Vitiello Massimo, BSc

User Prediction in MOOCs

Master's Thesis

to achieve the university degree of

Diplom-Ingenieur

Master's degree programme: Computer Science

submitted to

Graz University of Technology

Supervisor

Assoc.Prof. Dipl.-Ing. Dr.techn. Denis Helic

Institute for Interactive Systems and Data Science
Head: Univ.-Prof. Dipl.-Inf. Dr. Stefanie Lindstaedt

Second supervisor: Assoc.Prof. Dipl.-Ing. Dr.techn. Christian Gütl
External supervisor: Prof. Vanessa Chang, Curtin University

Graz, October 2017

This document is set in Palatino, compiled with pdfLaTeX2e and `Biber`.

The LaTeX template from Karl Voit is based on KOMA script and can be found online: https://github.com/novoid/LaTeX-KOMA-template

Affidavit

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly indicated all material which has been quoted either literally or by content from the sources used. The text document uploaded to TUGRAZonline is identical to the present master's thesis.

Date	Signature

Abstract

Massive Open Online Courses (MOOCs) rapidly stirred up hype and interest in recent years. Advocates of this technology appreciate the possibility to enroll in massive online classes, with no need of physical presence, particular knowledge, nor high economic means. However, MOOCs suffer from low completion rates; a large number of people enroll, but only a few individuals complete a course. The high attrition of these systems is perceived by the Online Learning community as one of the most compelling issues.

In this thesis, we analyze MOOCs from two different platforms offered with various modalities. Specifically, we employ machine learning techniques, such as Support Vector Machines (SVM) and Boosted Decision Trees, to classify users as either Completers or Dropouts. Furthermore, we experiment with system-to-system and multisystem settings and verify to which extent different systems and domains can be analyzed with a general approach. Moreover, we evaluate the features used in our analysis to identify those that better predict whether users are likely to drop out.

Our results show that detecting dropouts across different systems is a tractable problem. Besides, we discover unique properties of both systems that affect the accuracy of the classification task. Furthermore, we identify the features of our model that best help us to classify users into completers and dropouts. We firmly believe that our presented work represents a solid foundation and stepping-stone towards more detailed analysis of attrition and user behavior in MOOCs.

Kurzfassung

Massive Open Online Courses (MOOCs) haben in den letzten Jahren sehr schnell an Relevanz und Interesse gewonnen. BefürworterInnen dieser Technologie schätzen die Möglichkeit an sogenannten "massive online classes" teilzunehmen, ohne Ortsgebundenheit, ohne spezielle Vorkenntnisse oder finanzielle Hürden. Dennoch weisen MOOCs geringe Abschlussraten auf; eine Großzahl von Menschen schreibt sich in Kurse ein, aber nur wenige dieser nehmen auch erfolgreich bis zum Ende teil. Die hohe Ausfallrate dieser Systeme wird von der Online Learning Community als eine der größten Herausforderungen gesehen.

In dieser Arbeit analysieren wir MOOCs von zwei verschiedenen Plattformen in verschiedenen Szenarien. Wir verwenden maschinelles Lernen, wie Support Vector Machines (SVM) und Boosted Decision Trees, um UserInnen als "Completers" und "Dropouts" zu klassifizieren. Weiter erforschen wir system-to-system und multisystem Szenarien um darzustellen, in welchem Ausmaß verschiedene Systeme und Domains mit einem einheitlichen Verfahren analysiert werden können. Ebenso bewerten wir die in der Analyse verwendeten Faktoren, um darzustellen, mit welchen dieser sich eine Klassifizierung am sichersten vornehmen lässt.

Unsere Ergebnisse zeigen, dass das Aufspüren von potenziellen "Dropouts" über verschiedene Systeme hinweg eine machbare Herausforderung darstellt. Ausserdem erschließen wir spezielle Eigenschaften beider Systeme, die die Genauigkeit der Klassifizierungsmaßnahme beeinflussen. Aus den Ergebnissen streichen wir jene Faktoren unseres Modelles hervor, welche für die Einteilung in "Completers" und "Dropouts" am besten funktionieren. Wir gehen davon aus, dass die vorliegende Arbeit ein Fundament und einen Meilenstein in der fortlaufenden Analyse der Ausfallrate und des UserInnen-Verhaltens in MOOCs darstellt.

Acknowledgements

First of all, I want to thank my main supervisor Dr. Denis Helic, who was always available to discuss my doubts and problems. I'm glad to have had the opportunity to work with him not only on my thesis but also on a certain number of papers. His expertise helped me a lot through my projects.

I also have to immensely thank Simon Walk for his support, insights and help through my work. Simon taught me not only how to do proper research but also how to write scientific papers. He would always find the time to talk even when we were on two opposite sides of the globe.

I am also incredibly grateful to my second supervisor Dr. Christian Gütl. His deep knowledge in Online Education helped me very much during my work. Thanks to him, I had the chance to publish my first paper and to attend my first conference. Again, it was thanks to him that I had the great opportunity to work at Curtin University in Western Australia.

It was also an honor for me to work with the people at Curtin Learning and Teaching. Specifically, I want to thank Vanessa Chang for welcoming me in Perth, letting me use Curtin data for my analysis and for taking me Kangaroo spotting. I also want to thank Simon Huband. He would always find time for a discussion and was very helpful during my time at Curtin.

Furthermore, I have to thank Universidad Galileo in the person of Rocael Hernández for allowing me to analyze their data and the MOOC Maker Project for the help and support during my research.

Additionally, I have to thank Corinna and Giorgio for bear with me every day. Their support and the fun we have together are immeasurable.

Last but not least, I thank my family from the bottom of my heart for always being there for me even when the distance makes it harder. I would never be able to thank them enough for everything they did and do for me.

Contents

List of Figures

1 Introduction

Massive Open Online Course (MOOC) is a relatively recent phenomenon of online education. MOOCs promise to offer free-of-charge or low-cost university-level learning contents in the form of online courses to a multitude of individuals, overcoming the geographical barriers by exploiting the penetrability of the Internet.
In the fall of 2011, Stanford University offered three courses that catalyzed the buzz around MOOCs [19, 70]. Students of Stanford University could take these classes for credits. Simultaneously, these courses were mirrored through the Stanford website and made freely available as no-credit courses for individuals outside of the university. The response was enormous, with over 200,000 enrolments from more than 190 countries.
The success of these courses led to the development of ad hoc platforms that universities could use to publish their MOOCs; 2012 saw the debut of Udacity[1], Coursera[2] and edX [3], later referred to as the "Big-Three" of online education [61]. Consequently, MOOCs received such a big media coverage and interest from higher education institutions and education professionals that 2012 was dubbed the "Year of the MOOC" [17, 51, 69].
The era of MOOCs started, and it has not ended yet. From the roughly 100 MOOCs existing in 2012, the number of courses available reached more than 6,800 with more than 700 universities worldwide involved in creating and offering MOOCs to a total number of enrolled individuals as high as 58 million as of the end of 2016 [56]. Prognosis for the coming years estimates a compound growth rate (CAGR) for MOOCs of 36%, which will make MOOCs an 8.5 billion USD industry by 2020. Since the invention of the printing press some 480 years ago, there has not been any innovation in the

[1] https://www.udacity.com/
[2] https://www.coursera.org/
[3] https://www.edx.org/

educational area with a comparable growth [54].
These numbers testify how the MOOCs movement hasn't stopped yet and is still growing.

1.1 The Limitations and Challenges of MOOCs

MOOCs hit the world of online learning as a tsunami of change [45] and initially gathered a lot of enthusiasm and advocates among education professionals and institutions. MOOCs have been praised for the dimension of the crowd they manage to reach, which allows such environments to deliver knowledge even in remote places and to individuals that cannot afford traditional university high education.
However, the more MOOCs were offered and used the more their flaws and drawbacks became clearer, further entrenching the stigma surrounding these systems. One of the most compelling issues of MOOCs is their high attrition rate. Regardless of the particular system considered, MOOCs usually experience attrition rates of 90% [34, 46], meaning that most of the users that initially enroll in a MOOC end up not completing the course. These users are referred to as *Non Completers* (in contraposition to the *Completers*, those users that successfully pass a MOOC), *Disengager* or *Dropouts* [43].
Dropout in MOOCs has been linked to various aspects, including user's motivation to enroll and their expectations [5, 25, 28, 38, 40], a lack of social interactions that leads to a sense of isolation [37, 50], and no penalty for dropping out among others [73].
All of these aspects combined complicate the analysis of online courses. Understanding the internal dynamics of MOOCs and the behavior of the users is a complex task. Even if the traditional metrics of higher education attrition and dropout rates could be misleading when applied to MOOCs [20, 48], it is widely agreed that an improvement of the overall dropout rates, a deeper understanding of reasons that lead users to abandon MOOCs and at what time such a drop out happens are directions that research should focus on [74].

1.2 Approach & Research Questions

This work focuses on the analysis of MOOCs logs using machine learning techniques, with the goal of constructing a model for the prediction of Dropouts. The questions this research aims to answer are two-fold.

First, we classify users as Completers or Dropouts focusing on the initial stage of MOOCs. For example, we collect information from the first days after the start of a MOOC or consider an absolute number of user interactions with the system and try to predict whether users will succeed in the MOOCs. Alternatively, we classify the users considering the number of interactions (in percentage) with the MOOC and how much time they spend on it.
Specifically, we conduct these experiments for each MOOC in our datasets, and we inspect if considering more days or number of interactions influences the accuracy of the classification.
The final premise of this thesis is to investigate similarities and differences between MOOCs of the same system and verify how well we can predict dropouts at an early stage of MOOCs.

Second, this work also aims at analyzing logs of MOOCs coming from different sources. Intuitively, MOOCs with different domains, scopes, goals, and audiences lead to different course structures and organizations.
Specifically, we analyze MOOCs organized according to a well-defined schedule with deadlines for assignments, submissions, and exams (fixed schedule) and MOOCs that do not have deadlines where the registered users can interact at their own pace (self-paced).
Additionally, the portal or website on which a MOOC is available also carries peculiarities due to design settings and system constraints that further increase the heterogeneity of courses. Due to these differences, analyzing MOOCs from various sources is a challenging task. For example, information existing for one system may be irrelevant or even not available at all for another system.
Despite the particularity of each course and system, all MOOCs suffer from high dropout rates [34]. This shared issue is the driving factor for a more homogeneous and rigorous analysis that targets and includes MOOCs from

various sources. To investigate this aspect, we consider MOOCs from different systems to find analogies that can help build up a baseline for an analysis of MOOCs at a multisystem level.
The joint study of different systems, together with the analysis and evaluation of the used features, is the second research question of the here presented work.

1.3 Impact

Correctly separating the completers from the dropouts allows for a precise class profiling, which offers further details about the different behaviors of the users in MOOCs. Furthermore, once the potential dropouts are identified, course administrators have the chance to intervene in real-time to help them and keep those at risk of fail users engaged. For example, administrators of MOOCs could directly reach out to these potential dropouts to understand their difficulties and, thus, provide specialized help where needed.
Alternatively, for MOOCs that include group activities, it would be reasonable to pair up these potential dropouts with better performing users. With this approach, the presence of strongly motivated users could benefit the potential dropouts, virtually boosting their motivation and engagement in the course. Overall this approach could also provide users the possibility to compare their methods with their peers, thus offering a fast channel for self-assessments and promoting peer-to-peer help. Therefore, peer-to-peer collaboration also represents a way to diminish the amount of work of instructors in favor of a more open and social learning experience, which could directly influence users' success.
Moreover, techniques for early detection of dropouts could be embedded in existing systems to create real-time autonomous models able to identify users at risk of failing during the initial part of MOOCs. Such models could be used to automatically contact these potential dropouts (for example by sending emails or showing notification in their personal page).
Furthermore, this would be a tremendously powerful tool in the hands of instructors, who would have information about the set of at-risk users at a point in time when this indication is still valuable. In fact, the sooner

indications about the soon-to-be dropouts are available, the more time is left for instructors to intervene and potentially prevent those dropouts.

1.4 Thesis Outline

The rest of this thesis is structured in 6 chapters. After the introduction, Chapter 2 features an overview of the state of the art of research in MOOCs. Chapter 3 describes the analyzed systems, their considered datasets and explains and motivates the proposed experiments. Chapter 4 presents the results of the experiments. Chapter 5 focuses on the understanding and justification of the obtained results. Finally, Chapter 6 concludes the work by answering the research questions and outlining future works.

1.5 Contributions

All the experiments presented in this thesis have been designed, set up and performed completely by the author. The discussion and explanation of the obtained results have been done in collaboration with Denis Helic, Simon Walk, Christian Gütl, Rocael Hernandez and Vanessa Chang.
Particularly, initial results and experiments on a part of the dataset of Universidad Galileo have been accepted and presented at eMOOCs 2016 [68].
Results of attrition analysis on an extended set of MOOCs of Universidad Galileo have been published and presented at the 6th International Workshop on Learning Technology for Education challenges (LTEC 2017) [66].
Experiments on Curtin University's datasets and the combination of datasets from Universidad Galileo and Curtin University have been accepted and were presented at the European Conference on Technology Enhanced Learning (EC-TEL 2017) [67].
This thesis is based on the experience gained from these previous works and further extends on the obtained results.

2 Related Work

This chapter presents an overview of the relevant literature to the study and analysis of MOOCs, and it is logically organized into two parts. In the first section, we discuss research that investigated and analyzed MOOCs trying to understand the point of view of the users. The second section covers research tasked with predicting users participation in MOOCs, with an emphasis on dropout detection and mitigating practices.

2.1 Analysis of MOOCs

These works try to grasp a better understanding of the learning process in MOOCs. Notably, the considered aspects include reasons for users to enroll, their expectations, goals, and motivation towards the MOOC that might cause users to drop out eventually.

The *Funnel of Participation* is a well-trusted model to describe users engagement process in online learning scenarios. Originally proposed by Clow [14], it is one of the very first standards that has been introduced to characterize the interactions of users with web-based courses environment. Therefore, many subsequent works have expanded upon this initial model. It is inspired by the *Purchase Funnel Model*, widely used in marketing and sales contexts.
The process in which users enroll, interact and progress in a MOOC is described in 4 phases; Awareness—potential learners become aware of the existence of a MOOC; Registration—those interested in the course sign up for it; Activity—enrolled users engage in typical MOOC's learning activities; Progress—the action of remaining active and engaged until the end of the course.

Each of these phases is characterized by a certain quota of users who lose interest and eventually abandon the MOOC. This phenomenon is called *Attrition* and the percentage of users that fail to complete a course and drop out is denoted as *Dropout rate*. Although being a relatively simple model, the *Funnel of Participation* has been widely accepted for its simplicity and flexibility.

The works presented in Gütl et al. [25] and Guetl et al. [22] introduced the *Attrition Model for Open Learning Environment Setting (AMOES)*, a more detailed version of the *Funnel of Participation* intended for a deeper understanding of the reasons for Attrition.
The authors analyzed a 4-week MOOC offered by Universidad Galileo in Guatemala, characterized by a dropout rate of 92%. At the end of this MOOC, a survey was sent out to all users who failed to complete the course successfully. The purpose of the survey was to gather feedback from the users about their experience and opinions regarding the MOOC.
The analysis of users' answers led the authors to propose their AMOES model, which divides the Attrition into *Unhealthy* and *Healthy*. Healthy Attrition represents users that no longer have interest to progress through the course. Under this umbrella the authors further individuated 3 different classes of users: *Exploring User*, *Content Learner* and *Restricted Learner*. *Exploring Users* are those who subscribe driven by curiosity about a new learning environment rather than a pure desire to complete a certain course. In the same way, *Content Learners* are users interested only in particular topics offered during the course. Therefore, they can take advantage of the free access and no-entry barrier policies of the MOOC to gain the desired knowledge. Finally, *Restricted Learners* represent users that exhibit an audit behavior and do not seem to have a strong interest in any particular topic of the MOOC.
For these reasons, the authors did not perceive the Healthy Attrition as a negative factor, but rather a consequence strictly related to the very nature of MOOCs. On the other hand, Unhealthy Attrition represents users that, although willing to complete a MOOC, fail to do so. Reasons for this could be both external and internal. External factors include users' work and job commitments and limitation of technologies, while internal ones could be, for example, an absence of support and poor organization of the course, as well as a lack of prerequisite knowledge and the inability of the users

to correctly engage in the forum and organize their time. The authors concluded that Healthy Attrition is the most urgent issue when it comes down to lowering the dropout rates.

Kizilcec et al. [39] investigated a set of three computer science MOOCs, by analyzing assessment and video interactions and classified the learners using k-means clustering approach.
The authors identified four classes of learners: *Completing*, *Auditing*, *Disengaging* and *Sampling*. *Completing* are those learners who completed the majority of assignments or, at least, attempted to do so. *Auditing* users are those who engaged mostly by watching videos rather than solving assignments. Learners classified as *Disengaging* exhibited an engaging style similar to the *Completing* at the beginning of the course, but with a steady decrease in the number of interactions, the more time passed. *Sampling* learners were those who watched video lectures for a short period, either at the beginning of the MOOC or when it was already underway.

The authors in Anderson et al. [3] classified users from three successive offerings of a *Machine Learning* course and from three consecutive offerings of a *Probabilistic Graphical Models* course, for a total of six MOOCs all offered on Coursera.
Their approach included the analysis of two main activities, viewing a lecture and hand in assignments. These two aspects were blended in a so-called *assignment fraction*, which indicated the percentage of assignments interactions over the sum of viewing lectures and assignment activities. The authors analyzed the distribution of this variable and depicted five classes of engagement style: *Solvers*, *Viewers*, *Collectors*, *Bystanders* and *All-rounders*. Besides, the authors verified how the users from each of these classes participated in forum discussions and conducted an experiment involving the award of badges for forum activities. Their results indicated how a system of badges could directly affect the overall user activities.

Coffrin et al. [15] analyzed a set of two MOOCs offered by the University of Melbourne.
The first one, *Principle of Macroeconomics*, was an eight-week course with minimal prerequisites. During each week, users had to watch videos and respond to a set of quizzes, three of which contributing to users' final grade and the remaining five being for practice purposes only. Furthermore, users

had to write an essay on one of the topics covered during the course and review and grade the essays of three other users.
The other MOOC, *Discrete Optimization*, was a graduate level course, which required the registered users to have a strong background in computer science. With a duration of nine weeks, it was organized in a self-paced manner, with the complete assignments and lectures already available from the first week. In this case, seven programming assignments with unlimited attempts would constitute the final grade of the users.
A discussion forum was present for both MOOCs. Principle of Macroeconomics had a completion rate of 4.33%, while the one of Discrete Optimization was of 3.5%. The authors studied the behavior of the users with a linear regression model.
Considering the interactions from the first two weeks only, they individuated three classes of learners; Auditors—users who watch videos in a particular week but do not participate in any assessment; Active—users that participate in an assessment during a particular week; Qualified—users who watch a video or participate in an assessment and obtain marks higher than 60% in both assignments from the first two weeks.
Particularly, the authors showed that completion rates of Qualified users are higher than the one calculated over the whole users. The authors inferred that Qualified users were those learners with prior knowledge on the topics, which helped them to achieve high scores already in the assignments from the initial weeks.
Moreover, the authors focused on weekly assignments and video interaction transactions and represented these by State Transition Diagrams. They concluded that the State Transition Diagrams highlighted certain similarities between the same classes of users among the two considered MOOCs.

In Teusner et al. [63] the authors analyzed three successive re-runs of the MOOC *In-Memory Data Management (IMDM)*, developed by the Hasso Plattner Institute of Potsdam and available on the onpenHPli platform[1].
These re-runs were offered between 2012 and 2014, held in English and targeted academics and learners with a business background. Each re-run spanned over six weeks and included a final exam for a positive grade. More than 13,000 users enrolled in the first two interactions of the course, whose contents barely differed. The third re-run attracted more than 9,000

[1]https://openpli.org/

participants. Besides, the content of this third offering was updated and enhanced according to the feedback received from the users of the initial two runs of the MOOC. Notably, around 60% of the video underwent some editing, while part of the learning units was reshaped to ease and increase their readability.
The authors verified that using a stable material for successive re-runs required less effort for content creators but was enough to attract a wider audience. The authors also suggested that the sooner users' feedback is addressed and introduced to model and design the course, the higher their effects on user engagement are during the rest of the course.

A review of the available literature in MOOCs is presented in Hew and Cheung [27]. The authors analyzed more than twenty different research works on MOOCs, characterized by a variable number of participants and offered with different contexts and systems.
The authors also presented a summary of the perspectives of users and instructors. Motivations for users to enroll in a MOOC were mainly out of curiosity about such environment, desire to extend their knowledge and personal challenge. The authors denoted unclear course materials, lack of support and incentive and other personal obligations or priorities as the most common reasons for users to drop out.
The challenges of teaching MOOCs for the instructors included lack of user participation in the online forum and users' feedback. The quality of education provided by MOOCs and the accuracy of assessments of user learning emerged as two of the main pressing issues that still had to be resolved.

2.2 Prediction of user participation

In order to mitigate the number of dropouts of a course, it is necessary to distinguish those users more at risk of failing. The sooner it is possible to identify this set of users, the more time there is to take countermeasures to prevent them from turning into dropouts. Mostly, these classification tasks rely on the analysis of logs of user interactions by data mining techniques. These are an emerging alternative to traditional methods of analyses

such as surveys, questionnaires, and interviews, which are increasingly time-consuming the larger the surveyed population is and are, therefore, impractical for most scenarios.

Authors in Jiang et al. [33] analyzed a four-week MOOC offered on Coursera by the University of California (UCI) called *The Preparation for Introductory Biology*. The course was thought for incoming first-year students enrolling in the Biological Science Major and consisted of three units. Each of these included short videos, up to four multiple-choice quizzes as well as peer assessment modules.
The authors analyzed users' social interaction and assignments performances to train a logistic regression model. Namely, the set of features included the average quiz score, the number of completed peer assessments, the social network degree of users and whether users were undeclared majors, meaning users had to attend an entire year of biology and chemistry classes before being eligible to enroll in the major.
The authors reported that first-week assignment scores were a reliable indicator of users' performance. Furthermore, getting a Distinction certificate was positively correlated with the level of social interaction of the users.

Balakrishnan and Coetzee [4] tried to predict the likelihood users would drop out in the following week using Hidden Markov Models and an ensemble stacking approach similar to the one described in Xing et al. [72]. Furthermore, they aimed to individuate patterns in student behaviors.
Their dataset consisted of a single MOOC offered by Berkeley University on the edX platform, called *Software as a Service*. This course lasted six weeks and consisted of eleven lectures, each including several videos, programming homework, and graded multiple-choice quizzes. The course also included a basic discussion forum.
The authors used features such as the percentage of total available videos watched, the number of threads viewed, the number of posts made on the forum, and the number of times the Progress Page was visited.
Their results offered insight into users' interaction style and how this related to the probability of dropping out of the MOOC. For example, the authors found that users who frequently checked the Progress Page and watched more than 50% of the available videos were less likely to abandon the course. This information could be used by instructors to suggest changes in the interaction behaviors of those users who are at risk of failing.

Boyer and Veeramachaneni [8] experimented with the prediction of users that will drop out in a real-time scenario. For their experiments, they analyzed a set of three different offerings of the same MOOC (*Circuits and Electronics*) offered by MIT on edX.
By employing a variable amount of interactions, the authors tried to predict users who would drop out. Their model was based on a rolling window concept, whose size represented the amount of information (in their settings, the number of weeks) from the past to include when making a new prediction.
The results of this work suggested that a small window size could yield, in some cases, outcomes comparable to those obtained with a window size that included the whole available history.

