Universidad Autónoma de Madrid
Superior Polythecnic School

Master's Degree in ICT Research and Innovation, Computational Intelligence Track

Master Thesis

IDENTIFYING CHEATING USERS IN ONLINE COURSES

Author: Eng. Vicenzo Abichequer Sangalli
Tutors: Dr. Gonzalo Martínez-Muñoz
Dra. Estrella Pulido Cañabate

July 2020

IDENTIFYING CHEATING USERS IN ONLINE COURSES

Author:
Eng. Vicenzo Abichequer Sangalli
Tutors:
Dr. Gonzalo Martínez-Muñoz
Dra. Estrella Pulido Cañabate

Machine Learning Group
Informatics Engineering Department
Superior Polytechnic School
Universidad Autónoma de Madrid
July 2020

Abstract

Resumen

Los estudiantes interactúan con los cursos en línea principalmente de dos maneras: revisando los materiales del curso y resolviendo ejercicios. Sin embargo, hay casos en los que el comportamiento de los estudiantes difiere y tiende a centrarse más en resolver ejercicios sin mirar los materiales del curso. Este tipo de interacción podría ser indicativo de un comportamiento poco ético, como los estudiantes que colaboran compartiendo respuestas entre ellos o cuentas falsas que usan los estudiantes para obtener las respuestas correctas para los ejercicios.
En este trabajo, proponemos varias métricas para identificar estos dos tipos de trampas basadas en eventos concurrentes y medidas de interacción con el curso. Del conjunto de cuentas en el curso, los pares de cuentas que resuelven ejercicios muy cercanos en el tiempo se consideran cuentas potencialmente colaboradoras.
Las métricas propuestas se calculan para estos pares de cuentas y la agrupación de K-means se usa para separar pares de estudiantes reales que colaboran con respecto a los estudiantes que usan cuentas falsas para obtener las respuestas correctas a los ejercicios. Se logra una precisión de generalización superior al 95 % para clasificar estos tipos de trampas mediante el uso de una máquina de vectores de soporte (SVM)

Palabras llave

machine learning, agrupamiento, MOOCs, educación, trampas, CAMEO

Abstract

Abstract

Students interact with online courses mainly in two ways: by reviewing the course materials and by solving exercises. However, there are cases in which student behaviour differs and tends to become more focused on solving exercises without looking at course materials. This type of interaction could be an indicative of unethical behavior, such as students who collaborate by sharing answers with one another or fake accounts that are used by students to obtain the correct answers for exercises.

In this work, we propose several metrics to identify these two types of cheating based on co-occurring events and measures of interaction with the course. From the pool of accounts in the course, the pairs of accounts that solve exercises very close in time are considered to be potential collaborating accounts.

The proposed metrics are computed for these pairs of accounts and K-means clustering is used to separate pairs of real students who collaborate with respect to students who use fake accounts to harvest the correct answers to exercises. A generalization accuracy over 95% to classify these types of cheating is achieved by using a Support Vector Machine (SVM).

Key words

machine learning, clustering, MOOCs, education, cheating, CAMEO

"*If I throw a manuscript into the forest, did I just submit it to nature?*"

Alexander Lin

Acknowledgements

First and foremost, I'd like to thank my wife, Ana. She's been the cornerstone of my life for quite some time now and we've accomplished so much together. We've gone far and beyond in this world and I just wish to continue this journey alongside her. This work is just another step towards our future and the support she has always given me is what helps me to keep moving forward. *Meu anjo, sem ti não há nós e sem nós não há sentido. Eu te amo além da matéria.*

Second, I'd like to thank my best and dearest friends: Pedro and Tábata. They are, together with Ana and my pets, my family. They have given us all the support one could wish in this world: you guys are an inspiration for us and we will be forever grateful for all the love and care you've given us throughout all the time we've been in Europe and when we were still in Brazil. Me and Ana love you very, very much. *Muito obrigado, gente, vocês são inacreditáveis e os melhores padrinhos de casamento que alguém pode ter!*

I'd also like to thank my advisors, Gonzalo and Estrella for giving me an opportunity to work in Spain and start my Master's degree. They've helped me a lot in this project and it would not have been possible without you. *Gracias!*

Finally, the authors acknowledge financial support from the European Regional Development Fund and from the Spanish Ministry of Economy, Industry, and Competitiveness - State Research Agency, project TIN2016-76406-P (AEI/FEDER, UE). We would also like to thank the Universidad Autónoma de Madrid for providing us with the data from the MOOC.

Glossary

- **EDM**: Educational Data Mining
- **MOOC**: Massive Open Online Course
- **SVM**: Support Vector Machine
- **CV**: Cross-Validation

Contents

List of figures

List of tables

1
Introduction

When talking about online education today, it is impossible not to mention Massive Open Online courses (MOOCs). According to the definition by Gaebel (1), a MOOC is characterized by 5 premisses: it is online, no formal entry is required, there are no participation limits, they are free of charge and do not earn credits. They started as early as 2008, when the term did not even exist yet, through the CCK08 course from the University of Manitoba, in Canada. After successfully developing since 2008, MOOCs took a more professional shape, and different styles of MOOCs have appeared, such as cMOOCs and xMOOCs. As described by Fidalgo-Blanco et al. (2), the latter type is an instructivist and individualist type based on resources and that uses classic e-learning platforms, while the former is a connectivist type, based on social learning, cooperation, the use of web 2.0, and is more focused on a pedagogical approach. In the context of this thesis, we will focus on xMOOCs, which will be referred to as MOOCs.

Before delving further, it is important to have more context about how MOOCs appeared and who uses it. The study done by Glass et al (3) gives a broad outlook on these matters, starting by defining what is one of the most important objectives in MOOCs: democratization of learning. This is due to the fact that, according to the authors, most of those who are enrolled in the top classes, at top universities, are young, white males. Also, the economic meltdown provoked by the 2008 crisis left families looking at fewer opportunities and less income, which led to fewer people being able to study and motivated a need for a different model of learning. To give an insight into the demographics from MOOCs, however, we can take a look at the study from Li (4), which shows that most students in MOOCs are comprised of: bachelor's and master's students, between 24-35 years, with most students groups in English speaking and Latin America cultures and almost 60% of female participants, which is a difference from what Glass et al (3) reports, as having two times more males than females. Zhang et al. (5) reports that out of nearly 400 thousand students, there was a distribution of 48% of females and 52% of males, which can be seen as a positive progress from 2016 (the date Glass et al's study was published).

However, one of the main key problems in MOOCs is cheating. As the study from Glass et al (3) points out, MOOCs are cheating-rich environments, which enables users to create harvesting accounts in order to game the system and obtain answers. This not only is a problem for learning itself, it also dilutes the importance and credibility of a MOOC certificate.

The main reasons why students cheat in academic environments, with the goal of obtaining

better grades, are: procrastination, lack of time and lack of understanding of the subject at hand, as explained on the work of Jones (6). Jones also reports that students have a confused understanding of what cheating is, according to a survey he performed that showed different scenarios and asked students to tell whether some actions were ethically correct or not. The work of Tabsh et al(7) also stresses the lack of understanding from students as to what cheating exactly is. Lack of time, mild penalties and peer pressure are proposed as the main reasons behind cheating. As a way of counteracting this behavior, authors suggest educating the students about academic integrity during the classes and reinforcing it throughout the course. Also, faculty should provide a clear guideline as to the limits of cooperation for assignments conducted outside the class.

As one could expect, cheating occurs also in online environments, although some of the techniques used by the students differ, according to the work of Northcutt et al. (8). One of the techniques revised on the work of Northcutt et al. is the Copying Answers using Multiple Existences Online (CAMEO) strategy, which consists in using a "harvesting" account in order to solve exercises and copying the correct answers to a master account afterwards. It is shown that some courses use techniques to prevent CAMEO, such as in-person assessments taken for a fee or withholding the answers until all problems are graded.

In order to combat the ongoing problem of CAMEO users, the work of Alexandron et al. (9) proposes the use of some techniques. First, the use of randomization in some parameters of the question, so different accounts get a question with different parameters that lead to different correct answers. They state that they have seen this happen in logs from their database, as students tried to submit a correct answer from the harvesting account and have it graded wrong in the master account because the exercise was different. Another option is also using question pools, but this requires additional resources. Second, they propose delayed feedback since, according to them, there is two times less CAMEO cases with it. However, as discussed in this work, this raises questions as to the definition of a MOOC, where instant feedback is one of its prerogatives.

Another strategy to avoid cheating would be to automatically detect dishonest users based on the paths they follow in the course. As noted by Perez-Lemonche et al. (10), students follow different learning paths in online environments. Most of these paths converge into the following basic categories: students who go through all the material and exercises, students who mainly review the contents, and a less populated group of students who mainly do exercises. The latter case, as depicted in the work of Ferguson and Clow (11), could be related to students who are not doing a true effort to pass the course. These students can be further divided into more intricate categories, such as collaborators and harvesters. The former defines students who solve exercises together, while the latter defines students who get the answers from harvesting accounts, as explained by Northcutt et al. (8).

In addition, the trend of open courses has shifted to a more controlled environment, in which part of the course contents and exercises is made available only to students who pay to obtain a certificate. Under these conditions, institutions not only need to improve the student educational experience, but also need to assure that students acquire the contents corresponding to the issued certificates. This is also the case for both online Science, Technology, Engineering and Mathematics (STEM) and Computer Science (CS) courses. As described in the work of Chuang et al. (12), more than 55% of the 4.5 million users in the period from 2012 to 2016 that participated in 290 courses of HarvardX and MITx were on STEM or Computer Science courses. Also, when looking at the percentage of certified CAMEO users by area in the study of Northcutt et al. (8), it is shown that 0.7% of STEM and 0.1% of CS users were awarded certificates while cheating, amounting to nearly 7000 and 1500 students, respectively.

One other method courses usually utilize as a countermeasure to cheating is signing an honor

code. According to LoSchiavo and Shatz (13), the main factor implicating in the compliance to the honor code is the perceived social distance of the instructor. They performed 3 studies to determine the impact of honor codes in online studies. In all 3 studies, students were not monitored in any way, but in study 2 and 3, students were asked to sign an honor code digitally or physically, respectively. Regarding results, all studies were comprised of 14 quizzes followed by an anonymous survey asking students whether they cheated or not. In study 1, 40 undergraduates have gone through the tests and 72.5% of them reported to have cheated at least once while doing the quizzes. It is important to note that, in this context, cheating refers to consulting a textbook, notes, friends, family, the Internet, etc. In study 2, an honor code is introduced to test whether it would reduce cheating with 84 undergraduate students. Considering the ones that signed the honor code, 62% reported cheating. From the others who did not sign the honor code, 50% reported cheating. The authors conclude that no significant difference emerged from the self-reported cheating when an online honor code was introduced, possibly because the necessary social dynamic might be missing in a completely asynchronous online exam. In the last study, which employed a physical sign of the honor code, 165 students were tested in the same 14 quizzes. Comparing the students who did sign the honor code to the ones that did not, they found that the ones who did sign were 30% less likely to report cheating (57.6%) than the ones who did not (81.8%). Therefore, it is a straightforward assumption that social and contextual factors influence an honor's code effectiveness.

In this context, the goal of this master thesis is not only to detect cheating but also to distinguish between students that use harvesting accounts and pairs of students who collaborate. The thesis proposes novel metrics to detect these two types of cheating. All the proposed metrics are designed for pairs of accounts. The metrics are based on the timeline in which different users submit their exercises, in addition to attributes that measure their *compromise* with the course, such as: answer patterns, final score and content visualization. In addition, information about the IP addresses is exploited to identify these users. By using these metrics, we narrow down the search to a few hundreds of suspicious pairs of accounts. These pairs are then clustered into two groups to identify the pairs of users who share their answers (collaboration) and users that have a harvester account.

This master thesis is structured as follows: Chapter 2 discusses work related to cheating detection in online environments and STEM courses, while also introducing the fundamental terms related to this study, such as MOOCs and how students interact with it. In Chapter 3, data processing and extraction of relevant information, useful to identify suspicious users, is described. This chapter also explains the proposed metrics. In Chapter 4, the results obtained from the experiments are discussed. Our approach and Northcutts's are explained in detail and a comparison between them is performed. The results of testing with a new dataset are also shown. In Chapter 5, a discussion is proposed to list possible modifications. Another aspect of this chapter is the exposure and comments on the feedback received from the presentation of this work in the EDUCON 2020 conference. Finally, the conclusions of this study are presented in Chapter 6, comprised of an analysis over the metrics.

2

Theoretical background and related work

2.1 Related work

Before dealing with all the works related to this study, it is important to introduce the topic, starting with a question: what exactly is a MOOC? A formal definition was given in Chapter 1, but it can be synthesized as how Kaplan and Haelein(14) define: "an online course aimed at unlimited participation and open access via the web". Although it is a fantastic initiative that provides education for many people, there are problems such as the dropout rate, the gender, financial and race bias and also cheating. This section will discuss the dropout rate and cheating in more depth, as the biases were already introduced in Chapter 1.

One important aspect that needs to be taken into account when creating a MOOC is the high dropout rates. As pointed out by several studies (2), (15), (16), (17), only about 10% of the students who enroll in MOOCs actually finish them. Some solutions to this problem include personalizing the students' experience, as suggested by Assami et al. (18), or by using a gamification approach, described in the work of Ortega-Arranz et al. (19). Lately, however, online courses are shifting to a more controlled environment, in which you can get access to all exercises only if you pay for a certificate. In this context, cheating has become a prominent problem that needs to be tackled by institutions.

One of the main works that define cheating formally in an online environment was done by Northcutt et al (8). The work consists in the definition of the Copying Answers using Multiple Existences Online (CAMEO), which is a rather comprehensive framework that defines how harvesters can be detected. A CAMEO user is a single user that manages two accounts: a master account - used to log the correct answers and obtain a certificate - and a harvesting one - used to harvest the correct answers by trial and error or any other method. These two account types, when used in conjunction, perform what is defined in Northcutt's study as **harvesting**. Our study encompasses the harvester interaction type and also introduces a new type called **collaboration**, where two students (which can be seen as two master accounts) interact directly to share answers. Going back to Northcutt's definition of CAMEO, it bases the detection of this type of interaction on defining multiple metrics derived from the time difference between each submission and applying 5 filtering criteria to classify a user as a CAMEO (or harvester, in our case). The first condition states that the time difference between the answers of the master and harvesting accounts must be positive (meaning that the master

answers always after the harvesting account, thus introducing the information gathered from there). However, considering only the positive part only tells half of the story: when looking at both the negative and positive parts, one can identify collaboration, as the interaction between the pair of accounts might actually be divided almost equally into positive and negative time differences. For example, the time differences between the answers to each exercise by the pair might show that both accounts answered before and after the other in almost the same amount of times, thus having nearly the same number of negative and positive time differences, respectively. Second, the magnitude of the time difference must be small, a criterion to which we also abide by expanding their implementation: they state that 90% of all the time differences between the users must be under 5 minutes. We modified it so that different time windows can be considered instead of just one. Also, we used fixed amounts of exercises in common in order to compare to this approach, as we believe that by enforcing 90% between all time differences would exclude people that use the CAMEO technique just for a few exercises. A comparison between these two approaches is made in Chapter 4. The third criterion is that the harvesting account, in contrast to the master, must not be certified. Although we agree this is a potentially useful metric and could diminish the amount of false positives, the dataset used did not emit a certificate right away, the student had to request it. The fourth criterion states that the master and the harvesting should share at least one IP throughout the course, which we also implemented, but not enforced, as they can mask their IP. Lastly, the fifth criterion states that there should be less than 10 accounts that share an IP with the master and harvesting, in order to exclude cafes and school networks, which we did not implement. In Northcutt's study, it is reported that from 103 thousand people with only one certificate, 657 of them ($< 1\%$) could have obtained it through CAMEO. It is important to note that the percentage of CAMEO users rises with the number of certificates obtained: for users with at least 20 certificates, 18 out of 73 users (25%) used the CAMEO strategy.

One way to identify cheating is by locating accounts that only interact with exercises in the course and not with the material, which may by an indicative of cheating, as indicated by Ferguson et al. (11). In the work of Perez-Lemonche et al. (10), K-means clustering based on event transitions is used to identify the different learning paths the students follow in online courses. One of the identified groups in that study consisted of students that only do exercises, which could correspond to harvesting accounts that use the CAMEO technique to retrieve correct answers as described by Northcutt et al. (8).

