, badges or points.

Lee et al. (2011) noted that the motivation factor acquired from games results from the impact of the

game on the cognitive, emotional and social areas of players [24]. The cognitive area is stimulated by the rule system provided by the game along with a set of tasks and goals that are useful to guide the players through it. Gee et al. (2003) referred to these tasks as *cycles of expertise* where the players repeatedly try to complete them but may fail until the necessary skill is acquired [25]. The task sequence is not necessarily linear, as it is preferred that the player can have the freedom to choose which ones he wants to accomplish, allowing him to be aware of his skills and to act according to his personal preferences [14].

In the emotional area, the impact results from success or failure. As mentioned before, games try to ensure that the player is having fun and being engaged in order to trigger positive emotions. This is generally granted by reward systems, where players may receive points, badges, or items which might be used in the game, to give immediate recognition to the success of the player's actions. However, when players fail, the emotions tend to be negative, such as becoming anxious, angry or despaired. It is not desirable that these feelings evolve into frustration which is why the tasks are carefully planned to fit the player's skills and the penalties on failed tasks are generally low, in order to encourage the player to try again and experiment other ways to succeed.

Finally, the social area is stimulated by the possibility of interacting with other players, which is possible in a multiplayer environment. In these types of games, players are able to cooperate with each other towards a common goal, to compete with them in order to achieve a top position in the leaderboard or even to chat [14].

2.3 Gamification in an Educational Context

Along the years, the number of papers published on gamification in an educational context has been growing suggesting that the theme is becoming a popular subject for academic inquiry. In 2015, a search made by Dicheva et al. (2015) returned around 5.000 results for papers that discuss explicitly gamification in educational contexts [26]. There are several studies that unveiled many advantages of implementing game elements in education, as students are able to get immediate feedback, have access to information on demand, develop an important role in a community, are able to take control of and evaluate their own learning behavior and even improve their capacity of working as a team [14]. In online learning, gamification solves the problem of the lack of motivation on the part of the student, related with the limited capacity of interacting with the teacher or with classmates.

On the other hand, there are also some issues on this matter such as the need to understand how to gamify the educative content, the possibility that students may not be able to apply the knowledge and skills learned outside the gamified environment or even to learn more in a self-regulated way afterwards, the need for the presence of a teacher in the environment in order to control the performance of the

students, and also the cost of implementation of a technological infrastructure to support gamification [14].

Hamari et al. (2016) studied the engagement, flow, and immersion in game-based learning to understand if challenging games effectively help students learn [27]. Although games are designed for entertainment and leisure, serious games and game-based learning are designed for training and educating. Flow refers to the psychological state that results from the integration of both work and play [28] and reveals elevated enjoyment during the practice of an activity, affecting consequent basic outcomes. But flow and engagement do not rely only on the student. Students' concentration and engagement raise according to the level of the challenge in the classroom. Although some students consider challenges unpleasant or arduous, the majority prefers to face a challenging task and value themselves more when achieving their goal. The perception of their competence leads to a feeling of accomplishment and raises motivation [29]. Another important aspect is that the level of the challenge shall be adjusted to the skill of the student. Video games allow the player to adjust the level of the challenge according to the increase of the player's skills. During our educational path, we started with something as simple as learning letters and digits, then learned to build simple phrases and from that moment on were able to understand math, science, history and many other subjects and could study each of these in much deeper levels. Our level of skill increased to match the challenge which increased flow, learning outcomes and even satisfaction. Also, the challenge is responsible for increasing learning since the student applies a wide range of strategies in order to succeed.

So, game-based approaches combined with motivation techniques are a promising replacement for overly theoretical classes with possible unfavorable schedules [4]. Cheong et al. (2013) conducted a study where he used a gamified quiz to evaluate IT undergraduate students in which they reported that the quiz improved their grades, learning effectiveness and also their enjoyment and engagement [30]. Domínguez et al. (2013) proposed a gamified approach to an e-learning course where students had the alternative to take exercises either by the conventional education system, in which they had to read a PDF file, or via a gamified system. Results showed that students opting by the gamified approach obtained better exam grades and reported higher engagement [14]. Glover et al. (2016), in a similar study, chose to add achievement badges to an online learning environment to support and encourage participation in learning activities and also obtained positive results [31].

All previously mentioned studies used a "one-size-fits-all" approach and did not explore the pedagogical effects of gamification in terms of its impact on different students in different ways. To try to bridge that gap, Barata et al. (2013) gamified an MSc course by adding game elements such as experience points and levels, badges and a leaderboard [5]. Also, the course included a set of challenges and quests which granted the students experience points when completed. Using machine learning techniques, and taking into account the experience points accumulated over time, two different student

types where identified: the Achievers, which tried to obtain every badge and performed the best, and the Underachievers, who just made an effort to pass the course.

3

Literature Review

Contents

3.1 Educational Data Mining

Studies based on students' interactions with instrumental educational software and online learning as the ones previously mentioned generate a considerable amount of data, which can be explored and exploited in order to understand how students learn [32] [6]. The exponential growth of educational data and its use to investigate scientific questions within educational research consists of one of the biggest challenges faced by educational institutions [6].

Educational Data Mining (EDM) is defined by the International Educational Data Mining Society (IEDMS) as "an emerging discipline, concerned with developing methods for exploring the unique types of data that come from educational settings and using those methods to understand the students better, and the settings which they learn in." The IEDMS is responsible for promoting scientific research and development in the field of educational data mining and for supporting collaboration between the members of the community through the organization of the EDM conference, which has occurred every year since it was first held in 2008, in Montreal, Quebec, and through the assembling of several journal articles submitted by the community in order to create the Journal of Educational Data Mining. This society supports the sharing of data techniques and brings together a community of computer scientists, learning scientists and researchers [32] [33].

3.1.1 Types of Educational Environments

Educational environments generate data which can be analyzed and converted through the process of EDM into useful information depending on its nature and on the specific problems and tasks that are meant to be resolved, having a significant impact on educational research and educational practice. These educational environments may be found not only in computer-based education, as previously mentioned, but also in traditional education [33].

3.1.1.A Traditional Education

Traditional Education refers to the conventional classrooms with long-established customs that have been deemed to be appropriate by society. These environments consist of well-structured lectures composed by a relatively small group of students responsible for their seat work, and base themselves on face-to-face contact between them and the responsible teacher. With this kind of educational system, it is possible to gather information not only from individual classes, such as student attendance and marks, but also from the administrative data in the traditional databases of the educational institution, where it is possible to access information about the student, its educator, his classes and schedule, and from the online information on the school web page and the course content page. In conventional classrooms, the analysis of student performance and the monitorization of the student's learning process is made

through observation, but the implementation of computer-based educational systems can complement face-to-face sessions [33].

3.1.1.B Computer-Based Educational Systems (CBE)

CBE consist of the use of computers in education to provide direction, instruct the student or manage the given instructions. Initially, these CBE systems ran on a local network and were simple stand-alone educational applications. With the globalization of the Internet, e-learning, online training, and online instruction systems emerged leading to a plethora of web-based educational systems and increased the need for the application of artificial intelligence techniques for student modeling or to develop new intelligent and adaptative educational systems [33].

3.1.2 Goals of EDM

Data gathered from schools, colleges, universities, and other learning institutions that work modern forms and methods of teaching is not restricted to the interactions between the student and the system, such as the results of online quizzes or its behaviour while navigating on the platform. There are many ways of obtaining relevant information, and that may include data from collaborative students, extracted from text forums where they communicate and clarify doubts, administrative data, related to the school or the teacher, demographic data, such as gender, age, and student grades, and data from student affectivity, which can be measured by its motivation, for example. So, there are many different types of data available for mining that although being specific to the educational area, have intrinsic semantic information and multiple levels of hierarchy, which means that can be related to the student, to the assignments, or to the questions [6] [33].

The existence of data from thousands of institutions and students with similar learning experiences but in different contexts gives leverage for studying contextual factors on learning and learners [34]. EDM has the potential to analyze important questions in individual differences [34] and enables data-driven decision-making to improve the current education methods and learning materials [33]. However, depending on the viewpoint of the final user and on the problem that needs solving, it is possible to consider many specific objectives in EDM such as understanding how to structure or restructure classes, organize the evaluation methods and distribute the materials based on the usage of the platform and on the performance data; identifying students that need more feedback, study advice or other type of help; deciding which kind of feedback, advice or help would be more effective; and even understanding how to help learners in finding useful material, either individually or in collaboration with colleagues [33].

3.1.3 Tasks

Baker et al. (2010) and Romero et al. (2013) assess some popular tasks within EDM where some of them are acknowledged to be universal across types of data mining, which is the case of classification/prediction, clustering, outlier detection, social network analysis, process mining, and text mining. Others have particular prominence within EDM, such as distillation of human judgment, discovery with models, and knowledge tracing [34] [6].

- **Classification/Prediction -** Prediction aims to infer a target attribute (predicted variable) from a combination of other aspects of the data (predictor variables). Classification, regression, and density estimation are some types of prediction methods. In classification, the predicted variable is a categorical value, in regression, a continuous value and in density estimation, the predicted value is a probability density function. It has been used for forecasting student performance and to detect student behavior.

- **Clustering -** The goal of clustering is to group instances and identify those groups according to their similarity. In order to decide how similar are the instances, a measure of distance is defined. It is particularly useful when the most common categories within the data are not known in advance. Once the instances are grouped, or the clusters are formed, it is possible to classify new instances by determining the closest cluster. In EDM, clustering is used for grouping similar course materials or students with similar learning and interaction patterns.

- **Outlier Detection -** An outlier is a data point significantly different from the rest of the data. Outlier detection can be used in EDM for detecting students with learning difficulties, deviations in the learner's or educator's actions or behaviors, or even to detect irregular learning processes.

- **Relationship Mining -** The goal of this data mining technique is to discover relationships between variables in order to understand which variables are most strongly associated. Relationship mining techniques include association rule mining, to find if-then rules of the form that if some set of variable values is found, another will have a specific value, sequential pattern mining, to find temporal associations between variables, correlation mining, to find linear correlations between variables, and causal data mining, to find causal relationships between variables. This technique is used to identify relationships in learners' behavior patterns and finding learning difficulties or mistakes that frequently occur together.

- **Social Network Analysis -** The use of social network analysis is useful to understand and measure the relationships between the entities of the network. The entities, or individual actors, are represented by the nodes, and the links between them represent relationships, such as friendship, or organizational position. It is used in EDM for analyzing the structure and relationships in

collaborative tasks and interactions with communication tools.

- **Process Mining -** The goal of process mining is to extract process-related information from event logs recorded in a computer-based environment to have a clear representation of the whole process. It can be used for reflecting students behavior concerning their examination traces consisting of a sequence of the course, grade and timestamp triplets for each student.

- **Text Mining -** Text mining is the technique of deriving high-quality information from text and include text categorization, text clustering, extraction of concepts, production of granular taxonomies, sentiment analysis, document summarization, and entity relation modeling. It is useful to analyze content from discussion boards, forums, chats, and documents.

- **Distillation of Data for Human Judgment -** When data is appropriately presented, human beings can make inferences about it that are beyond the scope of fully automated data mining methods. The goal of distillation of data is to represent data in clear ways using summarization, visualization, and interactive interfaces to help the user detect the relevant data and to facilitate the decision-making process. This method has been used to help educators visualize and analyze the students' learning process.

- **Discovery with Models -** This method uses a model of a phenomenon developed via prediction, clustering or manual knowledge engineering as a component in another analysis such as prediction or relationship mining. It supports the identification of relationships between student's behaviors and personal characteristics, the analysis of research questions depending on the context, and the integration of psychometric modeling frameworks into machine-learning models.

- **Knowledge Tracing -** It is useful to estimate student mastery of skills that have been used in effective cognitive tutor systems and relates those skills with the logs of the students' correct and incorrect answers.

3.1.4 Applications

Romero et al. (2010) consider a variety of applications that have been resolved through the application of data mining in educational systems [6]. These educational data mining techniques include:

- **Analysis and Visualization of Data -** where the objective is to highlight useful information, such as reports and statistics, to help educators analyze the students' course activity and his interaction with the learning process in order to obtain a general view of his learning and to support decision making about how to modify the educational environment to improve that learning.

- **Providing Feedback for Supporting Instructors -** to support the course authors/teachers/administrators in decision making, such as how to improve the students' learning and organize the existing resources more efficiently, and help them take the appropriate action.
- **Recommendations for Students -** taking into account the students' personalized activities, the objective of this task is to make recommendations and also to adapt learning contents and interfaces to each particular student.
- **Predicting Student's Performance -** prediction estimates the unknown values of variables that describe the student. In the education field, it is interesting to predict values of performance, knowledge, score, and mark. Depending on the nature of the values, either being numerical/continuous or categorical/discrete, different procedures may be applied which will be explained further on this report.
- **Student Modeling/Profiling -** to develop cognitive models of students, considering their characteristics, such as motivation, satisfaction, learning styles, affective status, among others, and their learning behavior. This includes models of their skills and declarative knowledge.
- **Detecting Undesirable Student Behaviours -** undesirable behaviors include wrong actions, low motivation, misuse of the platform, cheating, dropping out of the course and academic failure. This task aims to discover or detect those students who present some problem, related to the previously listed ones.
- **Grouping Students -** this task's objective is to group students according to their personal characteristics. The obtained groups of students can be used by educators to develop a personalized learning system to promote active group learning.
- **Social Network Analysis -** a social network consists of a group of people connected by social relationships such as friendship, cooperative relations or informative exchange. It aims to study relationships between the individuals of that group.
- **Developing Concept Maps -** a concept map is a graph with concepts that present relationships between them and expresses the hierarchical structure of knowledge. The development of concept maps helps educators building concept maps using an automated process.
- **Constructing Courseware -** to help educators and developers to carry out the construction process of courseware and learning contents automatically and to promote the exchange of existing learning resources among the users or even different systems.
- **Planning and Scheduling -** to enhance the traditional educational process by planning future courses, helping the students with scheduling, planning the allocation of resources, developing

curriculum, amongst others.

3.2 Student Profiling: Related Work

Along with the increase of the popularity of educational games and game-based learning, increased the interest in education. Given the amount of data collected in these environments, it is possible to apply advanced Educational Data Mining techniques to better understand and predict student behavior. Knowing the set of methods usually applied in EDM and its applications in educational systems, several studies have been conducted to create student profiles automatically.

In the last decade, there have been several efforts to explain how the behavior of the student affects its performance and to predict that performance, so that the educator can know as soon as possible with which type of students he is dealing with, to identify if the student needs guidance or only to provide feedback. In these studies, data was mainly explored through simple formulas [7] or tabular tools [8], disregarding intrinsic temporal data.

Liang et al. (2017) analyzed the behavior characteristics of online learners in Massive Open Online Courses (MOOCs) and the factors affecting the student profile [7]. The authors started by classifying the student according to its age, defined twelve kinds of learning behavior, such as making notes, putting questions online or download courseware, and considered a duration set (timeslot) to be an essential parameter to evaluate the quality of online learning. To perform student profile analysis, the authors calculate the similarity in the behavior set of different students using the Jaccard similarity coefficient algorithm. This process is followed by a comparison with the online behavior characteristics and duration set of the learners, and if the properties are similar, the student is classified as a class, otherwise, if there are different properties, the student is added as a new class to the knowledge base. The Jaccard coefficient is then responsible for labeling the students as different types, depending on the behavior and the duration of the performance. The authors further performed learning attitude and duration of online learning behavior analysis with simple statistics based on students' opinions. Finally, in order to predict whether the student gets the certificate for being successful in the course or not, the authors propose the use of three classification models: Linear Discriminant Analysis (LDA), Logistic Regression (LR), and Linear Support Vector Machines (LSVM).

Results of this study regarding profiling show that students are labeled according to their learning performance such as "depth learning type" in case of a performance duration of 60 minutes, or "tested type" if it takes less than 10 minutes. Additionally, taking into consideration the frequency of online questions, students are labeled as "inquisitive type," "application type," amongst others. Concerning learning attitude, this study points out that e-learning courses require specific objectives, inner motive, synchronous feedback and shall allow the learners to be independent. Finally, regarding achievement

prediction, the study shows that students that present a specific behavior during the first half of the semester are more likely to maintain it until the end.

Bydzovská et al. (2017) address student performance prediction by searching for patterns using classification and regression algorithms and by analyzing students' social behavior data [8]. The two main tasks are predicting students' success or failure and predicting final grades.

For the first task, the authors first considered solely study-related data and employed the following classification algorithms: Support Vector Machines (SVM), Random Forests (RF), Rule-based classifier (OneR), Trees (J48), Part, IB1, and Naïve Bayes (NB). As for regression algorithms, the selected ones were SVM Reg, RF, IBk, RepTree, Linear Regression, and Additive Regression. In order to measure how close the predictions were to the real outcomes, the evaluation techniques applied consisted of Mean Absolute Error (MAE), where lower values represent better results, sensitivity, and F1 score. Tabulated the results for study-related data, the authors then added social behavior to the original dataset. This added data represented interaction among students and included posts and comments in discussion forums, e-mails co-authoring in publications, and even file sharing. Both results were further compared.

For the second task, the authors' considered student-course-grade triplets and focused on the similarities among student's grades using several metrics including Mean Absolute Difference (MAD), Root Mean Squared Difference (RMSD), Cosine Similarity (COS), and Pearson's Correlation Coefficient (PC) to compare grades of student's shared courses. The final grades could be estimated from the grades of similar students belonging to the computed neighborhood.