In Guruler et al. [24] the authors attempted to classify users with the help of a Microsoft Decision Tree (MDT). This classifier was used to obtain a ranking of the features, thus gaining more insights about their relations and relative importance.
The authors used a set of twenty-four features, which included dimensions related to the family's living conditions and financial status, high school information and university's entrance exam placement.
Their results indicated some degree of correlation between certain features, such as the family's income and the type of academic registration.

Similar studies undertaken by colleagues and I [68] have depicted dropout prediction over a set of five MOOCs. Each MOOC lasted eight weeks and was offered by Universidad Galileo in Guatemala on an own e-learning portal.
Initially, we attempted to classify users on each MOOC singularly and, afterward, employed a unique dataset as a combination of the five courses' logs to detect dropouts. We conducted the experiments using k-means and Support Vector Machines (SVM) classifiers.
Our set of features included session's information (e.g., the total number of requests per session, average sessions' length, average number of requests per day) and the amount of time of interaction with each available tool of the course. We also experimented with the same features but computed on the first half of the MOOCs only (four initial weeks) and conducted the experiments using different combinations of features alternatively.
In our results, SVM always outperformed k-means, and we noticed im-

provement in the prediction scores when using particular combinations of features.

Successively [66], we analyzed an extended dataset of 11 MOOCs offered from Universidad Galileo in Guatemala. In this work, we attempted to detect dropouts on a weekly basis using cumulative features. The features we used included general information, such as the number of active days and the average number of interactions per day, the total session length in second. Furthermore, we also extracted features by counting the number of times users interacted with each of the available tools the system offered. We used SVM to run an experiment for each week of duration of the MOOCs and evaluated the results using per-class and average Precision, Retention and F1 Score.
We obtained results in line with those from our initial work, with an averaged F1 Score higher than 0.85 for more than the six weeks considered. Furthermore, we tried to classify users according to the Attrition Model for Open Learning Environment Setting (AMOES) proposed by Guetl et al. [22]. Specifically, we used SVM and attempted to classify users into Completers, Healthy Attrition, and Unhealthy Attrition. However, we obtained an F1 Score lower than 0.7 for both Healthy Attrition and Unhealthy Attrition class, indicating some flaws and possible improvement in the proposed model and approach.

In our most recent work [67], we analyzed MOOCs from two different systems. Specifically, we used a set of MOOCs of Universidad Galileo offered on their own portal Telescopio and a set of MOOCs of Curtin University available on the edX website. The two systems differed concerning intended audience, the number of enrolled users and the modalities these MOOCs were offered. Specifically, MOOCs of Universidad Galileo were designed in a synchronous mode, while those from Curtin University were proposed in a self-paced mode.
We attempted to detect dropouts and analyzed the features of our model in a multisystem setting. We identified a set of features that always were the ones with the highest scores and obtained accuracy greater than 0.7 already when using the initial 20 interactions of each user. Furthermore, the accuracy of the classifier was always higher than 0.7 when we analyzed the initial days following the user's first interaction.

3 Datasets and Experimental Setup

As a first step, we introduce the considered MOOCs. Specifically, we describe and characterize the MOOCs from Universidad Galileo and the ones from Curtin University. Following the steps of the Knowledge Discovery Process [1], we preprocess the original data and derive the set of features from it. Further, we motivate and introduce the per-system and multisystem experiments. Finally, we present and describe the two classifiers that we use for our experiments: Support Vector Machine (SVM) and Boosted Decision Trees.

3.1 Datasets

3.1.1 Universidad Galileo

Universidad Galileo is a university located in Guatemala City in South America. Before joining edX in May 2016, the university offered online courses on their own e-learning portal Telescopio[1] (built upon the .LRN open source software[2]). The MOOCs we consider were offered on this platform between 2013 and 2014.
The dataset from Universidad Galileo includes logs from 11 different MOOCs, each differing in topic, audience and learning goals. These MOOCs are briefly described in Table 3.1. All these MOOCs have a fixed schedule;

[1] http://telescopio.galileo.edu/
[2] http://www.dotlrn.org/

throughout the duration of the MOOCs, users have to pass various assessments in the form of weekly activities and quizzes. Moreover, a final exam concludes each course. Eventually, some MOOCs (the more technical ones) also require users to hand in a course project (programming assignment). The final grade is the average of the weekly assignment scores, the project score, and the user's final exam score. Weekly quizzes can be employed by the users as self-assessment. These, although not mandatory in order to get a positive final grade, are highly suggested. Each of the MOOCs also implements a discussion forum, where users can ask questions, clarifications and discuss the topics considered during the course. All assignments and final exams for each MOOC have submission deadlines that have to be respected by the users. Although users can still organize their learning schedule autonomously, these settings impose a certain pace.

In order to collect feedback after the end of each MOOC, a survey is sent out by email to all users who do not manage to complete the course with a positive grade. The survey is intended to help system administrators and instructors of MOOCs to gather information about user opinions on the course. Particularly, the questions cover preferential aspects such as user

Table 3.1: Universidad Galileo MOOCs description. We consider a total of eleven different MOOCs. Three MOOCs are re-runs of the same course and are listed only once under *Cloud Based Learning (CBL)*. The details for these re-runs are reported in 3.2. The first column lists the name for each MOOC (translated from Spanish), together with its abbreviation between brackets. In the second column, we give a short description of the topic the courses focus on. The range of topics is wide and goes from technical ones, as in the case of *Android (AND)*, to more general and practical ones, as for example *Medical Emergencies (ME)*. The third column indicates the particular audience the MOOCs were thought for and offered to. Most of the MOOCs were not designed for students but targeted at Professionals and Teachers.

MOOC	Content	Target
Android (AND)	Introduction to mobile apps for Android development	Students
Authoring tools for E-Learning (AEL)	Tools for design of interactive virtual courses	Professionals, Teachers
Client Attention (CA)	Professional customer service	Professionals
Cloud Based Learning (CBL)	Cloud computing principles applied to education	Professionals, Teachers
Community Manager (CM)	Professional management skills and promotion on the web	Professionals
Digital Interactive TV (DITV)	Theory of interactive applications for digital TV	Professionals
Introduction to E-Learning (EL)	Content developing for online fruition	Professionals, Teachers
Medical Emergencies (ME)	First aid basics	No specific target
User Experience (UE)	Design and implementation of UI	Students

satisfaction with the MOOCs, perception of the course's offer and reasons for users to enroll and drop out. Additionally, these answers are used to create a classification model to investigate and describe reasons for attrition [22, 25].
However, only part of the users who received the survey returned it filled out. The percentage of the users that send back the completed survey varies between 40% (*Cloud Based Learning (CBL)*) to 81% (*Medical Emergencies (ME)*). For the purposes of this work, we consider two classes of users: *Completers* and *Dropouts*.
It is necessary to note, that due to the structure of the logs, information about those users who abandoned the course and did not click on the survey link are not available, and, therefore, we do not consider these users. Likewise, the log files only include interactions posterior to the MOOCs' enrollment phase. Therefore, users who enrolled in a MOOC but never

Table 3.2: Universidad Galileo MOOCs characteristics. After the MOOCs name, the *Enrollments* column represents the total number of users registered to the MOOC and the *Users* one accounts for those who had at least one interaction available in the logs. *Completers* and *Dropouts* columns represent respectively the number of users who successfully completed the course and those who did not succeed to do so. The column *Dropout Rate* reports the relative rate computed in relation to the *Users* column and the absolute rate in relation to the *Enrollments* column between brackets. Even considering only the active users, the dropout rates (except for three MOOCs) are always higher than 50%, reaching up to 87% in one case. When the whole number of enrolled users is considered, these rates increase dramatically and are always higher than 90%, in line with the values from the literature. The last three columns report the average interactions per user and for each class. From these columns, we see that for all the considered MOOCs the *Completers* on average, always engage more than the *Dropouts*.

						Average Interactions		
MOOC Title	**Enrollments**	**Users**	**Completers**	**Dropouts**	**Dropout Rate**	**Global**	**Completers**	**Dropouts**
Android (AND)	8852	583	77	506	87% (99%)	433	1597	260
Authoring tools for E-Learning (AEL)	1679	255	101	154	60% (94%)	722	1401	279
Client Attention (CA)	2915	89	60	29	33% (98%)	394	510	154
Cloud Based Learning (CBL)	2055	274	121	153	56% (94%)	2353	4423	747
Community Manager (CM)	9145	811	320	491	60% (97%)	850	1760	268
Digital Interactive TV (DITV)	847	117	63	54	46% (93%)	999	1582	319
Introduction to E-learning (EL)	2141	239	81	158	66% (96%)	1623	3804	545
Medical Emergencies (ME)	2112	118	49	69	59% (98%)	1671	3172	606
User Experience (UE)	2150	182	62	120	66% (97%)	499	1137	170
Web Tools and Educational Applications (WTEA)	1350	176	99	77	44% (93%)	265	369	131
Web Tools in the Classroom (WTC)	2045	313	131	182	58% (94%)	1044	2078	299
Total	35291	3157	1164	1993	64% (97%)	987	1985	343

interacted with it, do not appear in any of these logs and are not considered in this work.
For a more comprehensive overview, Table 3.2 summarizes the different types of users within each of the MOOCs and at a global system level. The entries *Web tools and educational applications (WTEA)* and *Web tools in the classroom (WTC)* represent a subsequent offering (*re-run*) of the MOOC *Cloud Based Learning (CBL)*. Despite the different names, their contents and audiences do not differ, and, therefore, we list these three MOOCs only once in Table 3.1 as *Cloud Based Learning (CBL)*.
The structure of the log files is the same among all different MOOCs. In each file, each row describes a single interaction of a user with the MOOC and consists of the following five fields:

- **User Id**: a system internal identifier to uniquely identify a user within the system (int)
- **Origin URL**: the URL from which the request originated (string)
- **Request URL**: the URL that was requested (string)
- **Timestamp**: the date and time at which the request happened (string in the form DD-MM-YY HH:MM:SS)
- **Tool**: indicates the type of the requested resource (string)

While the first four fields do not require any particular explanation, some clarification is needed for the **Tool** field. This field is a mapping from the **Request URL** to the type of resource that was requested. The list of all available **Tools** includes the following:

- **Assessment**: it is used for self-assessment mostly. It can be of two types: online quizzes and surveys (the latter being the most used one);
- **Assignment**: link to the assignments page. This page contains a list of assignments with their descriptions. Assignments can be of various types, including tasks, projects, participation, etc.;
- **Calendar**: a page that contains the calendar of the MOOCs;
- **Course Board**: a page that includes a per-week description of the discussed topics;
- **Course Members**: a page that includes the list of tutors, instructors, and experts that are involved in the MOOC;
- **Diploma**: a page that contains the requirements to obtain the MOOC's certificate;

- **Eduwiki**: educational wiki for .LRN;
- **Evaluation**: a tool used to upload, download or view the list of tasks of the MOOC. Furthermore, it also displays information about users' total and partial grades;
- **FAQs**: a page with the list of Frequently Asked Questions;
- **File storage**: a page including all the files, documents, and resources available for the course;
- **Forum**: a link to the discussion forum of the MOOC;
- **Hangouts**: a page that contains the summary of the hangouts done during the course;
- **Learning content**: access to the content uploaded by the instructors. The contents can include different resources such as videos, audio, mind maps, images, etc.;
- **Members**: a page that lists all enrolled users of the MOOC;
- **News**: a page that contains all the course's news;
- **One-Community & Page-Num**: an internal page of the MOOC;
- **Peer evaluation**: used to make student peer review;
- **Requirements**: a page containing the technical requirements (if any) to enroll in a MOOC;
- **Twitter**: a page that lists all the tweets related to the course;
- **Wiki**: link to the MOOCs wiki;

The Tool **Forum** only indicates a user clicking on the link to the forum. Interactions that happened within the discussion forum are saved in a separate log file, which consists of the following fields:

- **Forum id**: a unique identifier for each interaction (int);
- **Creation date**: the date and time at which the interaction happened (string in the form DD-MM-YY HH:MM:SS);
- **Type**: one of *answer*, *comment* or *question*. *Question* represents the creation of a new thread, while *comment* and *answer* indicate replies to an already existing thread (string);
- **Tags**: tags provided by the user (string);
- **Parent id**: if null it indicates a new topic (thus the type will be *question*), otherwise it indicates the *Forum id* to which it refers (int or null);
- **Character count**: count of the characters of the post, HTML formatting included (int);

- **Title**: the title given by the user to the thread, null for *comment* and *answer* (string or null);
- **Body**: the content of the posted message (string);
- **User id**: unique identifier of the user the interaction refers to (int);
- **Staff**: either *t* for teacher and admin posts, or *f* for students (string)

As a first step, we preprocess all the log files. We discard rows with meaningless or missing fields. Meaningless fields include, for example, interactions with wrongly encoded **Tool**, **Timestamp** or **User Id**. Missing fields are mostly **User id** and more rarely **Timestamp** and **Tool**. As it is not possible to recover a missing or wrongly encoded **User Id** or **Timestamp** we simply have to drop the corresponding entry in the log file. However, these interactions represent a small percentage of the total interactions and their removal has no strong implications for our analysis.
Second, we remove rows corresponding to certain interactions such as, for example, loading of images or CSS files. These interactions are of no interest as they are not actions of users but rather browser requests for resources to the server. Moreover, we also skip **User Id** appearing in both Completers and Dropouts log files. In the forums, we remove every interaction from staff members and keep only interactions from users, whose **User Id** also appears in the main log file.
After these basic cleaning steps, we blend all these files into a single one obtaining a file for each MOOC. Each of these files consists of the fields **Timestamp**, **User Id**, and **Tool**. For interactions within the forum logs, we set the **Tool** as one of *answer*, *comment* or *question*, while for interactions from the other files we keep the corresponding **Tool**. Furthermore, we add a field **Class** to label users as either Completers or Dropouts. As a general setting, we identify Completers by a **Class** value of one and Dropouts by zero.

3.1.2 Curtin University

Curtin University is located in the city of Perth (Western Australia). It joined and started to create online courses on the edX platform in 2015[3]. Initially,

[3]The discussion refers to the at-the-time-of-analysis version of edX. Since edX is continuously being developed and enchanted, some concepts might slightly differ in the

Curtin University made a set of two MOOCs available and extended it with two more MOOCs during 2016. Moreover, Curtin University also makes use of re-runs for some courses, which are offered several times a year.
The analyzed dataset consists of the first offering of a MOOC that, for convenience, will be called *MOOCC1* and its re-run, identified as *MOOCC2)*[4]. As with Universidad Galileo, Table 3.3 provides an overview of the considered MOOCs. In contrast to the logs from Universidad Galileo, the ones from Curtin University also include enrollment interactions. Therefore, for this system, the **Users** column coincides with the **Enrollments** one, which is omitted. Similarly, we also omit the relative dropout rate. We give a detailed overview of the structure of the logs in Section 3.1.2.
Each of these MOOCs is characterized by an initial time window during which the course is available for enrollment only (*enrollment phase*), and a second time window lasting from the MOOC's official start date until its official end date (*running phase*). All the considered MOOCs from Curtin University are *self-paced*; on the start date, the course content and materials are uploaded and become available to all enrolled users who can engage with the system at their own pace. Enrollments are possible anytime and mainly two modes are supported: *audit* and *verified*. Enrollment in audit mode is free of charge, but no certificate is issued after successful completion

Table 3.3: Curtin University MOOCs characteristics. After the MOOCs name, the *Users* column represents the total number of users registered to the MOOC. *Completers* and *Dropouts* columns represent the number of users who completed the course and those who did not succeed to do so respectively. The column *Dropout Rate* represents the percentage of registered users who did not complete the MOOC. For both MOOCs, we have rates higher than 90%. The last three columns report the average interactions per user globally and for each class. We can see that on average the *Completers* engage more than the *Dropouts*.

					Average Interactions		
MOOC Title	**Users**	**Completers**	**Dropouts**	**Dropout Rate**	**Global**	**Completers**	**Dropouts**
MOOCC1	21948	1500	20448	93%	93	683	49
MOOCC2	10368	208	10160	98%	58	760	44
Total	32316	1708	30608	95%	76	722	47

future. The latest available documentation can be found at http://docs.edx.org/

[4]The real names of the MOOCs have been removed as requested by Curtin University.

of the MOOC. However, audit users can switch to verified mode, which includes a certificate of completion of the course, anytime upon payment of a certain amount of money.
After the official end of the course, a MOOC can either remain closed for enrollment or enter in *Archive Mode*. This later modality, still allows users to enroll, browse through course materials, join forum discussions and any other activities usually available within the course. However, interactions happening after the official end date, will in no case allow a user to get a certificate. Therefore, users can request a certificate only if they manage to obtain a positive grade at any time before the official end of the course.
A user eligible for a certificate can request it by clicking on a dedicated *Obtain the certificate* button, which becomes available once the MOOC has been completed. Since there is no time limit to request a certificate, users may request certificates with a particular delay (even months). In this case, the interactions relative to the action of asking for a certificate has a delayed timestamp, potentially even later than the MOOC's end date. Due to this reason, we filter out the following interactions:

- the only considered enrollment interactions are those taking place before the official end of the course. Therefore, we drop enrollments of users when the course is in *Archive mode*;
- we consider certificates regardless of their timestamps. In fact, these always indicate users who completed the course before its end, regardless of the timestamp they exhibit;
- we consider all other interactions only if they take place before the official end of the course

The structure used by edX to manage the course structure and progress includes several files and SQL tables. Therefore, the next section offers some detailed explanation about the logs and the way we processed them.

edX logs

Each partner university of edX can, at anytime, request and download the logs of their MOOCs. These logs consist of an Event folder, a set of SQL dumps of the database and JSON files. The Event folder includes all

interactions coming from any of the MOOCs from the university, organized as daily files. Due to this setting, every file in the Event folder is generated once and never updated with interactions after its date.
On the other hand, the SQL dumps consist of separate tables for each of the MOOCs offered by the university. These SQL tables contain the latest status of users' progress, and, therefore, are constantly updated to reflect any change. For example, when a user gets a certificate the relative table is updated consequently. The *courseware* JSON file, which is used to save users' progress within the course, represents another example. When a user begins to solve a problem of a particular course's section but does not complete it, the status of the problem for this user is stored in this file. In the future, when and if the user will progress within the same problem, the corresponding entry in the table is updated consequently to map the current status.
The available set of instruments and resources that edX offers to shape and manage a course is referred to as *events* and includes twenty-one different tools. Each event comprises a set of subevents, which describe the particular interactions that take place. In the case of Curtin University's MOOCs, the set of used tools includes the following:

- **Enrollment**: includes course enrollment and unenrollment, as well as upgrade from audit to verified;
- **Course Navigation**: events happening on the main page of the course, such as clicks on links and tabs or selections of portion of text;
- **Video Interactions**: video related events, such as show or hide captions, pause of a video, change of a video speed, etc.;
- **Problems**: interactions related to solving and correcting course assessment. It includes events such as verification of the given answer, grading of the answer, display of hints and so on;
- **Bookmark**: users can create bookmarks to a particular course unit for an easier access at a later time. It includes add, view or delete of bookmarks events;
- **Discussion Forum**: typical forum interactions such as adding a comment, creating a new thread or searching within the forum discussions and posts;
- **Poll and Survey**: voting and viewing of the results of polls and surveys events;

- **Certificate**: includes certificate creation, sharing of the certificate on a social media website and clicks on the shared certificate link (which points back to the main page of the course on the edX website)

Besides the usual data cleaning tasks (removal of empty or malformed inputs) and the previously mentioned constraints, we need some extra preprocessing steps to prepare logs of Curtin University.

edX and Curtin's staff interactions. Event logs contain all type of interactions that take place in the MOOCs, regardless if those originated from users, staff members (referred to as *Course Team Members* in the documentation) or members of edX. Since the focus of the experiments is on the users, we filtered out the interactions from Course Team Members.
Removal of edX staff is easily carried out, as these have an *admin* flag set in the table listing all users of a MOOC. However, this is not the case for the remaining staff members (i.e., the professors and instructors that manage the MOOCs and the discussion forum). To correctly remove their set of interactions, we have to take particular attention regarding Enrollment events. Registration of users happens in two different modalities: either users register themselves by clicking on the *Enroll* button on the course main page or a Course Team Member can enroll one or more users at the same time, a procedure called *batch enrollment*. Registrations done through the latter have the Course Team Member as the initiator that performs the batch enrollment. We kept these interactions but substitute the original id of the initiator with the ones of the users that were enrolled. This way, we do not discard any user. Despite this situation, the remaining interactions from staff members do not present any particular problem and, therefore, we simply drop them.

Invalid username and user id. Some interactions in the logs can have a null or empty username or user id. This happens, for example, when the system logs out users because their session is expired. As it is not possible to associate these to the initiator users, we removed these entries from the logs.

Unmapped user id. After enrollment, an obfuscated id is generated and assigned to the newly enrolled user, and the mapping username-obfuscated

id is saved into an SQL table[5]. However, interactions from authenticated and not authenticated users who just browse the course main page without enrolling in the course, are also saved in the logs without obfuscated id. As these users are neither Completers nor Dropouts, we do not consider these interactions for our experiments.

Unmapped interactions. Besides the documented set of possible events, the logs might sometimes include some undocumented interactions. Such undocumented interactions could be due to, for example, updates to the edX platform, which suppressed specific events while the MOOC was still on-going. Since edX does not support such events any longer and no further information is available about them, we simply remove these interactions.

Duplicated interactions from different sources. Each of the available events is described by a precise set of fields. The set of fields varies from event to event but several common fields are always present. For example, the *event source* can be one of *server*, *browser* or *mobile*.
Interactions with *browser* and *mobile* as *event source* indicate actions of users that take place in a browser or in the edX mobile application respectively and are referred to as *Explicit Actions*. On the other hand, interactions with *server* as *event source* represent server responses to users' actions and are referred to as *Implicit Actions*.
Overall, some events always trigger *Implicit Actions*, while others can trigger a combination of *Implicit* and *Explicit Actions*. Particularly, it can happen that in response to an *Explicit Action* performed by the users (e.g. a click), the system triggers one or more *Implicit Actions* that are also saved in the logs. These *Implicit Actions* are not user interactions, rather they reflect server side actions.
For practical reasons, for events that only trigger either *Explicit* or *Implicit Actions*, we do not discard anything. Nonetheless, for events that can trigger both kind of actions we discard all *Implicit Actions* and keep the *Explicit Actions* only.

[5]Obfuscate ids are used by edX to provide anonymous logs for research purposes. However, Curtin University agreements with edX did not include the delivery of already obfuscated data. We performed the obfuscation of the ids and any other sensitive information following edX guidelines.

After the described preprocessing step, the interactions in the cleaned data set for Curtin University include the fields **User Id**, **Timestamp**, **Event**, and **Class**. A **Class** field with value zero indicates Dropouts, while a one indicates Completers.