Bao et al. (20) replicate Northcutt's study and test it on an edX MOOC dataset from the Delft University of Technology that includes 10 courses. They follow the same assumptions of Northcutt et al. (8) for harvesting accounts and found out that 1.9% of students are CAMEO accounts, which is a value similar to the 1.3% found in the original study. They describe a possible CAMEO pair of accounts as one with similar full names in both accounts; with similar emails; that uses the same IPs to answer questions; that submit answers within a short period of time from each other; and that the harvesting account submits answers for most questions, but the correctness is low.

In the work of Ruiperez-Valiente et al. (21), the same CAMEO technique is analyzed through a machine learning approach, using a Random Forest model and 15 features to detect submissions as fraudulent. The model achieved a sensitivity level of 0.966 and a specificity level of 0.996. The dataset comes from a single edX introductory physics MOOC called 8.MReV. Out of the 13500 students enrolled, 502 obtained a certificate. From 502 users with a certificate, 65 (12.9%) of them were considered CAMEO users. This number is different from the lower percentages of CAMEO users reported by Northcutt et al. (8) and Bao et al. (20), which may be a particularity on that dataset.

On one study from Alexandron et al. (22), they try to find cheating students with a more general model that does not focus only on a specific cheating strategy in order to detect students

who tried to bypass the CAMEO detectors. They use features that rely on the student's behaviour that is affected or associated with cheating, such as the amount of interaction with the course resources, the time to answer, the student's ability and two parameters from the Item Response Theory, designed by the psychometrics research community, which are used to classify unusual response patterns, including cheating: the Guttman error and the standard error from ability estimates. By using a probabilistic classifier (logistic regression) with four features, they consider only students who received a certificate and classify them into those who used multiple accounts and those who did not, with an AUC of 0.826 on the same dataset as the one used in the work of Ruiperez-Valiente et al. (21).

Another work of Ruiperez-Valiente et al. (23) tries to identify students who collaborate. In order to identify this different type of cheating, they record the time difference between all submissions of the course and users. With those time differences, they create a distance matrix between users by computing the mean absolute or the mean squared time difference. The distance matrix is then used to extract the nearest students, which are expected to be collaborating students. The first problem of this proposal is that it requires all users to do all submissions, which excludes many potential collaborating students as it is not rare to skip a few questions in an online course. In addition, a second limitation of the study is that the proposed metric neglects the order of the submissions. This means that this metric is not able to distinguish between harvesting users and collaborators. When CAMEO is being used, one user answers always before the other, but when two users collaborate, they alternate their answers as we have found in the present study.

In a follow up work from Alexandron et al (24), collaboration between users is identified as another method of cheating. The collaborating users are identified with the method described in the work of Ruiperez-Valiente et al. (23). They claim that a time-based anomaly-detection classifier can generalize from one type of cheating (CAMEO) to another (Collaboration) with an Area Under the Curve (AUC) mean of 0.85. However, the distribution of the data from both types of cheating, used to train and test their model, does not seem to have a significant difference, which renders a weak distinction between both types. Again, from our point of view the limitation is not considering the sign of the time difference between submissions, which we address in this study.

In another work from Alexandron et al. (9), they perform a rereading of the work from Northcutt, analyzing his parameters, the impact of CAMEO over online study environments and share some more statistics about the amount of harvesting account on their dataset, according to Northcutts solution. They start by commenting on the different users that benefit from CAMEO, including premeditated cases, where the user gathers the correct answer before even trying the problem and the ones who use it as a fallback strategy after failing to solve the problem themselves. As shown by their results, most CAMEO events were premeditated. The strategy to identify CAMEO behaviour follows almost the same directives as Northcutts work, the major changes are a fixed interaction of at least 10 questions to act as a sort of high-pass filter and some other metrics, such as 5% of the total answers must be harvested and the presence of inhumanly fast submissions, characterized by a small delay between opening the problem and submitting an answer. For results, the algorithm found 4 times more cheaters when compared to the approach proposed by Northcutt.

Finally, an inherent problem of cheating detection is how to establish the ground truth, since one cannot rely solely on the honesty of the students when asked about cheating. It is naive to think that a questionnaire would suffice, which imposes that a new way of determining it must be developed. One solution, proposed by another study from Ruiperez-Valiente et al. (21), is to deploy a cookie to identify when a student is running multiple accounts, but it may fail to address it if a student runs multiple accounts on multiple devices. A possible solution would be to limit the number of single accounts on the same course in one network, for example. A

future solution could come from this line of research, as models get better at predicting when a student is cheating and could be implemented to work in real time.

2.2 Theoretical background

In this part, the techniques used in this work are discussed. These are composed of mainly three branches of techniques: data mining, programming and machine learning. These areas were studied in depth throughout the courses in this master's program, in many classes. The following sections will not broad the explanation of every aspect related to each subject, instead the specific techniques that were used will be explained in detail.

2.2.1 Data mining

As proposed by David Hand (25), data mining derives from Information Retrieval, dealing with the discovery of interesting, unexpected or valuable structures in large datasets, be it in a global or local aspect.

In a more specific manner, Romero and Ventura (26) define the Educational Data Mining term, or EDM. It is the formal definition of data mining applied to an educational environment, with the purpose of explaining education phenomena and help improving educational systems. They also state that EDM is the union of education, computer science and probability. It has many resources available to evaluate data and find meaning in large datasets, but the most common ones are classification, clustering, Bayesian modeling, relationship mining and discovery with models. EDM is a growing body of research, as evidenced by a survey performed by Romero and Ventura (27), which related the subject all the way back until 1995, a time in which the first study about EDM was performed by Sanjeev and Zytkow (28), where they proposed a study over a university database to make strategic decisions over institutional policies. With the introduction of MOOCs, this area grew even bigger, encompassing sentiment analysis, pattern recognition and many other sophisticated data mining techniques.

2.2.2 Programming

As the programming language of choice for this project, Python 3 was used, along with many of its libraries, being the most important ones: NumPy[1], pandas[2], scikit-learn[3] and multiprocessing[4].

The reason to choose Python 3 is its readily available frameworks to develop programs for machine learning approaches. A study by the analytics company SlashData [5], performed with more than 17 thousand programmers, has shown that in the last quarter of 2019 to the first quarter of 2020, Python was ranked as the most popular language for data science and machine learning. In fact, it remains like that since at least 2017 as shown in Figure 2.1.

Python 3 alone, however, would not suffice to finish this work in a timely manner, which is where the libraries come in. NumPy, pandas, scikit-learn and multiprocessing were used to, together, enable the project to pre-process the data with NumPy and pandas, create a distributed architecture to lower the processing time of the algorithm with the multiprocessing

[1]https://numpy.org/
[2]https://pandas.pydata.org/
[3]https://scikit-learn.org/stable/
[4]https://docs.python.org/3/library/multiprocessing.html
[5]https://s3-eu-west-1.amazonaws.com/vm-blog/uploads/2020/04/DE18-SoN-Digital-.pdf

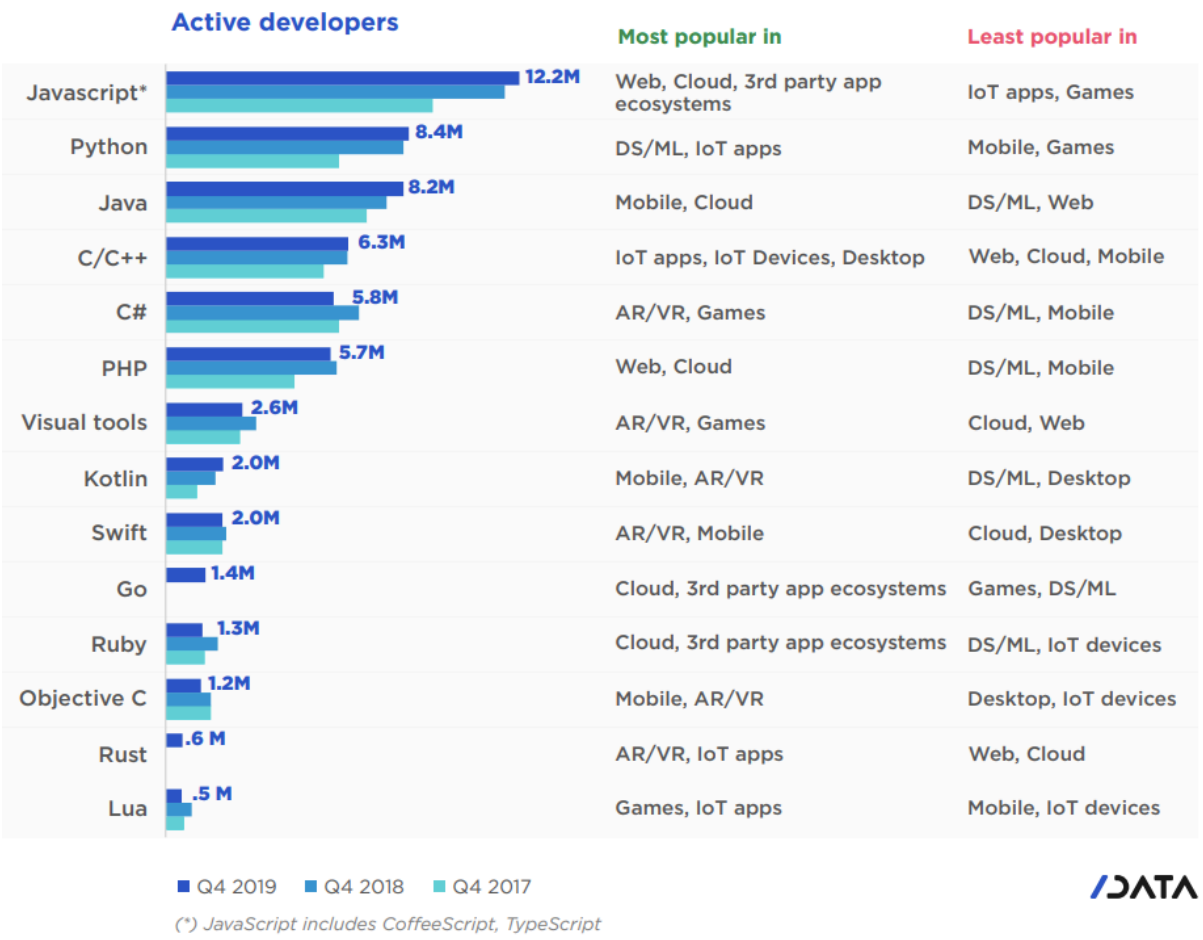

Figure 2.1: Usage and purposes of each language, as defined by the surveyed users. **Source:** This image was extracted from the survey file.

library and also perform a probabilistic analysis through machine learning approaches with scikit-learn. More details about each library are given in the following subsections.

NumPy

Python 3 supports a great variety of data types, but not optimally. In the sense of machine learning, Python acts almost as a wrapper for C/C++, as is the case for the NumPy library: by performing vectorized operations in C/C++, the algorithm avoids executing them natively in Python, thus outputting a faster program. This makes sense for a vast majority of machine learning operations, since it usually deals with large datasets that perform millions, if not trillions, of vectorized operations in order to classify a problem, for example. One of the most important use cases for NumPy is its ability to perform numerical operations (hence the name NumPy) with ease, being the go-to option when dealing with scientific programming and Python.

pandas

The pandas library is another framework built to deal with large datasets in Python but, differently from NumPy, it is not built in C/C++. It shares some of the operations the NumPy does, but it serves a different purpose: management of data. pandas does a remarkable job

when it comes to importing data, as it can do it in many different formats, such as CSV, JSON, SQL, and Microsoft's Excel. When importing data with pandas, the programmer can use the DataFrame object and merge, reshape, select and clean the data as desired, while having a handy way to display it as a table, as shown in Figure 2.2.

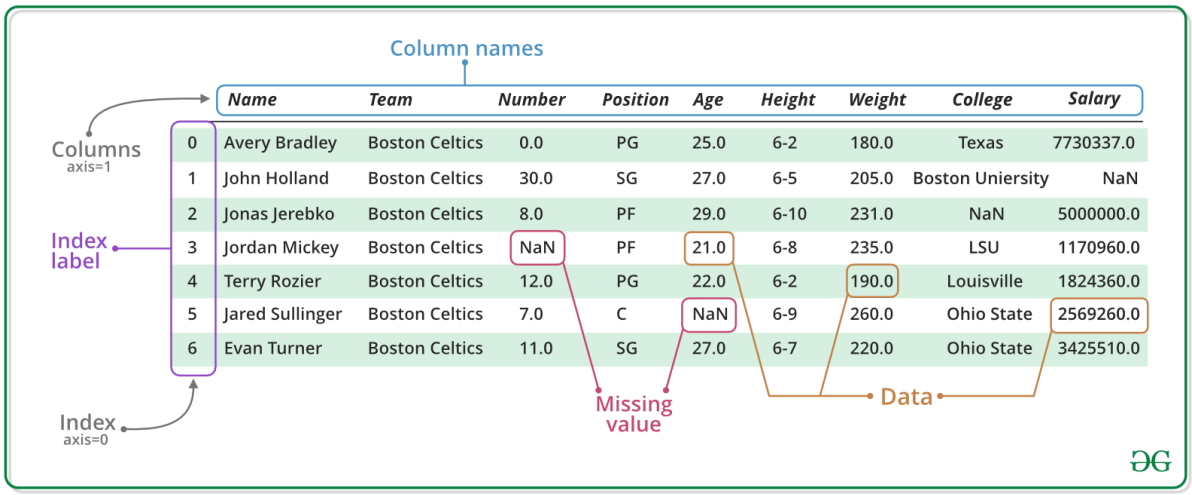

Figure 2.2: Each structure in a DataFrame is highlighted in a different color and serves as an example to how a programmer can use this object.

Source: This image was extracted from the GeeksforGeeks website (29).

A DataFrame is composed of different structures, but very similar to a matrix. The main advantage of using one is its readily available compatibility with many other libraries, as well as the ease to manage large datasets and extract information from them.

scikit-learn

The scikit-learn module is a widely known and adopted suite of functions for machine learning, designed to operate seamlessly with NumPy. It is available for Python and carries different functions designed for classification, regression, clustering, dimensionality reduction, model selection, preprocessing, including the support vector machines (SVM), random forests, gradient boosting and k-means algorithms.

In the context of this work, we use this library in all the evaluation and modeling phases by using its k-means algorithm for clustering, its SVM implementation for classification, its grid search function to tune the hyperparameters of the model and its cross-validation methods to evaluate the performance of the program.

multiprocessing

The multiprocessing library is used to implement parallelism in Python. It is done through the creation of different processes for the potentially parallelizable code. As one might wonder, a programmer could also use threads to perform the same thing using less resources, as new processes end up using more memory by copying the whole context of the program. However, in Python there is a particularity derived from the interpretative nature of the language, called global interpreter lock, or simply GIL[6]. This lock is in place due to the thread-unsafe memory

[6]https://wiki.python.org/moin/GlobalInterpreterLock

management from CPython, which handles the interpretation and compilation background from every code written in Python.

Since GIL is in place, the only possible way to circumvent it is by creating new processes, which will not have to deal with memory management in the same way threads do, since the whole context of the program is copied, hence the aforementioned increase in resource use. This imposes the need to copy the results back once finished, which can be accomplished with the Manager class, which creates a server process to hold Python objects and allow other processes to manipulate them using proxies.

2.3 Machine learning

In an intuitive sense, machine learning is basically composed of methods that learn patterns from data to be able to predict future events from unseen data. There can be good and bad predictions, which depend on a series of many different aspects when training a model. These aspects are so variable that, in fact, anything related to it can have an impact: from the actual data, to preprocessing and also the myriad of parameters that regulate the model. Before reflecting over the different aspects that can influence a prediction, however, it is important to mention and define some types of models and learning algorithms that were used.

2.3.1 Learning algorithms

If we were to divide learning types into all existing categories, it would be necessary to include semi-supervised and reinforcement, as well as some other, less common ones. In the context of this thesis, it is sufficient to divide it into a more general approach by defining only supervised or unsupervised learning.

Supervised

Starting from the definition of supervised learning by Christopher Bishop (30), it should be defined as "applications in which the training data comprises examples of the input vectors along with their corresponding target vectors". Therefore, it is the usual combo where a dataset has both examples and labels, considering examples as the set of variables that define an instance in the dataset.

One can think of this as a natural transition from the way humans learn. Think of an example: let's say there is a dataset composed of different variables that explain how different fruits are described. One of the variables is color, and a model with only that variable could potentially separate oranges, apples and bananas with relative ease. However, what happens if a lemon is added? It will not be as easy anymore to separate it from a banana. That is where complexity comes in, as a model can add many different variables to define its examples, always at the expense of more computational time to process.