Other studies, such as the ones performed by Silva et al. (2014) and Vale et al. (2014), explored the sequential nature of data to help profiling students [9] [10]. Silva et al. (2014) explored a multi-dimensional algorithm to find multi-dimensional patterns in educational environments and model student behaviors [9]. This multi-dimensional algorithm is responsible for discovering frequent relations that involve multiple tables. In this study, the authors applied it to a star schema which models student performance in the enrolled courses and teacher evaluation for their lectures. Educational data was collected from the ten most representative courses of advanced years of the Computer Engineering program. The authors tested the approach with two baselines. The first (B1) consisted of joint dimensions of student, course, and teacher, plus the student's average grade in the first two years, and the second (B2) also contained the grades obtained on the most significative courses of the first two years of the Computer Engineering program. Further steps included filtering all the patterns choosing the N best, using the best patterns as features for classification training, and finally, running the classification algorithms and observing the results.

Results show that it is possible to predict 50% of the students' grades taking only into account their characteristics and the average grade of the first two years (B1). Also, adding the patterns, classification accuracy improves in both cases.

Vale et al. (2014) show how to apply biclustering on educational data and how to use its results as features to predict student's performance [10]. In this study, the authors gathered data from a graduate program and considering that the students usually follow a master program after, the main task was to predict the grades of the first program when finishing the following. To do that, they analyzed a matrix where the rows consisted of students and the columns represented the subjects. Cells represented the student grade obtained in that specific subject. The experiment consisted of applying a set of bicluster algorithms. To obtain a training set a label regarding the mark obtained at the end of the semester was added to the data in the matrix. The attributes regarding the biclustering results were also added to the dataset, and each attribute consisted of a boolean value which was true whenever the bicluster contained the student instance and false otherwise. The authors performed classification with decision trees and evaluated the performance using 10 fold cross-validation. Finally, to obtain the best attributes, feature selection was performed. The results of this study showed that the best results were obtained using both biclustering and feature selection achieving an accuracy of 65.8%.

In other studies, McBroom et al. (2017) approached this subject by exploring temporal precedences among data [11]. The authors investigated techniques used to identify and follow the development of student behavior over the semester and focused specifically on the application of those techniques to a junior computer science course. They gathered the most common behaviors of students, followed how those behaviors changed over time, and studied the relationship between the behaviors and the final exam outcomes.

In this study, data was gathered from an auto-grading system for computer programming which contained information about all the submission attempts, passed and failed tests, the correspondent timestamps and the obtained mark. The first step consisted of analyzing student behavior and identifying essential features, such as the percentage of submissions in the system, categorized in "early", "neither early or late", and "late", the number of compilation errors, the percentage of submissions that passed on a first or last attempt, the number of attempts, and the time taken. In the next step, clustering was performed on all the data, and the results showed that 6 clusters were necessary to obtain a set in which the distance between all the cluster was assumed to be equal, and each cluster represented the typical behavior for the submission. After this, the authors presented a visual representation of the relationship between the clusters and the final exam and of the evolution of students from a given cluster over the weeks. To better understand the stability of the cluster over time, the authors conducted clustering in the middle of the semester and compared the accuracy of each cluster in the middle of the semester relative to the end of the semester. Another interesting task proposed by the authors was to comprehend behavioral evolution in time. To explore this question, they performed additional clustering, using K-means, to identify groups of students with similarities on the submission behaviors over the weeks, followed by an examination of the relationship between the behavioral cluster and the final exam

marks. Finally, they identified general trends based on the previously obtained results.

There were only a few attempts to use sequence classifiers [12] and clusters [13] to study profiling with promising results that should be thoroughly explored in the gamifying context. Lee et al. (2014) propose a data-driven approach capable of learning a player's movements in a sequential decision-making process [12]. To do so, the authors use a supervised method to predict the next movement of the player based on past gameplay data. The framework works on a model consisting of a finite set of states, a finite set of actions, which represent the different ways of interacting with the game, and rules of transitioning between states based on an action. For each state, in the training stage, the framework learns a set of features and a classifier. Next, a trajectory is converted into a feature vector, and further, in the prediction stage, into a stochastic policy. The stochastic policy describes what action a user will take on a particular state, based on the trajectory. The authors use a supervised learning approach in order to predict the player's next action given a dataset with thousands of possible trajectories in the training set. To evaluate the technique log-likelihood and accuracy rate metrics are used.

Finally, Klinger et al. (2016) propose an evolutionary pipeline which can be applied to learning data that aims to improve cluster stability over multiple training sessions [13]. The environment of the experiment consisted of a group of students and several sessions where each student needed to solve 20 tasks per session. The first step of this pipeline was the extraction of action sequences from log data and the transformation of those action sequences into Markov Chains per session. The number of actions per student depended on his performance. Next, pairwise similarities between students were computed using two common metrics, the Jensen-Shannon Divergence, the Hellinger distance, and the Euclidean distance, which return the distances between the expected transition frequencies of the Markov Chains. Clustering was then performed using agglomerative clustering techniques. Finally, to select the best model, the authors chose to use the Akaike Information Criterion (AIC).

4

Information System

Contents

Given all the research mentioned above, there are no studies where the pedagogical effectiveness of gamification and the different game elements have been assessed and tied to a predictive model of student behavior to timely adapt the teaching experience to cater to individual needs.

My thesis is part of the "Improving Learning in College through Gamification" project which is based on data collected from the Multimedia Content Production (MCP) course and which goal is to research how to do an early detection of student's learning profiles by using machine learning techniques on their interactions with the gamified course. In particular, my task consists on developing data analysis tools to automatically identify student profiles.

4.1 The Course

MCP is a gamified Master of Science (MSc) course, taught at Instituto Superior Técnico (IST), in Lisbon, yearly, during a whole semester. The course blends theoretical and lab lectures with a gamified Moodle platform and instead of the typical grading system, students are awarded Experience Points (XP) which vary from 0 XP to the top grade, 20.000 XP. For every 1.000 XP, the student increases a level, going form level 0 to level 20, which can be equated to the university's traditional grading system. In order to be approved in the course he or she has to reach level 10, that is 10.000 XP.

Students are awarded experience points every time they complete an evaluation item from a course activity. The current evaluation methodology admits the following course activities:

- *Lab assignments* – where students manipulate different media, such as image, audio and video, with open-source tools, and, in further lectures, use the Processing language to manipulate media, producing relevant contents. These consist of 15% of the final grade.
- *Multimedia Presentation* – which takes place at the end of the semester, and where students expose the contents produced in the course in a public presentation. Also consists of 15% of the grade.
- *Quizzes* – which consist of a set of questions about the previously addressed topics and take place at the end of theoretical lectures throughout the semester, making 30% of the final grade.
- *Continuous Evaluation* – instead of opting for a final exam or large project, students will perform several small tasks along the semester. These tasks consist of 40% of the final grade and there are two main activities:
  - *Skill Tree* – consists of a precedence tree where each node corresponds to a specific task regarding a multimedia theme and grants the student an amount of XP. In order to unlock further nodes, the two precedent nodes have to be complete. The tree is built in a way in

which students are able to choose their preferred path and use the skills they're best at to reach the top. The Skill Tree grants a total of 25% of the grade.

- *Achievements* – consist of a series of tasks which will have to be performed by the students in order to acquire the corresponding badge. These tasks encourage the students to engage with the course and include attending theoretical and lab classes, participate in them by asking relevant questions, participating in the Moodle platform by sharing relevant references to the topics explored in class, cooperate with other students by answering correctly to questions and by posting tutorials in the course platform, among many others. These achievements may grant a total of 15% of the final grade and students may work for an extra of 5% by concluding some more tasks.

Theoretical lectures cover multimedia topics, focusing on the various types of multimedia information and how to manipulate them to create content. The lab lectures allow the students to use tools to perform and improve image, audio and video manipulation.

The Moodle platform is the main environment where the interaction with the several activities of the course occurs. It is there where students go to access the course materials, take the weekly quizzes, submit the lab assignments, and take questionnaires regarding their player style and their opinion towards the gamified experience. The platform also includes a discussion forum where they can interact with colleagues and professors, access the latest announcements of the course and cooperate by answering each other's questions and discussing the course topics. Teachers may grade relevant posts from 0 to 5.

One of the main elements of the gamified experience is the *leaderboard*. It is a webpage that is accessible from the Moodle platform and it displays the list of enrolled students in descending order, according to their accumulated XP, showing the top scores in the first positions of the list. Besides this, each student has a *personal profile* where he or she can track the amount of XP collected since the beginning of the semester, the *collected badges* as well as the *completed achievements*, and access personal statistics about their progress in the course activities.

As mentioned, at some time during the semester, teachers provide *questionnaires* to understand the student's relationship with the course. In the first weeks, a questionnaire to predict a learning style is available and each question covers a dimension according to the possible learning style. There are four possible dimensions, with each dimension having two opposite categories: *Active/Reflective*, *Sensing/Intuitive*, *Visual/Verbal* and *Sequential/Global*. The reported score for a dimension indicates the student preference for one category or the other. Besides this questionnaire, there are more along the semester to understand the level of engagement with the course, to access some changes that may have happened since the beginning of the course and to improve the course in future editions.

Although the course follows the structure mentioned for the last few years, there are several changes

that occurred along the years. The course has been adapted to the needs of the gamified experience and according to the student's feedback by the end of the semester. Some of those changes include the replacement of a final exam by regular quizzes along the weeks of the semester, addition and replacement of achievements and corresponding badges, as well as skills from the Skill Tree.

4.2 The Data

The data provided was extracted from the Moodle platform and consisted of three main folders: one containing all the posts from the discussion forums from the academic year of 2010/2011 until 2018/2019, another one with the students' answers to questionnaires regarding their learning style, from the academic year of 2018/2019, and the last one containing the logs regarding every interaction with the platform along with some metadata from the last nine years.

4.2.1 Moodle Posts

The folder contained a series of *json* files where each file consisted of a post made by some person, student or teacher, in a certain topic on the discussion forum at some timestamp. Each *json* file consisted of an object with the attributes described in the following table:

Table 4.1: Description of the json file attributes

Attribute	Description
message	Content of the message posted
id	Id of the message
parent	Id of the message replied to
created	Timestamp of the post
username	Id of the user that made the post
userFirstLastName	First and last name of the user
discussion	Topic of the post
forum	Discussion forum where the topic is inserted
course	Code of the course for that year

4.2.2 Questionnaires

The folder regarding the questionnaires contained several *csv* files. For each questionnaire, there is a file containing the answers of each student to every question. These files may contain a column with the record of the timestamp of the submission. The questionnaires don't follow a default format: some may contain multiple choice questions, open answer or both. For some questionnaires, there is a file containing every question and its possible answers and another file with the meaning of the results.

4.2.3 Logs

The logs folder contains a very large number of files regarding all the interactions with the computerized part of the course, carrying maybe the most important data for the project. Besides the “logs” folder containing data from every interaction with the Moodle platform for each year, there is also another folder which contains metadata.

Regarding the folder with the logs, for each year there is a file named “moodlelogs.txt”. Each line of this file consists of an interaction with the platform and the data fields are described in the table below:

Table 4.2: Description of the ”moodlelogs.txt” file data fields

Data Field	Description
Course	Code of the course for that year
Time	Timestamp of the record
IP Address	IP Address of the machine that interacted with the platform
Full Name	First and last name of the user
Module Action	Type of action performed
Information	Web element interacted with
URL	Address of the web element

Besides the “moodlelogs.txt” file, another file named “moodleVotes.txt” may exist. This file records the grades for some of the posts of the students in the platform. It shares some data fields with the previous one and includes some different ones as shown in the following table:

Table 4.3: Description of the ”moodlevotes.txt” file data fields

Data Field	Description
Time Created	Timestamp of the post
Course	Code of the course for that year
Author	Name of the person that made the post
Topic	Topic of the post
Forum	Discussion forum where the topic is inserted
Grade	Grade attributed by the teacher
URL	Address of the page of the post

For every year of the course, there is a folder containing some metadata. This metadata includes information about that year’s set up and the common files are:

- “Achievements.txt” – which describes every available achievement for that year. One achievement may have one or more levels. Each line describes the tasks that have to be performed for each level of that achievement in order to obtain the correspondent badge, and also the amount of experience points granted by each of those tasks.
- “Levels.txt” – stores the amount of experience points needed to achieve some level, and the correspondent player title.
- “Students.txt” – gives information about each student: student identification number, first and last name, email and the campus he or she is enrolled at.

- ”Tree.txt” – describes each of the nodes of the skill tree: level, topic of the task, requirements to unlock that node, color of the node, and amount of experience points granted upon completion.
- “Teachers.txt” – which stores each teacher’s identification number, first and last name and email.
- “Gave up.txt” – contains the student identification number, first and last name and email of the students that stopped engaging with the course during the semester.

Another file contained by the metadata folder is the “Awards.txt”. Although it cannot be considered metadata, it contains relevant data to this project. In this file, each of the lines corresponds to a record regarding the award of XP to a student for some evaluation item with a certain timestamp. For the latter year, this includes experience points obtained by the lab tasks, the quizzes, the skill tree submissions, the tasks regarding the level of some achievement, the final presentation, and other possible items existing that year. For the first years of the course, some of the items may be different, since the evaluation methodology has been in constant adaptation of the course’s needs. Besides the described ones, in the folders of the first years there were other files that described some items that don’t match the current structure of the course.

4.3 Data Challenges

The platform used for the course has always been the same since it has been gamified so the files regarding the logs have always had the same structure, except for some minimal changes. However, the evaluation methodology has suffered some changes along the years, so the available data is very sparse and cannot easily be arranged in a simple set of rows and columns to further perform data mining techniques.

In order to overcome this limitation, it is required to perform an additional step of data preprocessing which can be challenging in several ways:

1. In terms of data extracted from the Moodle forums, there are minor changes which have been applied along the years, such as column renaming or data format representation which should be rectified. Also, there are separated files for all the logs, graded posts and posted messages. This data is related, so it should be merged to prevent data loss. However, this will highly increase the number of missing values because besides most of the attributes are common to each file, there are some that aren’t.

2. As for the data related to the experience points obtained by the students over the semester, there are columns that represent more than one dimension. Elements from the same column may consist of, for example, the amount of experience obtained from a skill tree element submission,

and at the same time, of the level of an obtained badge. Since there are different evaluation items, such as laboratory evaluations, quizzes, skill tree submissions, among others, each column should represent individually the type of item, the obtained experience from that item, the level of the item, and so on.

3. Some data, such as the discussion forums, the topics or the type of interaction performed, are neither categorical nor numerical. When converting these entries into categorical values, and since the number of possible values in the case of forum topics increases every time a new topic is created, it is expected to be needed a large amount of memory to store these values.

4.4 Solution: Data Warehousing

Taking a look at the previously presented challenges, it is possible to admit that the solution should be finding a more efficient way to store data, by consolidating it in a suitable format for further student profile prediction.

A *Data Warehouse (DW)* consists of "a collection of consistent, subject-oriented, integrated, time-variant, non-volatile data in support of management's decisions" [35]. This type of system makes data easily accessible, as its contents should be understood either for the developer and the user. It also presents data consistently since it is previously cleansed, maintaining its quality for further user consumption, and is a system that must adapt to change and to the user's needs. One of the most important characteristics of this system is that it has a timely way of delivering data due to its structure. It relies on dimensional modeling which ensures that the databases are simple enough so users can easily understand the data and be able to deliver results quickly and efficiently [36].

Usually, data warehouses are modeled as sets of star schemas. A *star schema* is a dimensional model which consists of a fact table, surrounded by a set of dimension tables associated by foreign keys to that central fact table. When an attribute from a dimension table is divided into separate normalized tables, it's called *snowflaking*. The *fact table* stores the performance measurements resulting from a specific business process. *Dimension tables* contain descriptive attributes which are useful for filtering and grouping the facts.

In the following pages, I propose an implementation of a data warehouse, following the Kimball methodology [37].

4.4.1 Defining the Business Processes

After a deep exploration of the provided data, and since the main goal of this project is to understand the commitment of the student with the course along the semester, regarding its participation in the course

Moodle, its responses on the questionnaires, and its performance on the course evaluation items, it is possible to identify three important elements: the *student*, the *activity*, and *time*.

The process regarding the engagement with the course considers the fact that there is a student that performs some kind of activity in a specific time which has an impact on his performance along the course. Each of these elements can be described by a set of attributes and components. Since a student can perform different types of activities during the course, it is possible to distinguish them:

- *Student Evaluation* - he or she may take an evaluation item and be awarded experience points which will have an impact on the final grade.
- *Learning Style Questionnaires* - teachers can provide questionnaires which will be taken by the students, to have an idea about the student's type of player.
- *Moodle Participation* - students will interact with the course platform by consulting resources, posting on forums, viewing pages or submitting assignments.

4.4.2 Identifying the Dimensions

Since we are dealing with interactions between students and a course along the semester, we can admit that two crucial dimensions will be the *Student* dimension and the *Date* dimension. Connected to the *Date* dimension, a *Semester* dimension will also be considered for further possible aggregations. For the different types of activities, more dimensions have to be considered.

Regarding the evaluation and consequent gain of experience points, it is assumed that a certain student is evaluated on a specific item at a certain date, therefore, an *Evaluation Item* dimension will be useful.

When answering the available questionnaires, the *Question* dimension is needed in order to record the possible available questions and the corresponding answers in case of being multiple choice, and since each question may cover one dimension, I will also consider a *Question Theme* dimension. I will also consider a degenerate *Questionnaire* dimension which will be responsible to register the identification number of the provided questionnaire.