3.1.3 Multisystem

We perform two types of experiments in a multisystem scenario.
First, we seek to predict dropouts of one system using information from another system. Second, we want to predict dropouts among different systems using mixed information coming from these same systems. To conduct these experiments, we construct some extra datasets.
Particularly, we create a first dataset by merging all users from the MOOCs of Universidad Galileo. We process this dataset in the same way we process each single MOOC from this system (see Section 3.1.1) and we describe each user using the fields **Timestamp**, **User Id**, **Tool** and **Class** and call this dataset *Galileo*.
Likewise, we merge the users from both MOOCs of Curtin University into a single dataset and process it accordingly (see Section 3.1.2). Interactions of this dataset include the fields **User Id**, **Timestamp**, **Event**, and **Class**. We call this second dataset *Curtin*. We use *Curtin* and *Galileo* datasets to perform system-to-system dropouts prediction.
Finally, we construct a third dataset by merging the datasets *Curtin* and *Galileo* into a single one. Particularly, we rename the **Event** field of *Curtin* before the merge, and we obtain a dataset that includes the fields **User Id**, **Timestamp**, **Tool**, and **Class**. We call this new dataset *MIX*, and we use it for the multisystem dropouts prediction experiments.

3.2 Datasets Processing

After the cleaning of the datasets from both systems, as described in the Chapters 3.1.2 and 3.1.1, we process the data and transform it into the right format for the classification task. We apply these steps to each MOOC of

each system in the same way, and we describe all of them together in the following subsections.

3.2.1 Feature Extraction

As a first step, we translate the interactions in our dataset into a format suitable for the classification task.Specifically, we analyze the interactions and extract a set of features, which are going to be the input for our machine learning algorithms.
Each feature represents an aspect of the dataset that we want to consider when predicting if a user will drop out or not. To construct our set of features, we first sort the interactions in chronological order according to their timestamp. Further, we use this sorted list of interactions to create user sessions. We define a session in relation to an idle time during which a user is not active [7]. Time sorted interactions whose timestamp difference is lower than the specified inactivity window, are part of the same session. Otherwise, if the difference between their timestamps is greater than the inactivity window, we create a new session, and the later interaction becomes the first one of this new session.
We repeat this process for each user and process all their interactions. Throughout this work, when creating user sessions, we always use an inactivity window of thirty minutes, which is a common choice thoughtfully used by many other systems; for example, this is the standard idle time for session also employed by Google [62, 52].
These per-user sessions represent our starting point to calculate the following set of time-based features for each user:

- **Sessions**: total number of sessions;
- **Requests**: total number of requests;
- **Days**: total number of days in which a user interacts at least once with the MOOC;
- **Active Time**: the total amount of time a user is active. We calculate it as the sum of the duration of each session of the user;
- **Session Length**: an averaged measure, which we calculate as **Active time** divided by **Sessions**;

- **Timespan Clicks**: an averaged measure, which we calculate as the sum of the per-session average time between two consecutive clicks divided by **Sessions**;
- **Session Requests**: the average number of requests per session, which we calculate as **Requests** divided by **Sessions**;
- **Active Days Requests**: the average number of requests per day, which we calculate as **Requests** divided by **Days**

Furthermore, we also construct features considering the Tool available for each system. For each Tool, we count how many requests for each user refer to that Tool. It is worth taking into consideration that the set of Tool of the two systems does not intersect. This is not an issue for the experiments on single MOOCs, but it has some consequences for the multisystem experiments as described in Section 3.3.2.

With our experiments, we want to improve our understanding of how the users engage with the MOOCs. Furthermore, we seek to gain insights on the very initial phase of each course, as we want to understand whether initial interactions are indicative of the users' behavior in subsequent parts of a MOOC.
In this regard, we have to be careful and consider the system's differences when deciding on a features extraction strategy. Particularly, all MOOCs from Universidad Galileo have a fixed calendar with hard deadlines for assignments submissions and exams. On the other hand, Curtin University's MOOCs are self-paced, defined only by the start and end date.
Due to these differences, we define four distinct feature extraction approaches: *Initial absolute interactions per user (**Absolute**)*, *First seven days after users first interaction (**Days**)*, *Percentage of interactions per user (**Percentage**)* and *Percentage of active time per user (**Scaled Time**)*. For practicality, we denote and refer to each setting using their abbreviations.

Absolute. First, we investigate whether the initial interactions of the users already indicate whether they will drop out or not in the future. In this setting, we construct the features considering only an initial absolute number of interactions for each user. Particularly, we consider 1 up to 100 absolute initial interactions per user, with an increment of 10. Therefore, in this approach, we conduct a total of ten experiments for each value of the considered absolute interactions.

Days. Second, we explore the engagement of the users during the first 7 days, starting with their first interaction, ignoring the official start of each MOOC. For this setting, we create features ranging from 1 up to 7 days after users' first interaction, with a 1-day increasing step. Therefore, w means we conduct a total of seven experiments with this approach.

Percentage. As a third approach, we analyze how increasing the percentage of considered user interactions, affects the accuracy of the classification. To do so, we compute the features from 1% up to 100% of total interactions per user with an increment of 10%. With this approach, we run a total of ten experiments.

Scaled Time. As a final setting, we inspect if considering a longer amount of time influences the accuracy of the prediction. Therefore, we use an increasing percentage of active time per user. This active time is defined relatively for each user, without considering the MOOCs' official start and end date. Precisely, the start date is given by the timestamp of the first interaction of users, while the end date corresponds to the timestamp of their last interaction. That way, the focus is on the users real active time rather than on the actual duration of the course. Again, we range from 1% to 100% of the per-user active time when extracting the features. We set a 10% increasing step and conduct a total of ten experiments with this setting.

We extract features in a similar way for both systems. In fact, we create features using only three information: the **Timestamp** associated with each request, the type of action performed by the user (the field **Tool** for the dataset of Universidad Galileo and the **Event** field for the dataset of Curtin University) and the **User Id** of the user. As this information is usually available for every log, we believe that our set of features can be easily extracted also from other systems.

3.2.2 Feature Scaling

A common pre-processing step is to scale and standardize the features [58]. This procedure is particularly necessary for certain learning algorithms such as K-Means and Support Vector Machines, which are not invariant under affine transformations.

For example, in the case of features with variances significative larger than the others, these algorithms might be unable to correctly learn in the training phase because the features with greater variance dominate over the other features. [12, 18, 21]. In the case of SVM, scaling of the features also prevents numerical problems when calculating kernel functions [31].
Practically, we consider each feature independently and scale it by mean value removal and with a division by its standard deviation. Hence, the distribution of each feature resembles a typical Gaussian distribution with zero mean and unit variance.

3.2.3 Class Balancing

Another difference between the datasets from Curtin University and Universidad Galileo is the number of active users.
The eleven MOOCs from Universidad Galileo include a total of 3,157 active users, while the two MOOCs from Curtin University have up to 32,316 enrolled users, with a ratio between the two systems of 1:10. Besides, MOOCs of Curtin University also have a higher ratio of Completers-Dropouts than MOOCs of Universidad Galileo.
For example, the MOOC **MOOCC2** of Curtin University has a 1:48 ratio of Completers-Dropouts with only 208 Completers over a total of 10,368 users. The ratio for the other MOOC of Curtin University, **MOOCC1**, is similar (1:13). On the other hand, the ratios of the MOOCs of Universidad Galileo range between 1:1.4 for **ME** and 1:6 for **AND**.
The large difference in the number of users of each class needs particular attention as it could create problems such as the *Class Imbalance Problem* (see Guo et al. [23]). As a consequence, we do not use the constructed features directly, but we first balance the class distribution so that the number of users of both classes is equal.
The balancing of the classes is a widely-studied topic, which can be solved using different techniques (see Japkowicz et al. [32] and He and Garcia [26]). The general idea is to introduce some bias and obtain a ratio of 1:1 of the classes. Intuitively this can be achieved with two basic approaches; either by reducing the number of samples in the bigger class or by increasing the number of samples in the smaller class. These two approaches are respec-

tively called *undersampling* and *oversampling*.
Undersampling can be performed by randomly discarding some samples from the bigger class until the size of the classes is equal. This simple approach comes with several drawbacks. First of all, discarding samples is generally not a good approach because the fewer data we use to train a classifier, the less accurate the model that represents the distribution. Furthermore, it could lead to situations in which the downsampled class has more samples in the test set than in the training set (see Section 3.2.4). The probability of these situations to occur is directly proportional to the imbalance of the classes.
Oversampling works the other way around, by adding samples to the smaller class until its dimension equals the one of the bigger class. The advantage oversampling over undersampling is that no sample is discarded. Thus, the constructed model is closer to the original dataset and, therefore, more accurate.
The most intuitive way to oversample a class is by *random oversampling*; we randomly pick and add samples to the smaller class until its number of samples equals the one from the bigger class. Even though MOOCs of Universidad Galileo do not have a class imbalance as high as those of Curtin University, to have the most general approach possible, we apply random oversampling to both systems. A more detailed analysis of the *Class Imbalance Problem* and of other oversampling techniques can be found in Weiss [71], He and Garcia [26] and Chawla et al. [13].

3.2.4 Training

The goal of a classifier can be summarized as a means to accurately classify new unseen examples by learning from already available ones. Therefore, the efficacy of the classifier is determined by how well it copes with the classification of new examples. Practically, from the available data two sets are created: a **test set** and a **training set**.
The training set is used by the classifier as initialization data to train a model. A classifier learns from the training set and then the hold-out data, which is represented by the test set, is used for performance evaluation. Correctly splitting the data into these two sets is an important step.

A first requirement is for the training set to contain at least one example from each class. This is necessary as the number of classes, that a classifier is trained on, implicitly determines the number of classes into which the classifier predicts the examples in the test set.
Moreover, the split has to be done in a way that both sets accurately represent the whole dataset. Clearly, there is a trade-off: we want to have as many examples in the training set as possible to fit the classifier with a high number of data. On the other hand, we also want the training set to be maximized in its dimension to get a proper validation of the effectiveness of the classification.
To reach both goals, we employ Cross Validation (CV). We split the data into K different subsets or folds. For each of these folds, we use K-1 for training and the remaining for testing the classification. With this procedure, we measure K different performances and take their average as the final measure. This approach ensures that each example is used both in the test and in the training set at least once. Thus, we avoid overfitting and obtain a more precise assessment of the machine learning algorithm.
There are different methods to create such a split, with the most common ones being the following:

- **K-Fold**: split into K folds and, in turn, use K-1 folds as training set and one as test set;
- **Random split**: randomly split the examples into two sets of a precise size given as input;
- **Stratified K-Fold**: similar to K-Fold but in this case, the percentage of examples of each class is preserved;
- **Stratified random split**: a combination of Random split and Stratified K-Fold. It creates K folds by randomly picking examples and keeping the percentage of examples of each class. The ratio between the training and the test set is needed as input.

As an example, given a dataset of 100 users, 10 of which are Completers and the remaining 90 are Dropouts (1:9 ratio) and k=10.
A **K-fold** split produces 10 folds each containing 10 examples and without any constraints on the classes (theoretically, one fold could contain only Completers or only Dropouts).
A **Stratified K-Fold** creates 10 folds each characterized by a 1:9 ratio between the classes.

A **Stratified random split** with a train size = 80% creates 10 folds each composed of 18 Dropouts and 2 Completers. Practically, the training set consists of 72 Dropouts and 8 Completers and the test set includes 18 Dropouts and 2 Completers, with both sets randomly picked.

For the experiments presented in this work, we always perform the split into training and test set using a **Stratified Shuffle Split** approach. We set the number of re-shuffling and splitting iterations (folds) to 10 and the training size to 0.8. That is, we use 80% of the examples for the training set and the remaining 20% for the test set, for which we predict the labels. Besides, we use Stratified Shuffle Split to preserve the balance of the classes in both sets. It can be argued that such strategy could fail to use all examples in both training and test set alternatively (as, for instance, is always the case with a Stratified K-Fold approach). Although this is theoretically correct, for sizable datasets as in our scenario, a random split is still very likely to produce different folds. However, to further ensure the adequacy of the settings, we repeat the whole cross-validation process for a total of 20 rounds: for each round, we create 10 different folds and construct training and test sets accordingly. The round score is the average of the 10 folds, and the overall score is the average of the 20 rounds.

3.2.5 Evaluation

Evaluating the performance of a classifier is a crucial part toward understanding how well the feature engineering and the classifier work. Evaluation is the proven method to appropriately assess the performance of the algorithm, to identify where the problems (if any) are and represents the starting point for optimization techniques.
Metric functions are the first form of model evaluation. The result of a classification experiment is a set of class labels for each of the examples in the test set. These are called predicted values (classes) of the examples. In contrast, the truth values of the examples represent the correct class they belong to. By comparing the predicted with the true values, we can define the following concepts for each class:

- **true positive (t_p)**: number of examples that belong to the class and are labeled correctly;
- **false negative (f_n)**: number of examples that belong to the class but are labeled incorrectly;
- **true negative (t_n)**: number of examples that do not belong to the class and are labeled correctly;
- **false positive (f_p)**: number of examples that to do not belong to the class and are labeled incorrectly

We can summarize these quantities in the form of a *Confusion Matrix*. As an example, for a dataset of 100 users, 10 of which are Completers and the remaining 90 are Dropouts, as a result of a classification experiment, the Confusion Matrix could look like the one in table 3.4. The Confusion Matrix represents the starting point for the evaluation of the performance of the classifier. In fact, any evaluation metric is simply a different way to summarize the information contained in this matrix.
All these metrics assume values within the range $[0, 1]$, where 0 indicates a classifier that misclassifies all examples, and 1 represents a classifier that correctly predicts all the examples. According to the characteristics of the dataset and on the aspects that we want to investigate, we choose a suitable measure. Since we are dealing with balanced datasets, we use *accuracy* and *baseline*.
As each example is classified either correctly or incorrectly, an intuitive way to evaluate the results is to count the number of correctly and incorrectly classified examples. *Accuracy* is nothing more than the total number of correctly classified examples divided by the total number of examples considered. Although being a rather simple measure, misleading in the case of unbalanced classes, it works well with our settings.
On the other hand, the *baseline* can be understood as the accuracy of a

Table 3.4: Confusion Matrix. This matrix summarizes the result of the classification experiment. In this example, the dataset consists of 100 users of which 10 are Completers and 90 are Dropouts.

		Predicted Class	
		Completers	Dropouts
Right Class	Completers	7	3
	Dropouts	5	78

dummy classifier, which is a classifier that labels examples by randomly picking one or another class. Therefore, the baseline represents the probability of randomly picking the right class; in the case of balanced classes, the baseline will be 0.5 when the classes are two, 0.33 for a scenario with three classes and so on. For this reason, a good classifier is one that has an accuracy higher than the baseline. If that is not the case, the classifier is no better than randomly predicting examples.
We express all the results and plots presented in Chapter 4 in terms of these two measures.

3.3 Experimental Setup

3.3.1 Single System

Generally, we want to be able to correctly detect potential Dropouts at a moment in time close to the start of the MOOC. Such a prediction is a more challenging task compared to using logs that cover the whole MOOC's duration. With a shorter amount of time, we have less information available as input and the model built by our classifier might be less accurate. On the other hand, early identification of at-risk users is a situation close to real-world scenarios.
The correct detection of a set of potential Dropouts is more convenient when we perform it shortly after the course start rather than towards its end. The sooner this information is available, the more time instructors and administrators have to take actions to keep these users on the right track and prevent them from turning into Dropouts.
For these aforementioned reasons, we propose experiments that focus on the early prediction of dropouts. Additionally, we also aim at better understanding our datasets and how the users interact with the system at the beginning of MOOCs. Notably, we classify dropouts of each MOOC from both systems independently, and we compute the features using the four extraction approaches explained in Section 3.2.1. We conduct these classification experiments using Support Vector Machines (SVM).

Support Vector Machines

The basic concepts of Support Vector Machines (SVM) were introduced in Vapnik [64] and Vapnik and Chervonenkis [65]. SVM is effective in high dimensional spaces (that is when the examples are described by a high number of features), can handle not linearly separable features and can perform binary and multiclass classification.
In this thesis, we always refer to the binary case, mainly because the proposed experiments have this setting. However, the introduced concepts also apply to a multiclass scenario. Initially, we present the underlying concepts in the case of linearly separable features. Although this situation is relatively unusual when dealing with data from real-world scenarios, this formulation is necessary to understand how SVM works. We also discuss the case of nonlinear SVM. For convenience, we use plots in a two-dimensional space. However, the dimensionality of these problems is always higher and is decided by the number of features.
For example, concerning a MOOC and the process by which we can detect the class of its users (dropouts or completers), we describe each of the users with features such as "number of post in the forum", "number of attempts to solve a problem" and so on. If the number of features is only two, we can represent those examples in a Cartesian plane where x is the first feature and y is the second one as in Figure 3.1.
SVM is a class of methods whose task is to estimate a functional relationship from a training set of examples described in terms of feature sets. Namely, the formulation for SVM includes the following:

- input: a set of examples (x, y), where $x_1, x_2, ..., x_n$ are the features and $y = f(x_i)$ is a function defined on the set of features. The cardinality of the domain of $f(x)$ defines the number of classes or labels;
- output: set of weights $w_1, w_2, ..., w_n$ for each feature, whose linear combination predicts the label value y

In the example from Figure 3.1, we can see that a line easily separates the classes. The same concept applies in the most general case of a higher number of features and thus, a greater number of dimensions. In this case, the separating line becomes a separating hyperplane. The way SVM classifies examples follows the same idea.

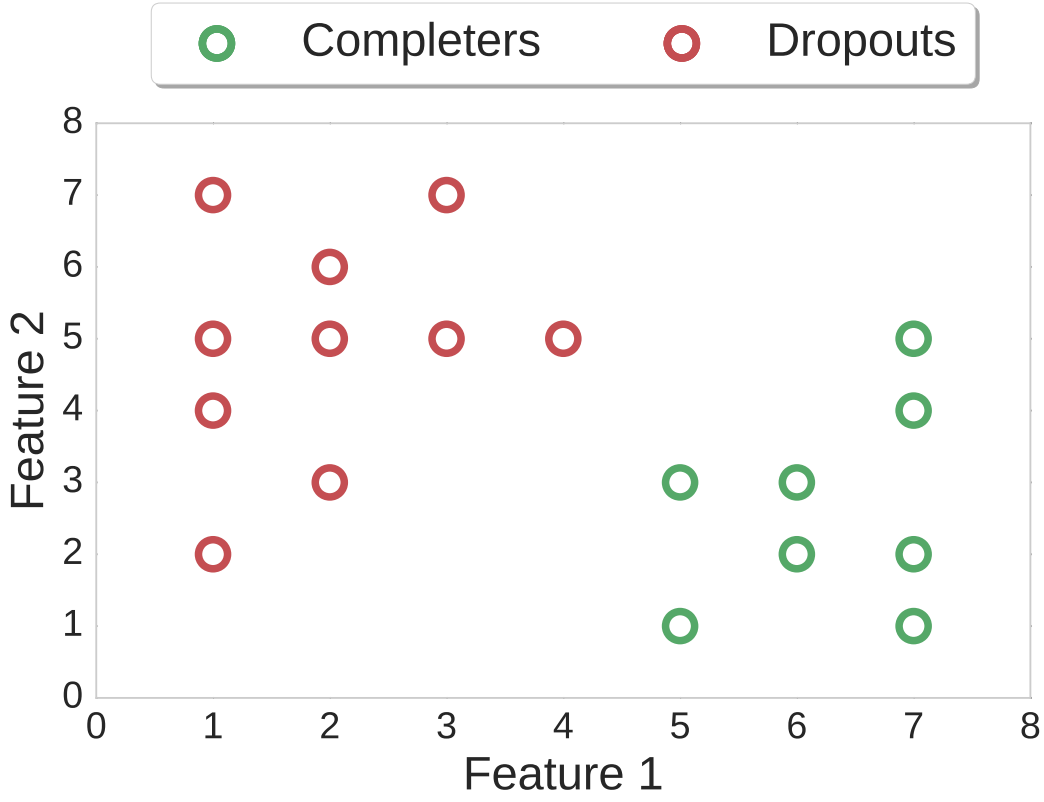


Figure 3.1: Users in a two-dimensional space. Each axis represents one of the considered features. In this case, there are two classes, one plotted in red and the other in green. This example is a simple representation, as each example is normally described by a higher number of features, which machine learning algorithms can easily handle.

Notably, SVM tries to solve the optimization problem of finding a unique separating hyperplane that separates the two classes. In general, for any set of points exist many (infinite) hyperplanes that separate them. SVM finds the optimal one, which maximizes the distance from the nearest point of each of the two classes. This distance (between the two closest points) is called *margin* and the set of points of each class that lies on the margin constitutes the *support vector*. Thus, the support vector is identified by a hyperplane for each class, H_0 and H_1, that can be defined as

$$w^T x + b = 0 \tag{3.1}$$

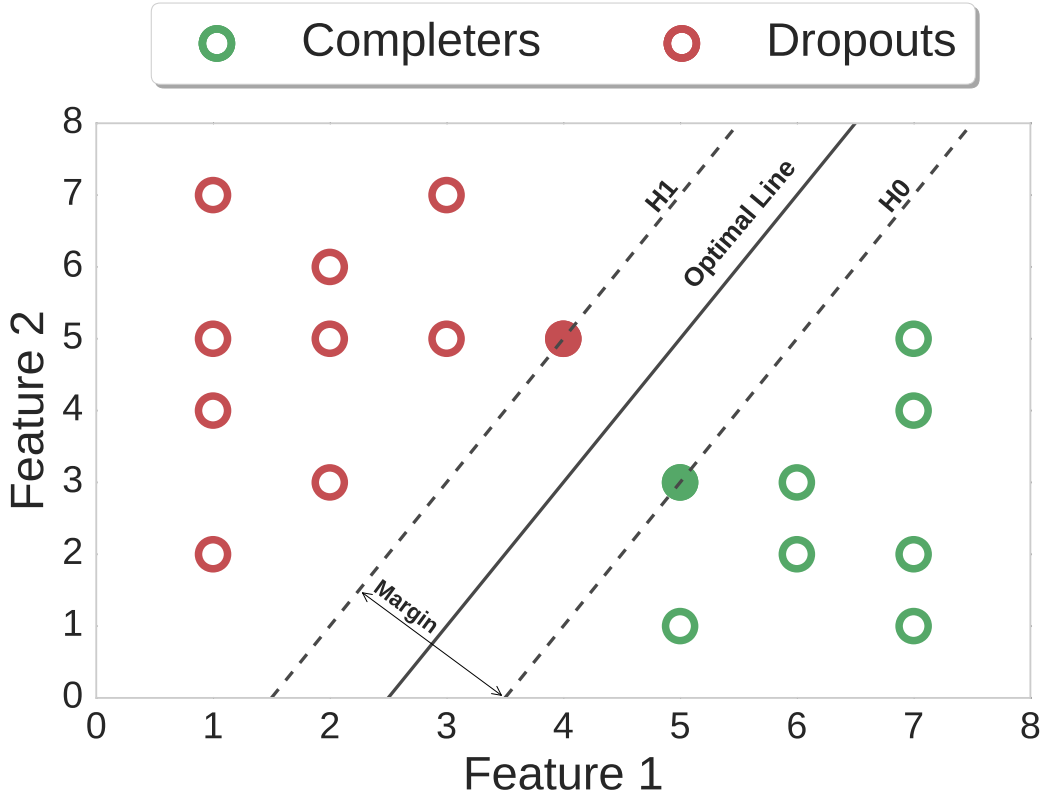


Figure 3.2: SVM optimal separating hyperplane. The optimal separating line is the one that yields the wider margin from the support vectors H_0 and H_1.