In supervised learning, mainly two types of problems can occur: classification and regression. The first one deals with the fruit example, and the idea is quite simple: find enough examples to classify new data into finite categories, or better, **classes**, hence the name **class**ification. Regression, on the other hand, deals with almost the same principles as classification, but it is used for continuous target variables. In the context of predicting the value of a car, for example, a regression model could use many different variables related to already valued cars to approximate a model that predicts the value of new cars.

Many different algorithms exist for supervised learning, each with its unique characteristics and uses: SVMs, Linear regression, Logistic regression, Naive Bayes, Linear discriminant analysis, Decision trees, Random Forest, K-nearest neighbour algorithm and Neural Networks (Multilayer perceptron). For this work, we have used SVMs, as it is one of the best performing methods for classification.

With that in mind, there is a fundamental concept in supervised learning that makes it all happen: validation. Any model can be trained on a dataset and will predict, with 100% of accuracy, if shown the same data. However, the main purpose of a model is to be able to generalize its predictions and still correctly identify new data. In order to do that, a model needs to have training and testing datasets.

Unsupervised

Unsupervised learning, contrary to the paradigm of supervised learning, does not have labels. Again, from the definition of Christopher Bishop (30), the rationale behind unsupervised learning "may be to discover groups of similar examples within the data, where it is called clustering, or to determine the distribution of data within the input space, known as density estimation, or to project the data from a high-dimensional space down to two or three dimensions for the purpose of visualization".

As in supervised learning, many algorithms exist to implement the techniques under unsupervised learning: Hierarchical clustering, K-means clustering, K-nearest neighbour algorithm, Principal Component Analysis, Singular Value Decomposition, Independent Component Analysis.

In the context of this work, the K-means clustering algorithm was chosen as a way to classify the data and generate a ground truth, as there was none.

2.3.2 Models used

Support vector machines

SVMs were first introduced in the seminal work of Cortes and Vapnik (31). As the authors define, SVMs are capable of high generalization by utilizing polynomial input transformations.

This algorithm is capable of performing both regression and classification analysis, although only classification is used in this work. In order to perform classification or, in fact, any of its functions, an SVM creates what are called *hyperplanes*. Considering SVMs are essentially binary classifiers, these exist in order to separate the classes into whatever they may be, but into 2 possible cases. However, as illustrated in Figure 2.3, not every hyperplane is able to optimally separate the classes. A reasonable argument to separate both classes and ensure minimum generalization error is by selecting the hyperplane that is able to have the biggest margin possible between the classes, i.e., the one in which the nearest points are the most far apart possible from it. In the case of Figure 2.3, this would be the hyperplane number 3 (H3), as can be seen by the gray lines representing the distance from the points.

Until now, only classic linear problems were dealt with. What if there was a nonlinear problem? SVMs can deal with that too through a technique called *kernel trick*. Instead of using a different technique to solve a nonlinear problem, a kernel trick can circumvent this by mapping data into a different dimensional space where a hyperplane that separate the classes can be easily found, as illustrated in Figure 2.4, where the data on the left is not linearly separable in 2D, but when using a kernel trick to move it to 3D, a hyperplane can be easily found, as the green hyperplane shows.

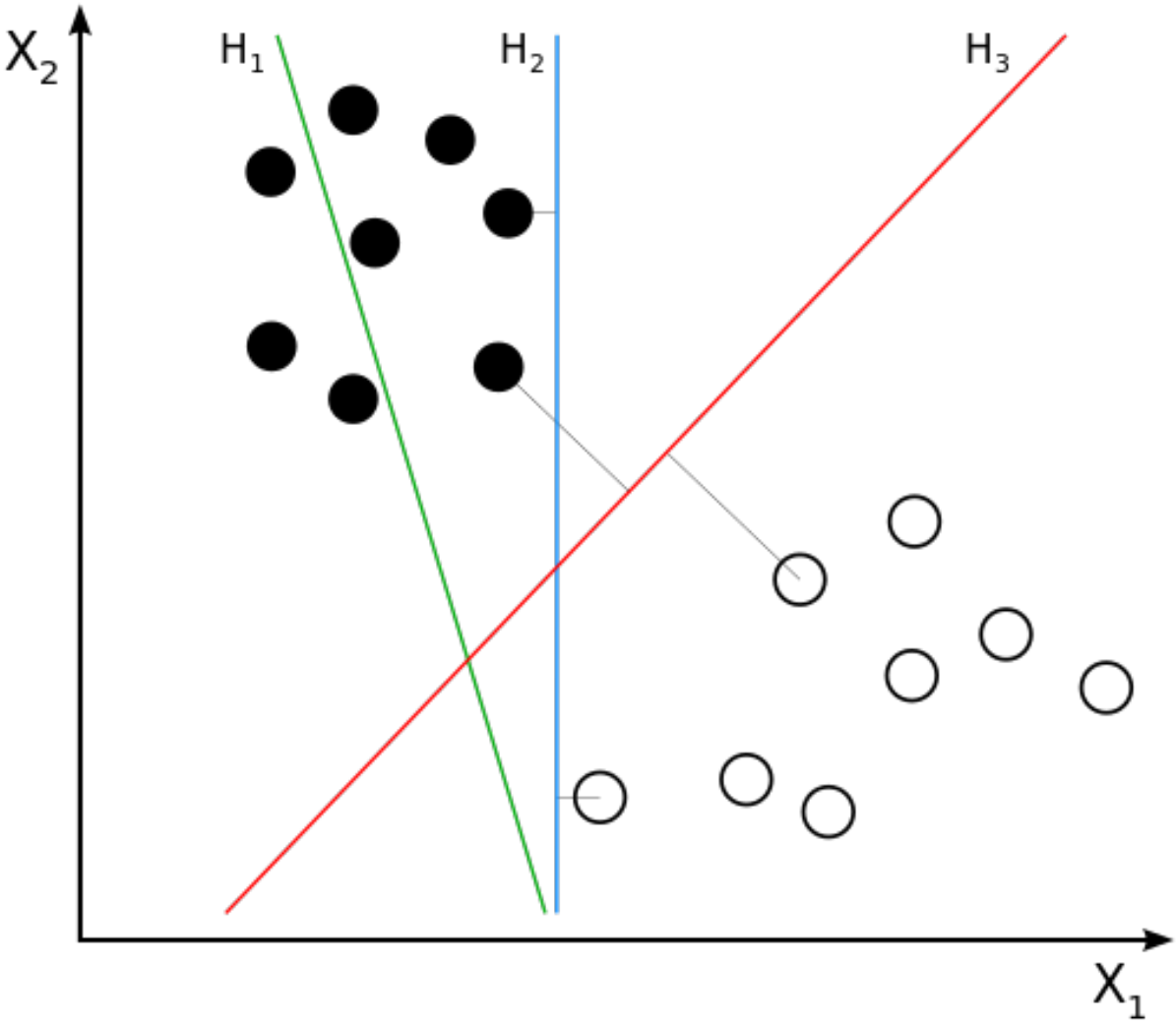

Figure 2.3: Three possible separating hyperplanes.
Source: This image was extracted from Wikipedia (32).

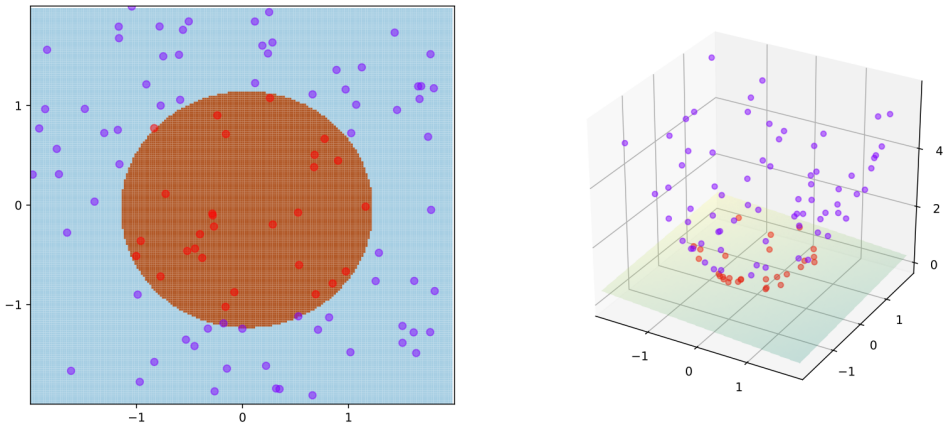

Figure 2.4: Example of the kernel trick.
Source: Image extracted from Wikipedia (33).

K-means clustering

Lloyd's work (34) was the first to formally define the k-means algorithm for clustering. The idea behind the method is simple: group similar points together and search for patterns by looking at k groups in the dataset. These groups, called clusters, are defined by the euclidean distance to the centroid and assigned to the nearest one. The class attributed to a data point depends on which centroid the point was assigned to. Figure 2.5 illustrates this idea.

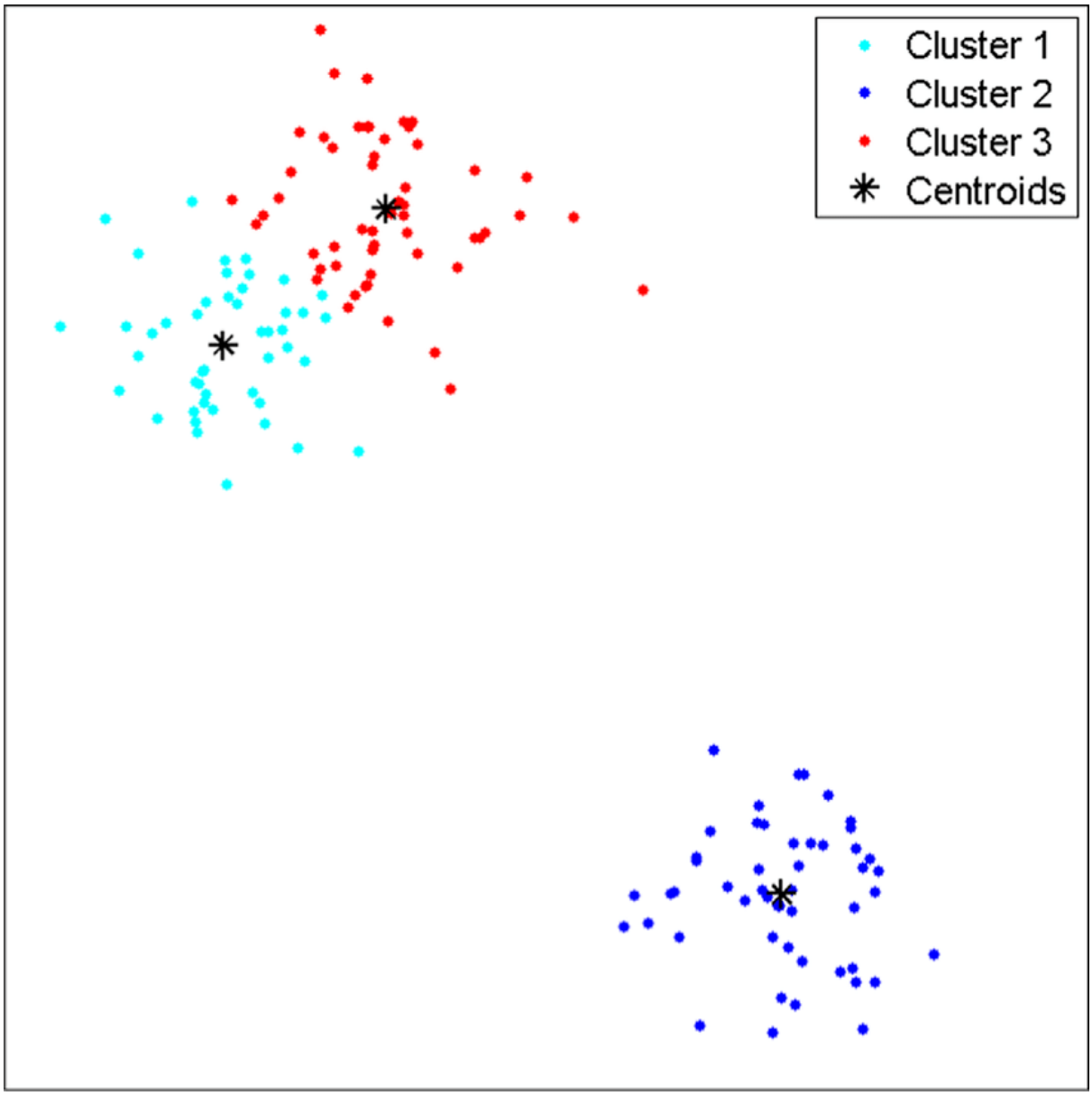

Figure 2.5: Example of the k-means clustering algorithm with k = 3.
Source: Image extracted from the work of Wen et al (35)

2.3.3 Validation

When developing algorithms to predict any kind of value, there is the inherent need of validating the performance of a trained model, since it is not advisable to train a model and start using it right away. However, validating a model can be a complex task, as there are wildly different

methods to accomplish this. A discussion over the different approaches is done in the following subsections.

Cross-validation

A validation process can be based on cross-validation (CV) which, in fact, is an extensive topic and generate numerous discussions due to the variations and the complexity involved in it. In a general panorama, there are many types of CV: k-fold, leave-one-out and leave-one-group-out. The right type depends a lot on the dataset and problems involved, but a general CV process consists of different phases:

1. Definition of a train and test split of the available dataset
2. Evaluation of the best validation technique to employ
3. Actual validation
4. Comparison of the generated models

It is important to keep in mind that in this work, k-fold CV, as well as nested CV, were the chosen approaches, and the first thing to do is go into details of what k-fold CV means. According to the work of Christopher Bishop (30), k-fold CV is the process of creating different batches of data within the same dataset, in order to test and train with different portions of the dataset to estimate the generalization of the model. This is rather limiting, as there are some pitfalls when adopting only this method to validate a model, such as overfitting. When evaluating models with the k-fold CV method, generalization error is the metric to define which one will be chosen. However, as Cawley and Talbot point out in their work (36), generalization error can be broken down into bias and variance and, while a low bias is a good model selection criterion, they demonstrate that a low variance is at least as important, as a non-negligible variance introduces the potential for overfitting in model selection as well as in training the model. The general idea, though, is to generate the maximum number of unseen data for the model and is exemplified in Figure 2.6.

Nested CV is a more robust method of validating a model, generally employed when the hyperparameters of a model need to be tuned. It is, essentially, a CV inside of a CV, to help mitigate the shortcomings of using only a simple CV. When using this method, the inner CV tunes the hyperparameters and finds the best ones for the respective fold, while the outer CV is used to estimate the generalization error of the model, by averaging the test score over the folds.

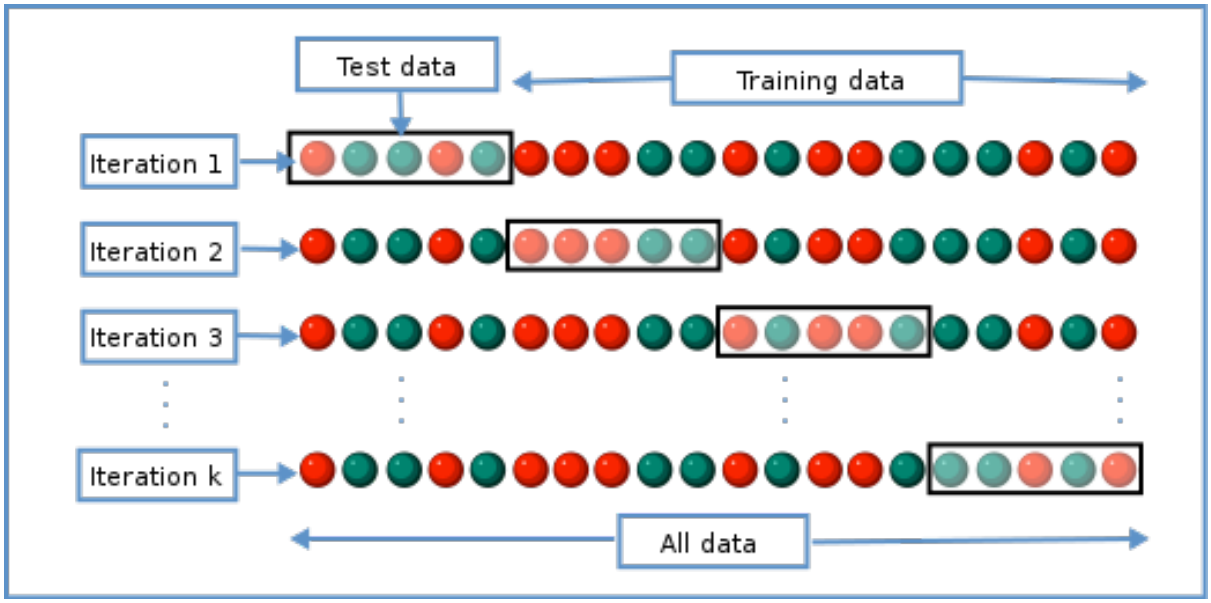

Figure 2.6: Explanation of the CV technique.
Source: Image extracted from Wikipedia (37)

3

Data processing

The data consists of a log of actions performed by the students in a course entitled "Jugando con Android: aprende a programar tu primera app" [1], taught by professors from the Universidad Autónoma de Madrid. The logged actions include: interaction with the course material, exercise submission attempts, posts on the forum, etc. The log was processed to select information mainly about actions related to exercises, of which we keep: the timestamp of the actions performed, the student answers and whether the answers were correct or not. This information is then used to create novel metrics that can be useful to identify suspicious users, as explained in Section 3.4 in more detail. In addition, every event was characterized by a unique label, such as "load_video", "problem_check" or "textbook.pdf.chapter.navigated", but there were many others, which were all categorized according to their meaning to count the distribution of the user's interaction and create a pattern about how he interacted with the course. To account for correct and incorrect exercises, there is another step that needs to be taken, as looking for the "problem_check" label will only return the event of a student attempting an exercise. To actually evaluate how the student assessed the exercise, the label "correcto" needs to be searched within what is returned by the "problem_check" label, which in turn will give a boolean result of whether the answer to the exercise was correct or incorrect.