Finally, regarding the participation on the Moodle platform, when dealing with the logs, we have to consider an action performed by some student at a certain time, on a specific web element, so the *Web Element* and the *Action* dimensions will also be considered. In this case, there are situations where it is possible to detect that the web element is enclosed in some topic, so I also propose the creation of the *Content Topic* dimension.

Still on the matter of the Moodle platform, but regarding the discussion forums, a student may post a message regarding some topic in a discussion forum at a certain time. It is interesting to keep a record of the posted messages, and this will be stored in the *Message* dimension.

So, in order to address data at the right level of granularity, the provided data will be organized around eleven dimensions: *Student*, *Date*, *Semester*, *Evaluation Item*, *Question*, *Question Theme*, *Questionnaire*, *Web Element*, *Action*, *Content Topic*, and *Message*.

4.4.3 Dimension Details

- The *Student* dimension will be used to collect identification data, which is usually the student number, and personal data, such as full name, email and campus of enrollment. It is the main entity of this project, since the goal is to perform student profiling.
- The *Date* dimension is the most detailed one in our context and describes the day of the course: the ordinal number in the month and in the year, the day on the week, if it's a weekday or a weekend, and so on. It will also include a foreign key for the *Semester* dimension.
- The *Semester* dimension is responsible for storing the year of the course.
- An *Evaluation Item* consists of an item that a student has to conclude in order to obtain experience points. These items may be lab assignments, skill tree elements, quizzes, among others. Each of them has a type, a maximum of XP that may be awarded to the student and some requirements.
- The *Question* dimension stores not also the question but also the possible answers in case of being multiple choice.
- The *Question Theme* dimension records the two opposite categories of the four possible dimensions of the learning style mentioned earlier on this document.
- The *Questionnaire* degenerate dimension has already been described as an empty dimension table containing only the identifier for each questionnaire.
- A *Web Element* dimension describes the page that the student acted with, like its URL or if it's a student profile, a discussion forum, or the course slides.
- An *Action* is registered in the logs as a type of interaction with the platform. These actions may be performed towards the many available elements of the platform, such as the chat, the discussion forums, the calendar, and so on. Each of these elements have possible actions, such as viewing, reporting, or adding. As an example, regarding the forum, the possible interactions may be "forum view", "forum add post", "forum subscribe", among others. This dimension will consist of a hierarchy of these interactions where each column will represent a level of this hierarchy.
- The *Content Topic* represents data about the contents of the course and the attributes will consist of a topic and the discussion forum it is inserted in.

- Finally, the *Message* dimension will record the content of the messages of the whole discussion forum.

4.4.4 Identifying the Facts

According to Kimball [37], each business process will correspond to a star schema, with a central fact table.

Regarding the *Moodle Participation* business process, and since the provided log files consist of interactions with the Moodle platform at a specific timestamp, it is possible to admit that the fact table shall be transactional. In these types of fact tables, each row corresponds to a transaction. In this case, each row corresponds to an action of a student on a specific forum at a given date. One of the fact tables considered will be *Access/Logs* and by grouping data regarding students in a semester another one which will be named *Moodle Participation*. This second table will consist of an aggregate fact table which will store a variety of measures which can be extracted from the first one, like the total number of posts made by some student during a semester or the number of times he or she downloaded the course resources.

Although the posted messages are integrated in the *Moodle Participation* business process, it is important to keep this data separated from data from the logs because it consists of a different view of the process, therefore, it will be stored in the *Posts* fact table where each row corresponds to a message posted by some student regarding some topic in a certain date. For each message, there may be an attributed rating, and a polarity regarding the positivity or negativity of the content. To further analyze information regarding this polarity, the length of the messages of some student, or even the total number of posts of a student in each discussion forum in a semester, I also suggest a *Message Analysis* aggregate fact table.

Concerning students' evaluation business process. For the records of the student's evolution in terms of experience when evaluated: a student is awarded an amount of experience points for any evaluation item taken someday. For this matter, the fact table will be *Student Evaluation* and the facts will include the collected XP for each of the evaluation items concluded by the student at some timestamp. To store data from the performance of some student in the semester, such as the total amount of XP collected by succeeding in each evaluation item or the number of collected badges, another aggregate fact table will be *Student Result*.

Finally, regarding the student's answers to the questionnaires, I will consider the *Student Answers* fact table, where each answer of the students for each question are contemplated. The aggregate fact table *Questionnaire* will store relevant data regarding the answers of some student during some year.

4.4.5 Bus Matrix

A Bus Matrix resumes the dimensions involved in each of the proposed processes and their relations with the identified facts. Since I'm following Kimball's methodology for designing a data warehouse, each of the proposed business processes will result in a star schema.

Table 4.4: Bus Matrix

	Dimensions										
Business Processes	Student	Date	Semester	Evaluation Item	Question	Question Theme	Questionnaire	Web Element	Action	Content Topic	Message
Student Evaluation	**X**	**X**		**X**							
Student Result AGG	**X**		**X**								
Student Answers	**X**	**X**			**X**	**X**	**X**				
Questionnaire AGG	**X**		**X**								
Access/Logs	**X**	**X**						**X**	**X**	**X**	
Moodle Participation AGG	**X**		**X**								
Posts	**X**	**X**								**X**	**X**
Message Analysis AGG	**X**		**X**								

4.4.6 Student Evaluation Schema

The constellation schema represented in Figure 4.1 show the links between some of the previously proposed dimension and the selected fact tables for the *Student Evaluation* business process. With this constellation schema it is possible to see in detail which attributes will be considered for each table.

The *Student Result* fact table, as already mentioned, consists of an aggregation of the *Student Evaluation* one and the proposed measures were defined according to the current evaluation method of the course. These measures consist mainly of the total amount of experience points obtained by concluding evaluation items of some type, the number of concluded items of that type and the total amount of collected experience points, which correspond to the student's final grade. During the preprocessing phase it is possible that some evaluation items which were considered in the first years of the course are found and that some measures are added to the aggregation.

4.4.7 Moodle Participation Schema

Figure 4.2 represents the *Moodle Participation* business process. This constellation schema combines data retrieved from the logs and from the discussion forum posts folders and relates the proposed fact and aggregated fact tables with the corresponding dimensions.

Although the *Logs* fact table doesn't contain any measure, there are several ones that may be extracted from the aggregation of its data and that are represented in the *Moodle Participation* fact table.

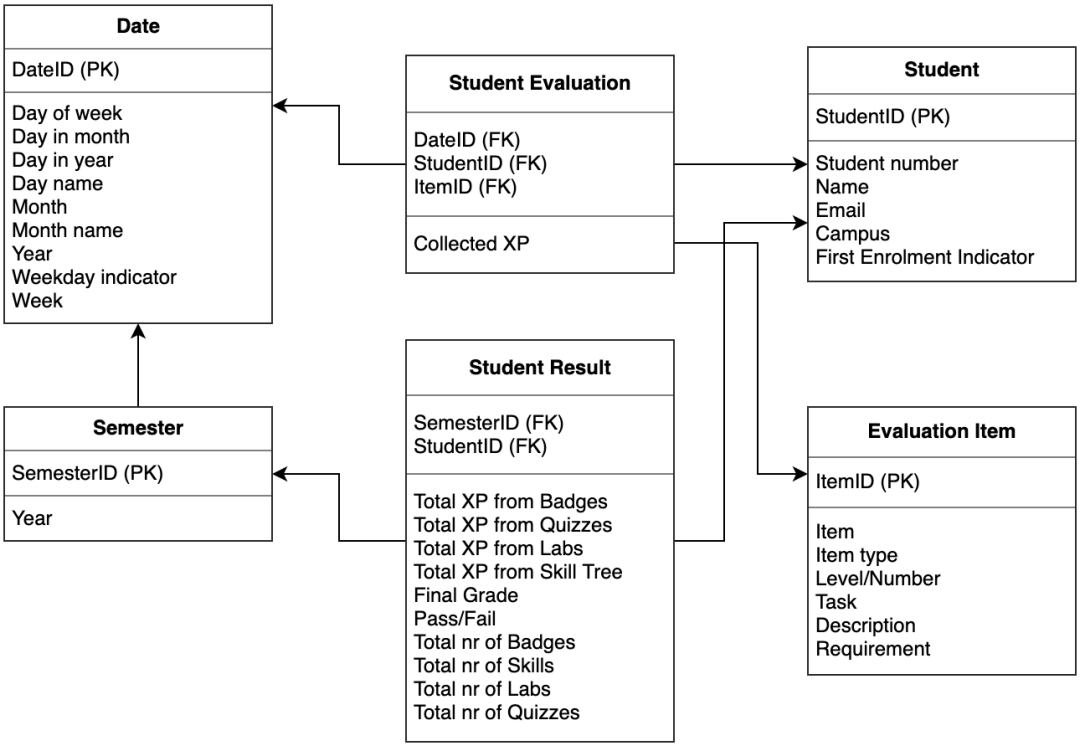

Figure 4.1: Student Evaluation Schema

Besides the large amount of possible actions towards the Moodle platform, the ones represented by the defined measures may be the most interesting ones to be analyzed regarding their impact on the final student profile.

Concerning the student posts, student messages posted on some topic at a certain date are stored in the *Posts* fact table. The aggregated fact table *Message Analysis* stores all the measures regarding the student's sent messages in that semester, such the count of positive, negative or neutral messages or the total number of posts made in each discussion forum.

4.4.8 Learning Style Questionnaires Schema

In Figure 4.3, there's a representation of the *Learning Style Questionnaires* business process. It consists of the *Student Answers* fact table connected to the corresponding dimensions and of an aggregation of that same table to store measures related to the performance of a student in a certain questionnaire. Since each question is inserted in a specific category which determines the student's learning style, some interesting measures are the total number of answers towards each category and the corresponding learning style according to those scores.

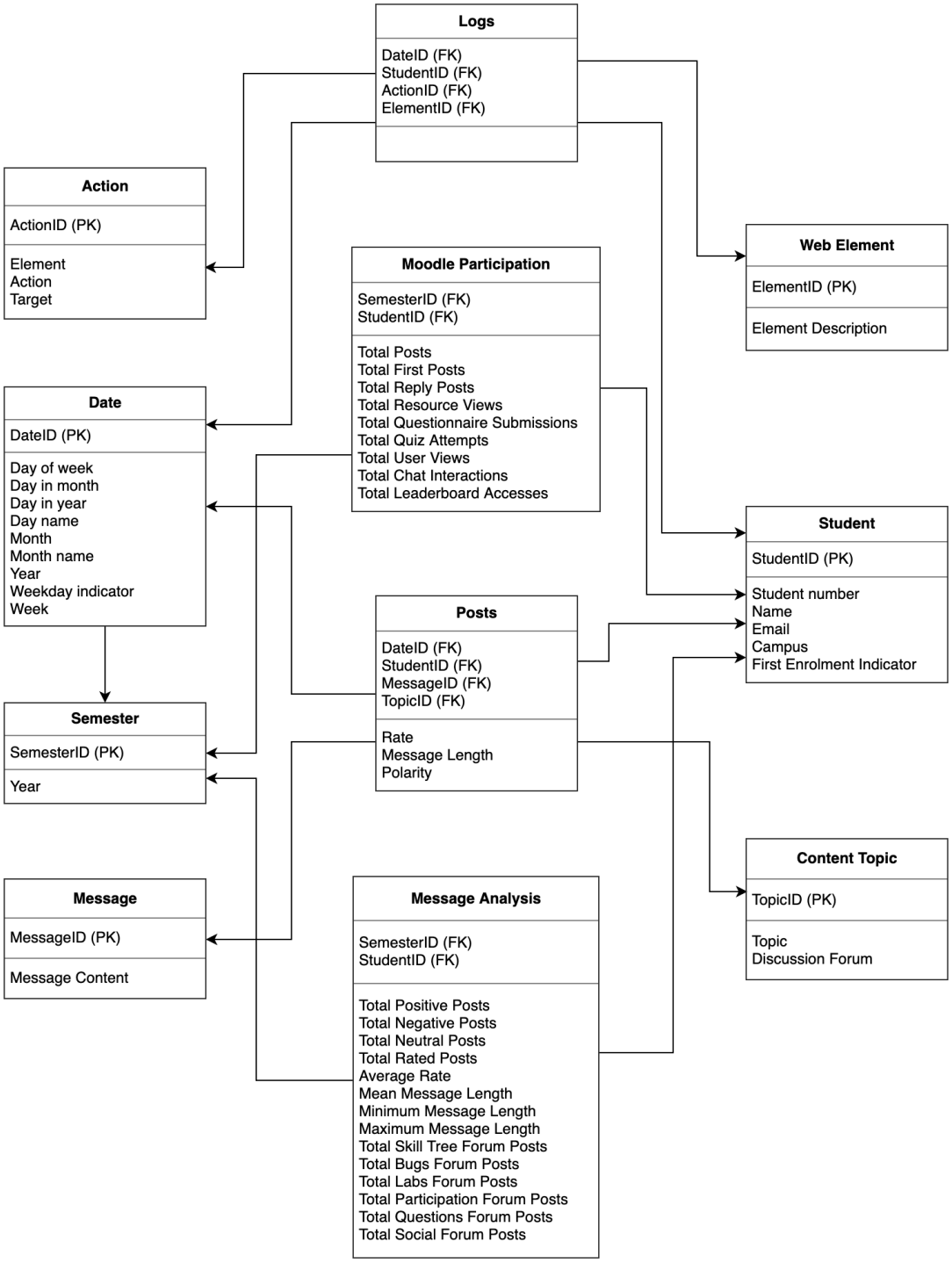

Figure 4.2: Moodle Participation Schema

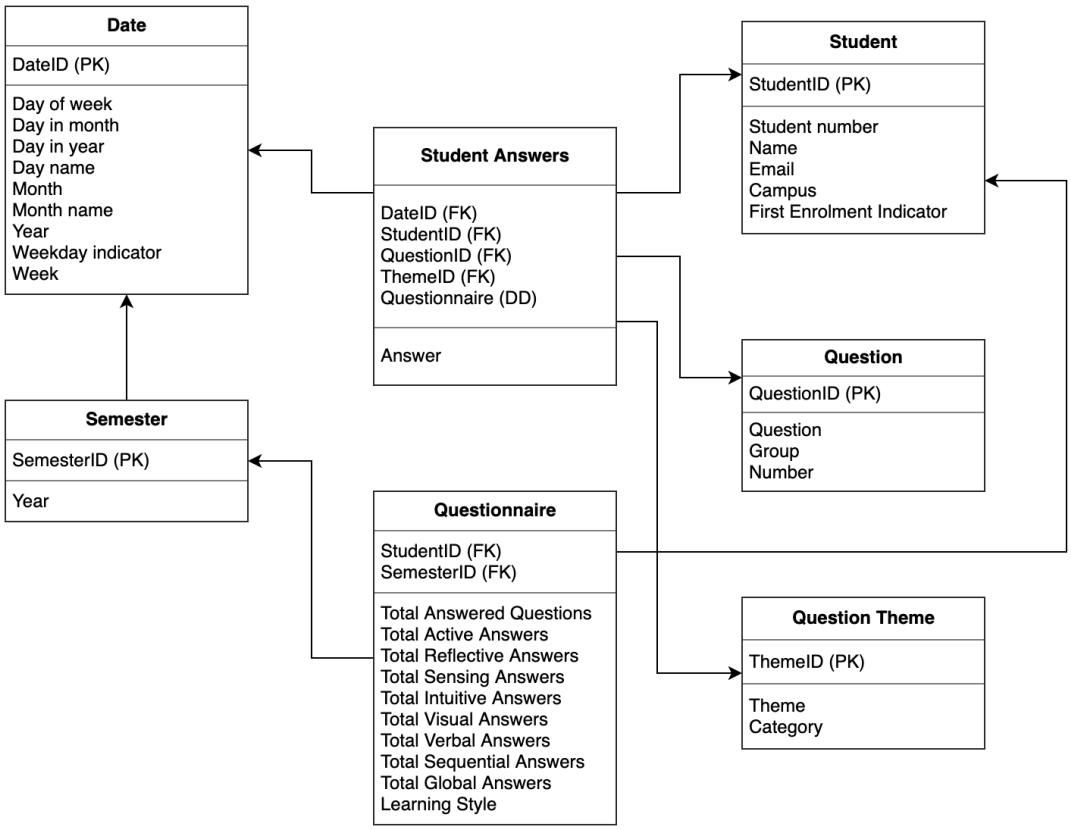

Figure 4.3: Learning Style Questionnaires Schema

4.5 Building the Data Warehouse

In order to populate the Data Warehouse, we have to resort to the use of Extraction-Transformation-Loading (ETL) tools. These tools are responsible for:

1. Extracting the data from the appropriate sources
2. Transform the source data and compute new values, or even records, in order to obey the structure of the data warehouse relation proposed
3. Load the cleansed, transformed data to the appropriate relation in the data warehouse

I opted to use pandas, and open-source, BSD-licensed library which provides high-performance, easy-to-use data structures and data analysis tools for the Python programming language as the tool to extract and manipulate the available data and build the final data warehouse.

4.5.1 Student Evaluation Schema

To create the tables regarding this business process, I started by analyzing the "Awards.txt" from each year's folder. Despite minor changes from some years to others, the extracted data consisted of a set of columns, where some of them were easier to recognize than others. From the datasets, it was possible to understand that each line consisted of an evaluation item concluded by some student at a given timestamp which grated some kind of reward (Table 4.5). Further data exploration helped to understand that column 4 entries consisted of experience points when the item in column 3 was other than an item of the type *Badge*, and the achieved level of that badge otherwise. Column 5 consisted of a count or a Boolean entry regarding the items of type *Badge*, a number for the *Quiz* and *Lab* item types, and the name of the completed skill node of the Skill Tree for that item type.