Figure 3.2 illustrates these concepts in a 2-dimensional case. The distance between the two hyperplanes H_0 and H_1 is given by $\frac{2}{||w||}$. Thus, to maximize this distance means to minimize $||w||$. The higher the margin between the hyperplanes of the support vector, the clearer the separation of the classes. For each of the classes, all points except those characterizing the support vector have to be on the correct side of the margin. We can write this constraint as

$$\begin{cases} w^T x_k + b \geq +1, & \text{if } y_k = +1 \\ w^T x_k + b \leq -1, & \text{if } y_k = -1 \end{cases} \tag{3.2}$$

which we can rewrite as

$$y_k(w^T x_k + b) \geq 1, \quad k = 1, ..., N \tag{3.3}$$

This s a minimization problem in which the objective function is convex defined and, therefore, the solution is easy to find.
However, in most real case applications, the set of example is not directly linearly separable. For example, this happens when the distribution of the two classes overlap and a linear decision boundary does not correctly separate the examples of the two classes. In Figure 3.3 we show an example of this situation.
In these cases, a linear SVM is bound to misclassify some examples. Intuitively, a simple ellipse will be a perfect decision boundary for the presented example. This can be showed by considering a nonlinear SVM implementation. The idea is to model the decision boundary by introducing slack variables that map the examples to a higher dimensional space in which they are linearly separable.
Considering the decision boundary being the following ellipse

$$2(x-3)^2 + 5(y-2)^2 - 5 = 0 \tag{3.4}$$

$$2x^2 - 12x + 5y^2 - 20y + 33 = 0 \tag{3.5}$$

we can introduce the following variables

$$z_1 = x \tag{3.6}$$

$$z_2 = x^2 \tag{3.7}$$

$$z_3 = y \tag{3.8}$$

$$z_4 = y^2 \tag{3.9}$$

and obtain

$$2z^2 - 12z_1 + 5z_4 - 20z_3 + 33 = 0 \tag{3.10}$$

The last equation is linearly separable in the transformed space. Basically, a transformation ϕ is applied to obtain a linearly separable set of examples

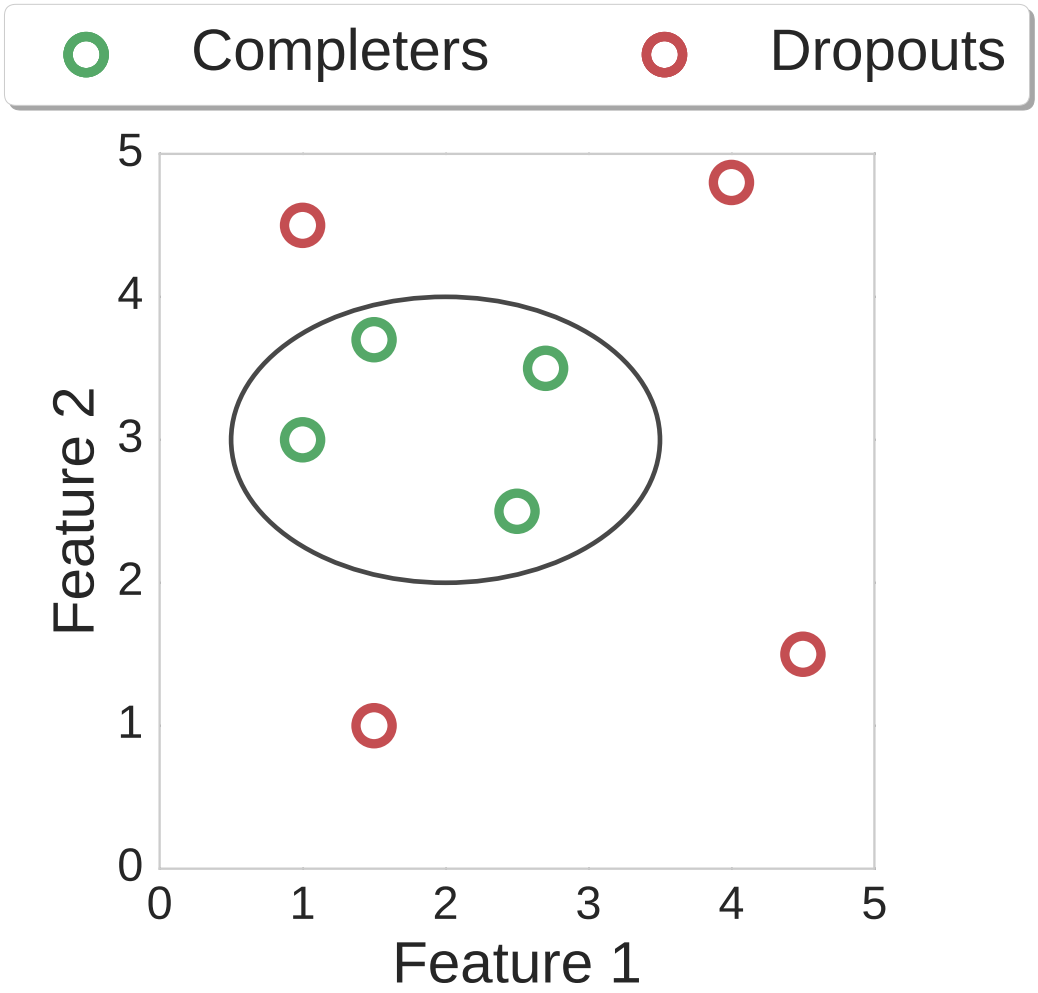


Figure 3.3: SVM nonlinear separable classes. Linear separation is unusual in real case scenarios. When this is not the case, we have to take different approaches in order to separate the set of examples of the classes. In this case, a linear function as a hyperplane from the previous example is not able to correctly separate the examples of the two classes. For this example, a quadratic function as an ellipse correctly separates the examples.

which SVM can easily solve. Nonlinear SVM finds the optimal solution by maximizing the dot product of the transformation

$$\phi(x_i) \cdot \phi(x_j) \tag{3.11}$$

Generally, this transformation is always possible, as we can always approximate the polynomial decision boundary into higher dimensions by the addition of extra variables.
The advantage of this approach also represents its main disadvantage; the higher the polynomial decision boundary, the more variables we must introduce and, consequently, the computational complexity grows.
Moreover, if we have too few examples that are not linearly separable and attempt this approximation, we might incur *Overfitting*; fitting too few examples with an over-complicated function could introduce too many constraints.
Concerning the previous example, imagine that new examples become available and that, the new distribution looks like the one in Figure 3.4 In this case, except for the two obvious outliers, we do not necessarily need a nonlinear SVM to separate the classes correctly. A simple linear SVM can perform the same task with fewer constraints. The nonlinear SVM adds unnecessary dimensionality, and the model becomes too elaborate and too specific. Thus, its predictive performances are poor, and that is why we should avoid overfitting.
An alternative approach in the case of nonlinearly separable examples is the so-called *kernel trick*, which exploits the similarities between data points. A Kernel is a function that given two vectors x_i and x_j defined on R^N, which implicitly computes their inner product in a higher dimensional space R^M without the need of transforming the vectors into the new dimensional space. Therefore, all dot product computations within the optimization problem solved by SVM can be substituted by a kernel function $K(x_i, x_j)$. Thus, there is no need to compute the transformation itself, and there is no need for extra memory or computation time, except for the calculation of the kernel function itself.
This trick allows to quickly approximate the nonlinear decision boundaries on a higher dimension's domain when the features are not linearly separable. Although we can create a custom kernel function to model the similarity between the examples, some standard off-the-shelf solutions already exist as

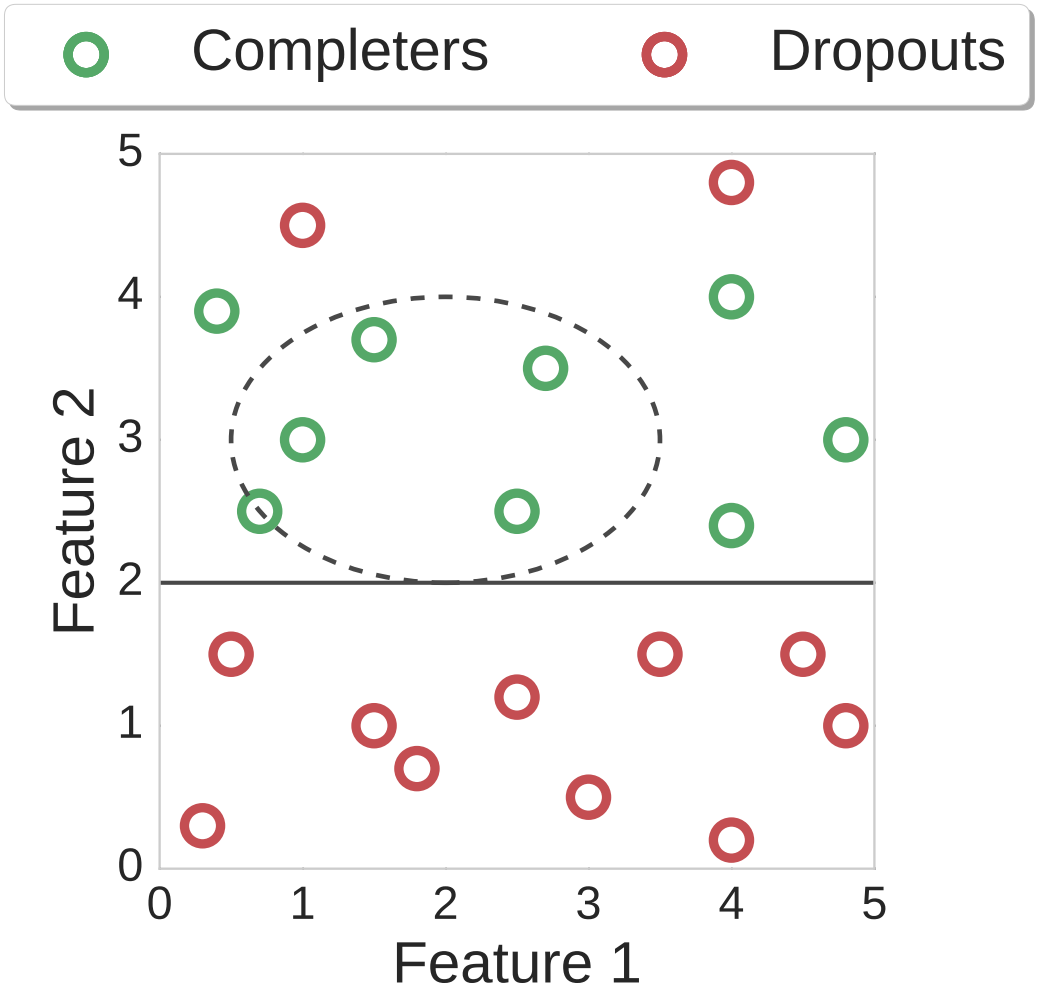


Figure 3.4: Overfitting. In this case, new examples are available, but the quadratic ellipse we used before is not able to correctly separate them. A linear SVM can find a line that correctly separates the examples except for the two outliers.

- polynomial:

$$K(X_i, X_j) = (X_i, X_j + c)^d \tag{3.12}$$

- radial basis function (RBF):

$$K(X_i, X_j) = e^{-\frac{||X_i - X_j||^2}{2\sigma^2}} \tag{3.13}$$

Each of these kernel functions is defined in relation to certain parameters that influence the shape of the decision boundary and, therefore, we have to choose these parameters appropriately.
The polynomial kernel is defined in terms of its degree d, while σ is the regulation parameter for the RBF kernel, and we can think of it as a variance control. With RBF kernel we search the decision boundary using a combination of Gaussian distributions centered on the support vector. Small values of γ will force a Gaussian distribution with a significant variance, meaning that points that belong to the support vector have widespread influence; the class assigned to points that are distant from the support vector will still be largely affected by it.
On the other hand, the higher the value of γ, the more local the influence of the support vector. Thinking of a typical Gaussian bell distribution, lower values of γ produce a more pointed distribution, while higher values result in a broader shape of the bell.
Furthermore, all the kernel functions (as well as the linear SVM) are also influenced by a parameter C that represents a trade-off for misclassifications. Practically, this parameter controls the margin of the separating hyperplane. Large values of C force SVM to choose the smallest margin possible for the separating hyperplane. Thus, it tries to separate as many examples as possible correctly. On the other hand, small values of C relax the classification task, as SVM searches the optimal hyperplane with a larger margin and, therefore, allows for more misclassifications.
All these configuration parameters heavily depend on the problem and the domain. Also, correctly deciding whether to use linear SVM or make use of one of the available kernels is not always easy to determine. Even if we know the right kernel for a particular domain, we still need to set its parameters correctly.
We can tune these parameters employing a Grid Search approach. This method is basically a brute force search on a set of specified classifier parameters and values. Grid Search takes, as input, a dictionary of parameters and a set of their values to be tested and returns their best combination.
The Grid Search approach requires more computation time but allows the estimation of the best settings exhaustively and can be run in parallel to decrease the needed computation time.
Practically, we perform Grid Search on each of the training sets for the following variables and values:

- C: (0.01, 1, 10, 100)
- kernel: (linear, polynomial, RBF)
- degree: (1, 2, 3)
- gamma: (0.01, 0.1, 1, 5)

Then, we can use the best combination to classify examples of the test set and evaluate the performance of the classifier.

3.3.2 Multisystem

Domain-specific analyses and predictions are useful for various reasons. They can give precise information about how a MOOC is offered and perceived by its users.
For example, if during the initial weeks, a high percentage of registered users leave the course or if the system records a low number of interactions, the way the learning content is delivered is probably not optimal. Possible causes might be a poorly developed User Interface (UI) or misleading and hardly understandable course materials.
In the same way, administrators and content creators of MOOCs can get information regarding which tools and modules are most used and which are not considered at all.
However, analyzing MOOCs only in relation to the domain they belong to fails to consider the big picture of E-learning, which includes ranges of various courses for different and peculiar areas. It is true that to run experiments correctly a model has to be created in relation to the underlying domain, but maybe the problem is the domain definition itself.
Results of a system-dependent model hardly have an impact nor do they provide results for different systems. This is undoubtedly a negative aspect when, for example, comparing results from researchers. It can be unproductive to compare results from systems that are characterized by particular constraints and defined on different features. A potential risk is for these results to be of value only for that particular domain or system, bringing only slight improvements to the general understanding of the field of MOOCs that the community has.
Despite the growing number of published work within this area, there are no examples of testing and validating results from a particular domain

across different systems and datasets. Additionally, there is a lack of literature that also covers the final step of the Knowledge Discovery process, the validation phase.
Validation is a crucial task because, either by consolidation or refutation of existing evidence, we can discover aspects that require further analysis and clarifications.
Shifting the focus of experiments to a wider domain becomes particularly meaningful when we want to consider problems shared among various systems: no matter how different MOOCs are, they all struggle due to high dropout rates. Being a common concern, we should approach it more homogeneously.
Due to these reasons, we propose a second set of experiments in which we investigate the two presented systems and their MOOCs together. We think of this as a first attempt towards a homogeneous and multisystem model for the analysis of such systems.

Analyzing and comparing different systems to each other homogeneously requires some preliminary steps. Specifically, we need to qualify the examples from both datasets over the same set of features. Thus, we use the intersection between the features of both datasets for our multisystem classifier.
Moreover, both systems include information about the type of event that an interaction triggers; the field *Tool* for Universidad Galileo and the *event type* field for Curtin University. Although these fields could be used to increase the number of common features of our model, a correct mapping is not straightforward for various reasons.
Specifically, the granularity of the two fields is highly different; each of Curtin University's eight events is described in a detailed way by a set of subevents. The overall number of these sums up to around one hundred and, therefore, mapping each of these events to one of the eight tools available for Universidad Galileo is not a trivial task.
Furthermore, it is not clear that such a mapping even exists. For example, the tools *File storage* or *Peer evaluation* of Universidad Galileo are completely missing within Curtin University's system. Vice versa, the tools *Bookmark* or *Poll and Survey* are not part of Universidad Galileo's MOOCs.
Other tools of Curtin University such as *Video Interactions* represents resources which are equally available for Universidad Galileo. However, the

way interactions are described in the log of Universidad Galileo does not allow us to distinguish interactions that refer to video interactions apart from those that refer to audio or other components.
Due to these limitations, we completely drop the Tool information and focus mostly on the time aspects of the interactions. The common set of features consists of the following features (see Section 3.2.1): **Sessions**, **Requests**, **Days**, **Active Time**, **Session Length**, **Timespan Clicks**, **Session Requests** and **Active Days Requests**. We calculate each of these according to the four metrics previously discussed.

We propose two settings; a system-to-system and a multisystem one.
In the system-to-system setting, we train a classifier with information from Completers and Dropouts from one system and predict Dropouts from the other system. Therefore, we conduct two experiments for this setting. Initially, we use the *Galileo* dataset for training and the *Curtin* one for prediction. Then, we switch the datasets and train our classifier on *Curtin* to predict on *Galileo*. We call these two experiments *Galileo on Curtin* and *Curtin on Galileo* respectively.
For the multisystem setting, we want to train the classifier with a mixture of users from both systems and try to predict Dropouts on both systems. To this end, we employ the *MIX* dataset and refer to this experiment as *MIX*.
As we are interested in identifying which are the most indicative features for both domains, we use Boosted Decision Trees as a classifier. Boosted Decision Trees can be used for prediction experiments, and they can return a ranking of the features according to their importance for the classification task. In this way, we aim to understand more about every feature's importance and their mutual combinations.

Boosted Decision Trees

Boosted Decision Trees are classifiers which can be used for both, regression and prediction tasks. They build upon the same concept of a single Decision Tree classifier and further improves the performance through Boosting technique.
Basically, a **Decision Tree** is a flowchart used to illustrate the possible outcomes of an event. It comprises nodes which are connected by edges.

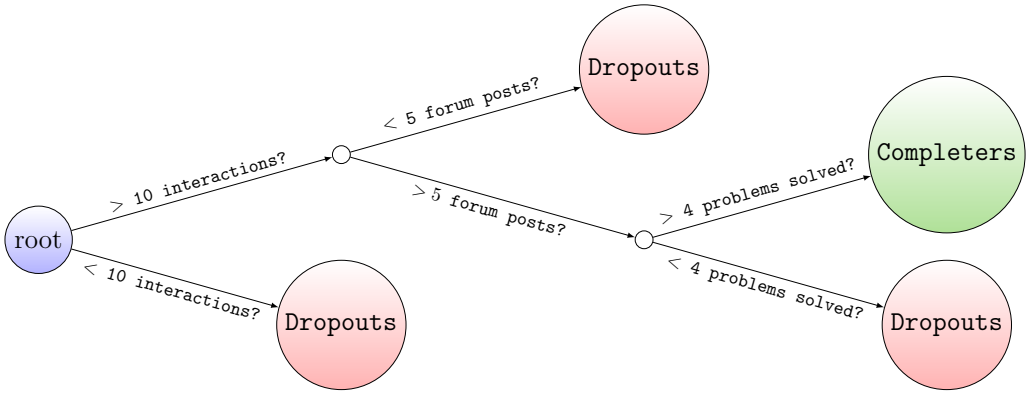


Figure 3.5: Decision Tree. It consists of a unique root node, internal node, and leaves. Each edge is a test on a particular feature. The path from the root node to each leaf gives the classification rule. We represent the root in blue, the internal nodes in white and the leaves that correspond to Dropouts in red, while those of Completers in green. In this example, a user with more than 10 interactions, more than 5 forum posts and more than 4 problems solved is classified as Completers.

The root of the tree is the only node having only outgoing edges. The remaining nodes can be either internal or leaves. Internal nodes always have exactly one ingoing edge and at least two outgoing edges. They represent tests on a particular attribute. Leaves have only one ingoing edge and no outgoing ones, and their values represent the taken decision.
Following the paths from the root node to each of the leaves, we obtain the classification rule. Figure 3.5 shows an example of a Decision Tree.
The application of a Decision Tree for classification and regression tasks is straightforward. Internal nodes define a test on one of the features, and each of the outgoing edges of the nodes represents the possible values of the targeted feature. The leaves contain either the class label or the probability distribution over the classes.
In order to obtain a tree from a dataset, the approach is the following:

- calculate the best split criterion for each feature;
- from all the features, select the one with the best split criterion and create two branches: examples failing the criterion go into one branch, those passing it into the other one;

- repeat these steps recursively on every produced branch until the stopping criterion is reached.