To give an objective idea of how the data looks like, an example of the raw data was given in the Appendix.

3.1 Co-occurrence in exercise answers

Co-occurrence of responses to exercises is one of the important bases on which suspicious students will be identified. We believe that the co-occurrence of events is an indicator of how related a user is to another, in the sense of how dependent one user is on another in his/her responses to exercises. For this, pairs of accounts answering to the same exercises, during the same period of time, will be considered as suspicious. In order to select dependent pairs of users, we first count the number of times that every possible pair of users have answered the same exercises within a small time frame. Then, for a followup analysis, we keep the pairs of students that have answered at least to a certain number of exercises within the given time frame. This

[1] https://www.edx.org/es/course/jugando-con-android-aprende-a-programar-tu-primera

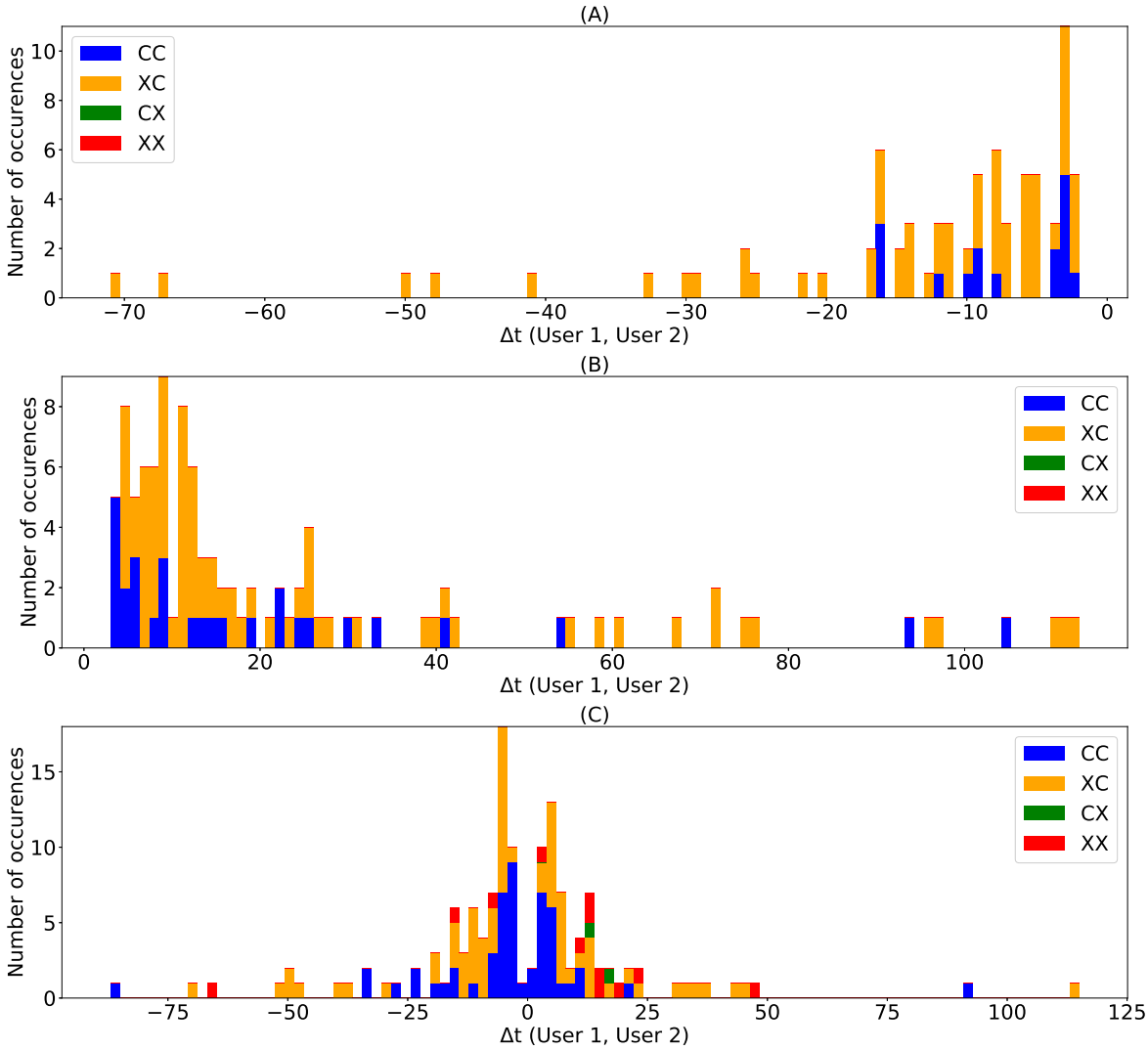

Figure 3.1: Example A shows a case in which user 1 answers always after user 2; example B is a similar case but now user 1 is the one answering before; example C presents a case in which both users answer very close in time

minimum number of exercises in common is set to rule out users who have answered by chance to the same exercises in the same time window. We consider time frames of 1, 2, 5 and 10 minutes and a minimum of 10 and 20 exercises in common within those time frames. The time difference between the response of user u_1 and user u_2 when performing exercise k is

$$\Delta t(u_1, u_2, k) = t_{u_2}(k) - t_{u_1}(k) \quad , \tag{3.1}$$

where $t_u(k)$ is the time, converted to epoch time, when user u answered to exercise k.

3.2 Interaction types

The time difference between exercise responses, as shown in (3.1), is necessary to identify which user answered first to a given exercise. By taking into account the time difference to order the responses of a pair of users to a given problem, four patterns can be identified: first answer in time is correct and second is correct (CC), wrong-correct (XC), correct-wrong (CX) and wrong-wrong (XX). From these cases, we have discarded the sequences when a user answers

Table 3.1: Unique users in every time window

Number of exercises in common	Time windows			
	1	2	5	10
10	92 (2.77%)	147 (4.43%)	437 (13.17%)	864 (26.03%)
20	69 (3.3%)	70 (3.35%)	77 (3.68%)	135 (6.46%)

correctly to an exercise and, after that, another user answers it incorrectly (case CX), since this is no indicative of unethical behaviour. It is important to stress that we do not discard XX cases. Even though it does not seem to produce any immediate benefit, it might be the case that students are ruling out wrong answers before finding out the correct one.

The interaction pattern and co-occurrence of answers to exercises are shown in Fig. 3.1 for three pairs of accounts. The plots show the number of times each pair of users responded to the exercises in common with respect to the time difference in their responses. In the top plot, there is a clear pattern in which the values of the time difference of responses (Eq. 3.1) are all negative but small (Note that the bulk of the answers are below 20s). The second plot is equivalent to the first one in the opposite direction. In both cases, all answers are of type XC (first response in time incorrect and second answer correct) or CC. These data show a clear pattern of an user cheating with a harvesting account (CAMEO technique). In addition, looking at the proportion of incorrect first responses, it seems that the first try is performed almost at random. The bottom plot shows a different behaviour. In this case both users answer to the same exercises almost at the same time (answers to most exercises have a difference in time below 30s) alternating in who answers first. Despite this fact, there are very few cases of a first answer correct and a second incorrect (CX). The absence of this pattern and the co-occurrence of answers seem to be clear indicators of collaboration between two users.

3.3 Amount of pairs of users and shared exercises

When processing data from the original 7171 users, the first chosen criterion was to keep only pairs of accounts with at least a minimum number of co-occurring responses to exercises, in order to eliminate useless data. The results show that 3318 users had done at least 10 exercises of the course and 2087 had done at least 20. When considering the time windows, these users are reduced to the number of pairs shown in Fig. 3.2. As for the number of **unique** users, the 3318 and 2087 pairs, are reduced to the values shown in Table 3.1. This result means that many users are involved in more than one pair of suspicious accounts, as most users are not unique. It could be the case that a user has several harvesting accounts or that a group of users are collaborating between each other.

3.4 Metrics

In this section we describe the proposed metrics that can help in the identification of harvesters and collaborators in online courses. Harvesters are users who utilize harvesting accounts to acquire answers to the exercises for their master accounts. In this case, we expect a clear one-way interaction pattern. Collaborators are pairs of students that share answers while doing the exercises. The interaction in this case is expected to be more symmetrical with possibly a two-way answer sharing channel. The plots in this section that illustrate the different metrics are computed using a minimum of 10 exercises in common within a 2 minutes time frame. Similar

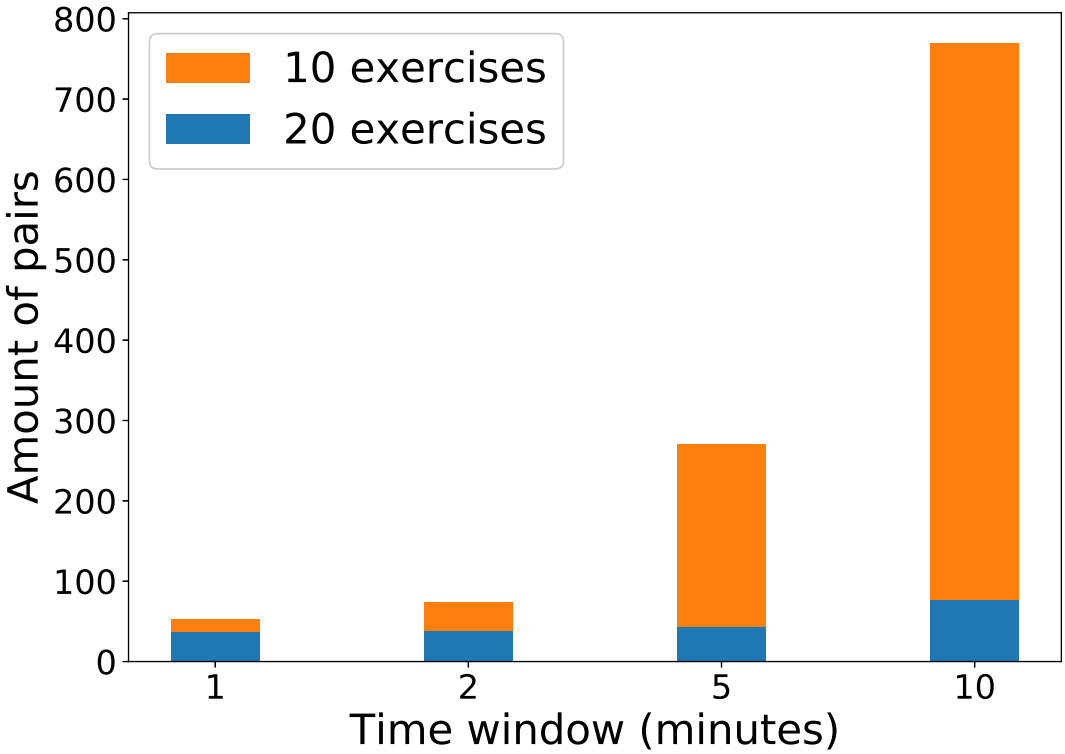

Figure 3.2: Number of pairs of accounts with at least 10 (orange bar) and 20 (blue bar) exercises in common done within 1, 2, 5 and 10 minute time frames

results are obtained for the other time frames and thresholds for the number of exercises. The proposed metrics should be applied assuming that the user receives an instant feedback when finishing a task. In environments where the answer is given only at the end of the assignment, this strategy would be hindered due to the time windows that are used to identify cheating behaviours.

3.4.1 User bias

Based on the interactions described in Section 3.2, we define the metric *user bias* as an indicator that helps identify the direction of the copy/collaboration —i.e., to identify who copied from whom if it is the case— as

$$UB(u_1, u_2) = \frac{\sum_{k\ s.t. |\Delta t(u_1,u_2,k)|<\tau} sign(t_{u_1}(k) - t_{u_2}(k))}{\sum_{k=1}^{N} \mathbb{I}(|\Delta t(u_1, u_2, k)| < \tau)} \tag{3.2}$$

where $\mathbb{I}$ is the indicator function such that $\mathbb{I}(true) = 1$ and $\mathbb{I}(false) = 0$ and N is the total number of exercises. A positive user bias indicates that the first user tends to answer after the second one to the same exercises and within the time window denoted by τ. A user bias value of 1 means that the first user answered after the second user for all the identified exercises. A negative user bias means that the second user tends to answer before the first one. A user bias value close to 0 means that there is no preferred order in the timestamps of the responses of both users. This could indicate that students are collaborating by solving the exercises together. It is important to note that $UB(u_1, u_2) = -UB(u_2, u_1)$.

3.4.2 Final score difference

The final score difference is defined as the difference between the final score obtained by any pair of accounts. This metric is useful to discern whether the pair has a harvesting account or if they are two collaborating accounts, especially when coupled with other metrics, such as the user bias. For example, if a pair of users solves the same exercises within a time window and has a high final score difference, it suggests that this pair might be composed of a master and a harvesting account since there is one account that does not care about the final score. On the contrary, if they present a low final score difference, it could be an indicative of two collaborating students, as both are seeking to pass the course.

The combination of *user bias* and *final score difference* can be helpful to individualize pairs of users that could be collaborating. In Fig. 3.3, the final score difference from pairs of users with respect to their *user bias* is shown. Both plots show the same pairs of users. In the top plot, the number of exercises solved by each couple within the considered 2 minutes time frame is represented in gray scale. The darker the color, the higher the number of times both users answered to the same exercises in the given time frame. As it can be observed, the most active part is the line along a score difference equal to zero and the vertical lines for user bias equal to 1 and -1 in the first and third quadrants. The regions of interest are located around the curve x^9, shown in the plot. These regions of interest are more clearly shown in the bottom plot marked with a 1, 2 and 3 labels. Region 3 has a score difference close to zero, which means that both users are interested in obtaining a final score. In this region, a wide range of values for the *user bias* can be observed. This means that, if collaboration occurred, the flow of information could be reciprocal (points around 0) or asymmetrical, with one user being the source and the other the recipient. In any case, as the score difference is small, we believe that this region corresponds to collaboration between users. On the contrary, in regions 1 and 2, the user bias is very close to 1 (or -1), which means that one of the account in the pair is answering after the other for most exercises. In addition, in these regions, when the first user answers after the second (positive user bias), the score difference is also positive, meaning that the first user gets a higher score than the second user. The same situation happens in region 2. These two regions can be an indicative of an user using a harvesting account. Note that a pair that falls in region 2 could be in region 1 just by computing the metrics after inverting the user inputs to the functions, since $UB(u_1, u_2) = -UB(u_2, u_1)$.

3.4.3 Material interaction rate

The material interaction rate (MIR) tries to capture how much time a student spent trying to solve exercises versus the total amount of time he/she spent taking the course. We propose to measure it as the fraction of interactions with the materials, which means the revision of documents and visualization of videos, with respect to the total number of interactions. The total number of interactions includes, in addition to the interactions with the materials, the attempted exercises (correctly answered or not), disregarding repetitions. A user who only answers exercises will have a low value of MIR, as the interactions with the materials are zero. On the other hand, a value of MIR equal to one indicates a user that only interacts with the materials. Since we are analysing pairs of accounts, and not individual users, the metric we will consider is the minimum MIR (mMIR) between every pair of accounts. The mMIR metric will be low if at least one of the accounts is used only to solve exercises. The minimum is taken because the average shadows the behaviour of the pair when the disparity between both measurements is high.