Table 4.5: "Awards.txt" file representation

Column 1	Column 2	Column 3	Column 4	Column 5
1367534837.49	73425	Guild Warrior	2	2
1367534837.49	72560	Lab Master	1	2
1367534837.5	77312	Postmaster	2	31
1367534837.55	76899	Grade from Lab	600	9
1367534846.07	62550	Skill Tree	200	Publicist
1402428378.04	70843	Grade from Quiz	510	10
1402428379.09	79476	Hollywood Wannabe	1	FALSE

The first step of the data preprocessing consisted in separating column 4 whether it contained XP or a level and creating the corresponding columns "Collected XP" and "Level/Number". To this second column, entries for column 5 regarding the Quizzes and the Labs numbers were also added. Then, I created the column "Item Type" where each entry in column 3 was compared to the entries in the

text files from the “metadata” folder of that year. If the entry of the column existed in any line from the “achievements.txt” file, the type of the item would be “Badge”. For entries which contained “Grade from”, the item type considered was its following words: “Quiz”, “Lab” and other items that appeared on the dataset but are not represented in the table. For the Skill Tree entries, the considered item type was “Skill Tree” and these entries were replaced by the corresponding ones in column 5 and this column was dropped. Finally, since students also receive XP from completing achievements and that information is contained in the “achievements.txt” file, the corresponding XP was extracted and inserted in column 4. All columns were renamed according to the held values. These changes resulted in a dataset which is partially represented in Table 4.6 for better understanding of the process.

Table 4.6: Transformed ”Awards.txt”

Timestamp	StudentID	Item	Item Type	Collected XP	Level/Number
1367534837.49	73425	Guild Warrior	Badge	50	2
1367534837.49	72560	Lab Master	Badge	150	1
1367534837.5	77312	Postmaster	Badge	80	2
1367534837.55	76899	Grade from Lab	Lab	600	9
1367534846.07	62550	Publicist	Skill Tree	200	NA
1402428378.04	70843	Grade from Quiz	Quiz	510	10
1402428379.09	79476	Hollywood Wannabe	Badge	150	1

Since the final structure of the *Student Evaluation* fact table is supposed to contain the primary keys *DateID*, *ItemID*, and *StudentID*, there are some changes that still had to be performed. The timestamp was converted to a date identification number with the format *YYYYMMDD*, and each unique value of the column was further used to create the *Date* dimension, where a number of attributes were inferred from that ID and added to the dimension (Table 4.7). Note that the *Date* dimension does not contain the attribute *Year*, as it is supposed to be stored in the *Semester* dimension (Table 4.8). Instead, the attribute *SemesterID* will create the relation between both of the dimensions and the *Semester* dimension will only contain the *Year* attribute along with the corresponding ID. As for the *Item*, *Item Type* and *Level/Number* columns, these were used to start building the *Evaluation Item* dimension. From the *StudentID* column, it was possible to start creating the *Student* dimension.

Table 4.7: Date dimension table

DateID	Date	Day	Day Name	Day of Week	Day of Year	Month	Month Name	Week	Weekday Indicator	SemesterID
2013-05-02	20130502	2	Thursday	3	122	5	May	18	Weekday	2
2014-06-10	20140610	10	Tuesday	1	161	6	June	24	Weekday	3
2015-02-28	20150228	28	Saturday	5	59	2	February	9	Weekend	4
2018-03-21	20180321	21	Wednesday	2	80	3	March	12	Weekday	7
2017-03-15	20170315	15	Wednesday	2	74	3	March	11	Weekday	6

As for the *Item*, *Item Type* and *Level/Number* columns, these were used to start building the *Evaluation Item* dimension. Regarding this element, there are plenty of attributes that can be added to the table if extracted from the metadata. Skills also have a level associated, and to add the levels to the

Table 4.8: Semester dimension table

SemesterID	Year
2	2013
3	2014
4	2015
6	2017
7	2018

corresponding column some comparisons had to be performed between each entry regarding the obtained skill and the file containing all the skills for that year. Some skills require the acquisition of others, and to represent that, a *Requirement* column was created. As for the obtained badges, each of them has a description, and each level of a badge requires the completion of a task which is defined in the "achievements.txt" file. The *Evaluation Item* dimension consists of the unique entries of every recorded item along the course which are identified by the *ItemID* and is represented in Table 4.9.

Table 4.9: Evaluation Item dimension table

ItemID	Item	Item Type	Level/Number	Description	Task	Requirements
182	Guild Warrior	Badge	2	You helped your Guild	All students in your lab had 4 or better grade (2 times)	ND
53	Lab Master	Badge	1	Excel at the labs	Top grade in two graded classes	ND
12	Postmaster	Badge	2	Post in something in the forums	Make thirty posts	ND
56	Grade from Lab	Lab	9	ND	ND	ND
231	Publicist	Skill Tree	2	ND	ND	Course Logo+Movie Poster

From the *StudentID* column, it was possible to create the *Student* dimension. Data regarding the enrolled students is stored in the "students.txt" file. From there, it was possible to extract the student ID, first and last name, email, and campus of enrollment. A column indicating if it was the first time the student enrolled in the course was added. I will not represent the final dimension table in this document to protect the student's identity. Finally, to put the *Student Evaluation* fact table in its final form, entries represented by the dimensions in the initial table (Table 4.5) were replaced by the corresponding IDs resulting in a table represented by Table 4.10.

To build the aggregated fact table *Student Result*, the measures were calculated by grouping the dataset by *SemesterID*, *StudentID* and *Item Type*. Doing so, it was possible to sum the total XP obtained for each evaluation item, as well as the total number of unique items and the final XP. By checking if the final XP summed over or under 10.000 points, a Boolean attribute was also added depending on whether the student passed or failed the course (Table 4.11).

Table 4.10: Student Evaluation fact table

DateID	StudentID	ItemID	CollectedXP
20130502	73425	182	50
20130502	72560	53	150
20140610	77312	12	80
20140610	76899	56	600
20150228	62550	231	200
20180321	70843	183	510
20170315	79476	80	150

Table 4.11: Student Result aggregated fact table

SemesterID	StudentID	XP from Badges	XP from Labs	XP from Quizzes	XP from Skills	...	Total Badges	Total Labs	Total Quizzes	Total Skills	Final XP	Pass/Fail Indicator
0	59031	2375	1950	3450	2600	...	22	4	9	11	12765	Pass
2	62535	1475	2500	4088	100	...	15	5	9	1	11078	Pass
4	64735	3500	2400	4463	1100	...	28	5	9	5	14243	Pass
5	64744	4710	2500	3950	4000	...	37	5	9	13	18255	Pass
5	64788	3535	2600	4200	500	...	27	5	9	5	13480	Pass

4.5.2 Moodle Participation Schema

The *Moodle Participation* schema concerns two main groups: the messages, and the logs. I started by creating the tables for the data regarding the messages. First, I had to concatenate all the *json* objects extracted from each post made by the students and to convert it into a *csv* file (Table 4.12). The extracted messages contained HTML tags, and these had to be cleaned in order to obtain an understandable string.

Table 4.12: Data from the *json* files concerning the student posts

Message	Date	StudentID	Topic	Discussion Forum	Course
#messagecontent	2019-04-26 03:49 PM	78029	Movie Barcodes	Skill Tree	PCM1819
#messagecontent	2014-03-01 03:43 PM	79419	Proficient Tool User Challenge #3	Labs Forum	PCM1314
#messagecontent	2013-06-05 10:36 PM	57681	Class Annotator - Lecture 4	Participation Forum	PCM1213
#messagecontent	2018-06-02 03:39 PM	91200	Slides Lesson 19 - Bug4	Bugs Forum	PCM1718

In the logs folder there was a file containing entries for rated posts, and I decided to include them in the tables regarding the messages and not in the logs because although close to 23.000 messages were extracted, only around 11.600 were written by students, and the “moodleVotes.txt” file contained data from posts that were not included in the provided *json* files. When merging these two files, only around 2.600 of the 7.400 entries from the “moodlevotes.txt” files were related to the messages’ entries, which means that from the 23.000 posts made available, only 2.600 were rated. I decided to keep all of

the entries from both files.

Merged the *csv* files, the attribute *Rate* was added to the table. Since I also want to understand if the measures regarding the messages consist of important features to further predict the student profiles, I decided to add two more attributes, the *Message Length* and the *Message Polarity*. The polarity of a message consists of the sentiment transmitted by it. In this case, I decided to label the messages as having a positive, negative or neutral tone. To do so, I used a lexicon and rule-based sentiment analysis tool called *VADER* (Valence Aware Dictionary and sEntiment Reasoner) which evaluates the string and returns how positive or negative the content is.

The Message attribute was used to create the *Message* dimension (Table 4.13), *Topic* and *Discussion Forum*, to create the *Content Topic* dimension (Table 4.14), and the *Date* was converted to *YYYYMMDD* and its unique values that were not yet in the *Date* dimension were added to it. The *Course* attribute was dropped, and the foreign keys for the *Message* and *Content Topic* dimension were added, thus concluding the creation of the *Posts* fact table (Table 4.15).

Table 4.13: Message dimension table

MessageID	Message
0	#messagecontent
1	#messagecontent
2	#messagecontent
3	#messagecontent

Table 4.14: Content Topic dimension table

TopicID	Topic	Discussion Forum
399	Movie Barcodes	Skill Tree
853	Proficient Tool User Challenge #3	Labs Forum
408	Class Annotator - Lecture 4	Participation Forum
489	Slides Lesson 19 - Bug	Bugs Forum

Table 4.15: Posts fact table

MessageID	DateID	StudentID	TopicID	Message Length	Polarity
0	20190426	78029	399	356	positive
1	20140301	79419	853	58	neutral
2	20130605	57681	408	81	positive
3	20180602	91200	489	152	negative

Finally, to build the aggregated fact table *Message Analysis*, the *Posts* fact table was grouped by *SemesterID* and *StudentID* to get the intended measures. From the message length the *Minimum*, *Average* and *Maximum Length* were created. The number of rated posts was counted, and the mean rate was calculated to get the *Total Rated Posts* and *Average Rate*. The total number of positive negative and neutral posts, as well as the total number of posts made in each discussion forum were also added to complete the final aggregated table (Table 4.16).

Table 4.16: Message Analysis aggregated fact table

Semester ID	StudentID	Total Negative Posts	...	Total Positive Posts	...	Mean Message Length	...	Total Graded Posts	Mean Post Grade	Total Bugs Forum Posts	...	Total Skill Tree Posts
8	48148	1	...	15	...	80.52	...	40	1.77	0	...	6
3	55329	1	...	2	...	107.9	...	10	1.8	0	...	0
2	55461	0	...	0	...	2.0	...	2	3.5	0	...	0
7	55906	4	...	7	...	107.12	...	56	2.60	1	...	14

Regarding the tables that concern the logs files, I started by concatenating all of the files and dropping every row that didn't consist of an action performed by a student. The *IP Address*, the *URL* and the *Course* attributes were dropped since there was no interest in keeping them, and the Information attribute was used to create the *Web Element* Dimension (Table 4.18). The *Full Name* was replaced by the *StudentID*, the *Time* attribute was once again transformed in *YYYYMMDD* format, and each new unique value, added to the *Date* dimension.

The *Module Action* column was used to create the *Action* dimension. As previously referred, a hierarchy was created, since each action was performed on an element with a specific target (Table 4.19). The *Logs* fact table contained only the foreign keys for the corresponding dimensions and no facts (factless fact table) (Table 4.17).

Table 4.17: Logs fact table

StudentID	DateID	ActionID	ElementID
70493	20160607	9	2203
67095	20140217	4	2370
65742	20150406	6	19
79208	20160220	29	122

Table 4.18: Web Element dimension table

ElementID	Information
40	Cartoonist
89	Sofia Gonçalves
157	Re: Multimedia Presentation Subject
208	Typo in the Pixel Art Skill

Again, to create the aggregated fact table *Moodle Participation*, the *Logs* fact table was grouped by *SemesterID*, *StudentID* and *ActionID*, and all the actions were counted. For further analysis, only some of them were kept for understanding their impact on the profile prediction. The chosen measures were the number of first posts, reply posts, and posts in general, the total resource views, questionnaire submissions, quiz attempts, student profile views, chat interactions, and accesses to the leaderboard. Each of this measure was gathered by counting the number of times an action of that type was performed.

Ended the construction of the Data Warehouse, to get data ready for further student profiling we only

Table 4.19: Action dimension table

ActionID	Element	Action	Target
29	forum	add	post
21	questionnaire	submit	ND
47	resource	view	all
17	quiz	view	summary

Table 4.20: Moodle Participation aggregated fact table

SemesterID	StudentID	Total Resource Views	...	Total Quiz Attempts	Total User Views	Total Leader-board Accesses	Total First Posts	Total Reply Posts	Total Posts
0	53883	443	...	8	38	3	3	50	53
3	70493	327	...	18	23	0	1	11	12
4	67095	565	...	12	13	0	4	16	20
2	65742	225	...	1	3	0	1	5	6
6	56851	759	...	12	98	25	17	37	54

need to gather all defined measures from every aggregated fact table created in a final table. In this final table, each row represents the performance of some student regarding the gamified course and each column contains a specific attribute concerning the student's performance.

One of the main advantages of this information system is that it is possible to gather the attributes concerning the performance of the students for several phases in a semester. This is an important aspect for this work as the main goal is to predict profiles according the students' performance along the course. The final table will allow us to classify each of the students as belonging to a specific type of student depending on factors that will be explored in the following chapter. Classified the students, when extracting from the data warehouse the attributes concerning the performance of the students in other phases of the semester, each student type will already be known and attributed as a class.

5

Student Profiling

Contents

In order to cover the goals of this project, in this chapter I present my proposal for an early prediction of students' profiles taking into account their performance and interactions with the gamified course. The very first thing to do was to read about previous work on this subject and one of the most important papers was written by Barata et al (2014) who present an experiment based on student's performance where they try to understand which data better characterizes a student type [38]. Using cluster analysis, four main student types were identified: Achievers, Regular, Halfhearted, and Underachievers. In a further publication, the same authors repeated the procedure with a new batch of students and identified the same number of clusters. They also identified that besides the different levels of participation and performance, the accumulated XP over time was also an indicator of each type of student [39].

The data regarding this study was also extracted from the MCP course in its first years of gamified experience, so it has high relevance to the task of student profiling of this work.

To perform the profiling of the students, I first started by extracting the data from each of the previously described aggregated fact tables concerning the proposed business processes and combining it into a dataset. This dataset contains all the students as records, and each of the defined measures as attributes, representing the performance of the students by the end of the semester.

I proceeded by labeling the dataset based on Barata's studies. The labels consist of one of four student types: Achiever, Regular, Halfhearted and Underachiever. Known the correspondent student type, I used the same labels in seven other datasets extracted from the data warehouse just as before for the final performance, but now for the performance in different weeks through the semester. In total, the number of datasets that will be used for classification is eight, and consist of the students' performance after three, five, seven, nine, eleven, thirteen and fifteen weeks, and also by the end of the semester.

The next step was to preprocess the data to further feed the machine learning algorithms. The chosen classifiers were Naïve Bayes, Decision Trees, Random Forests and Gradient Boosting which had to be built and tuned to achieve the best results for each moment in the semester.

Finally, with the built classifiers, it was possible to perform profile prediction in different phases of the semester to understand how soon a student type can be predicted with high accuracy.

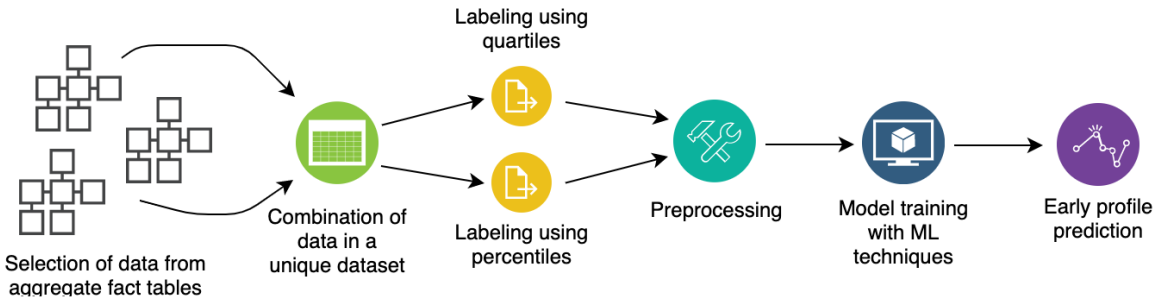

Figure 5.1: Representation of the thesis proposal

5.1 Labeling Criteria

Since the provided data does not include an assignment of a profile to each student, I started by labeling the dataset which contained the calculated measures for each student based on the XP accumulation curves presented by Barata et al (2016) (Figure 5.2) .

Given the rough similarity in the evolution of the curves and the fact that the points of each curve representing the final XP are close to equidistant from each other, I opted by taking into account only the final XP and label the students using two methods: quartiles and percentiles. The first consist of a separation of data into four bins where the first contains the lowest 25% of numbers, the second between 25,1% and 50%, the third between 51% and 75%, and the fourth the highest 25% of numbers. The second separates data according to percentiles and in this case four bins where considered: the first contains all the students with final XP below 63%, the second between 63,1% and 75%, the third between 75,1% and 85%, and the fourth with XP above 85%. In both cases, the students placed in the first bin are labeled as Underachievers, in the second bin as Halfhearted, in the third as Regular, and in the fourth as Achievers. For convenience, the student types were converted to numbers, where 0 represents the Underachievers, 1 the Halfhearted, 2 the Regular, and 3 the Achievers.