This algorithm depends on two parameters that need to be defined; how to find the best way to create new branches (split criterion) and the stopping criterion.
Generally, the metrics used to find the best split measure the homogeneity of the produced branches, the so-called node impurity; the more examples of a single class a node contains, the purer it is. In other terms, the impurity of a node is a measure of the reduction of uncertainty.
For example, if a split on a feature value yields a node containing only Completers and another node containing both Completers and Dropouts, the first node is pure while the second one as a certain impurity as it includes examples of both classes. Therefore, when we select a feature for a split, what we are searching for is a feature that produces branches with the lowest node impurity.
One common metric to find the best feature is **Information Gain** (**IG**), which is a measure based on the concept of entropy. Given a set of examples S, a set of label $L = 1, 2, ..., c$ in the number of classes C, and a probability p_i, $\forall i \in L$ of the examples S belonging to the class i, we can define the entropy of a node as:

$$H(S) = -\sum_{i=1}^{C} p_i \log_2 p_i \tag{3.14}$$

which in the case of only two classes ($L = 0, 1$) reduces to:

$$H(S) = -p_0 \log_2 p_0 - p_1 \log_2 p_1 \tag{3.15}$$

We calculate the IG of a split for a feature $feat$ as the difference between the entropy of the node $H(S)$ and the weighted sum of the entropy of the newly created nodes:

$$IG(feat) = H(S) - \sum_{i=1}^{C} \frac{S_i}{S} H(S_i) \tag{3.16}$$

The best split is then the one which provides the highest information gain:

$$feat^* = argmax\ IG(feat) \tag{3.17}$$

As an example, suppose we have a dataset consisting of 3 Completers and 7 Dropouts described by the features *Interactions (IT)* and *Problem Solved (PS)* as reported in Table 3.5 and plotted in Figure 3.6.
We indicate with IT_1 the set of examples with a value $IT < 10$ and with IT_2 those with a value $IT > 10$. In the same way, PS_1 includes examples with a value $PS < 5$, while examples with $PS > 5$ are in PS_2. To decide which of the two features to use for the split, we compute the entropy of the new nodes produced by a split of each feature as follows

$$H(IT_1) = 0 \tag{3.18}$$

$$H(IT_2) = -\frac{1}{8}\log_2(\frac{1}{8}) - \frac{7}{8}\log_2(\frac{7}{8}) = 0.544 \tag{3.19}$$

$$H(PS_1) = -\frac{1}{5}\log_2(\frac{1}{5}) - \frac{4}{5}\log_2(\frac{4}{5}) = 0.72 \tag{3.20}$$

Table 3.5: Dataset and Features for Decision Tree. We have a total of ten users, of which three are Completers and seven are Dropouts. For each user, we report the class and the values for the features *Interactions (IT)* and *Problem Solved (PS)*

Class	Interactions (IT)	Problem Solved (PS)
Completers	14	10
Completers	16	8
Completers	9	3
Dropouts	2	1
Dropouts	4	4
Dropouts	9	2
Dropouts	5	3
Dropouts	3	7
Dropouts	3	10
Dropouts	6	6

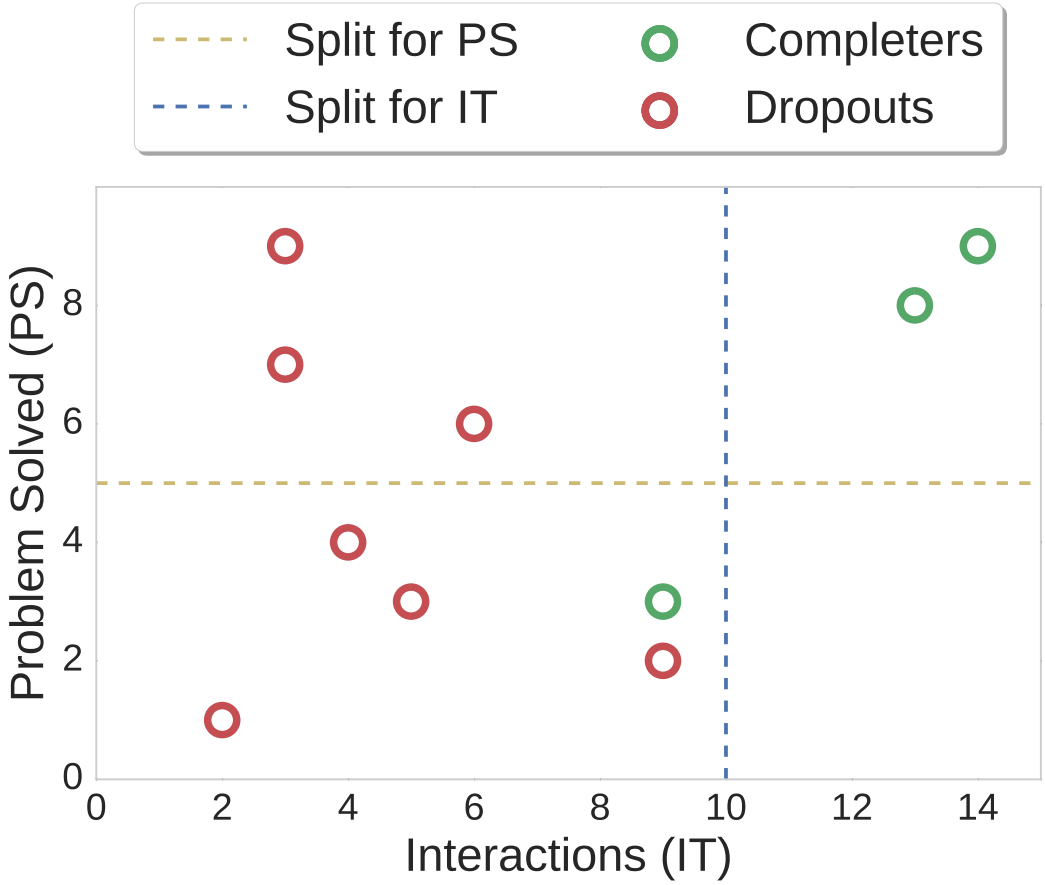


Figure 3.6: An example of a split for a Decision Tree. In this example, we plot the three Completers in green and seven Dropouts in red with regard to the features *Interactions (IT)* and *Problem Solved (PS)*. Furthermore, we also indicate the values of each feature at which we compute the split as dotted lines.

$$H(PS_2) = -\frac{2}{5}\log_2(\frac{2}{5}) - \frac{3}{5}\log_2(\frac{3}{5}) = 0.972 \tag{3.21}$$

With IT_1, IT_2, PS_1, and PS_2 as the sizes of the obtained splits and the entropy $H(S)$ of all examples computed as

$$H(S) = -\frac{3}{10}\log_2(\frac{3}{10}) - \frac{7}{10}\log_2(\frac{7}{10}) = 0.879 \tag{3.22}$$

the IG of the features is then:

$$IG(IT) = H(S) - \frac{IT_1}{S}H(IT_1) - \frac{IT_2}{S}H(IT_2) = 0.79 \tag{3.23}$$

$$IG(PS) = H(S) - \frac{PS_1}{S}H(PS_1) - \frac{PS_2}{S}H(PS_2) = 0.03 \tag{3.24}$$

The feature *Interactions (IT)* has the higher IG and, therefore, we select it and create the new nodes accordingly.
The stopping criterion could include different aspects such as reaching a fixed minimum leaf size or a maximal tree depth, insufficient improvements when an additional split is done or when all examples from a node have the same value and, thus, belong to the same class.
Decision Tree has certain advantages in comparison to other classifiers. First, the results are easy to interpret and understand as they can also be plotted. Furthermore, it can handle multivariate data and big datasets without extensive pre-processing steps such as the normalization of features.
However, a single tree built from a dataset usually has not a lot of predictive power, and we might not find the most suitable model.
We can improve the performance of a Decision Tree by so-called Ensemble methods. Instead of using a single Decision Tree, we perturb the training data and obtain several models. Then, we combine their outputs to get the ensemble prediction. There are several possibilities to extend Decision Tree through Ensemble methods.
As introduced in Breiman [9], with *Bagging* (Bootstrap aggregating), we can uniformly sample a training Data D of size n in order to generate m new training sets D_i each of size n' called bootstrap. By fitting each one of these m bootstraps, we learn a model. This method lowers variance and overfitting of the classifier. Plus, since the models are independent of each other, we can compute them in parallel.
Random Forests were introduced in Ho [29] and then further developed in Breiman [10]. This ensemble method is employed following a procedure similar to Bagging. In this case, the idea is to further add randomization to allow search on a larger function space. For example, instead of selecting at each split the feature with the highest IG overall, we can randomly select a subset of the features and pick the feature that maximizes the IG from this subset. This extra randomization helps to handle larger datasets and

also addresses the events of trees correlation. For example, if a few features are already robust predictors, it is likely that they would always be selected over most of the generated trees, thus becoming correlated.
Boosting is another ensemble method that helps to reduce bias and variance. It is based on the concept of weak and strong learners. A weak learner is a classifier which has better performance than a dummy one, but that is poorly correlated with the true classification. On the other hand, a classifier well-correlated with the true classification is called a strong learner. Whether it is possible to transform a set of weak learners into a single strong learner is a problem introduced in Kearns [35], in Kearns and Valiant [36] and later solved in Schapire [55].
The idea is to reweigh training examples so that the next learned classifier focuses on those examples that have been misclassified in the previous models. Every time a new model is learned, misclassified examples get a higher weight, while correctly classified ones get a lower weight. Therefore, we cannot learn the models in parallel. Although susceptible to noise, this method generally has good accuracy and is the one we also employ for our multisystem experiments.
As in the case of SVM, we estimate some classifier parameters with a Grid Search approach also with Boosted Decision Trees. In this case, we use the following variables and values:

- max_depth: (2, 6, 9, 12)
- subsample: (0.5, 1.0)
- colsample_bytree: (0.9, 1.0)

max_depth (default = 6) represents the maximum depth of a tree. If this value is too high, we risk overfitting the model as this could learn relations peculiar to the considered sample.
subsample (default = 1) represents the fraction of examples that are randomly sampled for each tree. The lower this value, the more conservative the classifier.
colsample_bytree (default = 1) represents the fraction of features that we randomly sample for each model.

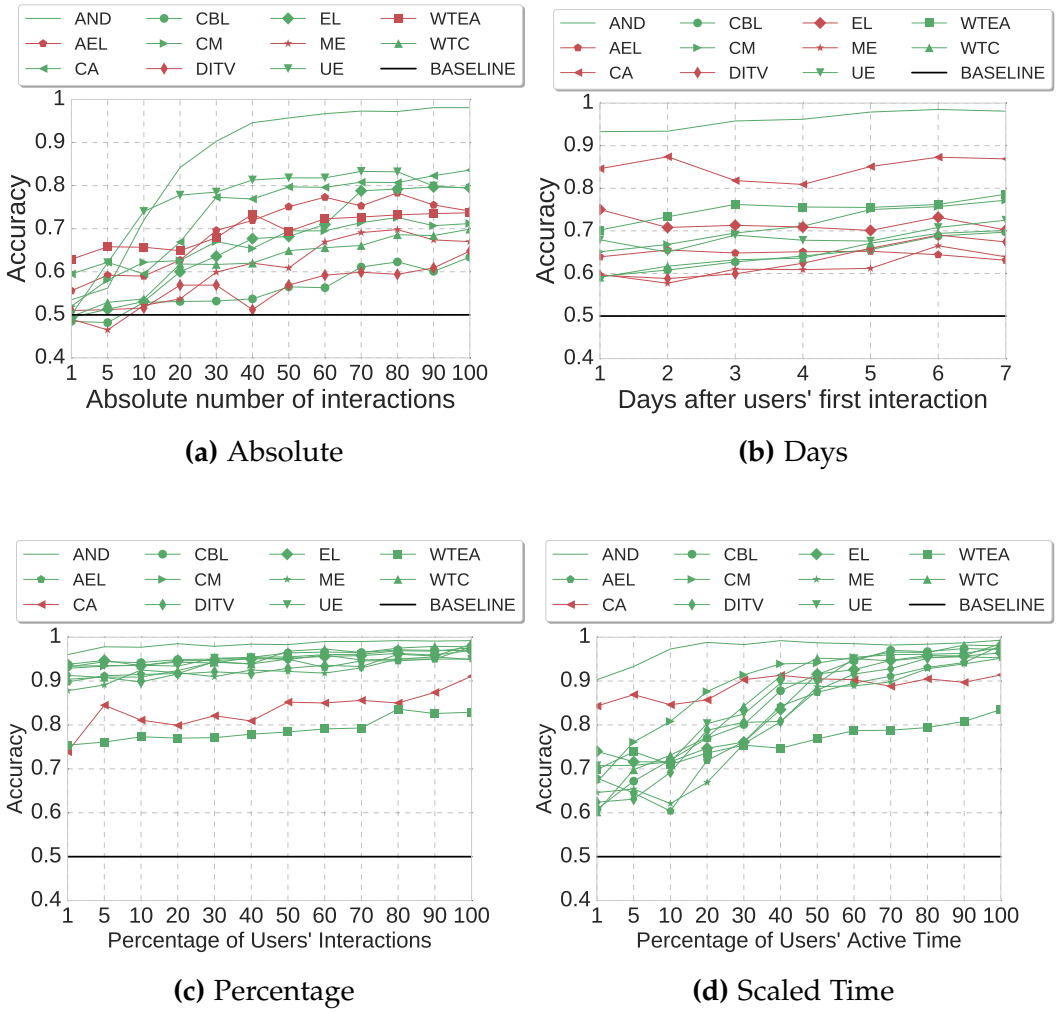
4 Results

In this chapter, we present our dropout prediction results. It is organized into three parts.
In the first section, we report the results for the MOOCs of Universidad Galileo. The MOOCs are analyzed one by one using accuracy as the evaluation metric. Furthermore, for each MOOC we report the results for all four presented approaches **Absolute**, **Days**, **Percentage** and **Scaled Time**.
The second section is organized in the same way, and we present the results for MOOCs of Curtin University. We report the results of the dropout classification for both MOOCs according to our four approaches.
In the third section, we consider the two systems together and present the results of the three multisystem experiments. Also in this case, we show the results of all multisystem experiments according to the four approaches previously described.
A detailed interpretation and discussion of the obtained results will be discussed in Chapter 5.

4.1 Single system experiments

4.1.1 Dropout prediction on Universidad Galileo

In Figure 4.1 we plot the accuracy (averaged over ten folds) for the dropout classification experiments of the MOOCs of Universidad Galileo. Each subfigure refers to one of the four proposed approaches for the construction of the features. Therefore, the x-axis always represents a different dimension. In Figure 4.1a it indicates the initial absolute interactions per user that we consider in the prediction experiment. In Figure 4.1b it indicates the number



(a) Absolute

(b) Days

(c) Percentage

(d) Scaled Time

Figure 4.1: Dropout prediction results for MOOCs of Universidad Galileo. Each subfigure depicts the accuracy results for one of the approaches. Figure 4.1a refers to the **Absolute** approach and its x-axis indicates the considered number of initial absolute interactions per user. Figure 4.1b reports the results of the **Days** approach, with the x-axis reporting the considered number of days from users' first interaction. We show the results of the **Percentage** approach in Figure 4.1c, whose x-axis represents the considered percentage of interactions per user. The Figure 4.1d depicts the results of the **Scaled Time** approach. In this case, the x-axis indicates the considered percentage of active time per user. The y-axis of each figure indicates the accuracy and is always bounded between 0.4 and 1, with the baseline plotted at 0.5. The accuracy of the MOOCs plotted in green is increasing with respect to the considered approach, while those of the MOOCs plotted in red is not.

of days from users' first interaction with the MOOC. The x-axis of Figure 4.1c represents the percentage of total interactions considered per user, while the x-axis in Figure 4.1d denotes the considered percentage of a user's active time for the prediction experiment.
For all these figures the y-axis represents the accuracy value that is bound between 0.4 and 1. Moreover, we plot the baseline as a solid black horizontal line at the accuracy value of 0.5. We denote each MOOC using a particular marker that remains the same in every subfigure.
Furthermore, we distinguish MOOCs with an accuracy increasing with respect to the specific metrics reported on the x-axis from those having an accuracy profile that presents both decreases and increases instead. Therefore, we plot the first group in green and the second group in red.

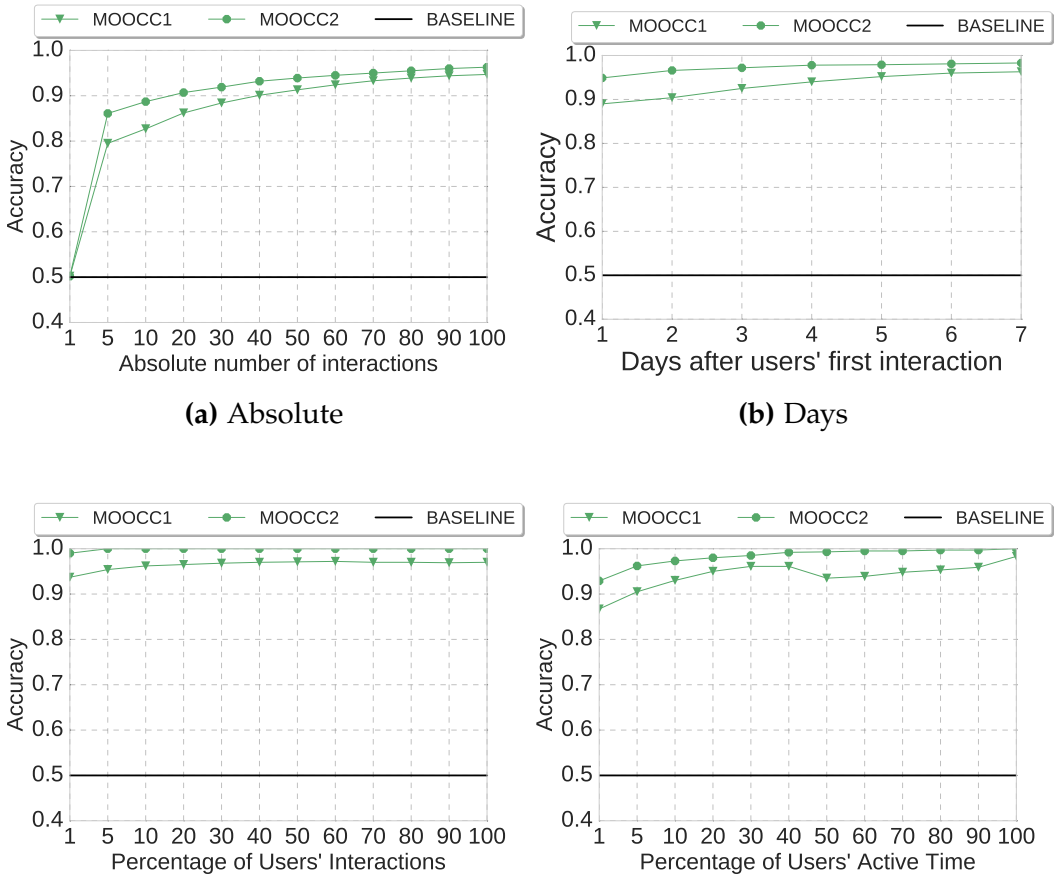
As shown in Figure 4.1a, with the **Absolute** approach, considering a higher number of interactions does not necessarily increase the accuracy for all MOOCs. There is a set of five MOOCs highlighted in green, whose accuracy either steadily increases or remains stable after an initial growth.
Additionally, the accuracy of the second group of six MOOCs plotted in red, do not seem to be directly related to the number of considered initial absolute interactions per user.
Figure 4.1b reports the result for the **Days** approach. Again, we can split the MOOCs into two groups. In total, six MOOCs plotted in green, have an increase in accuracy the more days we consider. This increase of accuracy is absent for the group of MOOCs highlighted in red, whose accuracy decreases the more days analyzed.
When we examine the results from the **Percentage** approach, as shown in Figure 4.1c, we notice a general increase in accuracy. Indeed, ten MOOCs out of eleven show an increase in accuracy the greater percentage of interactions per user taken into consideration. The only exception is the MOOC *CA*. However, the accuracy of this MOOC increases when the percentages are higher than 40%. *WTEA* is the MOOC with the overall lowest value of accuracy, even if we consider all interactions per user.
We can observe similar results for the **Scaled Time** approach as Figure 4.1d shows. Once again, *WTEA* is the MOOC with the lowest accuracy value when the considered percentages of active time per user are higher than 30%. The profile of *CA* becomes stable only when the considered percentages of active time per user are higher than 30%.

These results indicate that the **Percentage** approach is the one which yields the highest accuracy results for all MOOCs. In this case, for each MOOC the accuracy is higher than 0.7, including low percentages of users' interactions. With the **Scaled Time** approach, we see a constant increase in accuracy the higher the considered percentages of active time per user.
The accuracy for the **Absolute** and the **Days** approaches have wider scopes, ranging between 0.47 and 0.98 for **Absolute** and between 0.57 and 0.98 for **Days**. For most of the MOOCs, these two approaches yield high accuracy when classifying dropouts.

4.1.2 Dropout prediction on Curtin University

Figure 4.2 depicts the accuracy (averaged over ten folds) for the dropout classification experiments over the MOOCs of Curtin University.
As with the results of Universidad Galileo, we dedicate a subfigure to each of the four approaches we used to extract features. Therefore, the x-axis represents the number of considered initial absolute interactions per user in Figure 4.2a, the number of considered days from users' first interactions in Figure 4.2b; the considered percentages of total interactions per user in Figure 4.2c; the considered percentage of active time per user in Figure 4.2d.

From the results of the **Absolute** approach shown in Figure 4.2a, we notice a constant increase of the accuracy for both MOOCs, which reaches the highest value when we consider the first one hundred absolute interactions per user. Specifically, already five absolute interactions are enough to obtain an accuracy value of 0.8 for both MOOCs.
The **Percentage** approach, as shown in Figure 4.2c, is the one that yields the highest accuracy for both MOOCs. Even for small percentages, the accuracy values of both MOOCs are never lower than 0.93. In particular, for *MOOCC2*, the accuracy is stable at 1 when the considered percentages of interactions per user are higher than 5%. For *MOOCC1* accuracy remains stable at 0.97 when the analyzed percentages of interactions per user are larger than 30%.
For the **Scaled Time** approach, as shown in Figure 4.2d, we obtain different trends of accuracy for the two MOOCs. The accuracy of *MOOCC1* steadily increases in the percentage of users' active time, and it is always higher



(a) Absolute

(b) Days

(c) Percentage

(d) Scaled Time

Figure 4.2: Dropouts prediction results for MOOCs of Curtin University. In each subfigure, we depict accuracy results of each approach. In Figure 4.2a we plot the results of the **Absolute** approach, with the x-axis indicating the considered number of initial absolute interactions per user. Figure 4.2b depicts the results of the **Days** approach, with the x-axis indicating the considered number of days from users' first interaction. We report the results of the **Percentage** approach in Figure 4.2c. In this case, the x-axis indicates the considered percentage of interactions per user. Figure 4.2d depicts the results of the **Scaled Time** with the x-axis listing the considered percentages of active time per user. On the y-axis of each figure we indicate the accuracy, which is always bounded between 0.4 and 1. We also plot the baseline as a horizontal black line at 0.5. The accuracy of all MOOCs is increasing with respect to the considered metric. Therefore, we plot all MOOCs in green.

than 0.92. The accuracy of *MOOCC2* is also growing in the considered percentage of active time per user, with only a slight decrease between 40% and 50% of the analyzed percentage of active time per user. On the other hand, between 1% and 40% and from 50% to 100% of users' active time, the accuracy increases.
The results for the **Days** approach, depicted in Figure 4.2b, differ from those we obtain with the other approaches. In the case of *MOOCC2* the accuracy increases in the number of considered days, but only gets as high as 0.76 when we study the first seven days after users' first interaction. On the other hand, the accuracy of *MOOCC1* is constant at 0.5 regardless of the number of considered days. We believe there is a certain delay between the first interaction of users and their remaining interactions that influences the accuracy of the dropout prediction experiments for both MOOCs.

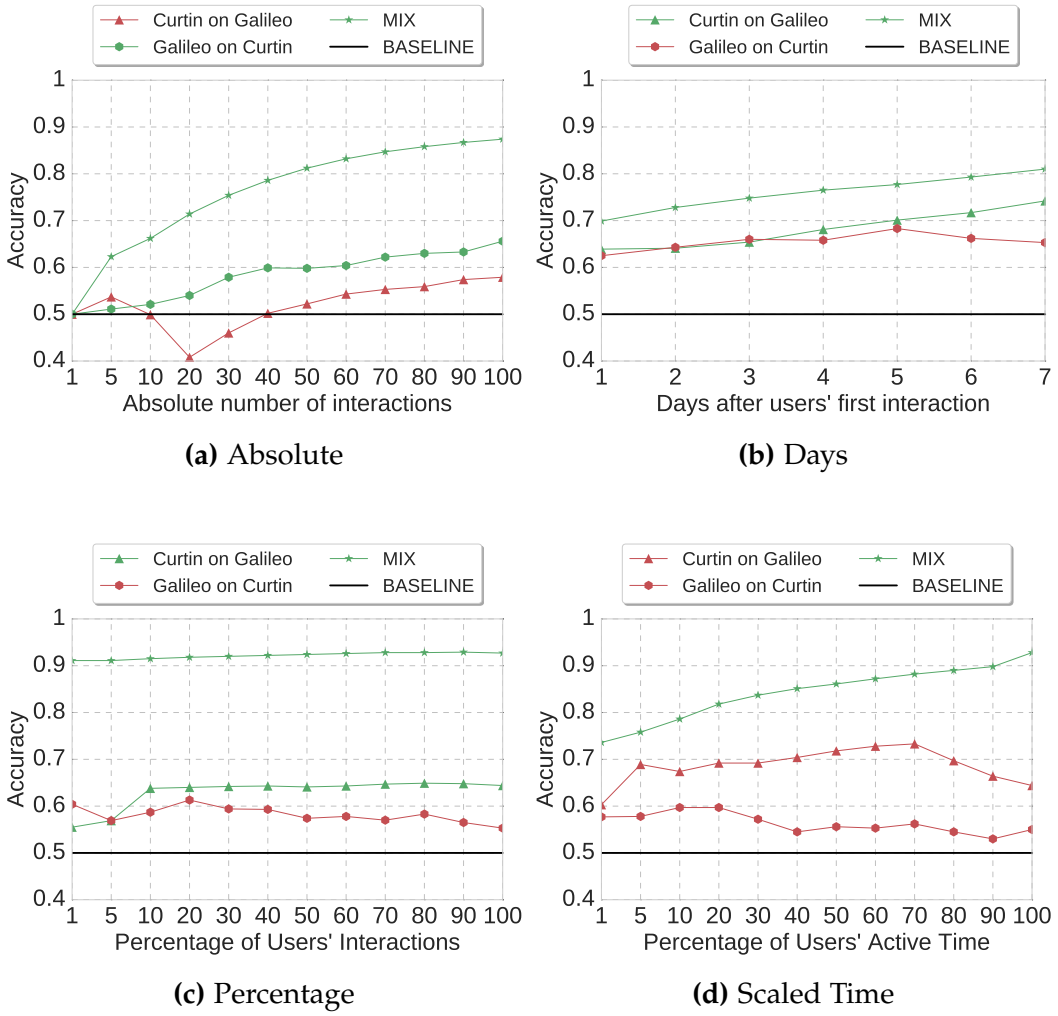
For the most part, we obtain high accuracy for the dropout prediction experiments on the MOOCs of Curtin University for each of the proposed approaches, except for the **Days** methodology.
Overall, dropout prediction experiments yield higher accuracy on Curtin University's MOOCs than on MOOCs of Universidad Galileo. The low accuracy we obtain with the **Days** approach might be related to the distribution of interactions of users over time.