Fig. 3.4 shows the mMIR for different pairs of users using a *score difference* vs. *user bias* plot. As it can be observed, pairs of accounts in regions 1 and 2 have low mMIR values, which

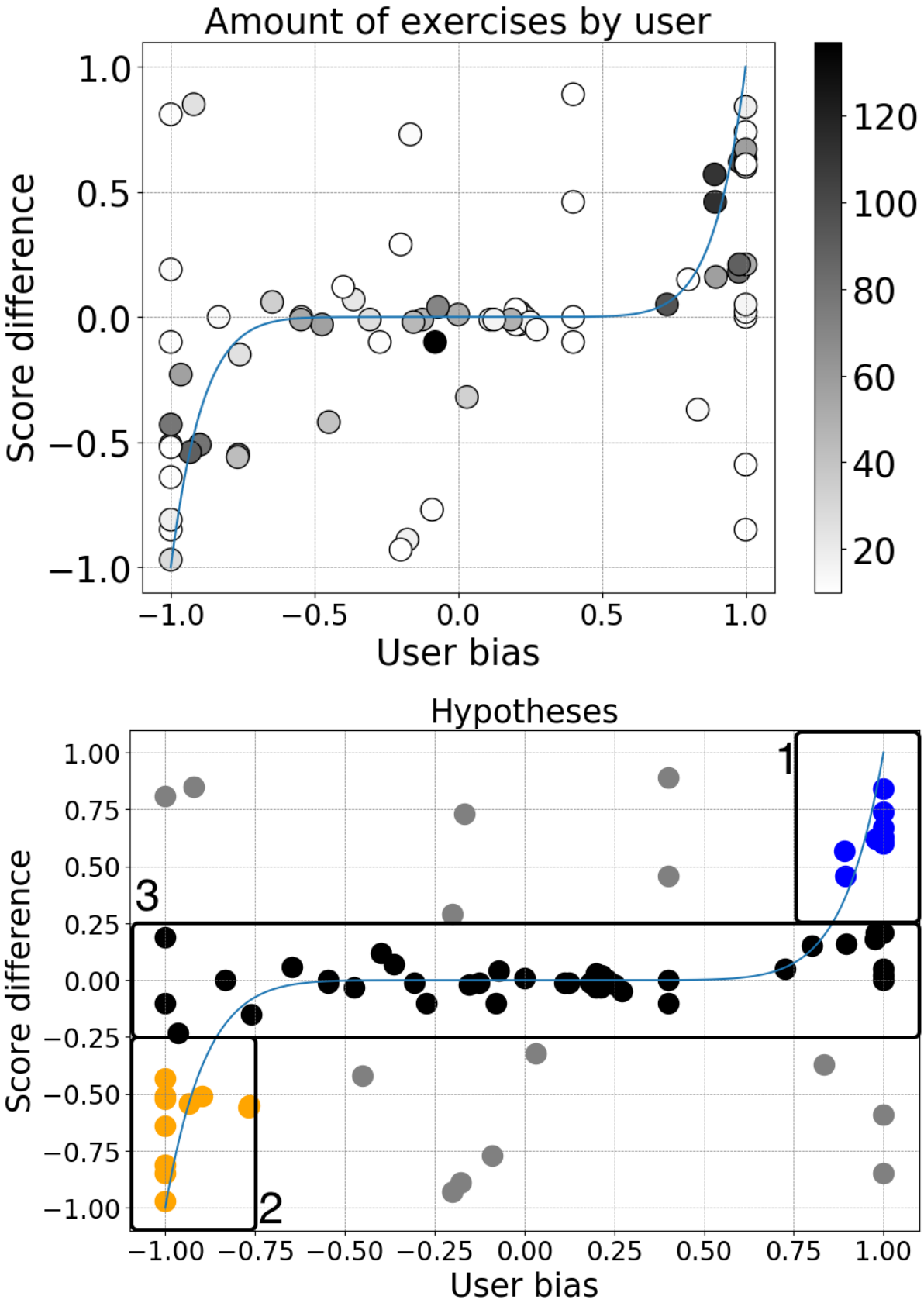

Figure 3.3: Score difference with respect to user bias for pairs of accounts that answered at least 10 exercises in common within a 2 minutes time frame. Top plot shows the number of exercises in common in gray scale and bottom plot shows harvesting and collaboration regions.

is another evidence of the use of harvesting accounts as these types of accounts are not used to review the materials.

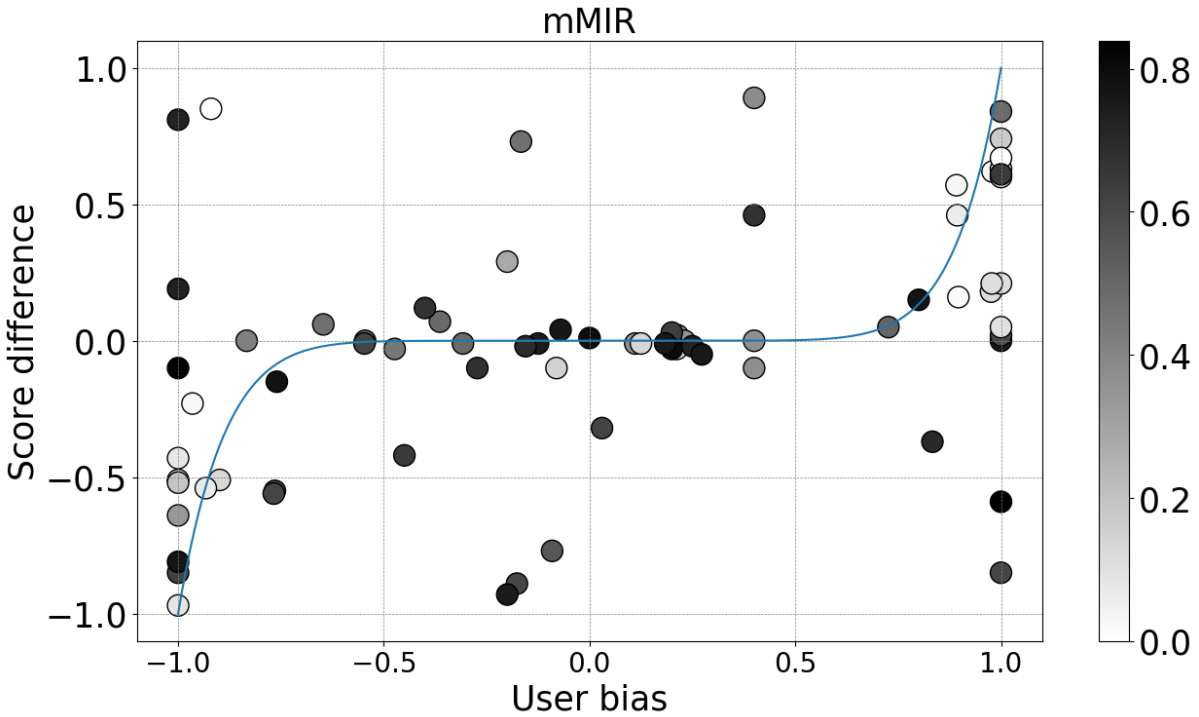

Figure 3.4: mMIR in gray scale for pairs of users for *score difference* with respect to *user bias*

3.4.4 Percentage of shared IP numbers

This metric is taken as the percentage of shared IP numbers both users have as

$$\%SharedIPs = \frac{IPs(u_1) \cap IPs(u_2)}{IPs(u_1) \cup IPs(u_2)} \quad , \tag{3.3}$$

where $IPs(u)$ is the set of IP addresses used by user u. This metric can also be useful to identify harvesters as CAMEO users usually share their IP addresses (8). A high percentage of shared IP addresses is an indicative of a CAMEO user, while a low percentage may be inconclusive.

3.5 Northcutt's approach

The work of Northcutt et al (8) utilizes an approach that differs from our strategy. First, they used an adaptive filter for the minimum number of exercises, as they consider that pairs of users are only able to be considered suspicious when they have at least 90% of all the time differences between their submissions sitting inside a 5 minutes window. This metric makes sense for finding harvesters, as harvesting accounts are supposed to have a close answering time to the master account and also do not complete exercises without an intent to find the answer. However, when looking at it from a collaboration standpoint, this does not make sense, as both are master accounts and will have a different answering dynamic. They also use only the positive portion of the time differences, which doesn't allow the use of the User Bias metric and hinders the detection of collaboration between the pairs. Another minor differences are considered, such as the IP number: they enforce it that a user and a master should share an IP at least once, but the user can mask its IP through a VPN of a different network and still manage to use a harvester. What came to the conclusion that it makes more sense to use it as another metric for evaluation, but not enforcing it. Also, they use a filter where the master account holds a certificate and the harvester account does not. Finally, they also consider that in order to diminish the number of false positives, a pair can share an IP number with at most 10 other users, otherwise they might be in the same classroom or cafe.

We implemented Northcutt's specifications as near as possible and provide a fair comparison to our approach. However, some differences are present. First, students in the course studied in our work had to request the certificate, which in turn lead to few certificates being issued. Therefore, this metric could not be implemented. Second, the IP metric in the original work is considered differently: the IP assigned to a user is what they call the "modal IP", consisting of the most common IP assigned to an user in a course. This IP is then put in a (name, IP) tuple and compared across the other 115 courses they evaluated. Since we only used 2 courses and the IP numbers were anonymized, we implemented it by counting the number of times an IP number was used by different users. If more than 10 users had used it, then the IP number would be considered a router, but a pair should still need to have at least one IP in common to be considered a cheater. The results are displayed in Chapter 4 and the comparison is detailed in Chapter 6.

4

Results

4.1 Our approach

In order to identify harvesters and collaborators in an automatic manner, we ran K-means on the previously defined variables. Specifically, for each pair of accounts identified when applying the 1, 2, 5 and 10 minutes time frames and the minimum of 10 and 20 exercises in common, we computed the metrics: *user bias*, score difference, minimum MIR and percentage of shared IP addresses. Since the user bias and score difference have different signs depending on the order the users are arranged, we chose to order the user input for those metrics such that the user bias is positive and we keep that order to compute the other metrics. This was done to avoid the ambiguity that comes just by the order of the user input. Furthermore, the percentages of CC and XC cases were added as attributes as well. Before processing the dataset, the features were standardized by removing the mean and scaling to unit variance. The K-means was executed using $k = 2$ and the results were used to tag the pairs of accounts into a *harvesting* pair and a *collaborator* pair, as shown in Fig. 4.1. As it can be observed in the plot, pairs of users with a low CC percentage and a high user bias (black dots in the plot) correspond to pairs of users in which the first one answers incorrectly more often. This means that this cluster correspond to users that are possibly using a harvesting account to obtain their answers. Pairs of users with a high percentage of CC answers indicate users that are collaborating. The centroids of both groups are shown in Table 4.1. From this Table, it can be observed clearly the different patterns of interaction of collaborations and harvesters. Collaborators have values much lower than harvesters in XC, user bias, score difference and shared IPs; and higher values in CC and mMIR.

Table 4.1: Centroids for the harvester and collaborating pairs of users

	CC	XC	User Bias	mMIR	Score Diff	Shared IPs
Collaborators	0.97	0.03	0.49	0.57	0.06	0.19
Harvesters	0.27	0.73	0.97	0.16	0.49	0.64

The tagged dataset was used to train a Support Vector Machine (SVM) classification model in order to automatically identify the two classes using the metrics described in section 3.4. This was carried out in order to be able to estimate a valid performance generalization of the

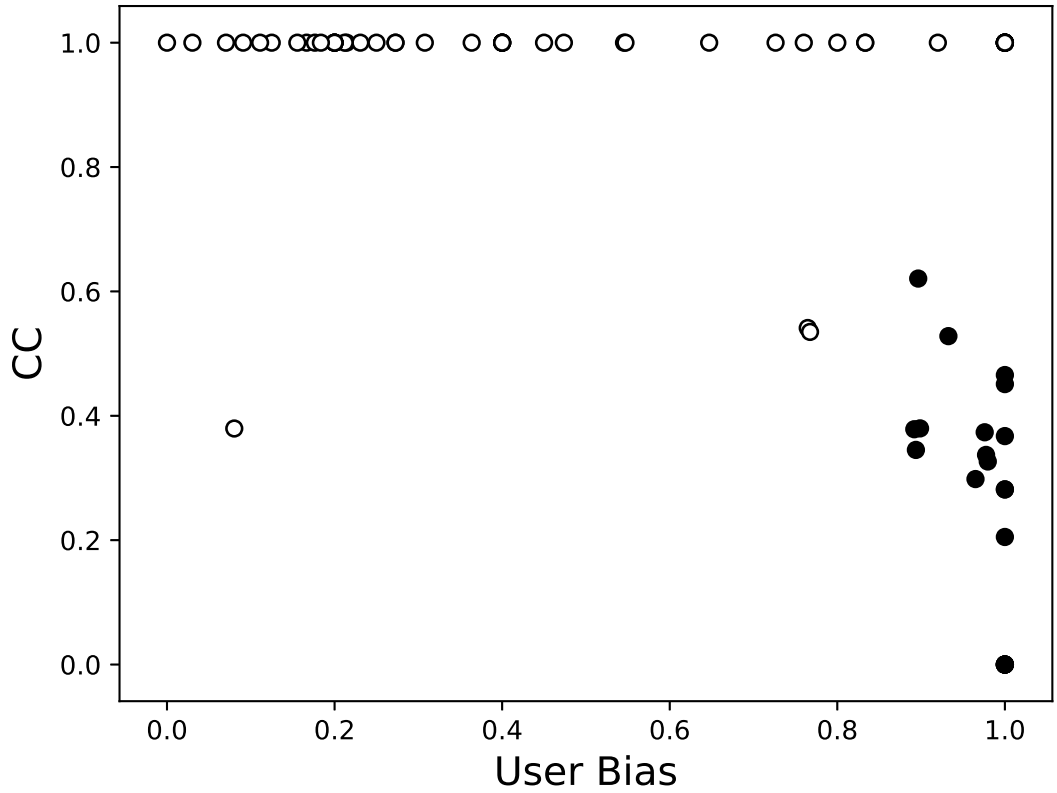

Figure 4.1: Pairs of accounts plotted for CC with respect to user bias metric. Black dots are possible harvesters, whilst white dots are possible collaborators

model, which cannot be acquired for clustering. The model's performance is evaluated using a 10-fold cross-validation. RBF and linear kernels were used for the SVM model and a within-train grid-search was performed to tune the values of C and *gamma* for the RBF kernel and C for the linear kernel. The grid search for C and *gamma* hyper-parameters is defined by 2^X, with $X = [-5, -3, ..., 11]$ for C and $X = [-1, 1, ..., 15]$ for gamma. The average accuracy with its standard deviation is shown in Table 4.2, accounting for the different time frames and minimum number of exercises. As it can be observed in Table 4.2, the generalization accuracy of the model is rather high, which means that the two types of suspicious pairs can be separated easily.

A further analysis of the pairs of accounts shows that in some cases a user can be in several pairs. In this dataset, 6 users are present in more than one pair when using a 10 exercises and 2 minutes time frame. In addition, we found that some of the users that appear in several pairs are tagged as a collaborator in one pair and as a harvester in another pair. This double role could be explained as one user harvesting the answers from a harvesting account (harvesting pair) and afterwards sharing those answers with a third user (collaboration). However, further analysis should be carried out to understand this double role.

In order to assess the importance of each feature, the Pearson correlation between the features and both types of cheating was computed. These values are shown in Table 4.3. As it can be seen in Table 4.3, mMIR and CC have a high negative correlation to the harvester cheating type. This type of cheating has a high correlation with user bias, score difference or shared IPs. These correlations are of opposite sign for the collaboration type. As we anticipated,

Table 4.2: Accuracy and standard deviation data from the analyzed time frames and minimum exercises in common

	Linear kernel			
	10 exercises		20 exercises	
Time frame	Accuracy	Std. dev.	Accuracy	Std. dev.
1 minute	98.2%	1.2%	98%	2.0%
2 minutes	98.9%	1.1%	95.7%	2.3%
5 minutes	100.0%	0.0%	98.4%	1.6%
10 minutes	100.0%	0.0%	100.0%	0.0%
	RBF kernel			
1 minute	95.9%	4.9%	96.7%	2.6%
2 minute	98.1%	1.2%	96.4%	3.0%
5 minute	98.4%	1.2%	99.1%	2.7%
10 minute	99.0%	2.0%	98.3%	1.8%

these correlation values make sense as we expected for harvesters a low mMIR, and high values for the user bias, the score difference and the common IPs percentage.