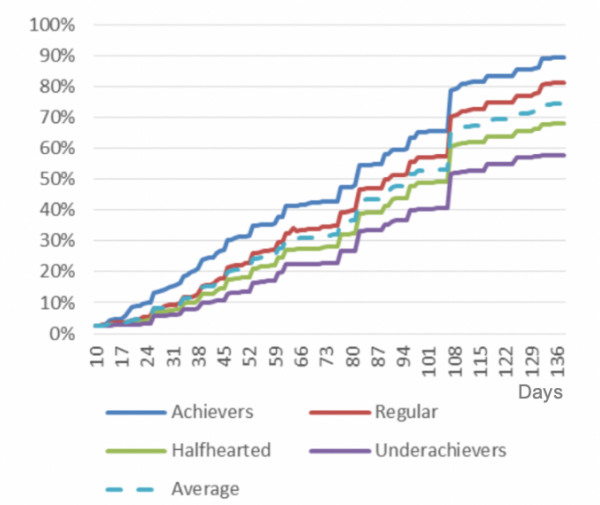

Figure 5.2: XP accumulation curves by student type

Although a limit of 63% of the final grade seems a high value for the class of the Underachievers, we assume that in a Masters' degree course there is a low failure rate, thus finishing the course with a grade below this value consists of doing the bare minimum get approved.

It is important to refer that the normalization of the *Final XP* attribute for each year was necessary since the minimum and maximum experience points obtained vary from year to year.

By the end of this phase, two datasets were created, in order to perform two different experiments:

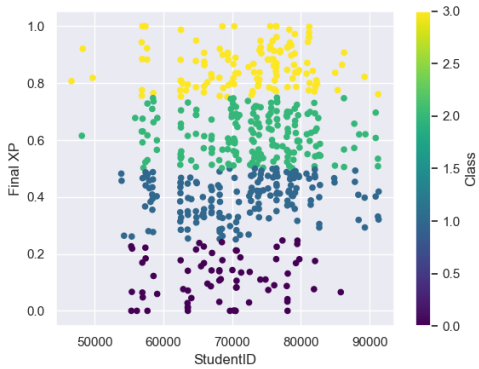

Figure 5.3: Distribution of the Classes according to quartiles

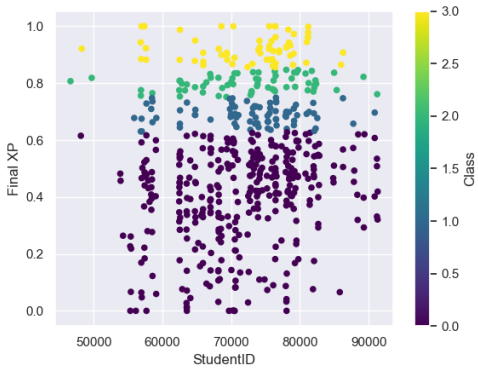

Figure 5.4: Distribution of the Classes according to percentiles

one regarding labeling according to quartiles, and other according to percentiles. The target variable "Class" referring to the correspondent student type was added to both the datasets.

The following step was to add the same "Class" attribute to the remaining datasets which represent the performance of the students in different phases of the semester. This way it is possible to understand how students of a specific type behave along the semester.

5.2 Explored Learning Techniques

In order to classify the students as having one of the four possible profiles, I will explore four classification techniques: Naïve Bayes, Decision Trees, Random Forests, and Gradient Boosting.

Given the fact that the Naïve Bayes algorithm is very fast and easy to build as it required no complicated iterative parameter estimation, it will be used as a baseline for the classification problem. This classifier relies on the assumption that features are conditionally independent of one another, although this assumption very rarely holds. However, this model can still achieve substantial classification accuracy and be relied on to be robust [40].

Decision Trees classify data by breaking it into smaller and smaller subsets. By partitioning the dataset, a tree is incrementally developed, being the final result a tree with decision and leaf nodes, where leaves represent class labels and branches represent conjunctions of features that lead to those class labels.

The incremental growth of the tree risks overfitting the training data and poorly generalizing to new samples. To avoid overfitting and to trim the Decision Tree to optimize the learning process, tree pruning is performed [40].

In this work, as we will be using the *DecisionTreeClassifier* available on *scikit* learn, the process of pruning will be done by tuning several hyperparameters, which can reduce the depth of the tree or

resizing the terminal, amongst others.

Random forests are an ensemble learning method for classification that works by constructing a multitude of Decision Trees at training time and outputting the class that is the mode of the classes. Random Forests use a modified tree learning algorithm that selects, at each candidate split in the learning process, a random subset of the features, rather than all of the features of the model.

In order to optimize Random Forests tuning the hyper parameters from the classifier available on scikit learn is also required.

Similarly to Random Forests, Gradient Boosting is an ensemble learning method, meaning it will create a final model based on a collection of individual models. Even though the individual models may be weak and prone to overfitting, combining these models in an ensemble will lead to an overall improved result. In Gradient Boosting machines, the most common type model to combine is Decision Trees.

In this work, we will be using a specific implementation of the Gradient Boosting method, XGBoost. XGBoost stands for "Extreme Gradient Boosting" and uses more accurate approximations to find the best tree model and is comparatively faster than other ensemble classifiers.

XGBoost contains a wide variety of parameters which with fine-tuning to optimize the model.

5.3 Evaluation Criteria

For cases where the target variable classes of a dataset are a majority of one class, that is, when the dataset is imbalanced, and also when dealing with a multi-class problem, using accuracy alone to evaluate the performance of a classifier can be misleading. Figure 5.4 represents the clear imbalanced distribution of the classes for the *percentiles* datasets.

In order to evaluate the results of the experiments, the confusion matrix was used.

A confusion matrix consists of a N-dimensional matrix, where N represents the number of classes, that summarizes the classification performance of a classifier with respect to the test data. The columns of the matrix represent the predicted classifications and the rows the actual defined classifications [41].

The test outcome can be positive or negative and may or may not match the actual sample.

- **True Positive (TP)** A positive sample is classified as positive.
- **False Negative (FN)** A positive sample is classified as negative.
- **False Positive (FP)** A negative sample is classified as positive.
- **True Negative (TP)** A negative sample is classified as negative.

The most commonly used performance measure is *accuracy* which measures the fraction of predictions that the model guessed as correct and is calculated in terms of positives and negatives as follows:

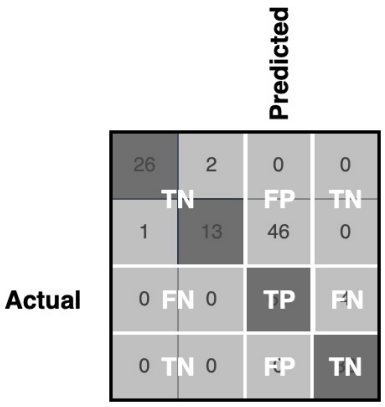

Figure 5.5: Confusion Matrix of a 4-class problem with relationship between the predicted and the actual samples regarding the third class

$$Accuracy = \frac{TruePositives + TrueNegatives}{TruePositives + TrueNegatives + FalsePositives + FalseNegatives} \tag{5.1}$$

Given the imbalanced nature of the datasets, in this work, we will be taking into account the *sensitivity* of the classifiers for each class, as the goal is to classify a student as being of a specific type correctly. This measure is formalized as follows:

$$Sensitivity = \frac{TruePositives}{TruePositives + FalseNegatives} \tag{5.2}$$

5.4 Preprocessing

Both the *quartiles* and *percentiles* datasets have 40 attributes and one target variable, which is one of the four student types (0, 1, 2 or 3). All attributes are numerical and continuous, except the "Pass/Fail" which is categorical. The datasets contain a total of 580 records each. All the actions performed in the preprocessing phase were applied to both datasets except when stated otherwise.

5.4.1 Data Cleaning

The first thing done in the preprocessing phase was "cleaning" the data. Since the attributes of the dataset correspond to the measures extracted from the aggregated fact tables of the data warehouse, we expect that the missing values are nonexistent, as most of the measures consist of sums and counts. However, when box plotting the distribution of the final XP per year, it is possible to detect some outliers. An outlier, in this case, is a student whose final XP does not fit the distribution for that year. The number of outliers consisted of only 3% of the data, and the impact on the Naïve Bayes classifier performance

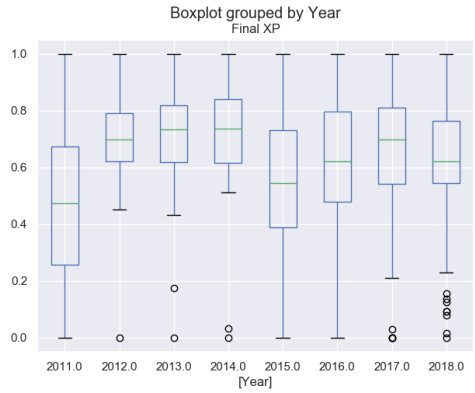

Figure 5.6: Boxplot distribution with outliers

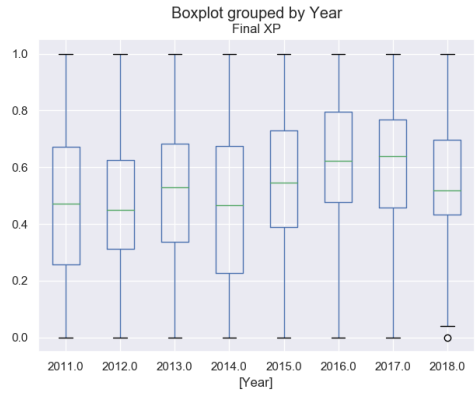

Figure 5.7: Boxplot distribution without outliers

was minimal, so the decision made was to remove them. The fact that all the outliers appear below the "minimum" of each boxplot also supported this decision. This puts as a possibility that the outliers consist of dropout students, which do not belong in the considered student-types.

These same outliers were also removed from the datasets regarding the remaining weeks.

Finally, I have removed the *StudentID* and *Year* attributes as they are considered irrelevant to the classification process.

5.4.2 Data Reduction

Since irrelevant data may contribute to a decrease in the accuracy of some models, I performed feature selection on both datasets. To do so, I tested both filter (by correlation with output variable) and wrapper methods. The Naive Bayes classifier performed better with Recursive Feature Elimination (RFE), where for each of the datasets, a regression model was built, and features were ranked according to when they were eliminated.

Both for the *quartiles* datasets and the *percentiles* datasets, the optimal number of features was selected according to how the students performed until then.

Feature selection was only performed when feeding data to the Naive Bayes classifier, as the remaining tree-based algorithms have their built-in feature selection methods and we are dealing with very few attributes.

5.4.3 Data Transformation

In this phase, the categorical attribute was binarized, as its values could only be "Pass" or "Fail".

The remaining attributes had to be normalized, since the chosen algorithm for the Naïve Bayes classifier was GaussianNB and it has to follow a Gaussian distribution. However, for the remaining tree-

based algorithms, scaling is not necessary because these classifiers are not affected by different scales, since node partitions happen by comparing a feature to a threshold value.

5.4.4 Resampling

By analysing the class distribution for the *quartiles* datasets, it is clear that the number of students labeled as some profile is very close to the number of students labeled as the others. Although the number of *Achievers* consists of a third of the number of *Underachievers*, it is possible to assume that the datasets are relatively balanced and that there is no need for resampling in this case. However, for the *percentiles*, by taking a look at Figure 5.9, the datasets are highly imbalanced, being the number of *Underachievers* close to 350 and the remaining student types between 60 and 80.

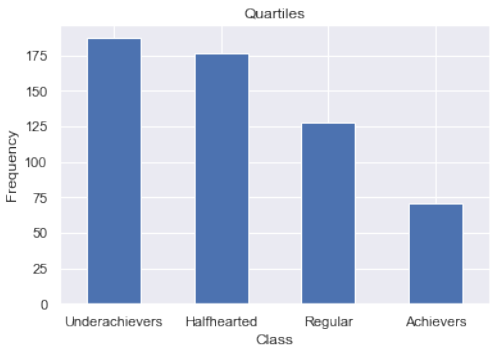

Figure 5.8: Class distribution for the *quartiles* dataset

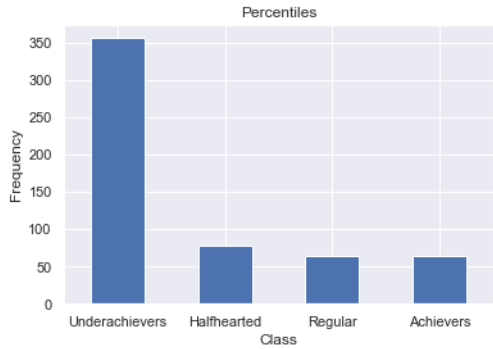

Figure 5.9: Class distribution for the *percentiles* dataset

In order to avoid the possibility of the classifiers being biased for the majority class, I used the Synthetic Minority Oversampling Technique (SMOTE) algorithm to generate synthetic data and randomly create a sample of attributes regarding all classes but the majority class. The performances of the classifiers were tested with and without this balancing technique to understand its impact on the performance of the classifier.

5.5 Classification

For both the *quartiles* and *percentiles* experiments, I followed the premise of dividing the data in a training set, to optimize the model's parameter values and build up the model, and a test set, to evaluate the optimized model. Since the datasets are medium-sized, with a total of 580 samples, I applied Cross-Validation with 10 folds. This way, it is guaranteed that all the data is used and that we get a more accurate estimate of the models' performance.

The graphic representations of the performance of each classifier in each experiment contain not only

the mean accuracy for the Cross-Validation process but also the 95% confidence intervals for which the performance may deviate from the actual value.

5.5.1 Baseline

The performance of the Naïve Bayes algorithm for the *quartiles* and the *percentiles* datasets is represented in Figure 5.10 and Figure 5.11, respectively. On the one hand, for the *quartiles* datasets, Figure 5.10 shows an increase of 45% of the accuracy, from 3 weeks until the end of the semester. Although it slightly increases until reaching 9 weeks, by 11 weeks a minor decrease happens, increasing immediately after. Before the end of the semester, the highest values for accuracy occur by 7 weeks, with 37%. On the other hand, Figure 5.11 indicates a less steep slope from 3 weeks until the end of the semester. This slope indicates an increase of around 15% for the classifier without resampling. In spite of this small increase, the Naïve Bayes classifier seems to present much better prediction accuracy for the earlier weeks, and a value of accuracy very close to the end of the semester already after 7 weeks. By this time, it would be possible to predict the type of student very closely to the type predicted by the end of the semester. However, accuracy is only a good measure when the target variable classes in the data are nearly balanced, which does not happen in this case.

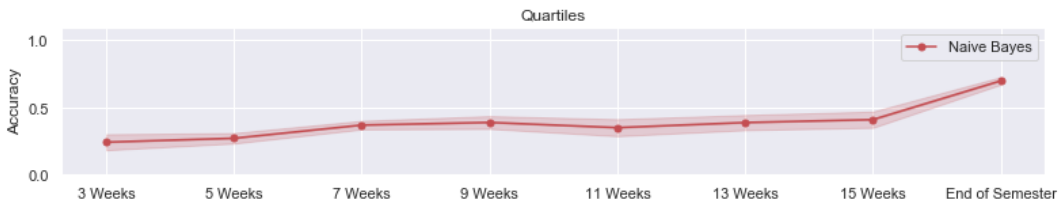

Figure 5.10: Performance of the Naïve Bayes classifiers for the *quartiles* datasets

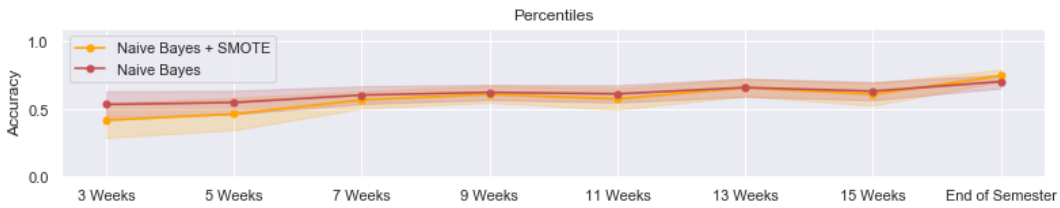

Figure 5.11: Performance of the Naïve Bayes classifiers for the *percentiles* datasets

In order to avoid a misleading analysis of the graphic representations, the performance for the Naïve Bayes algorithm with resampling is also represented. These values are very close to the ones for the classifier with no resampling, but Figure 5.12 shows that with SMOTE, on average, the sensitivity for the majority class decreases, and increases for the minority classes. Taking a look at the plot for 3 weeks, although the sensitivity for the majority class decreases, the percentage of the students who are

being correctly classified as *Regular* increases to 47%, when without SMOTE it was 30%. By the end of the semester, SMOTE contributes to an increase of around 14% of the sensitivity of the *Regular* and *Achiever* classes.