4.2 Multisystem experiments

4.2.1 Dropout prediction experiments

In Figure 4.3 we report the accuracy (averaged over ten folds) for the multisystem dropout prediction experiments *MIX* and for the system-to-system experiments *Galileo on Curtin* and *Curtin on Galileo*.
Once again, we show subfigure to each of the approaches that we proposed: Figure 4.2a for **Absolute**, Figure 4.2b for **Days**, Figure 4.2c for **Percentage** and Figure 4.2d for **Scaled Time**.

Among the considered approaches, we get the highest values of accuracy for the *MIX* experiments. Moreover, it is the only experiment for which



(a) Absolute **(b)** Days

(c) Percentage **(d)** Scaled Time

Figure 4.3: Dropout prediction results for Multisystem and System-to-System experiments. Each subfigure depicts the accuracy results for one of the proposed approaches. We show in Figure 4.3a the results of the **Absolute** approach and we indicate on its x-axis the considered number of initial absolute interactions per user. Figure 4.3b reports the results of the **Days** approach. The x-axis lists the number of considered days from users' first interaction. We depict the results of the **Percentage** approach in Figure 4.3c and indicate the considered percentage of interactions per user on the x-axis. We show the results of the **Scaled Time** approach in Figure 4.3d, whose x-axis indicates the considered percentage of active time per user. The y-axis of each figure indicates the accuracy and is always bounded between 0.4 and 1, with the baseline plotted at 0.5. MOOCs plotted in green have an increasing accuracy with respect to the considered approach, while those MOOCs plotted in red do not.

accuracy always increases. Indeed, the accuracy we obtain is consistently higher than 0.7 except for the **Days** approach, where it is limited to 0.68.
With the **Percentage** approach we obtain an accuracy profile invariably greater than 0.9, even when we consider low percentages of interaction per user.
For the **Absolute** approach, the initial five absolute interactions per users are enough to obtain an accuracy of 0.8, which increases for higher numbers of considered initial absolute interactions per user.
We see a constant increase in accuracy with respect to the considered percentage of active time per user also for the **Scaled Time** approach, whose accuracy is never lower than 0.73.

On the other hand, the accuracy for the system-to-system dropout prediction experiments *Galileo on Curtin* and *Curtin on Galileo* are generally low and, mostly, close to the 0.5 value of the baseline.
The **Absolute** and **Days** approaches are the only considered methodologies where we can measure a small increase in accuracy the higher the considered number of initial absolute interaction per user and the considered number of day from users' first interaction respectively.
For the **Percentage** and **Scaled Time** approaches we cannot identify how accuracy evolves when we increase the considered percentage of interactions per user and the percentage of active time per user. Notably, for the **Scaled Time** approach, we observe for both *Curtin on Galileo*, and *Galileo on Curtin* experiments lower values of the accuracy the higher the considered percentages of active time per user.
The accuracy of the **Percentage** approach for the *Galileo on Curtin* experiment also decreases for greater considered percentages of interaction per user. However, the same method for the *Curtin on Galileo* experiment yields an almost constant accuracy of 0.64.

Overall, system-to-system dropout prediction experiments *Galileo on Curtin* and *Curtin on Galileo*, where we use information about one system to predict dropouts in the other one, yield accuracy that outperforms the baseline.
On the other hand, we obtain higher accuracy results for multisystem dropouts prediction experiment *MIX*, in which we use a dataset that includes information from both systems.
The only exception is the **Days** approach. We believe this one is influenced

by the distribution of interactions of users with the MOOCs of Curtin University.

4.2.2 Feature analysis for dropout prediction

In this section we report the scores of each of our features for the **Absolute**, **Days**, **Percentage** and **Scaled Time** approaches in Table 4.1a-4.1d. Each of the tables have the same structure; in the first column, we list the features and report the scores of the features we used for the dropout prediction experiments. In the remaining columns, we list the feature scores for the system-to-system (*Galileo on Curtin* and *Curtin on Galileo*) and the multisystem (*MIX*) dropout prediction experiments.
We want to compare how the score of each feature varies between the start and the end of the MOOCs. Therefore, for each experiment, we list only the scores of the features for initial, middle and final values of each approach. Particularly, we report the scores for 5, 10, 50 and 100 considered initial absolute interactions per user for the **Absolute** approach in Table 4.1a; 1, 3, 5 and 7 days from users' first interaction for the **Days** approach in the Table 4.1b; 5%, 10%, 50% and 100% of interactions per user for the **Percentage** approach in Table 4.1c; 5%, 10%, 50% and 100% of considered active time per user for the **Scaled Time** approach in the Table 4.1d. We highlight the three features with the highest score for each setting and experiment in green.

We name the four features with the lowest score *low-scoring* features, in contraposition with the three features with the highest score that we call *high-scoring* features. Further, we notice that certain features always belong to the same group among the different approaches and experiments.
For example, the feature *Days* is always a low-scoring feature in each of our experiments. With a couple of exceptions for the **Days** (*Galileo on Curtin* first 5 days after users' first interaction), **Percentage** (*Curtin on Galileo* with 100% of considered interactions per user) and **Scaled Time** (*Galileo on Curtin* with 5% of considered active time per user) approaches, the feature *Active Days Requests* always belongs to the group of features with low scores.
The feature *Sessions* also belongs to the low-scoring features for most experiments, with an exception for the **Percentage** approach (*Galileo on Curtin*

Table 4.1: Feature scores of multisystem and system-to-system experiments. The first column in each table lists the features we used in the dropout prediction experiments. The remaining columns list the feature scores for each experiment; *Curtin on Galileo*, *Galileo on Curtin* and *MIX*. We focus on how the scores evolves from the beginning until the end of the MOOCs and, therefore, we report the scores for intial, middle and final values of each condisered approach. The three features highlighted in green are the ones with the highest scores for each experiment and value. The features *Session Length*, *Timespan Clicks*, *Requests* and *Active Time* are the one with the highest scores almost for each setting. *Days* is always the lowest-scoring feature. *Active Days Requests* and *Sessions* are rarely among the high-scoring ones and have, generally, low scores.

(a) Absolute

	Curtin on Galileo				Galileo on Curtin				MIX			
Absolute Interactions	5	10	50	100	5	10	50	100	5	10	50	100
Days	0.96	0.64	1.36	0.76	0.30	0.56	0.78	0.92	39.94	35.31	66.94	30.51
Active Days Requests	0.10	0.56	0.54	0.76	0.12	0.10	0.40	0.80	3.78	22.65	17.95	52.32
Session Length	1.76	2.38	2.50	1.72	2.36	2.44	3.12	2.64	98.41	109.59	132.30	105.26
Timespan Clicks	1.50	1.84	2.44	2.32	1.22	1.10	1.68	1.64	82.43	74.08	121.53	87.51
Requests	2.86	2.88	2.46	3.48	0.84	1.00	1.22	1.46	147.83	142.78	128.37	181.62
Sessions	0.64	0.82	0.80	0.98	0.76	0.62	0.58	0.66	33.69	31.89	41.18	59.31
Active Time	3.06	3.10	2.02	2.48	4.64	4.06	3.64	3.74	147.20	168.66	121.02	132.76

(b) Days

	Curtin on Galileo				Galileo on Curtin				MIX			
Days	1	3	5	7	1	3	5	7	1	3	5	7
Days	9.70	18.67	29.07	45.47	13.70	31.50	23.67	19.13	14.10	18.83	27.64	42.39
Active Days Requests	101.20	68.40	67.27	65.83	55.70	111.13	101.10	97.63	87.61	70.25	79.83	66.34
Session Length	99.17	92.43	111.37	108.50	117.60	120.43	116.63	137.90	103.07	99.46	127.60	107.88
Timespan Clicks	107.43	132.63	128.43	115.87	109.40	113.53	98.73	113.53	122.25	130.45	125.76	125.19
Requests	136.37	151.10	135.50	134.67	119.67	102.07	114.53	108.83	122.48	139.00	124.74	131.22
Sessions	36.87	30.53	47.60	27.97	20.57	15.83	38.07	29.73	40.08	28.24	36.88	24.47
Active Time	120.33	132.10	113.30	133.73	136.77	120.37	95.97	94.43	132.62	135.20	110.66	136.10

(c) Percentage

	Curtin on Galileo				Galileo on Curtin				MIX			
Percentage	5%	10%	50%	100%	5%	10%	50%	100%	5%	10%	50%	100%
Days	32.10	36.15	48.85	41.55	35.45	33.65	42.15	61.90	31.32	36.78	53.36	58.92
Active Days Requests	52.90	59.15	79.30	86.8	85.85	94.30	84.70	73.50	55.27	58.475	74.02	80.25
Session Length	116.00	102.70	92.90	85.75	64.75	81.70	72.15	62.60	112.88	101.66	86.84	88.47
Timespan Clicks	102.50	75.80	92.90	85.75	96.25	102.20	70.70	80.80	91.60	59.57	79.75	74.49
Requests	222.25	218.65	222.10	212.85	138.35	152.75	165.85	132.75	226.19	232.07	236.85	207.79
Sessions	23.05	29.75	24.45	25.40	36.95	37.30	57.35	98.60	19.47	33.86	22.39	26.74
Active Time	112.10	127.30	105.60	130.15	127.20	100.15	72.85	42.60	123.14	119.29	96.27	114.82

(d) Scaled Time

	Curtin on Galileo				Galileo on Curtin				MIX			
Percentage	5%	10%	50%	100%	5%	10%	50%	100%	5%	10%	50%	100%
Days	27.75	54.40	56.95	41.50	38.40	57.95	44.50	63.30	54.25	80.03	69.62	57.61
Active Days Requests	83.25	80.00	73.75	84.70	102.50	65.15	53.90	75.20	79.51	70.39	78.00	81.06
Session Length	146.40	108.10	99.15	87.80	97.10	81.95	98.35	57.20	125.19	107.43	94.88	89.47
Timespan Clicks	110.15	102.85	111.25	71.35	58.10	72.75	36.65	77.40	82.16	88.71	105.70	74.16
Requests	102.60	140.50	169.35	212.00	106.90	118.15	154.45	138.35	113.15	130.13	169.60	209.95
Sessions	45.25	50.35	40.60	24.95	49.25	75.35	70.05	100.05	62.99	58.19	29.82	26.86
Active Time	122.15	113.50	117.80	130.20	105.95	102.65	65.75	38.70	112.20	109.63	106.66	115.37

with 100% of considered interactions per user) and two exceptions for the **Scaled Time** approach (*Galileo on Curtin* with 50% and 100% of considered active time per user).
When we consider the results of the multisystem dropout prediction experiment *MIX* we notice that the features *Days*, *Active Days Requests* and *Sessions* are always low-scoring. On the contrary, the features *Requests* and *Active Time* are always among the high-scoring features. *Session length* belongs to the group of high-scoring features for **Absolute**, **Percentage** and **Scaled Time** approaches. For the **Days** approach *Timespan Clicks* gets higher scores than *Session Length*.

Generally, for the system-to-system dropout prediction experiments *Galileo on Curtin* and *Curtin on Galileo*, we can not distinguish between low- and high-scoring features as clearly as in the multisystem scenario. In particular, there is no feature always belonging to the high-scoring group.
However, we notice that the features *Session Length*, *Timespan Clicks*, *Requests* and *Active Time* belong to this group for most of the approaches and experiments. Moreover, the feature *Days* is the only feature that consistently scores low. This group also includes the features *Active Days Requests* and *Sessions* for the **Absolute** approach as well.

4.3 Summary of results

We can summarize our results as follows:

- for MOOCs characterized by a fixed length and schedule as those of Universidad Galileo, the **Absolute** and **Days** approaches yield mixed results. Except for the MOOC *AND*, there is an upper limit for the accuracy level that tends to stabilize the higher the considered initial absolute interactions per user and days after the first interaction per user. These situations are due to noise in the logs that influences the initial part of the MOOCs of this system.
  With the **Percentage** and **Scaled Time** approaches, the accuracy increases accordingly with the considered dimension. However, we recognize a peculiar accuracy profile for MOOCs *CA* and *WTEA*. We

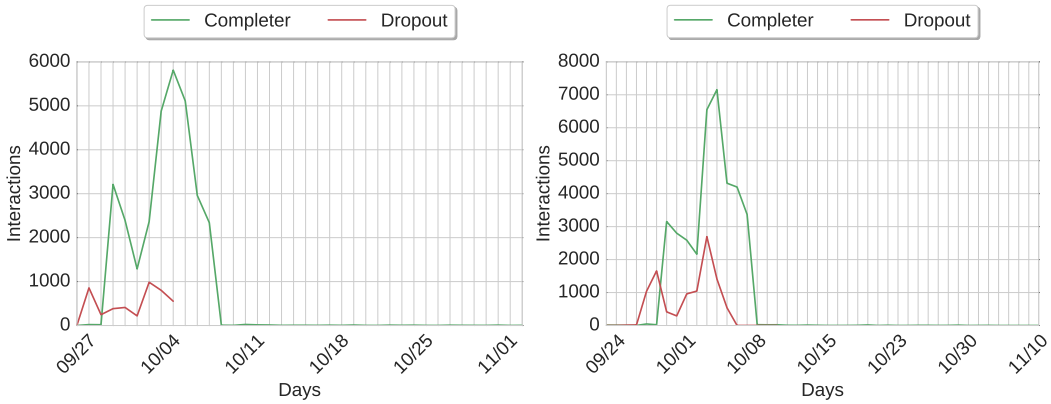
believe the settings, organization, and duration of these MOOCs differ from those of the other MOOCs of Universidad Galileo.

- the high level of details of the logs of Curtin University has a strong positive impact on the dropout prediction experiments. Particularly, the self-paced MOOCs of this system exhibit high accuracy for the **Absolute**, **Percentage** and **Scaled Time** approaches. As for Universidad Galileo, the **Percentage** and **Scaled Time** approaches yield the highest accuracy.
  On the other hand, we obtain the lowest results concerning accuracy with the **Days** approach. We speculate that the distribution of interactions during the initial days after the first interaction of users is negatively skewed and this could have an adverse impact on the accuracy of the dropout prediction experiment using this approach;
- the results of the multisystem and system-to-system experiments denote that different systems can be analyzed together. Generally, the use of a mixed dataset, as in the case of the *MIX* experiment, yields accuracy as high as 0.9 for all approaches except the **Days** one.
  To train a classifier on data from one system to predict dropouts on another system (as in *Galileo on Curtin* and *Curtin on Galileo* experiments) has been proven to produce less reliable results than those of which we were able to obtain with the *MIX* experiment.
  Potential explanations for this observation could be a result of the small set of features we used and the system differences, particularly, the fixed length of MOOCs of Universidad Galileo in contrast with the self-paced setting of Curtin University's MOOCs.

5 Discussion

5.1 Influence of duration of MOOCs on accuracy

When analyzing the accuracy of the dropout prediction experiments on the MOOCs of this system for the four proposed approaches, we notice that the MOOCs *AND*, *CA* and *WTEA* have peculiar accuracy profiles.
First, *AND* is the MOOC with the highest accuracy, which always increases and then stabilizes for each approach. Particularly, its accuracy is always higher than 0.9, except for the **Absolute** approach when we consider less than thirty initial absolute interactions per user. These results indicate that analyzing up to thirty initial absolute interactions per user, with the considered features, the behaviors of Completers and Dropouts are too similar for the classifiers to classify all users correctly. Therefore, this could also indicate that the Dropouts might stop engaging with the course after an initial phase in which they test the class and the learning environment.
Besides, from the results of the **Percentage** and **Scaled Time** approaches we see that the accuracy profiles of the MOOCs *CA* and *WTEA* differ from those of the other MOOCs. With the **Percentage** approach, the accuracies of *CA* and *WTEA* are bounded between 0.73 and 0.9 and are always lower than those from the rest of the MOOCs of this system. Moreover, the accuracy of *CA* does not strictly increase with the considered percentage of interactions per user.
We notice this same situation for this MOOC with the **Scaled Time** approach. In this case, its accuracy is higher than the rest of the MOOCs when the considered percentage of active time per user is lower than 40%. In contrast with the other MOOCs, the accuracy does not increase when we consider higher percentages.
With the **Scaled Time** approach, *WTEA* has an accuracy profile similar to those from the other MOOCs. Yet, its accuracy increases at a least significant



(a) *CA* **(b)** *WTEA*

Figure 5.1: Per-class interactions for MOOCs *CA* and *WTEA*. For both Figures, the x-axis indicates the date (every vertical line of the grid marks a day), while on the y-axis we report the number of interactions. We plot the interactions of the Completers in green and those from the Dropouts in red. Almost all interactions of both classes happen only for twelve consecutive days.

rate for greater percentages, getting as high as 0.83 when using the total percentage of active time per user. Interestingly, these MOOCs are the ones with the highest accuracy (together with *AND*) for the **Days** approach.
From Table 3.2 (see Section 3.1.1) we also notice that these MOOCs are among the three MOOCs with a dropout rate lower than 50%; *CA* is the MOOC with the lowest number of active users (89) and the lowest dropout rate (33%). *WTEA* has the second lowest dropout rate (44%) and has only 176 active users. Moreover, they have the lowest number of interactions per user among the MOOCs of Universidad Galileo (394 for *CA* and 265 for *WTEA*). *DITV* is the third MOOC with a dropout rate lower than 50% (46%), has a total of 117 active users and an average of 999 interactions per user.
Despite the common official eight weeks duration of the MOOCs of this system, we verify that the interactions in these MOOCs take place only for twelve consecutive days. We plot the overall interactions per days over the eight weeks duration for these MOOCs in Figure 5.1. Due to this situation, the high accuracy for these MOOCs with the **Days** approach is understandable because a single day for both of these MOOCs represents a higher

percentage of the overall time the users are active.
For the other approaches, the shorter duration of these MOOCs influences the features similarly. Most of the features (see 3.2.1) are based on the concept of user sessions. Shorter duration of a course signifies a shorter amount of time for the users to interact and, potentially, a lower number of overall user sessions and a low discrepancy between the number of sessions of Completers and Dropouts. Therefore, we report the average number of sessions for Dropouts and Completers for all MOOCs of Universidad Galileo in Figure 5.2.

We see a clear distinction between the average number of sessions of *CA* and *WTEA* and the remaining MOOCs. First, their average number of sessions for the Completers is the lowest among all MOOCs. Second, the difference between the average number of sessions of Completers and Dropouts is the lowest as well; 10.88 for the Completers and 2.66 for the Dropouts of *CA* and 9.71 and 6.6 respectively for Completers and Dropouts of *WTEA*.
As the features we use are based on the concept of sessions (see 3.2.1), when the difference between the number of sessions of the two classes is as small as in the case of these MOOCs, the whole set of features has little variance. Thus, it becomes more complicated for SVM to detect the class users belong to correctly.

5.2 Implication of later start of MOOCs

The accuracy for the remaining MOOCs with the **Percentage** and **Scaled Time** approaches increases the higher the percentage of users' interactions and the percentage of users' active time we consider. We do not see this relation with the **Absolute** and **Days** approaches.
To investigate the reasons for this situation, we plot the distribution of interactions of Completers and Dropouts over the duration of the course for some MOOCs in Figure 5.3. Particularly, we plot the distributions of the MOOC with the highest accuracy profile (*AND*) and compare it with three MOOCs whose accuracy does not constantly increase for both **Absolute** and **Days** approaches: *AEL*, *DITV* and *ME*.