Table 4.3: Pearson correlation between harvester (H) and collaborator (C) classes and all features

	CC	XC	User Bias	mMIR	Sc. Diff.	Shared IPs
C	0.95	-0.95	-0.61	0.70	-0.48	-0.60
H	-0.95	0.95	0.61	-0.70	0.48	0.60

4.2 Comparison to Northcutt's approach

To compare the number of identified users with the metrics from Northcutt, we generated the graphs in Figures 4.2, 4.3, 4.4 and 4.5. These figures represent the number of pairs of accounts (represented in the y axis) with exercises in common (represented in the x axis) for all time windows. Our analysis broadened the one made by Northcutt (8), in regards of time windows, since our approach encompassed the 5 minutes one used by Northcutt, but included the 1, 2 and 10 minutes time windows. Also, some metrics were changed to accommodate for the peculiarities of our dataset, as explained in Chapter 3.

To look at the images from a more objective point of view, one can look at Table 4.4 and see that there is a discrepancy between the number of suspicious cases in both approaches. This can be explained by the fact that, for a smaller number of users, which results from a smaller time window, our approach finds more users. However, when bigger time windows are used, our approach fails to find as many suspicious users. Considering that in our dataset 3853 users do not even attempt 10 exercisses and are, therefore, eliminated, over 50% of the possible users to combine are eliminated from the analysis. This might help explain why there are almost 20% less suspicious cases in our approach using the biggest window. Another reason may be related to a not exact implementation of Northcutt's algorithm: as explained in Chapter 3, the IP number filter was implemented differently, according to the limitations in our dataset. As Northcutt (8) states in his work, this filter was a great part of false-positive filtering, as it watched for the same IP numbers across 115 courses and eliminated the ones that were used by too many people. This can result in false-positives and, therefore, inflate the metrics.

Another thing to notice is that, when looking at Table 4.5, the amount of detected users

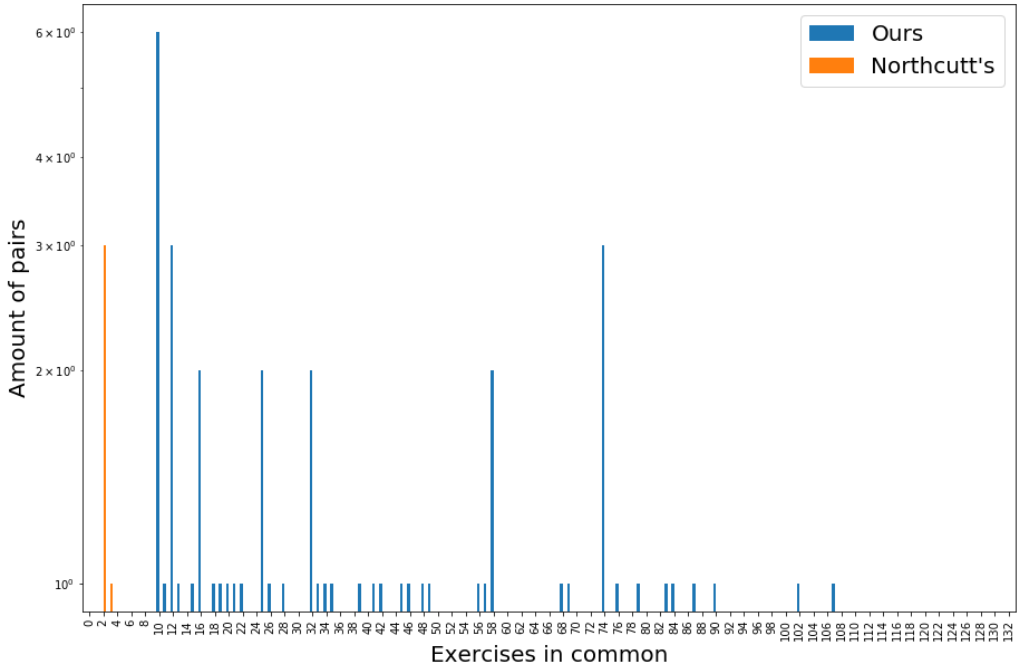

Figure 4.2: Number of pairs of accounts with exercises in common under a time tolerance of 1 minute

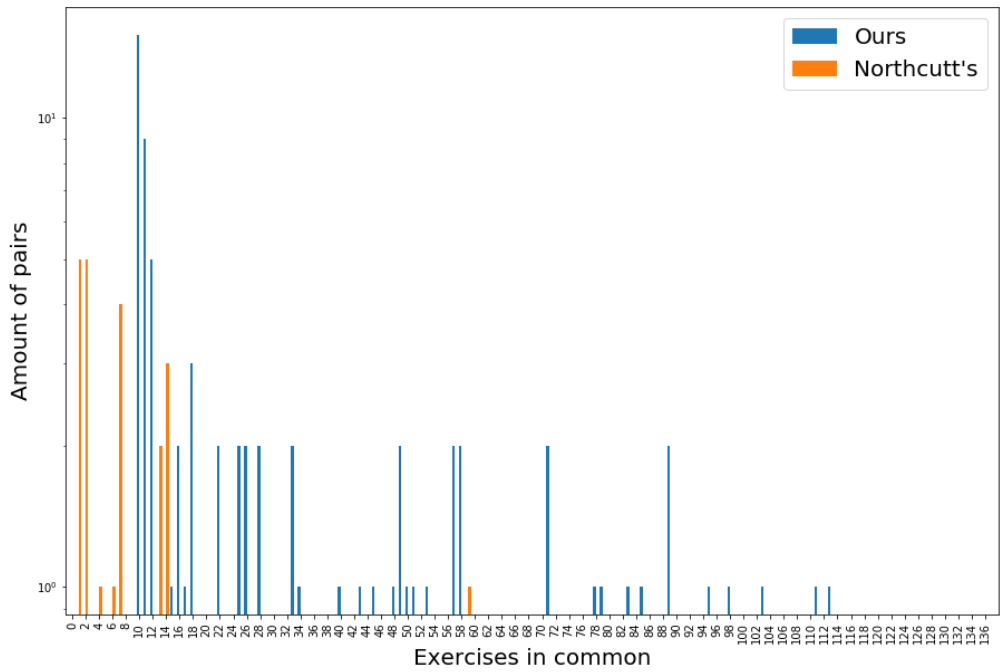

Figure 4.3: Number of pairs of accounts with exercises in common under a time tolerance of 2 minutes

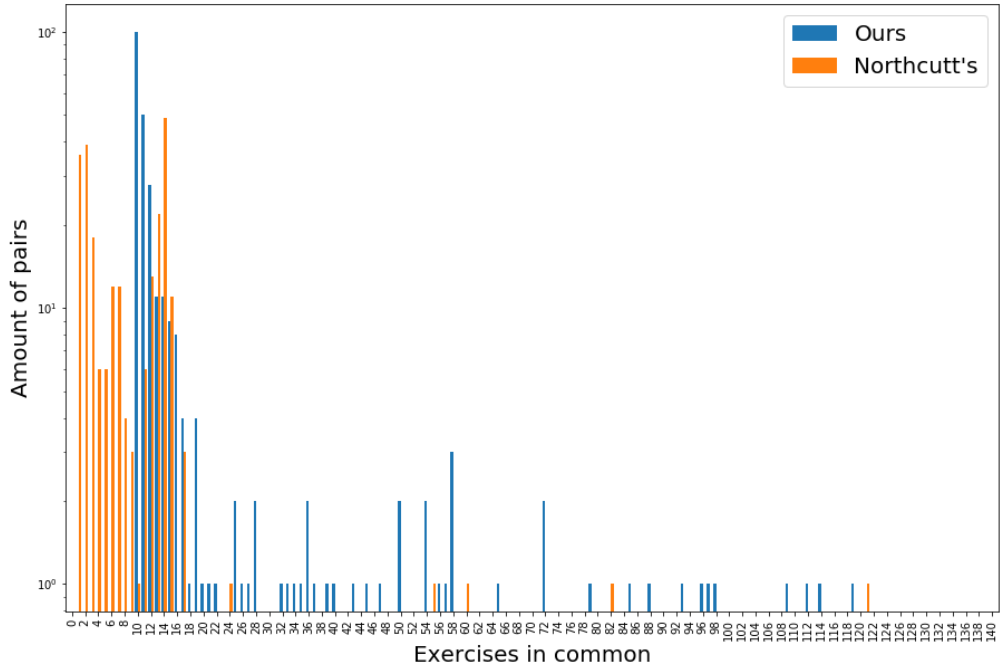

Figure 4.4: Number of pairs of accounts with exercises in common under a time tolerance of 5 minutes

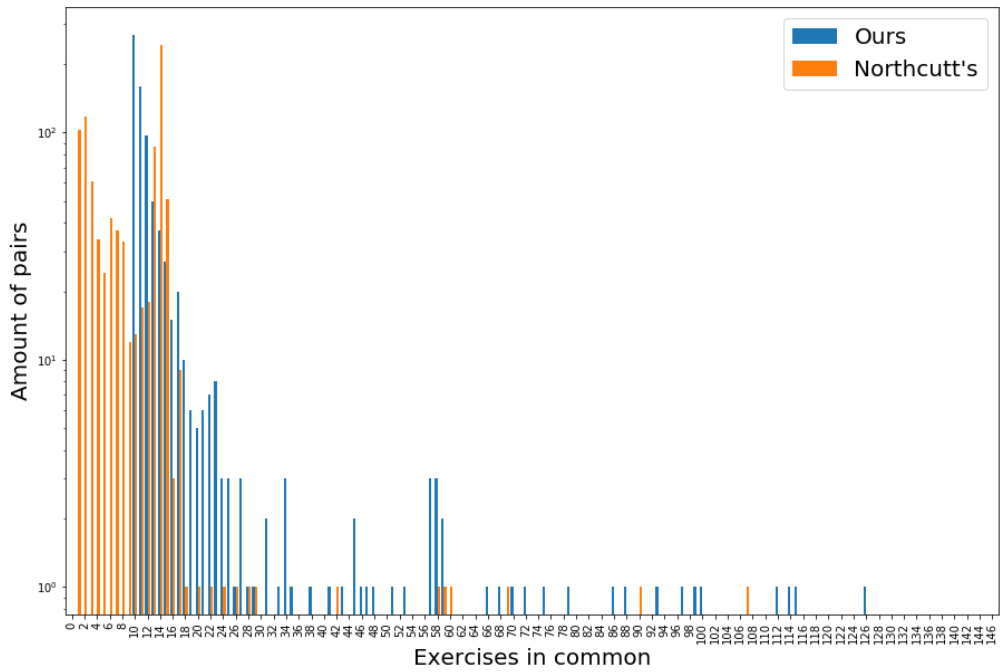

Figure 4.5: Number of pairs of accounts with exercises in common under a time tolerance of 10 minutes

Table 4.4: Amount of suspicious users found in each of the approaches

	Time window (minutes)			
Suspicious Users	1	2	5	10
Our approach	53	74	271	769
Northcutt's	4	22	246	918
Difference	-92%	-70%	-9.2%	19.3%

Table 4.5: Percentage of suspicious users found in each of the approaches

	Time window (part from total)			
	1	2	5	10
Portion from total users (Our approach)	0.73%	1.03%	3.77%	10.72%
Portion from total users (Northcutt)	0.05%	0.3%	3.43%	12.8%

orbits around 1% with a 5 minutes time window, which is similar to what we have found (3.43%). Although, again, it may be inflated due to a slight difference in the implementation or a particularity of the dataset.

4.3 New dataset

A new dataset, comprised of a different version of the same course, was obtained by the authors. This new dataset served as an entire new test target, which was classified with the model trained with the first dataset. The evaluation procedure is almost the same as before and is depicted in Figure 4.6: the first dataset is filtered to remove users without at least the minimum number of exercises, then features are extracted using the metrics explained in Chapter 3 (steps 1-3). After that, a k-means clustering algorithm is used to create the labels (step 4), and a nested cross-validation is used to find the best hyperparameters (step 5), while also estimating the generalization of the model (step 5.1). Finally, the model is trained with the first dataset using the hyperparameters (step 6) previously found and the model is stored (step 7). The mpodel is then tested using the second dataset and the results are analysed (steps 8-10).

Comparing the results obtained in the nested cross-validation, presented in Table 4.2, to the ones obtained by using the model over the new data, presented in Table 4.6, we can see that the results hold their accuracy and, therefore, the generalization of the model is present. This is particularly important, as the new data is totally unseen for the model and there is no risk of a leak when training, since both datasets are completely separated.

Table 4.6: Results on the new dataset

	Time windows (minutes)			
	1	2	5	10
Accuracy	97.12%	95.91%	97.14%	99.74%

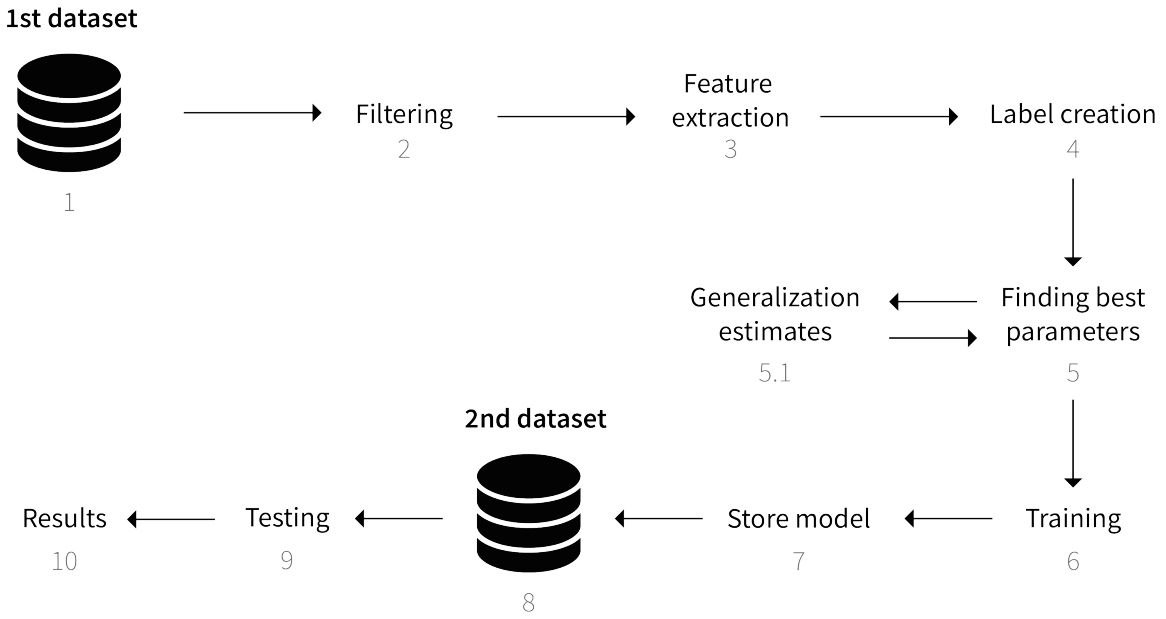

Figure 4.6: Scheme of the workflow developed to evaluate the second dataset

5 Discussion

In the presentation of this study on IEEE's EDUCON in 2020, some questions were asked regarding the application and the validation of this method. This chapter is devoted to address them.

5.1 Applications

One of the practical uses that could be employed for this study is to certify courses. Since there is a inherent problem in certifying that a student actually cheated, the first application could be diverted to a more subjective approach by providing a mean to indicate the percentage of users with suspicious activity in a course and display it to the future students, maybe printing it along the certificate. This is something that could try to circumvent the problem of identifying a specific user as a cheater and at the same time providing a more global perspective about how the course was designed and the trustworthiness of the certificates it generates. This, of course, incurs in a problem that if the number of participants in a course is small, a high percentage may be reported, but it could be expanded to a more broadened approach by providing data related to the amount of cheaters from the courses of each institute, which could give a more reasonable idea about the credibility of the course.

Another practical use could be to have it monitoring the students in the sense of a watchdog system. This proposes difficult questions:

- How would the system act when a cheater is identified?
  According to our discussions, we came to the conclusion that it would need to somehow propose new exercises for the user, generating exercises algorithmically, request an in-person final test or reveal the answers only after the course ends, for example. This leads us to our next question.

- How much a course can be interfered and still retain its MOOC status?
  Considering that generating exercises algorithmically can be challenging, as some exercises cannot be easily modified, this rapidly comes out as a problematic solution, leaving us with the others. When the definition of a MOOC is taken into account, we see that it is based in two primary principles: instant feedback and ease of access to spread education.