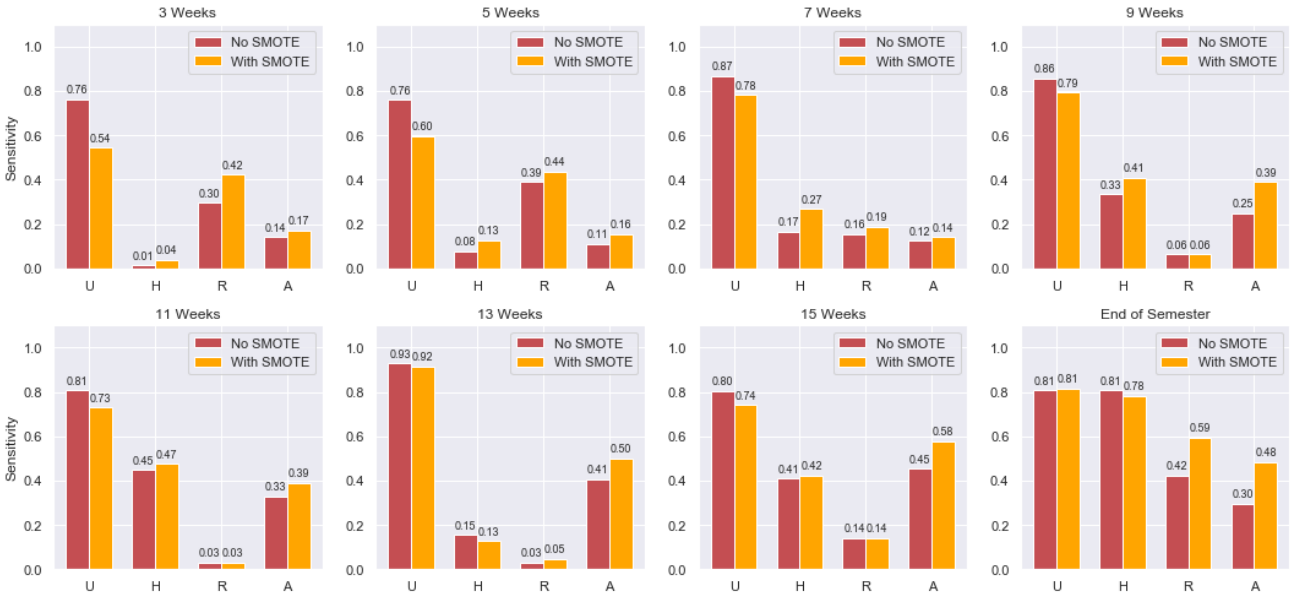

Figure 5.12: Naïve Bayes classifiers' sensitivity for each class along the weeks

5.5.2 Decision Trees

Without tuning any of the available parameters of the Decision Tree classifier, it may result in a tree with an unnecessary number of nodes, which means a highly complex algorithm that may result in low performance for unseen data and in an overfit model. In order to solve this issue, I observed the classifier behavior when changing some specific parameters and its impact on the predictive power of the model.

Decision Trees have a high number of hyperparameters which require fine-tuning in order to obtain the best model which reduces the generalization error as much as possible. In this case, I opted by focusing on four different hyperparameters:

- *max_depth* – when not tuned, this parameter allows the splits of nodes to happen recursively, capturing more information about underlying data and making it difficult for our model to generalize for unseen data. On the other hand, if the tree is too shallow, it may underfit the data, meaning that the model may not learn enough information, returning low values for accuracy.

- *min_samples_split* – consists of a value between 10 and 100% of the samples to be considered at each split of the node and the goal is to find a portion of samples that is not too large to the point

that almost a greedy approach is applied, and not too small that important features to learn are excluded.

- *min_samples_leaf* – consists of the minimum number of samples, or data points, that are required to be present in the leaf node, which is the last node of the tree.
- *max_features* – represents the number of features to consider when looking for the best split.

In order to obtain the optimal values for the hyperparameters of the model, I automated the process by using *GridSearchCV* from the *sklearn* library. By specifying the model that will be used for the hyperparameter tuning process and the list of the parameters and correspondent range of values, *GridSearchCV* performs a cross-validation process to determine the hyperparameter value set which provides the best accuracy. Both for the *quartiles* and *percentiles* datasets, the values for each parameter are represented in Table 5.1 where brackets represent a list of values and parentheses a range between the first and the second values.

Table 5.1: Values selected for each hyperparameter of the Decision Tree classifiers

Hyperparameter	Values
max_depth	[2, 3, 5, 7, 9, 11]
min_samples_split	(0.1, 1)
min_samples_leaf	[1,2,5,10]
max_features	["log2", "sqrt", "auto", None]

Figure 5.13 and Figure 5.14 represent the accuracy for each of the classifiers with 10-Fold Cross-Validation for the *quartiles* and *percentiles* datasets, respectively. As expected, tuning the hyperparameters contributed to an increase in the accuracy of the classifiers in an overall vision. With the tuned parameters, for the *quartiles* datasets, the classifier has an accuracy of around 40% by 3 weeks into the semester. The highest values for accuracy before the end of the semester occur at 7 weeks reaching around 55% and stand relatively stable until 15 weeks.

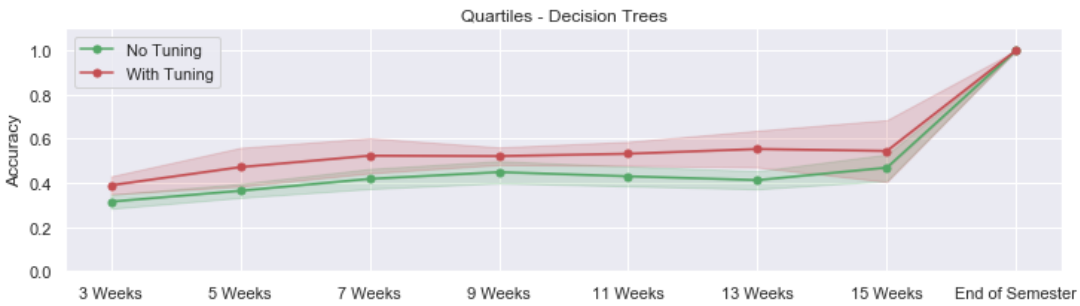

Figure 5.13: Performance of the Decision Tree classifiers for the *quartiles* datasets

As for the *percentiles* datasets, the improvement of the classifier with hyperparameter tuning is also clear. Figure 5.14 represents in detail the obtained results for the datasets. Without tuning, the earliest

phase when accuracy hits the highest values occurs by 7 weeks, which means that by this time, it is possible to predict the student profile with almost 56% accuracy. With no tuning and resampling, by 7 weeks the values for accuracy stabilize around 53% until 15 weeks. The classifiers present the best results with tuned hyperparameters. Between 3 and 15 weeks, there is a difference of only 8% of accuracy, starting at 64%. The line reaches its peak at 9 weeks with an accuracy of around 69%, however the difference between 7 and 9 weeks is about 1%, and between 7 and 15 weeks the values for accuracy remain stable around 68%, meaning that by 7 weeks, it is possible to predict the student profiles with 68% of accuracy. Applying SMOTE to the tuned hyperparameters results in a decrease of 30% for the 3 weeks dataset when comparing to the tuned classifiers, however, by 5 weeks the classifier reaches 50% of accuracy and the peak is hit also at 7 weeks just as before.

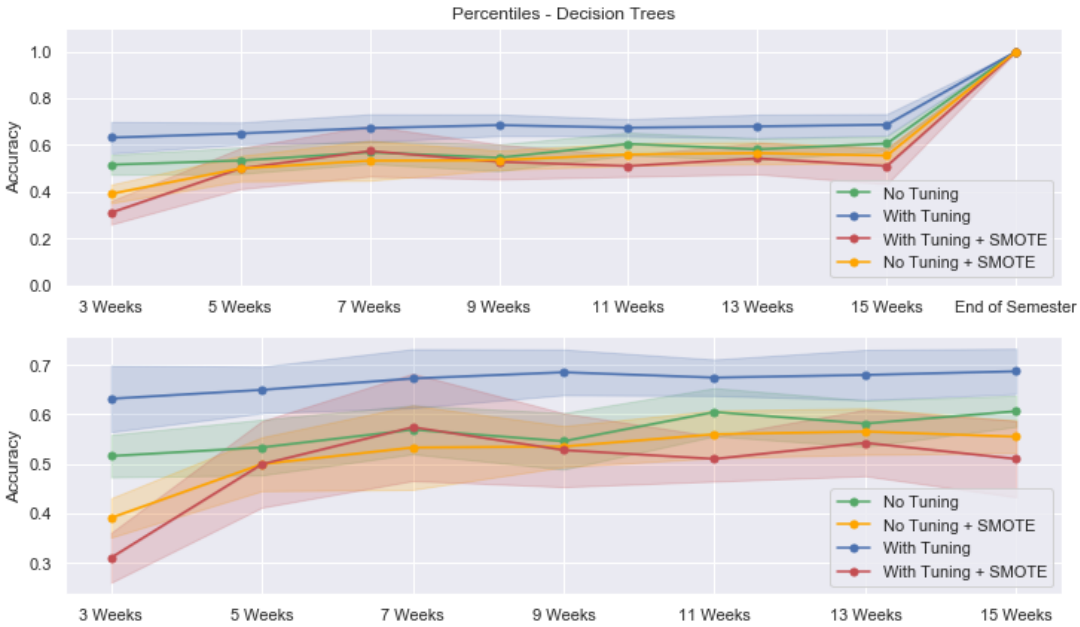

Figure 5.14: Performance of the Decision Tree classifiers for the *percentiles* datasets

Once again, the fact that SMOTE affects the performance of the classifiers with hyperparameter tuning, decreasing accuracy is deceiving. Figure 5.15 shows resampling the data leads to an increase of sensitivity for the minority classes, and it is visible that the percentage of correctly classified students is relatively balanced for all the classes. The impact of SMOTE stands out for the Halfhearted and Regular classes, which sensitivity for the first 11 weeks with no resampling is always close to 0. With SMOTE the values for sensitivity for these two classes increases to values between 15 and 35%.

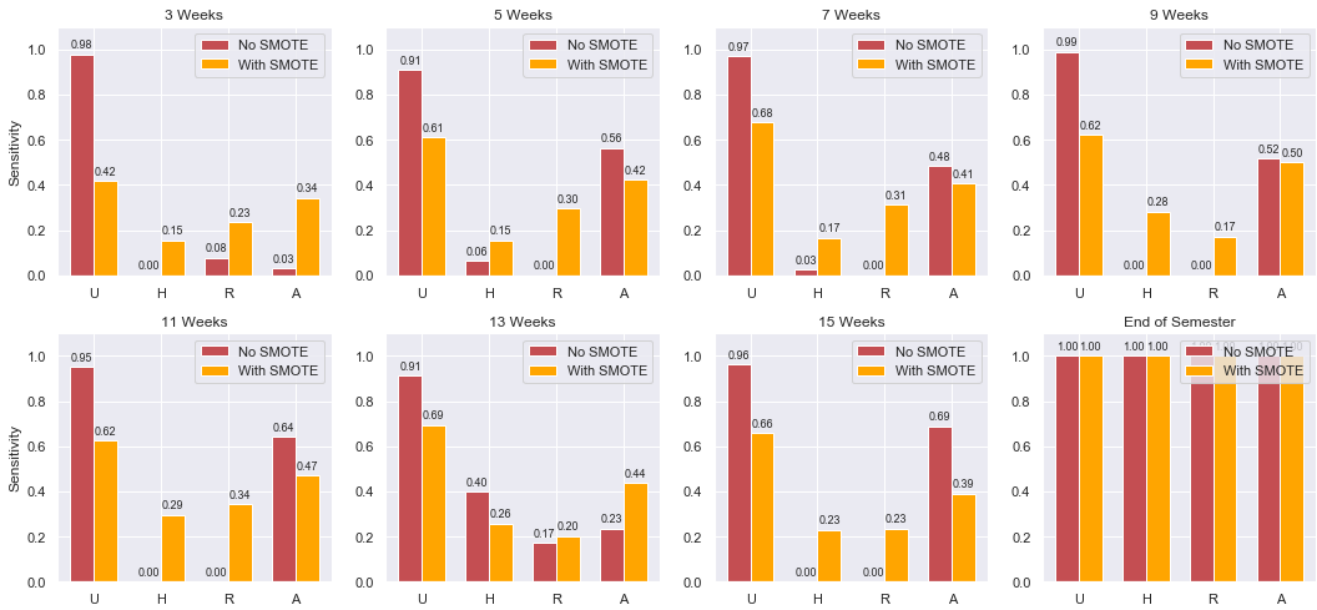

Figure 5.15: Tuned Decision Tree classifiers' sensitivity for each class along the weeks

5.5.3 Random Forests

As a Random Forest is an ensemble algorithm that takes the average of many Decision Trees to arrive at a final prediction, adding to the hyperparameters chosen for the Decision Trees, another main parameter to consider is the number of trees in the forest (n_estimators). In this case, the higher the number of trees, the better, however, adding a lot of trees can slow down the training process.

Similarly to the Decision Tree classifiers hyperparameter tuning method, I used *GridSearchCV* from the *sklearn* library to automate the process of obtaining the optimal values for the hyperparameters of the model. The values for each of parameter is represented in Table 5.2. Again, brackets represent a list of values and parentheses a range between the first and the second values.

Table 5.2: Values selected for each hyperparameter of the Random Forest classifier

Hyperparameter	Values
n_estimators	[10, 20, 30, 50, 80, 100, 200]
max_depth	[2, 3, 5, 7, 9, 11, 13, 15]
min_samples_split	(0.1, 1)
min_samples_leaf	[1,2,5,10]
max_features	["log2", "sqrt", "auto", None]

For the *quartiles* datasets, without or with hyperparameter tuning, the performance of the classifiers is very similar, being the values for the classifiers with tuned parameters more accurate in around 10% at each of the considered weeks. For these datasets, before the end of the semester, the peak is hit at 15 weeks which is very late in the semester and very close to the end. The evolution of the performance of the classifiers can be followed in Figure 5.16.

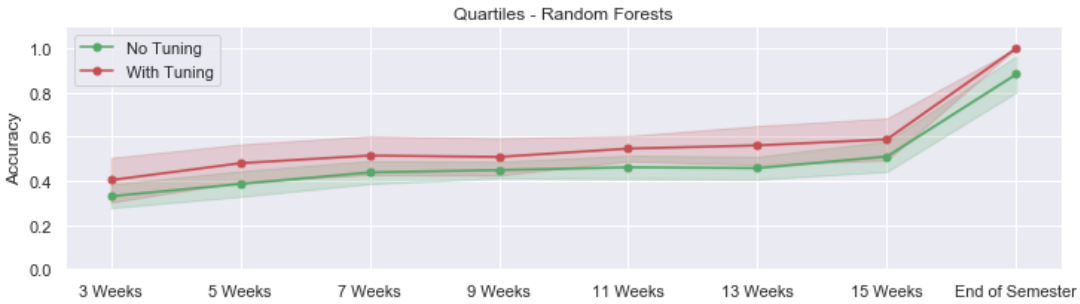

Figure 5.16: Performance of the Random Forest classifiers for the *quartiles* datasets

For the *percentiles* datasets, tuning the parameters without resampling the data shows very promising results as confirmed by Figure 5.17. By 3 weeks into the semester the classifier predicts the student types with 65% of accuracy slowly increasing and reaching a stable value of 69% by 7 weeks. In general, with or without hyperparameter tuning, the Random Forest classification algorithm performs very well, showing slightly worse accuracy (minus 10%) for the non-tuned classifier by 3 weeks and by the end of the semester. Both of the tests performed with SMOTE present higher values for accuracy by the end of the semester, and for the classifiers with tuning and SMOTE. However, just as for the Decision Tree classifiers, SMOTE highly increases the sensitivity of the Halfhearted and Regular classes. Before the end of the semester, the Random Forest classifiers with tuning and no resampling present no sensitivity for both of these classes. The implementation of SMOTE improves these values and decreases the sensitivity for the majority class.