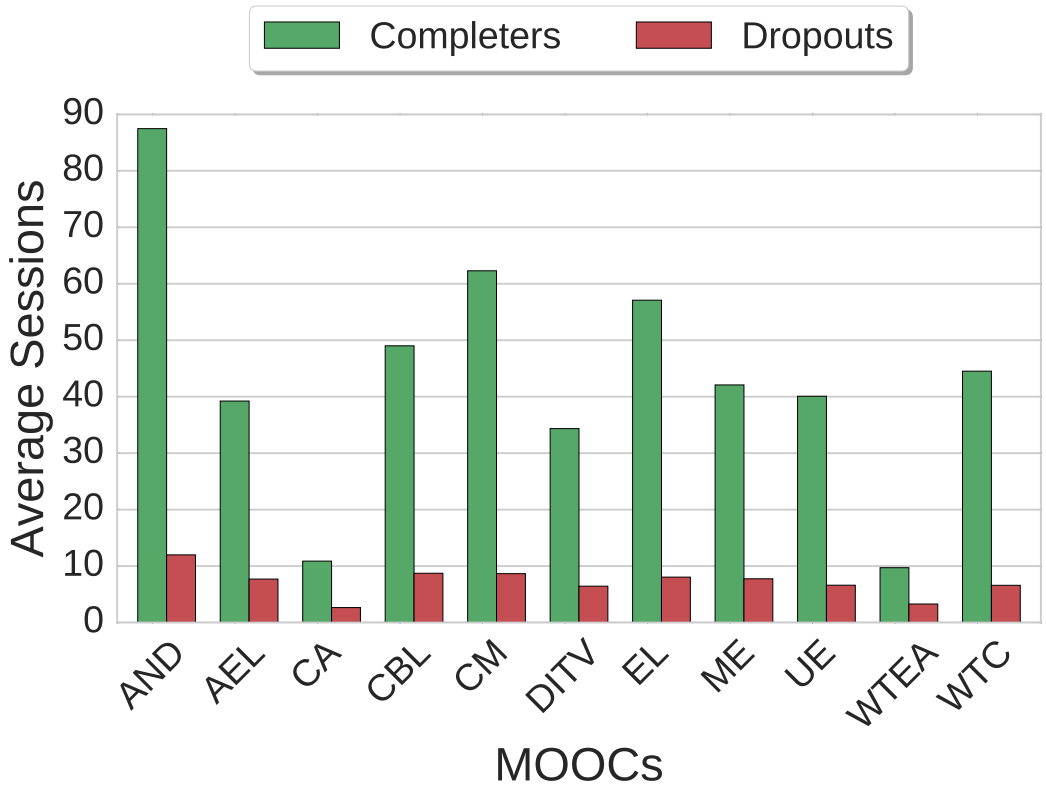
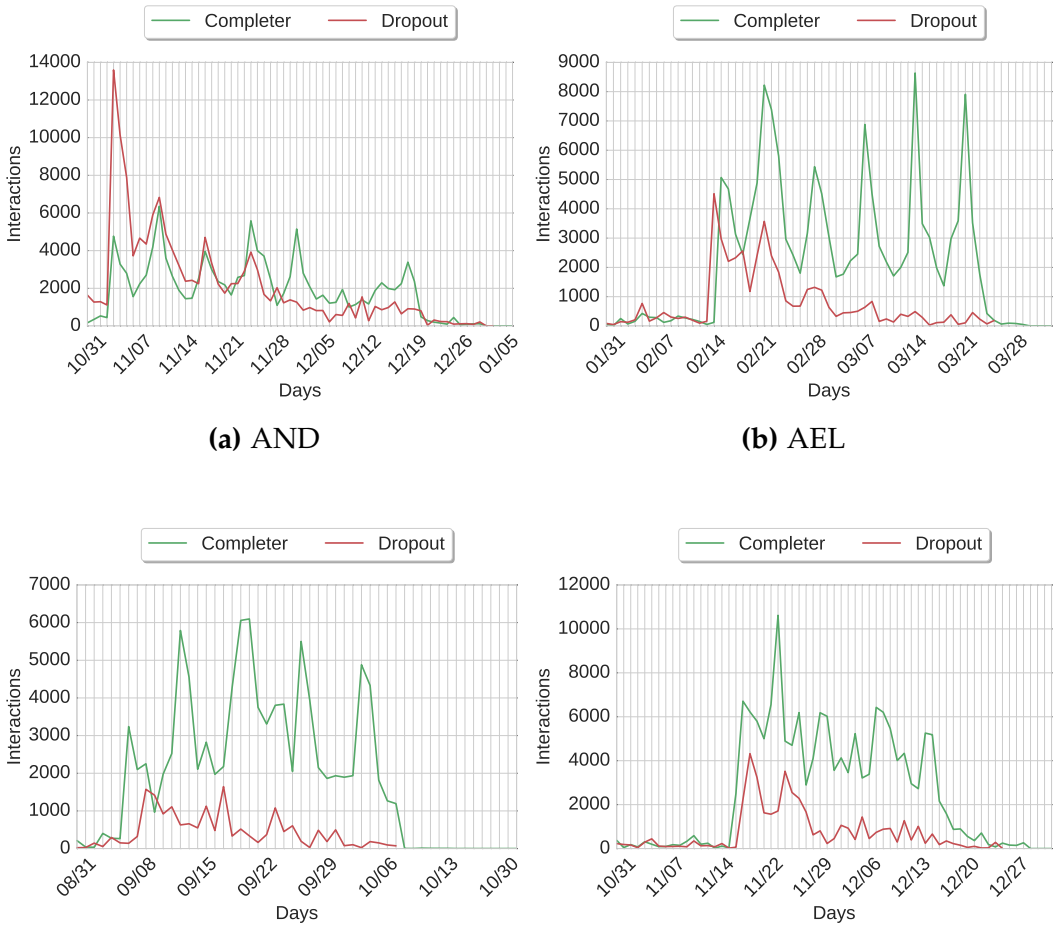


Figure 5.2: Per-class average number of sessions. We list on the x-axis the MOOCs of Universidad Galileo, on the y-axis the average number of sessions, and we plot Completers in green and Dropouts in red. For all MOOCs, the Completers have higher averaged numbers of sessions than the Dropouts. However, for MOOCs *CA* and *WTEA*, the difference between the two classes, is not as significant as for the other MOOCs. This situation is likely due to the shorter duration of these two MOOCs.

We see from Figure 5.3a that for *AND*, the initial day already accounts for a considerable number of interactions. The other MOOCs are characterized by a low number of interactions during the initial days and a burst in the number of interactions afterward. This initial period with low interactions consists of fifteen days for *AEL*, six days for *DITV* and seventeen days for *ME* (Figures 5.3b, 5.3c, and 5.3d respectively).
However, the distribution of the interactions of Completers and Dropouts during this initial period does not differ considerably, and they become



(a) AND **(b)** AEL

(c) DITV **(d)** ME

Figure 5.3: Per-class interactions over MOOCs duration. The x-axis represents dates and the y-axis the number of interactions. For each subfigure, we plot the Completers in green and the Dropouts in red. We see that *AND* is characterized by a high number of interactions during the initial days of the course (Figure 5.3a). The MOOCs *AEL* and *ME*, as shown in Figure 5.3b and 5.3d are characterized by the first fifteen and seventeen days respectively, during which the modest number of overall interactions causes the two classes to have a similar distribution of interactions. This situation leads to less precise classifications of the users. The delay for the MOOC *DITV*, as shown in Figure 5.3c, includes almost the first week of the course.

distinguishable only after this time. We believe that there is a definite delay of the actual start of the course for these MOOCs, which can happen, for example, when the course is already open to users, but the material is uploaded only with a delay of some days. Therefore, the initial days are characterized by few interactions that increase in number once the material is uploaded.
This particular situation explains the general low accuracies we obtain with the **Days** approach; in the initial phase users interact with the MOOCs only to verify if the course has started (that is, whether the material is already available or not). Therefore, these interactions do not strictly represent user engagement and, consequently, the features extracted from these interactions do not reflect the differences between Completers and Dropouts.
Therefore, we conclude that the lower accuracy for some MOOCs using the **Absolute** approach could be due to this situation.

5.3 Higher details of logs to increase accuracy of the prediction

The accuracy of the self-paced MOOCs of Curtin University is higher than those of Universidad Galileo for the majority of the approaches.
Particularly, with the **Absolute** and **Scaled Time** approaches the accuracy of both MOOCs of this system increases the greater the considered initial absolute interactions per user and considered percentage of active time per user.
With the **Percentage** approach, the accuracy of both MOOCs is always higher than 0.92 and it either grows slowly (*MOOCC2*) or remains stable (*MOOCC1*). These discrepancies in accuracy between the two systems are due to differences between the structures and organizations of the courses of each system.
First, MOOCs of Curtin University are self-paced; when the course starts, the entirety of course material, resources and problems are available to the users, who are free to interact with the course at their own pace. This implies that all interactions are representative of the users' engagement style (which is not the case for Universidad Galileo's MOOCs, as explained in

Section 5.1).
Second, there is a big difference in the average number of interactions between the two systems. From Table 3.3 we see that the average number of interactions per user of Curtin University system is 93 for *MOOCC1* and 58 for *MOOCC2*. Therefore, when we experiment with five initial absolute interactions per user with the **Absolute** approach we consider, on average, 5.38% for *MOOCC1* and 8.62% for *MOOCC2* of users' overall interactions.
The MOOC with the highest percentage of Universidad Galileo is *WTEA*, in which 5 interactions represents, on average, only 1.89% of each user overall interactions. Thus, the initial interactions in the MOOCs of Curtin University carry more information than those of Universidad Galileo's MOOCs.
Third, the level of detail of the edX logs is higher than the one of Universidad Galileo. As explained in Section 3.1.2, in the edX log, each available tool is described in higher detail by subcategories, and we derive a feature from each of them. As a result, Curtin University's interactions include almost hundred different tools, while those from Universidad Galileo comprise only twenty different tools.
This high variety of tools translates into higher number of features, which make it easier for the classifier to correctly identify and categorize users as either Completers or Dropouts.
Taking a step forward, we further investigate these situations by listing the most used tools for *MOOCC1* in Table 5.1. We divide the tools/features according to the group they belong to and compare experiments with 5, 10 and 100 number of initial absolute interactions per user. For each experiment and class, the column % represents the percentage of the whole interactions the feature accounts for. The *AVG* column indicates the number of interactions per user averaged over the total number of users of the class. We omit features with a percentage lower than 1%, which we indicate as empty cells.
We see that most used tools with 5 interactions belong to *Video* and *Main Page Links* components. The higher the number of initial absolute interactions per user we consider, the more users engage with *Video*, *Course Navigation*, and *Problem* components. *Discussion Forum* is scarcely used and mostly for visualization purposes only.
Therefore, we conclude that for this system the initial absolute interactions of users are strong indicators whether they will complete the course or not. Moreover, there is a particular set of components (*Video* and *Main Page Links*)

that strongly catalyze users' attention.

5.4 Distribution of interactions of self-paced MOOCs

The results of the **Days** approach represents an exception to the general high accuracy that we obtain for MOOCs of Curtin University.
With this approach, *MOOCC1* has a constant accuracy of 0.5, the baseline value. The accuracy of *MOOCC2* increases but with a slower pace compared to the other approaches. We believe that a possible explanation for this is represented by the Enrolment actions, which form part of Curtin University's logs.
MOOCs of Curtin University are characterized by an initial time window of

Table 5.1: Distribution of absolute interactions for MOOCC1. The table depicts the most used tools when we analyze 5, 50 and 100 initial absolute interactions per user. For each feature, we report its percentage of the class interactions in column % and the average requests per class in column *AVG*. The *Video*, *Course Navigation*, and *Main Page Links* are the most used components by Completers and Dropouts. The higher the considered number of initial absolute interactions per user, the more the users interact with the *Problem* component, while *Discussion Forum* is generally scarcely used.

		Absolute Interactions											
		5				50				100			
		Completers		Dropouts		Completers		Dropouts		Completers		Dropouts	
Feature Group	Feature	%	AVG	%	AVG	%	AVG	%	AVG	%	AVG	%	AVG
Course Navigation	Tab Selected	5	0.19	4	0.06	11	5.47	12	1.55	9	9.25	1	2.09
	Next Tab Click	4	0.14	3	0.06	19	9.11	17	2.30	15	14.42	14	3.05
	Previous Tab Click					1	0.48	1	0.11	1	0.95	1	0.19
Video	Video Loaded	21	0.77	20	0.33	10	4.85	11	1.53	10	9.71	11	2.27
	Video Played	9	0.34	8	0.14	10	4.88	11	1.44	11	11.16	11	2.40
	Video Paused	2	0.08	2	0.03	6	3.04	7	0.89	7	7.14	7	1.46
	Video Stopped					4	1.77	4	0.53	4	4.02	4	0.83
	Video Position Changed									2	1.90	2	0.38
Problem	Problem Check					4	1.72	2	0.25	7	6.69	4	0.79
	Problem Graded					3	1.67	2	0.24	7	6.63	4	0.79
Discussion Forum	Visualize Thread					4	2.07	3	0.41	3	3.32	3	0.56
	Visualize Forum Page					2	0.88	2	0.21	1	1.19	1	0.24
Main Page Links	Home	35	1.31	37	0.61	6	2.76	7	1.01	4	3.66	5	1.16
	Progress	5	0.18	3	0.05	2	0.78	1	0.17	2	1.86	13	0.29

up to a couple of months during which the course is available for enrollment only, and a second time window lasting from the MOOC's official start date until its official end date (See Section 3.1.2). Before the beginning of the course, users have only the possibility to visit the MOOCs main page, which contains a description of the course and its topics, but do not have access to lecture notes and problems, which become available only during the second time window.
This restriction on user's ability to interact with the system causes the features derived from it to be unable to distinguish between Completers and Dropouts accurately. Furthermore, we verify that 8552 Dropouts (39% of all users) from *MOOCC1* and 4436 (almost 43% of all users) from *MOOCC2* only interact once (Enrollment action) with the platform.
In this scenario, the classifier might correctly classify the users with only one interaction with the system as dropouts, but it will misclassify dropouts that interacted more than once with the system as completers. We speculate that these dropouts with a single interaction with the system further worsen the performance of our classifier.
By completely removing these users and re-running the experiments for the **Days** approach, the accuracy of the predictions increases substantially. These new results, depicted in Figure 5.4, are more in line with the ones from the **Absolute** approach. When we consider only the first day after users' initial interaction we obtain an accuracy of almost 0.9 for both MOOCs.
These new results indicate that there are strong differences in the engagement style of Completers and Dropouts, which are detectable already during the initial phase of the courses. This conclusion was also foreshadowed by the results of the **Absolute** approach, in which five initial absolute interactions per user already yielded an accuracy of 0.8 for both MOOCs.

5.5 Limitations of system-to-system dropout detection

When we conduct the multisystem dropout prediction experiment *MIX*, we use half of the users from Curtin University and half of the users from Universidad Galileo for training, and we evaluate the classifier by predicting

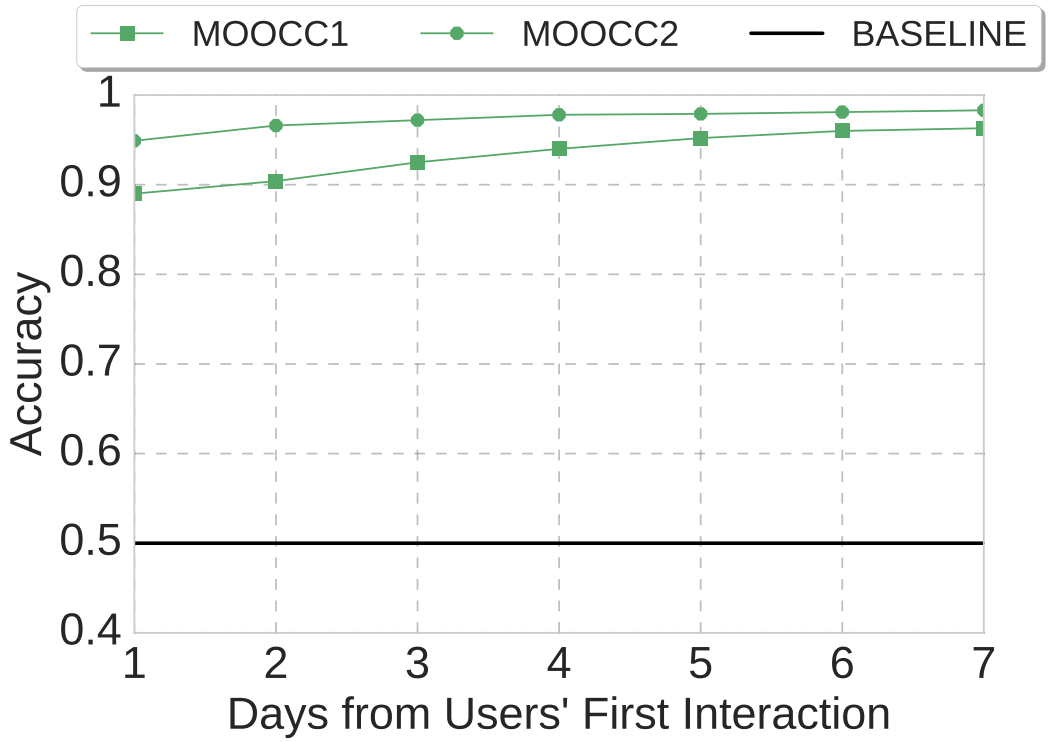


Figure 5.4: Curtin University Days approach without enrollment actions. On the x-axis, we list the number of days after the first interaction of users. The y-axis indicates the accuracy with the baseline plotted in black at 0.5. We plot the MOOCs in green as both their accuracies are increasing the more days from users' first interactions we analyze. Compared to the results of Figure 4.2b, we can see how dropping users with only one interaction (the Enrollment action) increases the accuracy of the prediction for both MOOCs, while considering only the first day after the initial interaction of users yields an accuracy of 0.89 or higher for both MOOCs.

the class of the remaining users. With this setting, we always obtain higher accuracy than the ones of the system-to-system experiments *Curtin on Galileo* and *Galileo on Curtin*.
As with the *MIX* experiment, the accuracy of the **Days** approach is relatively lower when compared to those of the other approaches. We suppose that this is due to the Enrollment actions that are part of the logs of Curtin University.

By removing users with only the Enrollment actions and re-running the experiments, we obtain the results shown in Figure 5.5. Removing users

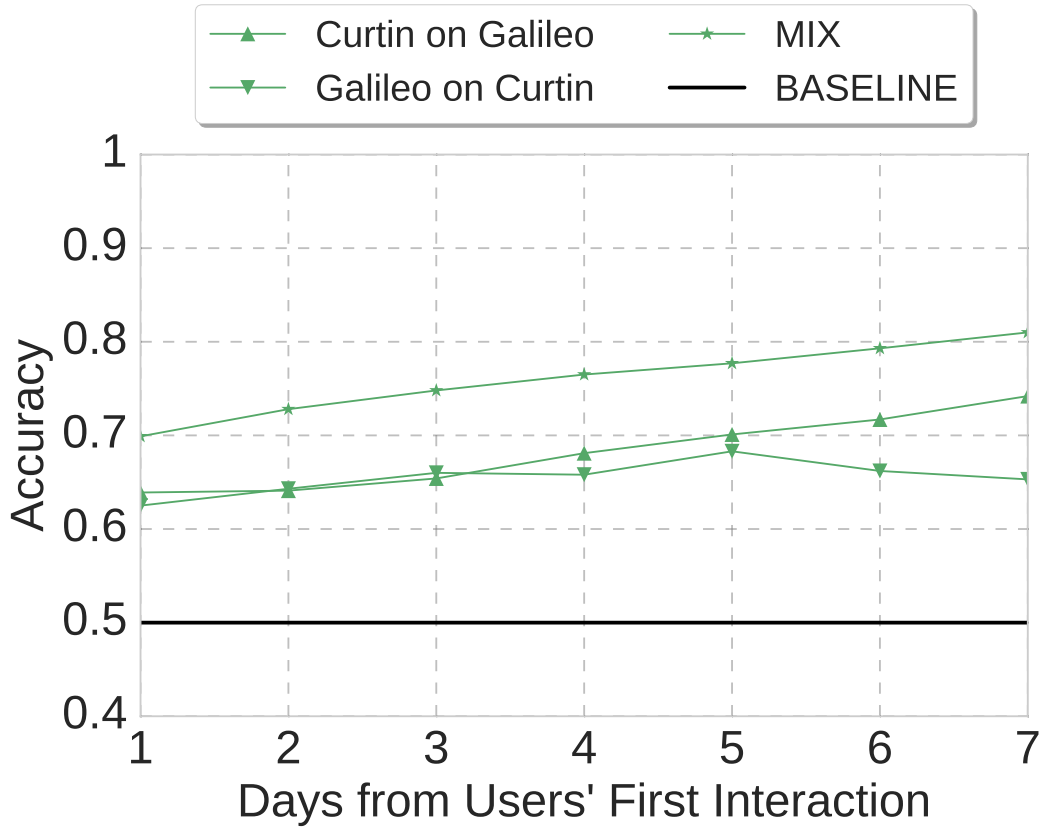


Figure 5.5: Multisystem Days approach without enrollment actions. The x-axis indicates the number of considered days (up to seven) from users first interaction. The y-axis indicates the accuracy; it is bounded between 0.4 and 1, and the horizontal black line at 0.5 is the baseline. All profiles are plotted in green as they are all either increasing or stable the higher the number of considered days from the user first interaction is. Compared to the results from Figure 4.3b there is an increase in the accuracy for all three settings, with *MIX* benefiting the most from the removal of Enrollment actions.

with only the Enrollment action increases the accuracy of the multisystem dropouts prediction experiment *MIX* too. The accuracy constantly improves with the number of days from the users' first interaction that we consider. The system-to-system experiments also benefit from the removal of these

users. The accuracy increases with the number of considered days with only a slight decrease for six and seven days for the *Galileo on Curtin* experiment. Instead, for the *Curtin on Galileo* experiment, the accuracy always increases. Generally, the results we obtain for the system-to-system experiments with the other approaches are variable.
One potential explanation for this observation is the delayed start of some MOOCs of Universidad Galileo (see Section 5.1) that introduces noise, which leads to a worsening of the accuracy. The main approaches affected by this situation include both the **Absolute** and **Days** ones.
Another possible explanation is a too small number of features that we use in the multisystem and system-to-system experiments. The set of features that we use consists of the seven features common to both systems. This restricted set of features seems to be too small to correctly describe the classes of users of the two completely different systems that we analyze in this work.
We see this in the generally lower accuracy of the system-to-system dropouts prediction experiments *Curtin on Galileo* and *Galileo on Curtin*. On the other hand, this limitation does not interfere with the multisystem dropouts prediction experiments *MIX*. For the *MIX* experiment, we create the training set by selecting 80% of users of Universidad Galileo's MOOCs and 80% of users of MOOCs of Curtin University. The remaining 20% of users of MOOCs of Universidad Galileo and 20% of users of Curtin University's MOOCs constitute the test set.
Therefore, we train the classifier with information about both systems, which results in a more precise classification of the users in the test set. This yields higher accuracy for the MIX experiment in all the approaches considered.
Furthermore, the dimensions of the training and test sets, particularly for the *Curtin on Galileo* and *Galileo on Curtin* experiments, could cause potential issues.
With the system-to-system experiments, we merge the MOOCs from each system into the dataset *Galileo* and *Curtin*. Then, we use one dataset as a training set and the other as a test set. Particularly, *Galileo* dataset includes 1,164 Completers and 1,993 Dropouts for a total of 3,157 users, while *Curtin* counts for 32,316 users, of which 1,708 are Completers and the remaining 30,608 are Dropouts.
When we balance the number of users of each class (see 3.2.3) we obtain a total of 3,986 in the *Galileo* dataset and 61,216 in the *Curtin* one, in both

cases, half are Completers and half are Dropouts.
Thus, in the *Galileo on Curtin* experiment, we use only 6% of the users for training and predict the class of the remaining 94%. Even by removing the users with only one interaction from Curtin University's logs only increases the number of users in the training set up to 10%.
Therefore, the number of users in the training set is significantly lower than the number of users in the test set, and we train the classifier on simply too few examples.
However, when the percentages are inverted, as is the case of the *Curtin on Galileo* experiment, the accuracy results are still generally low.
We conclude that there are strong differences between users from the different systems that our features are not capturing. As previously mentioned, the used set of features is likely too limited for these system-to-system dropout prediction experiments.

5.6 Averaged aspects of sessions and their importance

For each experiment, we attribute the three features with the highest scores as *high-scoring* and the remaining features as *low-scoring*. The group of low-scoring features for the **Absolute**, **Percentage** and **Scaled Time** approaches includes the features *Days*, *Active Days Requests* and *Sessions*.
The MOOCs of Curtin University are self-paced and, therefore, offer more flexibility to the users; as soon as the whole material is available, enrolled users start to engage and can, theoretically, complete a MOOC even within just a couple of days.
On the other hand, the MOOCs of Universidad Galileo include (weekly) assignments and projects that are subject to fixed deadlines. Therefore, even if users can still engage at their own pace these MOOCs still impose a predefined schedule.
To examine this situation further, we report the number of average active days per user with respect to the Completers and Dropouts class for all MOOCs from both systems in Table 5.2. Completers of Curtin University's MOOCs are active on average between 11.50 and 12.77 days, while the

Dropouts interact with the system between 2.15 and 2.24 days.
For MOOCs *CA* and *WTEA* of Universidad Galileo, Completers are active between 5.70 and 6.77 days and the Dropouts between 1.83 and 2.23 days. These MOOCs are the only two among those of Universidad Galileo with a duration of twelve days only, while the others lasted for eight weeks (see Section 5.1).
Users of remaining MOOCs of this system are active for more days. Specifically, Completers are active on average between 16.52 for (*DITV*) and 33.09 (*AND*) days. Dropouts interact on average between 4.19 (*UE*) and 6.11 (*AND*) days.
Therefore, users of MOOCs of Universidad Galileo are always active for more days than those of Curtin University's MOOCs. This is true for both, Completers and Dropouts. This can be a reason for *Days* to always be one of

Table 5.2: Per-class average active days. For each system (indicated in the first column) and each of its MOOCs, the averaged numbers of active days for Completers and Dropouts are indicated. The average is calculated with respect to the total number of users of the particular class of the MOOC. An active day for a user represents a day during which the user interacts at least once with the system. Users from Galileo Universidad's MOOCs have, on average, more active days than those from Curtin University's MOOCs. The only exceptions are the MOOCs *CA* and *WTEA*, which have the lowest average number of active days for both classes.