With that in mind, it is easy to realize that the other solutions clash with it, since it limits the spread of education when you need to potentially ask someone to take a test somewhere and it halts the learning progress when you need to wait for the course to finish to know the answers.

Given this scenario, a possible solution was to propose additional exercises for the student, should he be considered a suspicious user. But, then again, the student can simply replicate the behaviour and find the exercises on the harvesting account. The conclusions we draw from it are that no application will be perfect: should it be a certification or watchdog system. Another point to take into consideration is that CAMEO is not the only way of cheating, there are collaborations too and maybe other types, which limit even more these approaches. Therefore, a pilot test with users could give us a more palpable idea of how these systems could aid in the detection and prevention of cheating.

5.2 Validation

Another aspect approached in the conference was the validation of this model. At the time, there was not a new dataset, which meant that the only metrics we could give were estimates of the generalization of the model, as well as a test with some data holdout from the first dataset. As described in this thesis, the project has evolved since then and a new dataset became available to test the model. Therefore, this problem was addressed and discussed in Chapter 4.

Still, there is a different problem that is not constrained to this study, but to this area of study in general: establishing a ground truth. There is no clear way to say for certain that a student is cheating, unless we could rely on self-report from the student, but that is rather naive. A solution is proposed in this study, which is to use a clustering algorithm to predict the labels. However, basing the evaluation of our metrics and model on the predictions made over labels that were encountered by a clustering algorithm is debatable, as this is not a proper ground-truth and could also be listed as a limitation in this study. On the other hand, doing a manual analysis of the data is not different from finding it through clustering and also incredibly cumbersome. Basing ourselves on the literature shows that other studies (38), (9), (21), are using the same strategy. Northcutt's (8) study, however, appended a unique random string in the answers presented to every user. This string was a parentheses, a negligible decimal point at the end of a correct answer, or an expression that evaluates to 1. For example, an answer to the question "what is the final momentum of the particle?" could be 3.13, but the answer was displayed as "3.13556" to one user, and "3.13417" to another. By analysing the answers given by the students along with this modifications, they could verify the existence of CAMEO. Considering this, maybe a better alternative to verify the findings of this thesis would be to use the same strategy as Northcutt on the next version of the MOOC that originated our two datasets.

6
Conclusions

In this master thesis we present several metrics that allow us to identify two types of cheating behaviours from users in online courses. First, the proposed metrics are useful to individualize cheating users who take advantage of harvesting accounts to retrieve the correct answers to the exercises, as performed in other studies. In addition, these metrics are also valid to characterize another type of cheating, characterized by master accounts collaborating with each other, in order to obtain better grades for both.

The proposed metrics are computed for pairs of users and include: *user bias*, which measures the fraction of times a user answered before another and helps not only to identify which user copied from who but also to identify users who collaborate; *mMIR*, to detect users that do not interact with the materials; *final score difference*, used to identify users that are not interested in the final grade; and the *percentage of shared IP addresses* to help identifying harvesting accounts.

The user that utilizes a harvesting account to retrieve correct answers and the users that collaborate are identified by narrowing down the analysis, focusing on those accounts that answer the same exercises in a short time frame, while also applying K-means to the identified pairs of accounts using the proposed metrics as attributes. An SVM was applied to the data and resulted in an accuracy of over 95% under all tested settings when distinguishing between collaborators and harvesters.

When applying the model to the second dataset, trained on the first datset, the results are consistent with the nested cross-validation performed with the first dataset. This means that the model generalized the detection made from the first dataset to the second. However, it is important to note that the results tend to be optimistic, since the labels were obtained by running a clustering algorithm on the dataset, as there is currently no other way to do it when there is no self-declared cheating. On the other hand, the metrics seem consistent and an analysis on the behaviour of the students show that the detections make sense, especially on cases where there is no interaction with the course material and exercises are consistently answered right after another user.

Finally, more work needs to be carried to assess the potential of this model and its transition to a single user prediction, instead of pairs.

6.1 Suggested modifications

In a further analysis, we found that our approach finds more suspicious cases than Northcutt's (8) with smaller time windows, but the inverse is true for bigger windows, as depicted in the comparison between the two approaches in Chapter 4. This may indicate that a merge between the two techniques could enhance the detection of suspicious cases: as just an example, if the IP sharing rule from Northcutt changes to something more flexible, such as an indicator, and not an elimination clause, while also flexibilizing the strict 10 or 20 exercises rule for our algorithm, we believe that it could lead to an interesting merge from the two approaches.

One other modification is that this model should be tested with less data. Predicting cheating behaviour with all the data from the course is useful in its own ways, but in order to have a program that can monitor users in real time, this model needs to be modified. The way to start this would be to limit the data to 1 month after starting the course, which would present less data and, therefore, a bigger challenge to classify students into cheaters or not. However, a positive result in this scenario could lead to another, and potentially more valuable, evidence of the generalization of this model and the quality of the metrics.

7
Appendix

The following dump consists of a real example from the data gathered from a user. The identification of the user was the only thing that was removed.

7.1 Raw data from one user

"Usuario":"——-","Eventos":[
"evento": "load_video", "tiempo": "2015-02-24T15:30:21.298339+00:00", "id_video": "31",
"evento": "play_video", "tiempo": "2015-02-24T15:30:23.886024+00:00", "id_video": "31", "currentTime": "0",
"evento": "pause_video", "tiempo": "2015-02-24T15:30:24.863182+00:00", "id_video": "31", "currentTime": "0.073021",
"evento": "play_video", "tiempo": "2015-02-24T15:30:30.404615+00:00", "id_video": "31", "currentTime": "0.073021",
"evento": "pause_video", "tiempo": "2015-02-24T15:30:51.674912+00:00", "id_video": "31", "currentTime": "21.185001",
"evento": "stop_video", "tiempo": "2015-02-24T15:30:51.883413+00:00", "id_video": "31", "currentTime": "21.185001",
"evento": "load_video", "tiempo": "2015-02-24T15:31:04.417884+00:00", "id_video": "32",
"evento": "play_video", "tiempo": "2015-02-24T15:31:30.197024+00:00", "id_video": "32", "currentTime": "0",
"evento": "pause_video", "tiempo": "2015-02-24T15:32:14.834609+00:00", "id_video": "32", "currentTime": "41.321973",
"evento": "stop_video", "tiempo": "2015-02-24T15:32:15.425811+00:00", "id_video": "32", "currentTime": "41.321973",
"evento": "problem_check", "tiempo": "2015-02-24T15:32:49.680040+00:00", "id_problema": "1", "num_intentos": "1", "num_ejercicios": "1", "resultados": [
"id_ejercicio": "2_1", "correcto": "False", "respuesta": "cŏ0f3digo mŏ0e1quina"], "id_natural": "1",
"evento": "problem_check", "tiempo": "2015-02-24T15:33:16.298994+00:00", "id_problema": "2", "num_intentos": "1", "num_ejercicios": "1", "resultados": [
"id_ejercicio": "2_1", "correcto": "True", "respuesta": "cierto"], "id_natural": "2",

"evento": "problem_check", "tiempo": "2015-02-24T15:33:30.936247+00:00", "id_problema": "3", "num_intentos": "1", "num_ejercicios": "1", "resultados": [
"id_ejercicio": "2_1", "correcto": "False", "respuesta": "antes de la definici0̆0f3n de la clase."], "id_natural": "3",
"evento": "problem_check", "tiempo": "2015-02-24T15:33:43.019726+00:00", "id_problema": "4", "num_intentos": "1", "num_ejercicios": "1", "resultados": [
"id_ejercicio": "2_1", "correcto": "True", "respuesta": "transient"], "id_natural": "4",
"evento": "problem_check", "tiempo": "2015-02-24T15:33:58.750945+00:00", "id_problema": "5", "num_intentos": "1", "num_ejercicios": "1", "resultados": [
"id_ejercicio": "2_1", "correcto": "False", "respuesta": "definido fuera de una clase."], "id_natural": "5",
"evento": "problem_check", "tiempo": "2015-02-24T15:34:20.394101+00:00", "id_problema": "7", "num_intentos": "1", "num_ejercicios": "1", "resultados": [
"id_ejercicio": "2_1", "correcto": "True", "respuesta": "Versi0̆0f3n 1,Versi0̆0f3n 2"], "id_natural": "6",
"evento": "problem_check", "tiempo": "2015-02-24T15:34:37.758025+00:00", "id_problema": "14", "num_intentos": "1", "num_ejercicios": "1", "resultados": [
"id_ejercicio": "2_1", "correcto": "False", "respuesta": "A es un tipo especial de B."], "id_natural": "7",
"evento": "load_video", "tiempo": "2015-02-26T12:12:59.656281+00:00", "id_video": "32",
"evento": "load_video", "tiempo": "2015-02-26T12:13:06.462965+00:00", "id_video": "6",
"evento": "load_video", "tiempo": "2015-02-26T12:13:10.800942+00:00", "id_video": "1",
"evento": "play_video", "tiempo": "2015-02-26T12:13:14.371306+00:00", "id_video": "1", "currentTime": "0",
"evento": "problem_check", "tiempo": "2015-02-26T12:16:01.535568+00:00", "id_problema": "16", "num_intentos": "1", "num_ejercicios": "1", "resultados": [
"id_ejercicio": "2_1", "correcto": "True", "respuesta": "eclipse y plugin,sdk android"], "id_natural": "15",
"evento": "problem_check", "tiempo": "2015-02-26T12:16:10.084538+00:00", "id_problema": "16", "num_intentos": "2", "num_ejercicios": "1", "resultados": [
"id_ejercicio": "2_1", "correcto": "False", "respuesta": "eclipse y plugin"], "id_natural": "15",
"evento": "problem_check", "tiempo": "2015-02-26T12:16:30.571018+00:00", "id_problema": "15", "num_intentos": "1", "num_ejercicios": "1", "resultados": [
"id_ejercicio": "2_1", "correcto": "True", "respuesta": "Desarrollar para Android s0̆0f3lo es posible para plataformas Windows"], "id_natural": "16",
"evento": "problem_check", "tiempo": "2015-02-26T12:16:33.060210+00:00", "id_problema": "15", "num_intentos": "2", "num_ejercicios": "1", "resultados": [
"id_ejercicio": "2_1", "correcto": "True", "respuesta": "Desarrollar para Android s0̆0f3lo es posible para plataformas Windows"], "id_natural": "16",
"evento": "load_video", "tiempo": "2015-02-26T12:16:35.447280+00:00", "id_video": "2",
"evento": "problem_check", "tiempo": "2015-02-26T12:16:48.760128+00:00", "id_problema": "24", "num_intentos": "1", "num_ejercicios": "1", "resultados": [
"id_ejercicio": "2_1", "correcto": "True", "respuesta": "Un programa para descargar y actualizar los paquetes y herramientas del SDK"], "id_natural": "17",
"evento": "load_video", "tiempo": "2015-02-26T12:16:52.816874+00:00", "id_video": "3",
"evento": "problem_check", "tiempo": "2015-02-26T12:16:57.976665+00:00", "id_problema": "19", "num_intentos": "1", "num_ejercicios": "1", "resultados": [
"id_ejercicio": "2_1", "correcto": "True", "respuesta": "Un emulador que simula un dispositivo real"], "id_natural": "18",
"evento": "problem_check", "tiempo": "2015-02-26T12:17:08.973947+00:00", "id_problema": "17", "num_intentos": "1", "num_ejercicios": "1", "resultados": [

"id_ejercicio": "2_1", "correcto": "True", "respuesta": "Un emulador que simula un dispositivo m0̆0f3vil real y que permite ejecutar y probar las aplicaciones desarrolladas"], "id_natural": "19",
"evento": "problem_check", "tiempo": "2015-02-26T12:17:16.175746+00:00", "id_problema": "18", "num_intentos": "1", "num_ejercicios": "1", "resultados": [
"id_ejercicio": "2_1", "correcto": "True", "respuesta": "La carpeta donde se colocan los ficheros de todos los proyectos"], "id_natural": "20",
"evento": "load_video", "tiempo": "2015-02-26T12:17:20.040585+00:00", "id_video": "4",
"evento": "problem_check", "tiempo": "2015-02-26T12:17:39.028787+00:00", "id_problema": "20", "num_intentos": "1", "num_ejercicios": "1", "resultados": [
"id_ejercicio": "2_1", "correcto": "True", "respuesta": "La versi0̆0f3n m0̆0ednima del SDK en la que la aplicaci0̆0f3n puede funcionar"], "id_natural": "21",
"evento": "problem_check", "tiempo": "2015-02-26T12:17:48.964722+00:00", "id_problema": "21", "num_intentos": "1", "num_ejercicios": "1", "resultados": [
"id_ejercicio": "2_1", "correcto": "True", "respuesta": "Una clase de java que representa una pantalla de la aplicaci0̆0f3n"], "id_natural": "22",
"evento": "load_video", "tiempo": "2015-02-26T12:17:51.453854+00:00", "id_video": "5",
"evento": "problem_check", "tiempo": "2015-02-26T12:18:00.629356+00:00", "id_problema": "22", "num_intentos": "1", "num_ejercicios": "1", "resultados": [
"id_ejercicio": "2_1", "correcto": "False", "respuesta": "En la carpeta gen"], "id_natural": "23",
"evento": "problem_check", "tiempo": "2015-02-26T12:18:16.185574+00:00", "id_problema": "23", "num_intentos": "1", "num_ejercicios": "1", "resultados": [
"id_ejercicio": "2_1", "correcto": "True", "respuesta": "La carpeta src."], "id_natural": "24",
"evento": "problem_check", "tiempo": "2015-03-04T20:31:15.924372+00:00", "id_problema": "27", "num_intentos": "1", "num_ejercicios": "1", "resultados": [
"id_ejercicio": "2_1", "correcto": "True", "respuesta": "El zip sin limpiar ocupa entre 2 y 3 veces m0̆0e1s que el limpio"], "id_natural": "29",
"evento": "problem_check", "tiempo": "2015-03-04T20:32:17.604838+00:00", "id_problema": "31", "num_intentos": "1", "num_ejercicios": "1", "resultados": [
"id_ejercicio": "2_1", "correcto": "False", "respuesta": "ActividadPrincipal.java,actividad_principal.xml,strings
"id_natural": "30",
"evento": "load_video", "tiempo": "2015-03-04T20:37:13.866123+00:00", "id_video": "3",
"evento": "load_video", "tiempo": "2015-03-04T20:37:24.744484+00:00", "id_video": "7",
"evento": "load_video", "tiempo": "2015-03-04T20:49:37.317358+00:00", "id_video": "33",
"evento": "play_video", "tiempo": "2015-03-04T20:52:17.577736+00:00", "id_video": "33", "currentTime": "0",
"evento": "pause_video", "tiempo": "2015-03-04T20:53:28.934735+00:00", "id_video": "33", "currentTime": "71.18",
"evento": "stop_video", "tiempo": "2015-03-04T20:53:29.136482+00:00", "id_video": "33", "currentTime": "71.18",
"evento": "load_video", "tiempo": "2015-03-04T20:53:32.789126+00:00", "id_video": "8",
"evento": "load_video", "tiempo": "2015-03-05T16:42:11.355972+00:00", "id_video": "31",
"evento": "load_video", "tiempo": "2015-03-05T16:42:25.717470+00:00", "id_video": "32",
"evento": "load_video", "tiempo": "2015-03-05T16:42:46.465636+00:00", "id_video": "3",
"evento": "play_video", "tiempo": "2015-03-05T17:14:29.919708+00:00", "id_video": "3", "currentTime": "0",
"evento": "load_video", "tiempo": "2015-03-05T17:14:42.899498+00:00", "id_video": "1",
"evento": "play_video", "tiempo": "2015-03-05T17:14:45.280333+00:00", "id_video": "1", "currentTime": "148",
"evento": "load_video", "tiempo": "2015-03-05T17:15:26.268243+00:00", "id_video": "2",
"evento": "play_video", "tiempo": "2015-03-05T17:15:28.754890+00:00", "id_video": "2", "cur-