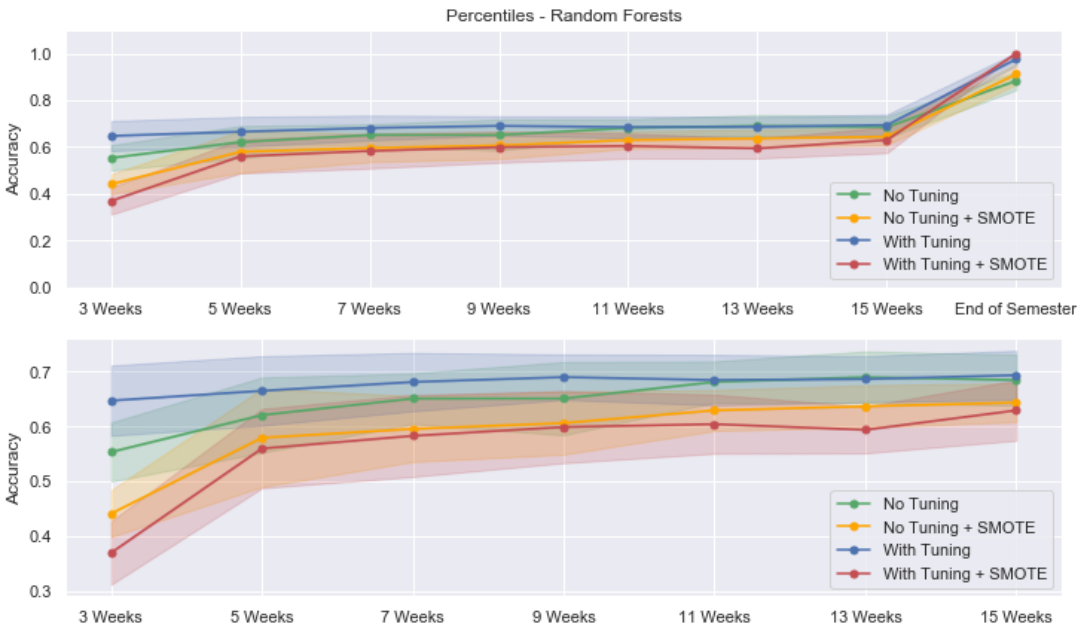

Figure 5.17: Performance of the Random Forest classifiers for the *percentiles* datasets

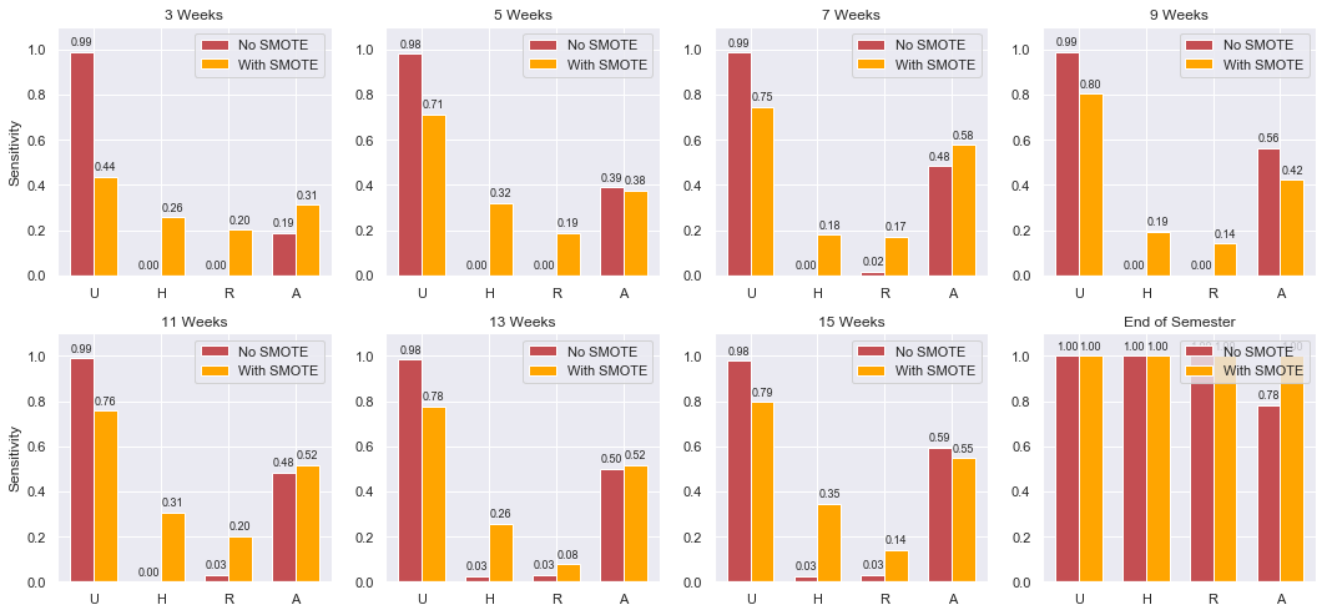

Figure 5.18: Tuned Random Forest classifiers' sensitivity for each class along the weeks

In order to explore the impact of the attributes on the classifier, we opted by representing feature importance in two phases of the semester: by 5 weeks and by 9 weeks. Figure 5.19 and Figure 5.20 show the 15 features with the highest impact on the classifier. By 5 weeks, besides the total XP obtained from quizzes and the total number of completed quizzes and skills, the total number of neutral and positive posts also consist of relevant features. However, by 9 weeks, although the number of positive posts still appears as a relevant feature, it falls from the 5th position to the 13th, which means that the number of posted messages in the forums may be an indicator for participation, but by the end of the semester it is not as relevant as attributes that represent the obtaining of experience or the completion of evaluation items.

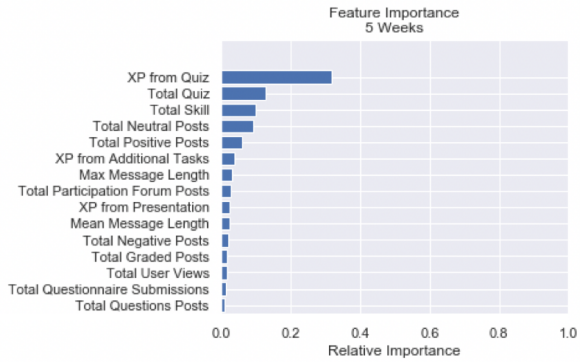

Figure 5.19: Feature Importance at 5 weeks for the *percentiles* dataset

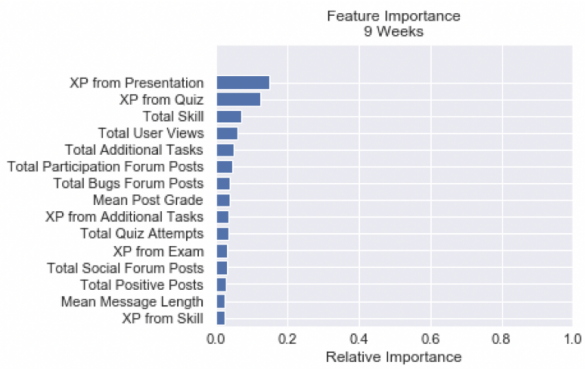

Figure 5.20: Feature Importance at 9 Weeks for the *percentiles* dataset

5.5.4 XGBoost

Similarly to the Decision Tree and Random Forest algorithms, in order to improve the model, hyperparameter tuning is required. There is a broad number of hyperparameters which require tuning in order to optimize the algorithm. In this work, I opted by choosing the following booster parameters:

- *max_depth* – just like for the Decision Tree and Random Forest algorithms, this parameter is used to control over-fitting as it prevents the nodes from splitting recursively and allowing the model to learn relations to a particular sample.
- *learning_rate* – increases the robustness of the model by controlling the weighting of new trees added to the model.
- *n_estimators* – as for the Random Forest algorithm, this parameter represents the number of Decision Trees (or rounds) of the classifier.
- *min_child_weight* – this parameter defines the minimum sum of the weights of the observations needed in a child node.
- *subsample* – denotes the ratio of training samples to be randomly sampled for each tree.
- *colsample_bytree* – similar to max_features, this is the fraction of features which will be randomly selected to be used to train each tree.

The process of obtaining the optimal values for each of the hyperparameters was automated by using the *scikit-optimize* package. I started by defining the search space for the model (Table 5.3) to get the best hyperparameter values, and defined a function that fit the model with these different hyperparameters and measured the model performance, returning the parameter values for which the model performed with higher accuracy. All the tested values consist of a range between the first and the second specified values except for the ones indicated for the *max_depth* which consist of a list.

Table 5.3: Values selected for each hyperparameter of the Gradient Boosting classifier

Hyperparameter	Values
max_depth	[2, 3, 5, 8, 15, 24, 30]
learning_rate	(10**-4, 10**-1)
n_estimators	(10, 400)
min_child_weight	(1, 20)
subsample	(0, 1)
colsample_bytree	(0.3, 1)

Figure 5.21 shows the performance of the XGBoost classifiers for the *quartiles* datasets. Overall, the difference of the accuracy for each of the weeks is minimal, being the most evident the difference of close to 10% by 7 weeks. Before the end of the semester, the best results for accuracy occur by 11 weeks for both tuned and non tuned classifiers.

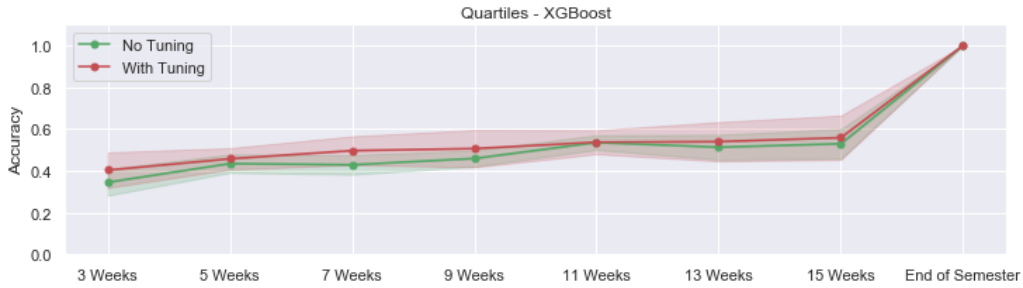

Figure 5.21: Performance of the XGBoost classifiers for the *quartiles* datasets

As for the *percentiles* datasets, the classifiers also performed very similarly, being the best results from the tuned model with no SMOTE, remaining relatively constant along the weeks but reaching a peak of around 70% of accuracy by 9 weeks. In order to clarify the differences between the classifiers, Figure 5.22 represents the performance along the first 15 weeks in detail. With no tuning, the peak is hit by 7 weeks and accuracy stabilizes on values around 67%. With SMOTE, both the plots represent a decrease of accuracy in general but by 5 weeks, both the classifiers' accuracy remains stable apart from a small decrease at 9 weeks for the classifier with no tuning. The increase of the sensitivity for each of the classes with SMOTE is represented in Figure 5.23. Although the performance of this algorithm is slightly better than Decision Trees' and Random Forests', the impact of SMOTE is not as significant.

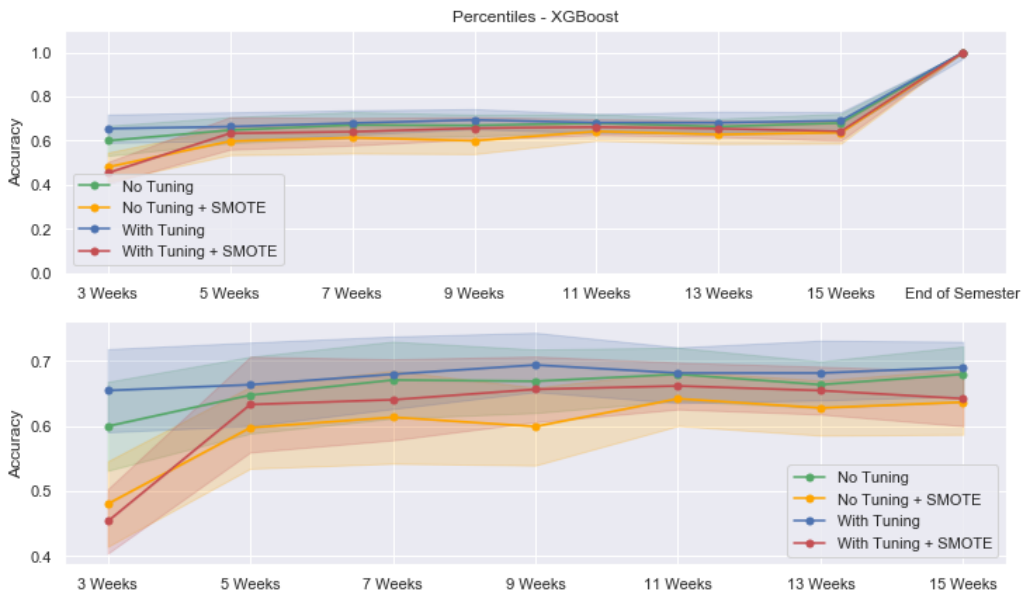

Figure 5.22: Performance of the XGBoost classifiers for the *percentiles* datasets

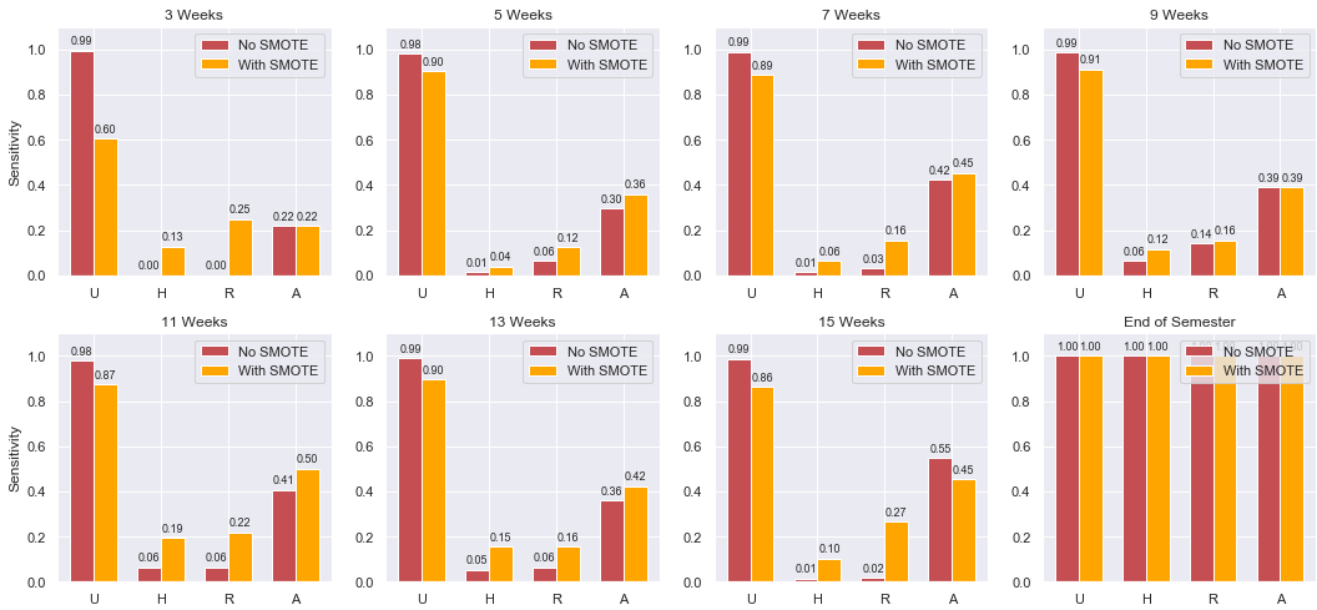

Figure 5.23: Tuned XGBoost classifiers' sensitivity for each class along the weeks

5.5.5 Overall Performance

From the analysis of the graphs of each of the algorithms, for both the *quartiles* and *percentiles* datasets, the highest values for accuracy belong to the classification algorithms with tuned hyperparameters and without SMOTE. However, as previously stated, despite the overall decrease of accuracy, the implementation of SMOTE contributed to an increase of the classifiers' sensitivity for the minority classes. As the *percentiles* datasets are imbalanced and contain more than two classes, in this overall analysis I will consider the performance of the classifiers with hyperparameter tuning and SMOTE as the best performance for the *percentiles* datasets.

Figure 5.24 and Figure 5.25 gather the performance of each of the best performing classifiers on the *quartiles* and *percentiles* datasets respectively. The performance of the Naïve Bayes algorithm is also represented as base line learning algorithm.

Observing the performance of the algorithms in detail for the first 15 weeks represented in Figure 5.24 it is possible to admit for the *quartiles* datasets that although the values for accuracy are not very high to take assured conclusions about the student type prediction, there is a noticeable growth of the performance of the classifiers, as it increased close to 20% for each of the algorithms between 3 and 15 weeks.

By 7 weeks, XGBoost, Decision Trees and Random Forests predict the student types with close to 50% of accuracy, with large confidence intervals. However, by 9 weeks, the Random Forest classifiers present a smaller confidence interval and a higher value of accuracy when comparing to the other tree-based classifiers.

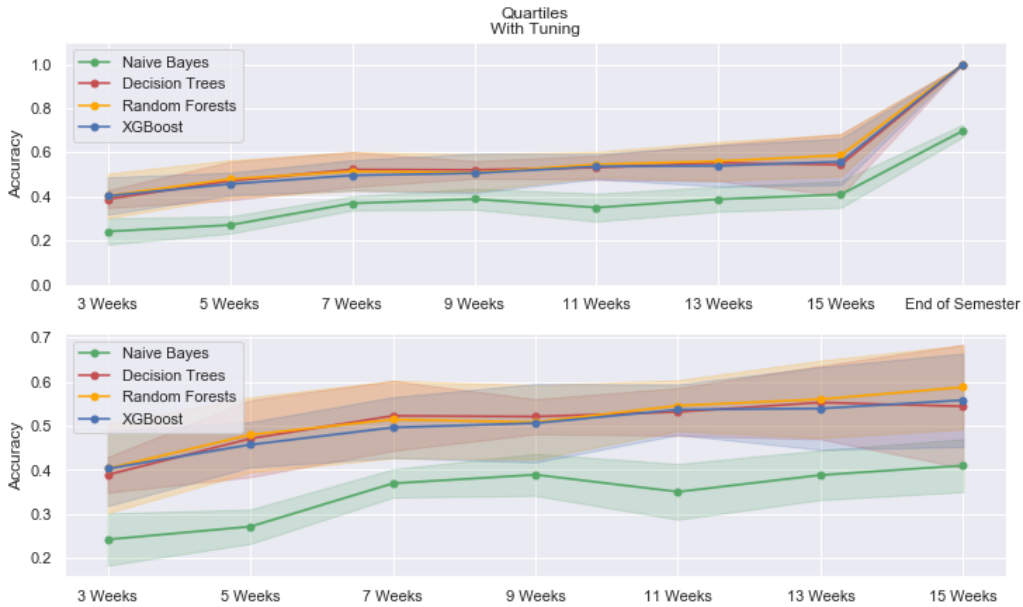

Figure 5.24: Classifiers' performance for the *quartiles* datasets with tuned hyperparameters

So, for these datasets the earlier phase where profile prediction can be done is by 9 weeks, with 52% accuracy for the Random Forest classifier. However, a fraction of right predictions of 50% by the middle of the semester is the same as a random choice of class by the classifiers and there is a risk of existing no connection between the features and the class.