System	MOOC	Completers	Dropouts
Universidad Galileo	AND	33.09	6.11
	AEL	20.93	5.19
	CA	6.77	1.83
	CBL	21.33	5.22
	CM	28.57	5.23
	DITV	16.52	4.26
	EL	23.81	4.55
	ME	20.06	5.29
	UE	18.89	4.19
	WTEA	5.70	2.23
	WTC	23.02	4.53
Curtin University	MOOCC1	11.50	2.24
	MOOCC2	12.77	2.15

the low-scoring features when we classify users from such different systems. As *Active Days Requests* is a feature closely related to *Days*, it is, consequently, a poor measure in such a scenario too.
The feature *Sessions* also belongs to the group of low-scoring features. However, the features that are strongly based on the concept of sessions such as *Timespan Clicks*, *Session Length*, *Session Requests* and *Active Time* are mostly among the high-scoring ones.
This indicates that the aspects that describe the behavior of the users during a session relate closely to the class users belong to rather than the simple total number of sessions that they have. This is particularly evident from the score obtained for the multisystem dropout prediction experiment *MIX*.
On the other hand, for the system-to-system dropout prediction experiments *Curtin on Galileo* and *Galileo on Curtin* the features that describe the behavior during a session are not necessarily the ones with the highest scores as in the multisystem experiments. This happens for the experiment *Curtin on Galileo* with the **Scaled Time** approach and in the experiment *Galileo on Curtin* with the **Days** approach.
These situations can be interpreted as a direct consequence of the settings imposed on these experiments, in which the features that we constructed have not always correctly identified Completers and Dropouts of the two systems.
Therefore, it is possible that Completers and Dropouts of the MOOCs of

Table 5.3: Feature Scores of the Days multisystem and system-to-system experiments without enrollments. In the first column, we list the considered features for the dropout prediction experiments. For each of our experiments, we report the scores for 1, 3, 5 and 7 days after the first interaction of users. For each column, we highlight the three features with the highest score for that particular setting in green. *Session Length*, *Timespan Clicks*, *Requests* and *Active Time* are the features with highest scores, while *Days*, *Active Days Requests* and *Sessions* always have low scores among the various experiments.

	Curtin on Galileo				Galileo on Curtin				MIX			
Days	1	3	5	7	1	3	5	7	1	3	5	7
Days	9.05	30.85	47.30	43.70	14.15	31.1	22.55	19.4	10.70	31.96	43.74	43.65
Active Days Requests	54.25	56.70	59.45	81.80	59.15	109.6	103.55	98.2	53.74	60.70	62.53	84.12
Session Length	126.20	133.80	110.30	123.60	118.55	123.40	118.5	136.05	120.15	133.97	112.8	129.93
Timespan Clicks	179.50	153.10	147.75	108.5	110.1	115.15	101.05	117.25	168.6	150.41	143.94	108.27
Requests	115.45	121.80	123.8	141.15	116.45	104.60	118.65	107.75	113.75	126.49	130.25	136.03
Sessions	34.2	34.90	28.2	36	19.4	16.20	39.05	28.3	37.11	31.29	33.37	37.80
Active Time	112.45	114.90	124.55	120.3	136.6	117.15	93.5	91.35	128.44	118.02	125.47	120.770

Universidad Galileo and those of Curtin University's MOOCs exhibit a different engagement style.
Otherwise, when we train the classifiers using examples from both systems, as can be seen in the multisystem experiments on the *MIX* dataset, we see a precise distinction between the two groups of features, except with the **Days** approach.
Particularly, for the multisystem dropouts prediction experiment *MIX*, the feature *Timespan Clicks* generally has a higher score than *Session Length*. This is also true for the system-to-system dropout prediction experiment *Curtin on Galileo*, while for *Galileo on Curtin* the two groups of features are less distinguishable, with up to five different features that belong to the high-scoring ones.
This could be a consequence of the noise that the Enrolment actions introduce (see Section 5.3). Re-running these experiments for the **Days** approach by removing the Enrolment actions for the MOOCs of Curtin University again, provides the results outlined in Table 5.3. In this case, the distinction between high- and low-scoring features is clearer, with the features *Days*, *Active Days Requests* and *Sessions* that always belonging to the low-scoring group.
We obtained similar results also with other approaches, which confirms that the distribution of interactions of users of MOOCs of Curtin University is negatively skewed due to the Enrollment actions included in the logs of this system.

6 Conclusion

In this thesis, we experimented with dropout detection for MOOCs in single and multisystem scenarios. In particular, we analyzed MOOCs of two different systems: Universidad Galileo and Curtin University.
These systems and their MOOCs differed concerning the number of enrolled users, topics, intended audience and structure of the courses. Besides, MOOCs of Universidad Galileo complied with a predefined timescale, with deadlines for assignments and project submissions.
On the other hand, MOOCs of Curtin University allowed the users the freedom to proceed through the course at their own speed. MOOCs of this system followed a self-paced structure, with the entire course material available from the very beginning of the course.
Initially, we predicted dropouts for each MOOC of each system independently. Second, we merged all the MOOCs of the same domain in a unique dataset and obtained two datasets, one for each system. We used these datasets to predict dropouts in system-to-system and multisystem settings. While doing so, we also analyzed which of the features of our model influenced the accuracy of the prediction the most, and we ranked them accordingly.
Throughout the various experiments, we constructed features using different approaches. In particular, we considered up to the first one hundred initial absolute interactions per user (**Absolute**), up to the first seven days after the first interaction of users (**Days**), a varying percentage of total interactions per user (**Percentage**) and a varying percentage of total active time per user (**Scaled Time**).

Results of MOOCs of Universidad Galileo exhibited a high level of accuracy for the **Percentage** and **Scaled Time** approaches. In these cases, the accuracy improved in proportion to the considered percentage of interactions per user and the analyzed percentage of active time per user.

The accuracy of the **Absolute** and **Days** settings were generally lower, even though some analyzed MOOCs showed an increasing trend in accuracy as we used higher numbers of considered initial absolute interactions per user and days after first interaction of users. When we further investigated these results, we discovered two potential explanations.
First, some MOOCs presented an initial phase (of up to seventeen days) with few interactions, possibly due to an actual later start of the courses and/or unavailability of materials to the users.
Second, despite an official duration of eight weeks, the majority of the interactions within some MOOCs were concentrated on shorter periods of consecutive days. This shorter length led to a similar distribution of the interactions of the classes and eventually resulted in features characterized by low variance, which in turn worsened the performance of our classifier.

For MOOCs of Curtin University, we obtained higher accuracy when compared with those of Universidad Galileo's MOOCs. We noticed such higher accuracy regardless of the chosen approach we used to construct the features. We identified a couple of possible explanations for the higher accuracy of the predictions on MOOCs of Curtin University.
First, the detailed logs of Curtin University allowed us to construct more features than the one we created for Universidad Galileo's MOOCs.
Besides, we also noticed that user interactions were strongly concentrated on a small set of tools, which included Video, Course Navigation, and Problem components. However, users interacted with the Forum component seldom. We believe this being more a sign of uncomplicated and easy courses or topics, rather than a general observation valid for other MOOCs and settings.
Second, in a self-paced setting, users do not have to comply with a fixed schedule and can interact with the MOOCs at their own speed. This autonomy seems to relate more to the final outcome of users than in the case of MOOCs with a course set over a predefined number of calendar days.
We obtained low accuracy only with the **Days** setting. However, we noticed that the logs of the two MOOCs included, on average, more than 40% of users who enrolled and never interacted any further with the system. We verified that, by not considering these users, the accuracy of the prediction increased and the results were in line with those obtained with the other approaches for this system.

The system-to-system dropout prediction accuracy was not particularly high

for all proposed approaches.
Such a result is likely due to the small set of features we used for these experiments. We can conclude that the seven features we used were too few to distinguish Dropouts from Completers among the two systems correctly. Moreover, MOOCs from Universidad Galileo had a fixed schedule, including deadlines for assignments and projects submissions, while the self-paced MOOCs of Curtin University let users more freedom on their learning schedule.
As a consequence, our classifier was not able to correctly identify the two classes of users on one system when the training set consisted only of users from the other system.

However, when we merged the datasets from Universidad Galileo together with the datasets of Curtin University and attempted a dropout prediction on a multisystem scenario, the accuracy increased for all approaches.
The only exception was the **Days** approach that suffered the most from users with only a single interaction, namely the enrolling action. By removing these users, the accuracy was comparable to the ones of the other settings, meaning that dropout detection on a multisystem setting is attainable and yields high accuracy.

Such result was also confirmed by the scores of the features that we obtained using Boosted Decision Trees. In particular, we were able to split the features into two groups clearly.
The first group of features always had high scores among the different experiments, meaning that the classifier could classify users with high accuracy by using these features. The second group of features had low scores, signifying that these features were not helpful to distinguish between Dropouts and Completers.
Overall, we found that the features with the highest scores were those that describe averaged aspects of sessions; *Session Length*, *Active Time* and *Timespan Clicks*. The total number of interactions users had with the MOOCs (*Requests*) was also among the high-scoring features.
Besides, we verified that certain features always got the lowest scores in almost every setting and experiments. This group of features included *Days*, *Active Days Requests* and *Sessions*. We also noticed that this division in high- and low-scoring features was clearer for predictions in the multisystem setting than in the system-to-system setting.

Our results represent the first step towards the early detection of dropouts in a multisystem setting.
As a first observation, we saw that in MOOCs with a fixed schedule Dropouts and Completers approached the first part of the course with similar behaviors. Therefore it is harder to detect dropouts during the initial days after the start of a MOOC. After the first deadline to hand in an assignment or project part of the users will, eventually, stop engaging with the MOOCs and, therefore, is easier to detect dropouts at this point.
We did not investigate reasons for users to abandon a course, but we speculate that an initial negative grade would be a possible reason that could have led to users dropping out of a course.
Furthermore, the fixed schedule forces users to engage with MOOCs during its whole duration. Therefore, when personal commitments and priorities of users outside of the MOOCs become demanding, it is hard for registered users to keep up with the course and, therefore, results in users dropping out. Besides, a delayed start of the course might lower the level of interest of users.
The idea of a self-paced MOOC and its availability on a wide-select of available platforms seems to appeal to many potential enrollees.
However, a high number of registered users does not signify more engaged users. Indeed, it is the self-paced setting of the courses, which have led to higher dropout rates. A possible consideration could be that a MOOC requires a proper organization and structure of the learning material to enhance the likelihood of a successful completion of the course.
The self-paced environment that many MOOCs are offering seems to be counterproductive for a high number of users that tend to interact less and are more prone to dropout. This can be evidenced by the fact, that some of these users subscribe to MOOCs without further interacting after the start of the course.
Therefore, it is easier to detect dropouts in these MOOCs even by analyzing only the interactions that take place at the beginning of the course.
The intrinsic differences among systems and domains can be overcome by the correct identification of the features with the highest scores. We discovered that some features always had high scores for both of the systems we analyzed. Thus, users' way of interacting in self-paced and synchronous MOOCs share some similarities.
However, these similarities are too few to correctly distinguish Completers

from Dropouts in system-to-system experiments, which is an application of our classifier that yielded mixed results. On the other hand, by training our classifier with information from both systems the accuracy of our prediction increased.

We identified the small number of features we used as one of the limitations of our model. Therefore, naturally, the next step should be to do a more in-depth analysis of the events and tools each system offers, with the goal of incorporating new features into our model.
Evaluation of these new features through ranking would help us to verify which features are more valuable to detect dropouts on a system. With this process, we would obtain a refinement of the set of features to enhance the accuracy of the predictions.
To deepen the gained insights and verify whether the obtained results hold at a general level, we also aim to incorporate new systems and domains into our model. It is only by finding general rules across various systems that it would be possible to improve the accuracy of our classifier and, consequently, lower the general trend of high dropout rates.
As this study has underlined, the need for a homogenous prediction model is strong. Early dropout detection stands out as a requirement for any system and domain and should, therefore, be addressed by a flexible homogeneous model.

Bibliography

[1] Knowledge discovery and data mining: towards a unifying framework., vol. 96 (1996)

[2] Aggarwal, C.C.: Data mining: the textbook. Springer (2015)

[3] Anderson, A., Huttenlocher, D., Kleinberg, J., Leskovec, J.: Engaging with massive online courses. In: Proceedings of the 23rd international conference on World wide web. pp. 687–698. ACM (2014)

[4] Balakrishnan, G., Coetzee, D.: Predicting student retention in massive open online courses using hidden markov models. Electrical Engineering and Computer Sciences University of California at Berkeley (2013)

[5] Belanger, Y., Thornton, J.: Bioelectricity: A quantitative approach duke university's first mooc (2013)

[6] Bishop, C.: Bishop pattern recognition and machine learning (2001)

[7] Boldi, P., Bonchi, F., Castillo, C., Donato, D., Gionis, A., Vigna, S.: The query-flow graph: model and applications. In: Proceedings of the 17th ACM conference on Information and knowledge management. pp. 609–618. ACM (2008)

[8] Boyer, S., Veeramachaneni, K.: Transfer learning for predictive models in massive open online courses. In: International Conference on Artificial Intelligence in Education. pp. 54–63. Springer (2015)

[9] Breiman, L.: Bagging predictors. Machine learning 24(2), 123–140 (1996)

[10] Breiman, L.: Random forests. Machine learning 45(1), 5–32 (2001)

[11] Brinton, C.G., Chiang, M., Jain, S., Lam, H., Liu, Z., Wong, F.M.F.: Learning about social learning in moocs: From statistical analysis to generative model. IEEE transactions on Learning Technologies 7(4), 346–359 (2014)

[12] Chapelle, O., Keerthi, S.S.: Multi-class feature selection with support vector machines. In: Proceedings of the American statistical association (2008)

[13] Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: synthetic minority over-sampling technique. Journal of artificial intelligence research 16, 321–357 (2002)

[14] Clow, D.: Moocs and the funnel of participation. In: Proceedings of the Third International Conference on Learning Analytics and Knowledge. pp. 185–189. ACM (2013)

[15] Coffrin, C., Corrin, L., de Barba, P., Kennedy, G.: Visualizing patterns of student engagement and performance in moocs. In: Proceedings of the fourth international conference on learning analytics and knowledge. pp. 83–92. ACM (2014)

[16] Cook, M.: State of the mooc 2016: A year of massive landscape change for massive open online courses (2016)

[17] Daniel, J.: Making sense of moocs: Musings in a maze of myth, paradox and possibility. Journal of interactive Media in education 2012(3) (2012)

[18] Forman, G., Scholz, M., Rajaram, S.: Feature shaping for linear svm classifiers. In: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 299–308. ACM (2009)

[19] Friedman, T.: Come the revolution. The New York Times 15 (2012)

[20] Gee, S.: Mitx-the fallout rate. Message posted 16 (2012)

[21] Grandvalet, Y., Canu, S.: Adaptive scaling for feature selection in svms. In: NIPS. pp. 553–560 (2002)

[22] Guetl, C., Chang, V., Hernández Rizzardini, R., Morales, M.: Must we be concerned with the massive drop-outs in mooc? an attrition analysis of open courses. In: Proceedings of the International Conference Interactive Collaborative Learning, ICL2014 (2014)

[23] Guo, X., Yin, Y., Dong, C., Yang, G., Zhou, G.: On the class imbalance problem. In: Natural Computation, 2008. ICNC'08. Fourth International Conference on. vol. 4, pp. 192–201. IEEE (2008)

[24] Guruler, H., Istanbullu, A., Karahasan, M.: A new student performance analysing system using knowledge discovery in higher educational databases. Computers & Education 55(1), 247–254 (2010)

[25] Gütl, C., Rizzardini, R.H., Chang, V., Morales, M.: Attrition in mooc: Lessons learned from drop-out students. In: International Workshop on Learning Technology for Education in Cloud. pp. 37–48. Springer (2014)

[26] He, H., Garcia, E.A.: Learning from imbalanced data. IEEE Transactions on knowledge and data engineering 21(9), 1263–1284 (2009)

[27] Hew, K.F., Cheung, W.S.: Students' and instructors' use of massive open online courses (moocs): Motivations and challenges. Educational Research Review 12, 45–58 (2014)

[28] Hill, P.: Emerging student patterns in moocs: A (revised) graphical view (2013)

[29] Ho, T.K.: Random decision forests. In: Document Analysis and Recognition, 1995., Proceedings of the Third International Conference on. vol. 1, pp. 278–282. IEEE (1995)

[30] Hollands, F.M., Tirthali, D.: Moocs: Expectations and reality. full report. Online Submission (2014)

[31] Hsu, C.W., Chang, C.C., Lin, C.J., et al.: A practical guide to support vector classification (2003)

[32] Japkowicz, N., et al.: Learning from imbalanced data sets: a comparison of various strategies. In: AAAI workshop on learning from imbalanced data sets. vol. 68, pp. 10–15. Menlo Park, CA (2000)

[33] Jiang, S., Williams, A., Schenke, K., Warschauer, M., O'dowd, D.: Predicting mooc performance with week 1 behavior. In: Educational Data Mining 2014 (2014)

[34] Jordan, K.: Initial trends in enrolment and completion of massive open online courses. The International Review of Research in Open and Distributed Learning 15(1) (2014)

[35] Kearns, M.: Thoughts on hypothesis boosting. Unpublished manuscript 45, 105 (1988)

[36] Kearns, M., Valiant, L.: Cryptographic limitations on learning boolean formulae and finite automata. Journal of the ACM (JACM) 41(1), 67–95 (1994)

[37] Khalil, H., Ebner, M.: "how satisfied are you with your mooc?"-a research study on interaction in huge online courses. In: EdMedia: World Conference on Educational Media and Technology. pp. 830–839. Association for the Advancement of Computing in Education (AACE) (2013)

[38] Khalil, H., Ebner, M.: Moocs completion rates and possible methods to improve retention-a literature review. In: World Conference on Educational Multimedia, Hypermedia and Telecommunications. vol. 1, pp. 1305–1313 (2014)

[39] Kizilcec, R.F., Piech, C., Schneider, E.: Deconstructing disengagement: analyzing learner subpopulations in massive open online courses. In: Proceedings of the third international conference on learning analytics and knowledge. pp. 170–179. ACM (2013)

[40] Koller, D., Ng, A., Do, C., Chen, Z.: Retention and intention in massive open online courses: In depth. Educause Review 48(3), 62–63 (2013)

[41] Koutropoulos, A., Gallagher, M.S., Abajian, S.C., de Waard, I., Hogue, R.J., Keskin, N.Ö., Rodriguez, C.O.: Emotive vocabulary in moocs: Context & participant retention. European Journal of Open, Distance and E-Learning 15(1) (2012)

[42] Krzysztof, J.C., Pedrycz, W., Swiniarski, R., Kurgan, L.: Data mining: A knowledge discovery approach. Springer (2007)

[43] Levy, Y.: Comparing dropouts and persistence in e-learning courses. Computers & education 48(2), 185–204 (2007)

[44] Margaryan, A., Bianco, M., Littlejohn, A.: Instructional quality of massive open online courses (moocs). Computers & Education 80, 77–83 (2015)

[45] McKenna, L.: The big idea that can revolutionize higher education:'mooc.'. The Atlantic 5 (2012)

[46] Meyer, R.: What it's like to teach a mooc (and what the heck'sa mooc?). The Atlantic (2012)

[47] Mitchell, T.M., et al.: Machine learning. wcb (1997)

[48] Oakley, B.: Why virtual classes can be better than real ones (2015)

[49] Onah, D.F., Sinclair, J., Boyatt, R.: Dropout rates of massive open online courses: behavioural patterns. EDULEARN14 Proceedings pp. 5825–5834 (2014)

[50] Palloff, R.M., Pratt, K.: Building online learning communities: Effective strategies for the virtual classroom. John Wiley & Sons (2007)

[51] Pappano, L.: The year of the mooc. The New York Times 2(12), 2012 (2012)

[52] Piwowarski, B., Zaragoza, H.: Predictive user click models based on click-through history. In: Proceedings of the sixteenth ACM conference on Conference on information and knowledge management. pp. 175–182. ACM (2007)

[53] Rice, J.: What i learned in mooc. College Composition and Communication 64(4), 695–703 (2013)

[54] Ryana, P., Williamsb, A.: Moocs or mocs?-revisiting the commercialisation of moocs (????)

[55] Schapire, R.E.: The strength of weak learnability. Machine learning 5(2), 197–227 (1990)

[56] Shah, D.: By the numbers: Moocs in 2016. retrieved april 2, 2017 (2016)

[57] Smola, A.J., Schölkopf, B.: Learning with kernels. Citeseer (1998)

[58] Soman, K., Loganathan, R., Ajay, V.: Machine learning with SVM and other kernel methods. PHI Learning Pvt. Ltd. (2009)

[59] Steinwart, I., Christmann, A.: Support vector machines. Springer Science & Business Media (2008)

[60] Suykens, J.A., Vandewalle, J.: Least squares support vector machine classifiers. Neural processing letters 9(3), 293–300 (1999)

[61] Taneja, S., Goel, A.: Mooc providers and their strategies. International Journal of Computer Science and Mobile Computing 3(5), 222–228 (2014)

[62] Teevan, J., Adar, E., Jones, R., Potts, M.A.: Information re-retrieval: repeat queries in yahoo's logs. In: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval. pp. 151–158. ACM (2007)

[63] Teusner, R., Richly, K., Staubitz, T., Renz, J.: Enhancing content between iterations of a mooc–effects on key metrics. Proceedings Papers p. 147 (2015)

[64] Vapnik, V.: The nature of statistical learning theory. Springer Science & Business Media (2013)

[65] Vapnik, V.N., Chervonenkis, A.J.: Theory of pattern recognition (1974)

[66] Vitiello, M., Gütl, C., Amado-Salvatierra, H.R., Hernández, R.: Mooc learner behaviour: Attrition and retention analysis and prediction based on 11 courses on the telescope platform. In: International Workshop on Learning Technology for Education in Cloud. pp. 99–109. Springer, Cham (2017)

[67] Vitiello, M., Walk, S., Chang, V., Hernandez, R., Helic, D., Guetl, C.: Mooc dropouts: A multi-system classifier. In: European Conference on Technology Enhanced Learning. pp. 300–314. Springer (2017)

[68] Vitiello, M., Walk, S., Hernández, R., Helic, D., Gütl, C.: Classifying students to improve mooc dropout rates. Research Track p. 501 (????)

[69] Watters, A.: Top ed-tech trends of 2012: Moocs. Hack Education (2012)

[70] Watters, A.: Mooc mania: Debunking the hype around massive open online courses. The digital shift 13 (2013)

[71] Weiss, G.M.: Mining with rarity: a unifying framework. ACM SIGKDD Explorations Newsletter 6(1), 7–19 (2004)

[72] Xing, W., Chen, X., Stein, J., Marcinkowski, M.: Temporal predication of dropouts in moocs: Reaching the low hanging fruit through stacking generalization. Computers in Human Behavior 58, 119–129 (2016)

[73] Yang, D., Sinha, T., Adamson, D., Rosé, C.P.: Turn on, tune in, drop out: Anticipating student dropouts in massive open online courses. In: Proceedings of the 2013 NIPS Data-driven education workshop. vol. 11, p. 14 (2013)

[74] Yuan, L., Powell, S., CETIS, J., et al.: Moocs and open education: Implications for higher education (2013)