For the *percentiles* datasets, the detailed performance of the best classifiers of each algorithm by the first 15 weeks is represented in Figure 5.25.

In this case, all the tree-based classifiers surpass 50% of accuracy after 5 weeks which is earlier than after 7 weeks for the *quartiles* datasets. The Decision Tree classifiers have a performance very close to the one for the same classifiers on the *quartiles* datasets. The peak is hit after 7 weeks with 53% accuracy. As for the Random Forest classifiers accuracy is maintained over 50% beyond 5 weeks. Beyond 7 weeks, the accuracy stabilizes in a range of 57 to 62%. Finally, for the XGBoost classifiers, after 5 weeks the values of accuracy are relatively stable in a range of 63 to 66%.

Given the fact that the fraction of right predictions for these datasets is slightly better than for the *quartiles* datasets, and that the XGBoost classifier presents a better overall performance, it is possible to assume that the earlier phase where it is possible to predict a student type occurs by 5 weeks. By then, 63% of the students are correctly classified, and since we are evaluating the classifiers taking into account the sensitivity for each of the classes, we're before more reliable results.

Figure 5.26 represents class sensitivity for each of the studied classifiers by 5 weeks into the semester. By this time, XGBoost has higher sensitivity for the Achievers and Underachievers, whereas Random Forests have more balanced sensitivity, with higher values for Underachievers, Halfhearted

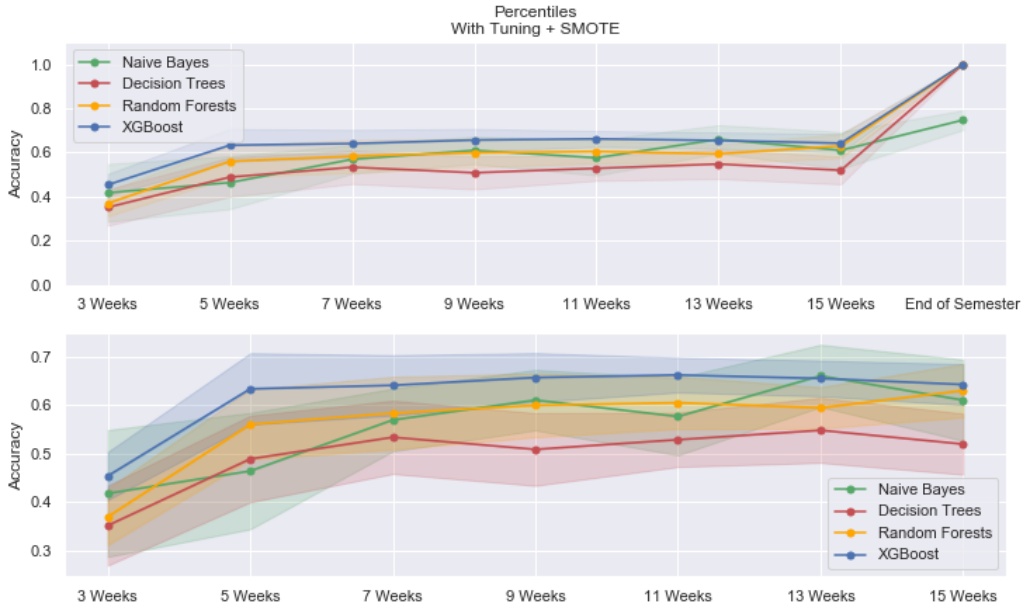

Figure 5.25: Classifiers' performance for the *percentiles* datasets with tuned hyperparameters and SMOTE

and Achievers. Decision Trees admit that by this time, the main percentage of students belongs to the Underachievers, Regular and Achievers. The Naïve approach is more sensitive to Underachievers and Regular students.

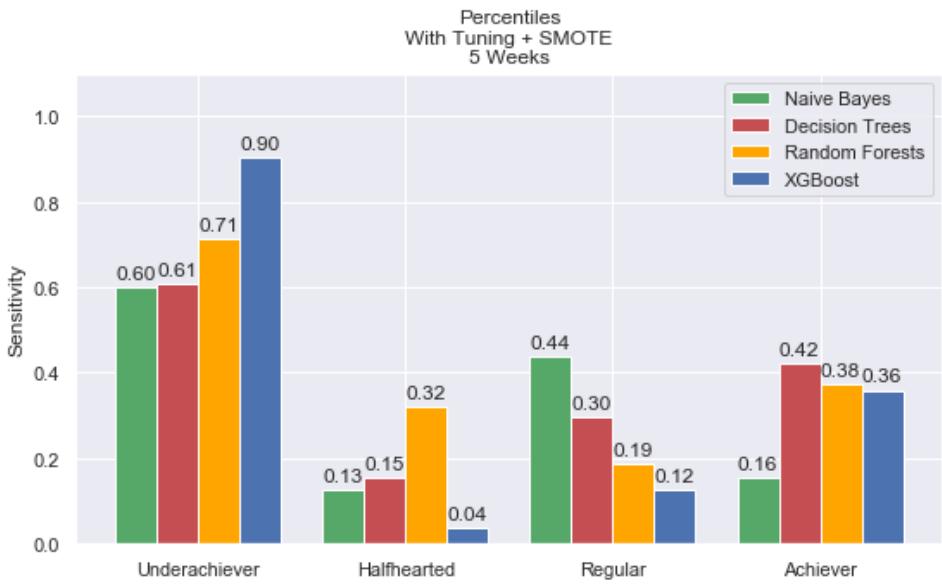

Figure 5.26: Class sensitivity for each of the classifiers for the *percentiles* datasets by 5 weeks

After 5 weeks, the values of sensitivity for the Underachiever and the Achievers remains relatively stable for each of the classifiers until the end of the semester. However, for the Halfhearted and Regular classes, the sensitivity for one may be higher than for the other in one week, and two weeks later may

be lower again, meaning that until the end of the semester, students are mainly fighting to belong to one of these two classes.

The possibility of predicting student profiles by 5 weeks into the semester is supported by the fact that access to activities, such as the Skill Tree and the Achievements, is granted since the beginning of the semester. These activities consist of a total of 40% of the final grade. Since most of the regular courses happen to hand project assignments by midterm, the opportunity for students to excel on this gamified course occurs before that time.

6

Conclusions and Future Work

Contents

6.1 Conclusions

This thesis aimed to build a predictive model to early detect students' learning profiles by exploring the knowledge discovery process, based on data collected from a gamified course.

The data extracted from the Moodle platform of the MCP course presented several challenges for the profiling task, not only because it was scattered across multiple files, but also because of the way it was stored in each of the files. I managed to create a Data Warehouse to save data more efficiently, and to consolidate it in a suitable format for further profile prediction.

The profile prediction phase comprehended the comparison of four machine learning algorithms on different datasets regarding different phases of the semester to find the best performing model in an earlier stage.

From the four evaluated models, results showed that the overall best performing model was XGBoost, which outperformed the other models, Naïve Bayes, Decision Trees, and Random Forests in both of the experiments, being the most effective method for performing student profiling.

The performance of each model on the student profiling task depends on several factors, such as the choices made on the preprocessing phase, the size of the dataset, how balanced the dataset is, and on the hyperparameter chosen for tuning. However, the experiments have shown that even when dealing with balanced datasets, the classification task is still hard to perform with satisfying results.

For the experiment regarding the imbalanced dataset, the best results for accuracy were obtained for tuned classifiers with no resampling. However, these classifiers presented low sensitivity for the minority classes, so by taking both accuracy and sensitivity into account, the most reliable results consisted of the ones obtained for the tuned classifiers with SMOTE.

The primary concerns of this work derive from the fact that this consists of a hard problem, where we are acknowledging four highly imbalanced classes and where the classifiers return low values of average accuracy and individual class sensitivity for each of the considered weeks, except for the end of the semester. However, an analysis of the evolution of the accuracy and class sensitivity along the weeks shows that the most significant difference between both measures occurs between 3 and 5 weeks. Beyond this phase, the evaluation measures tend to stabilize. Therefore, the earlier stage where it is possible to predict the students' learning profiles is after 5 weeks, with an accuracy of around 63%.

6.2 Future Work

There are several potential extensions of the scope of this thesis.

During the course, some activities require a minimum grade, while some of them are even not mandatory. It is assumed that the Underachievers would do the bare minimum to pass the course, while the Achievers would try to gather the most experience points possible until the end of the semester. As for

the Halfhearted and the Regular students, these may me motivated by different course activities. One potential extension of this thesis would be performing a detailed analysis of the built models in order to distinguish the activities that support each of the student profiles.

Since profile prediction in education support teachers to be aware of their students' performance to intervene whenever necessary, either to motivate them or to give feedback, another potential extension could be building two automated systems that would aid the teachers with these tasks. On the one hand, to stimulate the students, it would be interesting to create a recommendation system that, according to the current student profile, would present the main activities that would have to be concluded to achieve the next student profile. On the other hand, in terms of feedback, I also propose a system that would analyze the student's current profile and the tasks he or she concluded by then. The system could retrieve useful information such as all the course activities that a student belonging to that same profile would have completed by then to aid the students to be aware of what they may be missing.

Bibliography

[1] S. Deterding, R. Khaled, L. E. Nacke, and D. Dixon, "Gamification: Toward a definition," in *CHI 2011 gamification workshop proceedings*, vol. 12. Vancouver BC, Canada, 2011.

[2] G. Zichermann and C. Cunningham, *Gamification by design: Implementing game mechanics in web and mobile apps*. " O'Reilly Media, Inc.", 2011.

[3] S. Deterding, D. Dixon, R. Khaled, and L. Nacke, "From game design elements to gamefulness: defining gamification," in *Proceedings of the 15th international academic MindTrek conference: Envisioning future media environments*. ACM, 2011, pp. 9–15.

[4] B. Taspinar, W. Schmidt, and H. Schuhbauer, "Gamification in education: A board game approach to knowledge acquisition," *Procedia Computer Science*, vol. 99, no. October, pp. 101–116, 2016. [Online]. Available: http://dx.doi.org/10.1016/j.procs.2016.09.104

[5] G. Barata, S. Gama, J. Jorge, and D. Gonçalves, "Engaging Engineering Students with Gamification: An empirical study," *Vs-Games 2013*, no. January, pp. 1–8, 2013. [Online]. Available: http://web.ist.utl.pt/~daniel.j.goncalves/publications/2013/barata2013a.pdf

[6] C. Romero and S. Ventura, "Educational data mining: a review of the state of the art," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 40, no. 6, pp. 601–618, 2010.

[7] K. Liang, Y. Zhang, Y. He, Y. Zhou, W. Tan, and X. Li, "Online Behavior Analysis-Based Student Profile for Intelligent E-Learning," *Journal of Electrical and Computer Engineering*, vol. 2017, 2017.

[8] H. Bydžovská, "A comparative analysis of techniques for predicting student performance," in *Proceedings of the 9th International Conference on Educational Data Mining 2016*, 2016.

[9] A. Silva and C. Antunes, "Mining Multi-dimensional Patterns for Student Modelling," *Educational Data Mining 2014*, pp. 393–394, 2014. [Online]. Available: http://web.ist.utl.pt/andreia.silva/articles/silva2014edm.pdf

[10] A. Vale and S. C. Madeira, “Mining Coherent Evolution Patterns in Education through Biclustering,” *Proceedings of the 7th International Conference on Educational Data Mining (EDM)*, pp. 391–392, 2014.

[11] J. McBroom, B. Jeffries, I. Koprinska, and K. Yacef, “Mining Behaviours of Students in Autograding Submission System Logs,” *Edm ’16*, pp. 159–166, 2016.

[12] S. J. Lee, Y.-e. Liu, Z. Popović, and Z. Popovi, “Learning Individual Behavior in an Educational Game : A Data-Driven Approach,” *Proceedings of the 7th International Conference on Educational Data Mining (EDM)*, no. Edm, pp. 114–121, 2014. [Online]. Available: http://grail.cs.washington.edu/wp-content/uploads/2015/08/lee2014lib.pdf

[13] M. Gross, B. Solenthaler, S. Klingler, and T. Käser, “Temporally Coherent Clustering of Student Data,” *Proceedings of the 9th International Conference on Educational Data Mining*, pp. 102–109, 2016.

[14] A. Domínguez, J. Saenz-De-Navarrete, L. De-Marcos, L. Fernández-Sanz, C. Pagés, and J. J. Martínez-Herráiz, “Gamifying learning experiences: Practical implications and outcomes,” *Computers and Education*, vol. 63, pp. 380–392, 2013. [Online]. Available: http://dx.doi.org/10.1016/j.compedu.2012.12.020

[15] S. Deterding, S. Bjork, L. E. Nacke, D. Dixon, and E. Lawley, “Designing Gamification: Creating Gameful and Playful Experiences,” *CHI 2013: Changing Perspectives, Paris, France*, pp. 3263–3266, 2013.

[16] B. Reeves and J. L. Read, *Total engagement: How games and virtual worlds are changing the way people work and businesses compete*. Harvard Business Press, 2009.

[17] J. Hamari, J. Koivisto, and H. Sarsa, “Does gamification work?–a literature review of empirical studies on gamification,” in *2014 47th Hawaii international conference on system sciences (HICSS)*. IEEE, 2014, pp. 3025–3034.

[18] H. Cavusoglu, Z. Li, and K.-W. Huang, “Can gamification motivate voluntary contributions?: the case of stackoverflow q&a community,” in *Proceedings of the 18th ACM conference companion on computer supported cooperative work & social computing*. ACM, 2015, pp. 171–174.

[19] T. Althoff, R. W. White, and E. Horvitz, “Influence of pokémon go on physical activity: study and implications,” *Journal of medical Internet research*, vol. 18, no. 12, 2016.

[20] T. Reiners and L. C. Wood, *Gami cation in Education and Business*. Springer, 2015.

[21] K. S. Tekinbas, R. Torres, L. Wolozin, R. Rufo-Tepper, and A. Shapiro, *Quest to learn: Developing the school for digital kids*. MIT Press, 2010.

[22] A. Marczewski, *Gamification: a simple introduction*. Andrzej Marczewski, 2013.

[23] K. M. Kapp, *The gamification of learning and instruction: game-based methods and strategies for training and education*. John Wiley & Sons, 2012.

[24] J. J. Lee and J. Hammer, "Gamification in education: What, how, why bother?" *Academic exchange quarterly*, vol. 15, no. 2, p. 146, 2011.

[25] J. P. Gee, "What video games have to teach us about learning and literacy," *Computers in Entertainment (CIE)*, vol. 1, no. 1, pp. 20–20, 2003.

[26] D. Dicheva and C. Dichev, "Gamification in education: Where are we in 2015?" in *E-Learn: World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education*. Association for the Advancement of Computing in Education (AACE), 2015, pp. 1445–1454.

[27] J. Hamari, D. J. Shernoff, E. Rowe, B. Coller, J. Asbell-Clarke, and T. Edwards, "Challenging games help students learn: An empirical study on engagement, flow and immersion in game-based learning," *Computers in Human Behavior*, vol. 54, pp. 170–179, 2016.

[28] J. Nakamura and M. Csikszentmihalyi, "The concept of flow," in *Flow and the foundations of positive psychology*. Springer, 2014, pp. 239–263.

[29] S. Lamborn, F. Newmann, and G. Wehlage, "The significance and sources of student engagement," *Student engagement and achievement in American secondary schools*, pp. 11–39, 1992.

[30] C. Cheong, F. Cheong, and J. Filippou, "Quick quiz: A gamified approach for enhancing learning." in *PACIS*, 2013, p. 206.

[31] I. Glover, "Student perceptions of digital badges as recognition of achievement and engagement in co-curricular activities," in *Foundation of digital badges and micro-credentials*. Springer, 2016, pp. 443–455.

[32] G. Siemens and R. S. d Baker, "Learning analytics and educational data mining: towards communication and collaboration," in *Proceedings of the 2nd international conference on learning analytics and knowledge*. ACM, 2012, pp. 252–254.

[33] C. Romero and S. Ventura, "Data mining in education," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 3, no. 1, pp. 12–27, 2013.

[34] R. Baker *et al.*, “Data mining for education,” *International encyclopedia of education*, vol. 7, no. 3, pp. 112–118, 2010.

[35] W. H. Inmon, “The data warehouse and data mining,” *Communications of the ACM*, vol. 39, no. 11, pp. 49–51, 1996.

[36] R. Kimball and M. Ross, *The data warehouse toolkit: The definitive guide to dimensional modeling*. John Wiley & Sons, 2013.

[37] R. Kimball, M. Ross, W. Thornthwaite, J. Mundy, and B. Becker, *The data warehouse lifecycle toolkit*. John Wiley & Sons, 2008.

[38] G. Barata, S. Gama, J. A. Jorge, and D. J. Gonçalves, “Relating gaming habits with student performance in a gamified learning experience,” in *Proceedings of the first ACM SIGCHI annual symposium on Computer-human interaction in play*. ACM, 2014, pp. 17–25.

[39] G. Barata, S. Gama, J. Jorge, and D. Gonçalves, “Early prediction of student profiles based on performance and gaming preferences,” *IEEE Transactions on Learning Technologies*, vol. 9, no. 3, pp. 272–284, 2016.

[40] X. Wu, V. Kumar, J. R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu, S. Y. Philip *et al.*, “Top 10 algorithms in data mining,” *Knowledge and information systems*, vol. 14, no. 1, pp. 1–37, 2008.

[41] C. Sammut and G. I. Webb, *Encyclopedia of machine learning*. Springer Science & Business Media, 2011.