rentTime": "0",
"evento": "pause_video", "tiempo": "2015-03-05T17:17:42.408147+00:00", "id_video": "2", "currentTime": "132.217",
"evento": "stop_video", "tiempo": "2015-03-05T17:17:42.468791+00:00", "id_video": "2", "currentTime": "132.217",
"evento": "play_video", "tiempo": "2015-03-05T17:17:42.536687+00:00", "id_video": "2", "currentTime": "132.217",
"evento": "seek_video", "tiempo": "2015-03-05T17:17:42.729183+00:00", "old_time": "91", "new_time": "0.419", "id_video": "2",
"evento": "play_video", "tiempo": "2015-03-05T17:17:42.785159+00:00", "id_video": "2", "currentTime": "0.419",
"evento": "load_video", "tiempo": "2015-03-05T17:17:59.262030+00:00", "id_video": "4",
"evento": "play_video", "tiempo": "2015-03-05T17:23:57.130456+00:00", "id_video": "4", "currentTime": "208",
"evento": "pause_video", "tiempo": "2015-03-05T17:23:57.951608+00:00", "id_video": "4", "currentTime": "208.917",
"evento": "stop_video", "tiempo": "2015-03-05T17:23:58.265605+00:00", "id_video": "4", "currentTime": "208.917",
"evento": "play_video", "tiempo": "2015-03-05T17:24:00.384284+00:00", "id_video": "4", "currentTime": "208.917",
"evento": "play_video", "tiempo": "2015-03-05T17:24:01.424517+00:00", "id_video": "4", "currentTime": "0",
"evento": "pause_video", "tiempo": "2015-03-05T17:25:04.686325+00:00", "id_video": "4", "currentTime": "62.238",
"evento": "play_video", "tiempo": "2015-03-05T17:25:05.159766+00:00", "id_video": "4", "currentTime": "62.238",
"evento": "pause_video", "tiempo": "2015-03-05T17:25:05.935520+00:00", "id_video": "4", "currentTime": "63.557",
"evento": "play_video", "tiempo": "2015-03-05T17:25:44.026242+00:00", "id_video": "4", "currentTime": "63.557",
"evento": "seek_video", "tiempo": "2015-03-05T17:26:17.960539+00:00", "old_time": "97.319", "new_time": "82", "id_video": "4",
"evento": "play_video", "tiempo": "2015-03-05T17:26:18.379082+00:00", "id_video": "4", "currentTime": "97.319",
"evento": "pause_video", "tiempo": "2015-03-05T17:27:45.278759+00:00", "id_video": "4", "currentTime": "167.638",
"evento": "load_video", "tiempo": "2015-03-05T17:32:44.690732+00:00", "id_video": "5",
"evento": "load_video", "tiempo": "2015-03-05T17:33:00.578916+00:00", "id_video": "7",
"evento": "play_video", "tiempo": "2015-03-05T17:34:41.263433+00:00", "id_video": "7", "currentTime": "14",
"evento": "speed_change_video", "tiempo": "2015-03-05T17:36:35.203007+00:00", "new_speed": "1.50", "currentTime": "127.998", "old_speed": "1.0", "id_video": "7",
"evento": "pause_video", "tiempo": "2015-03-05T17:36:36.198491+00:00", "id_video": "7", "currentTime": "129.757",
"evento": "play_video", "tiempo": "2015-03-05T17:36:38.192963+00:00", "id_video": "7", "currentTime": "129.757",
"evento": "pause_video", "tiempo": "2015-03-05T17:36:41.066787+00:00", "id_video": "7", "currentTime": "134.437",
"evento": "seek_video", "tiempo": "2015-03-05T17:38:27.010401+00:00", "old_time": "134.437", "new_time": "125", "id_video": "7",
"evento": "seek_video", "tiempo": "2015-03-05T17:38:27.711867+00:00", "old_time": "125",

"new_time": "117", "id_video": "7",
"evento": "play_video", "tiempo": "2015-03-05T17:38:28.204123+00:00", "id_video": "7", "currentTime": "117",
"evento": "speed_change_video", "tiempo": "2015-03-05T17:38:30.612299+00:00", "new_speed": "1.0", "currentTime": "119.937", "old_speed": "1.50", "id_video": "7",
"evento": "pause_video", "tiempo": "2015-03-05T17:38:34.850120+00:00", "id_video": "7", "currentTime": "124.428",
"evento": "play_video", "tiempo": "2015-03-05T17:38:41.437329+00:00", "id_video": "7", "currentTime": "124.428",
"evento": "pause_video", "tiempo": "2015-03-05T17:38:42.344521+00:00", "id_video": "7", "currentTime": "125.528",
"evento": "play_video", "tiempo": "2015-03-05T17:38:43.401129+00:00", "id_video": "7", "currentTime": "125.528",
"evento": "pause_video", "tiempo": "2015-03-05T17:38:43.802745+00:00", "id_video": "7", "currentTime": "125.668",
"evento": "play_video", "tiempo": "2015-03-05T17:38:44.499876+00:00", "id_video": "7", "currentTime": "125.668",
"evento": "pause_video", "tiempo": "2015-03-05T17:38:48.696648+00:00", "id_video": "7", "currentTime": "130.128",
"evento": "seek_video", "tiempo": "2015-03-05T17:39:20.449114+00:00", "old_time": "130.128", "new_time": "120", "id_video": "7",
"evento": "play_video", "tiempo": "2015-03-05T17:39:22.348010+00:00", "id_video": "7", "currentTime": "120",
"evento": "pause_video", "tiempo": "2015-03-05T17:39:26.060585+00:00", "id_video": "7", "currentTime": "123.439",
"evento": "play_video", "tiempo": "2015-03-05T17:39:43.381509+00:00", "id_video": "7", "currentTime": "123.439",
"evento": "seek_video", "tiempo": "2015-03-05T17:42:53.120722+00:00", "old_time": "312.518", "new_time": "198", "id_video": "7",
"evento": "play_video", "tiempo": "2015-03-05T17:42:53.125455+00:00", "id_video": "7", "currentTime": "312.518",
"evento": "seek_video", "tiempo": "2015-03-05T17:42:58.756270+00:00", "old_time": "213", "new_time": "203.698", "id_video": "7",
"evento": "play_video", "tiempo": "2015-03-05T17:42:58.819481+00:00", "id_video": "7", "currentTime": "203.937",
"evento": "speed_change_video", "tiempo": "2015-03-05T17:43:04.248089+00:00", "new_speed": "1.50", "currentTime": "218.077", "old_speed": "1.0", "id_video": "7",
"evento": "pause_video", "tiempo": "2015-03-05T17:45:28.639830+00:00", "id_video": "7", "currentTime": "434.257",
"evento": "stop_video", "tiempo": "2015-03-05T17:45:28.695803+00:00", "id_video": "7", "currentTime": "434.257",
"evento": "problem_check", "tiempo": "2015-03-05T18:31:50.631130+00:00", "id_problema": "32", "num_intentos": "1", "num_ejercicios": "3", "resultados": [
"id_ejercicio": "3_1", "correcto": "False", "respuesta": "@string/etiqueta_boton1",
"id_ejercicio": "2_1", "correcto": "True", "respuesta": "etiqueta_boton1",
"id_ejercicio": "4_1", "correcto": "False", "respuesta": "R.string.etiqueta_boton1"], "id_natural": "31",
"evento": "problem_check", "tiempo": "2015-03-05T18:41:12.583146+00:00", "id_problema": "32", "num_intentos": "2", "num_ejercicios": "3", "resultados": [
"id_ejercicio": "3_1", "correcto": "True", "respuesta": "@string/hola_mundo",
"id_ejercicio": "2_1", "correcto": "True", "respuesta": "etiqueta_boton1",

"id_ejercicio": "4_1", "correcto": "False", "respuesta": "R.layout.activity_main"], "id_natural": "31",
"evento": "problem_check", "tiempo": "2015-03-05T18:42:25.187771+00:00", "id_problema": "33", "num_intentos": "1", "num_ejercicios": "2", "resultados": [
"id_ejercicio": "3_1", "correcto": "True", "respuesta": "21",
"id_ejercicio": "2_1", "correcto": "True", "respuesta": "15"], "id_natural": "32",
"evento": "problem_check", "tiempo": "2015-03-05T18:42:28.485690+00:00", "id_problema": "33", "num_intentos": "2", "num_ejercicios": "2", "resultados": [
"id_ejercicio": "3_1", "correcto": "True", "respuesta": "21",
"id_ejercicio": "2_1", "correcto": "True", "respuesta": "15"], "id_natural": "32",
"evento": "problem_check", "tiempo": "2015-03-05T18:45:13.202849+00:00", "id_problema": "34", "num_intentos": "1", "num_ejercicios": "3", "resultados": [
"id_ejercicio": "3_1", "correcto": "True", "respuesta": "2",
"id_ejercicio": "2_1", "correcto": "True", "respuesta": "6",
"id_ejercicio": "4_1", "correcto": "True", "respuesta": "5"], "id_natural": "33",
"evento": "problem_check", "tiempo": "2015-03-05T18:45:16.129198+00:00", "id_problema": "34", "num_intentos": "2", "num_ejercicios": "3", "resultados": [
"id_ejercicio": "3_1", "correcto": "True", "respuesta": "2",
"id_ejercicio": "2_1", "correcto": "True", "respuesta": "6",
"id_ejercicio": "4_1", "correcto": "True", "respuesta": "5"], "id_natural": "33",
"evento": "problem_check", "tiempo": "2015-03-05T18:47:27.292951+00:00", "id_problema": "28", "num_intentos": "1", "num_ejercicios": "1", "resultados": [
"id_ejercicio": "2_1", "correcto": "False", "respuesta": "[
1:src,
3:gen,
5:drawable,
4:layout,
2:values,
7:values]"], "id_natural": "34",
"evento": "problem_check", "tiempo": "2015-03-05T18:48:01.886651+00:00", "id_problema": "28", "num_intentos": "2", "num_ejercicios": "1", "resultados": [
"id_ejercicio": "2_1", "correcto": "True", "respuesta": "[
6:raiz,
1:src,
3:gen,
5:drawable,
4:layout,
2:values,
7:values]"], "id_natural": "34",
"evento": "problem_check", "tiempo": "2015-03-05T18:51:46.620046+00:00", "id_problema": "35", "num_intentos": "1", "num_ejercicios": "2", "resultados": [
"id_ejercicio": "3_1", "correcto": "False", "respuesta": "GetResources(R.id.otra_etiqueta_boton2)",
"id_ejercicio": "2_1", "correcto": "True", "respuesta": "findViewById(R.id.button2)"], "id_natural": "35",
"evento": "problem_check", "tiempo": "2015-03-05T18:52:34.613498+00:00", "id_problema": "35", "num_intentos": "2", "num_ejercicios": "2", "resultados": [
"id_ejercicio": "3_1", "correcto": "False", "respuesta": "getResources(R.id.otra_etiqueta_boton2)",
"id_ejercicio": "2_1", "correcto": "True", "respuesta": "findViewById(R.id.button2)"], "id_natural": "35",
"evento": "problem_check", "tiempo": "2015-03-05T19:14:36.619361+00:00", "id_problema": "36", "num_intentos": "1", "num_ejercicios": "1", "resultados": [

"id_ejercicio": "2_1", "correcto": "True", "respuesta": "[216,174]"], "id_natural": "36", "evento": "problem_check", "tiempo": "2015-03-05T19:16:00.928183+00:00", "id_problema": "38", "num_intentos": "1", "num_ejercicios": "1", "resultados": [
"id_ejercicio": "2_1", "correcto": "True", "respuesta": "[427,238]"], "id_natural": "38", "evento": "problem_check", "tiempo": "2015-03-05T19:16:05.686936+00:00", "id_problema": "37", "num_intentos": "1", "num_ejercicios": "1", "resultados": [
"id_ejercicio": "2_1", "correcto": "False", "respuesta": "[450,293]"], "id_natural": "37", "evento": "problem_check", "tiempo": "2015-03-05T19:16:07.923546+00:00", "id_problema": "37", "num_intentos": "2", "num_ejercicios": "1", "resultados": [
"id_ejercicio": "2_1", "correcto": "True", "respuesta": "[637,326]"], "id_natural": "37", "evento": "problem_check", "tiempo": "2015-03-05T19:16:24.683738+00:00", "id_problema": "41", "num_intentos": "1", "num_ejercicios": "1", "resultados": [
"id_ejercicio": "2_1", "correcto": "True", "respuesta": "[138,287]"], "id_natural": "39", "evento": "problem_check", "tiempo": "2015-03-05T19:17:18.873101+00:00", "id_problema": "40", "num_intentos": "1", "num_ejercicios": "1", "resultados": [
"id_ejercicio": "2_1", "correcto": "True", "respuesta": "colors.xml"], "id_natural": "41"]

Bibliography

[1] M. Gaebel, *MOOCs: Massive open online courses.* European University Association, 2014.

[2] . Fidalgo-Blanco, M. L. Sein-Echaluce, and F. J. García-Peñalvo, "From massive access to cooperation: lessons learned and proven results of a hybrid xMOOC/cMOOC pedagogical approach to MOOCs," *International Journal of Educational Technology in Higher Education*, vol. 13, no. 1, 2016.

[3] C. R. Glass, M. S. Shiokawa-Baklan, and A. J. Saltarelli, "Who Takes MOOCs?" *New Directions for Institutional Research*, vol. 2015, no. 167, pp. 41–55, 4 2016. [Online]. Available: http://doi.wiley.com/10.1002/ir.20153

[4] K. Li, "MOOC learners demographics, self-regulated learning strategy, perceived learning and satisfaction: A structural equation modeling approach," *Computers and Education*, vol. 132, pp. 16–30, 4 2019.

[5] Q. Zhang, F. C. Bonafini, B. B. Lockee, K. W. Jablokow, and X. Hu, "Exploring Demographics and Students' Motivation as Predictors of Completion of a Massive Open Online Course," *International Review of Research in Open and Distance Learning*, vol. 20, no. 2, pp. 140–161, 2019.

[6] D. L. R. Jones, "Academic dishonesty: are more students cheating?" *Business Communication Quarterly*, vol. 74, no. 2, pp. 141–150, 2011. [Online]. Available: http://journals.sagepub.com/doi/10.1177/1080569911404059

[7] S. W. Tabsh, H. A. El Kadi, and A. S. Abdelfatah, "Faculty perception of engineering student cheating and effective measures to curb it," in *IEEE Global Engineering Education Conference, EDUCON*, vol. April-2019. IEEE Computer Society, 2019, pp. 806–810.

[8] C. G. Northcutt, A. D. Ho, and I. L. Chuang, "Detecting and preventing "multiple-account" cheating in massive open online courses," *Computers and Education*, vol. 100, pp. 71–80, 2016.

[9] G. Alexandron, J. A. Ruipérez-Valiente, Z. Chen, P. J. Muñoz-Merino, and D. E. Pritchard, "Copying@Scale: Using Harvesting Accounts for Collecting Correct Answers in a MOOC," *Computers and Education*, vol. 108, pp. 96–114, 5 2017.

[10] . Pérez-Lemonche, G. Martínez-Muñoz, and E. Pulido-Cañabate, "Analysing event transitions to discover student roles and predict grades in MOOCs," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 10614 LNCS. Springer Verlag, 2017, pp. 224–232.

[11] R. Ferguson and D. Clow, "Examining engagement: Analysing learner subpopulations in massive open online courses (MOOCs)," in *ACM International Conference Proceeding Series*, vol. 16-20-Marc. Association for Computing Machinery, 2015, pp. 51–58.

[12] I. Chuang, "HarvardX and MITx: four years of open online courses – fall 2012 - summer 2016," *SSRN Electronic Journal*, 2017.

[13] F. M. Loschiavo and M. A. Shatz, "The Impact of an Honor Code on Cheating in Online Courses," Tech. Rep. 2, 2011.

[14] A. M. Kaplan and M. Haenlein, "Higher education and the digital revolution: About MOOCs, SPOCs, social media, and the Cookie Monster," *Business Horizons*, vol. 59, no. 4, pp. 441–450, 7 2016.

[15] T. Eriksson, T. Adawi, and C. Stöhr, "Time is the bottleneck: a qualitative study exploring why learners drop out of MOOCs," *Journal of Computing in Higher Education*, vol. 29, no. 1, pp. 133–146, 2017.

[16] D. Clow, "MOOCs and the funnel of participation," in *ACM International Conference Proceeding Series*, 2013, pp. 185–189.

[17] T. R. Liyanagunawardena, P. Parslow, and S. Williams, "Dropout: MOOC participants perspective," in *Proceedings of EMOOCs 2014, the Second MOOC European Stakeholders Summit*, 2014.

[18] S. Assami, N. Daoudi, and R. Ajhoun, "Personalization criteria for enhancing learner engagement in MOOC platforms," in *IEEE Global Engineering Education Conference, EDUCON*, vol. 2018-April. IEEE Computer Society, 2018, pp. 1265–1272.

[19] A. Ortega-Arranz, M. Kalz, and A. Martinez-Mones, "Creating engaging experiences in MOOCs through in-course redeemable rewards," in *IEEE Global Engineering Education Conference, EDUCON*, vol. 2018-April. IEEE Computer Society, 2018, pp. 1875–1882.

[20] Y. Bao, G. Chen, C. Hauff, and Y. Bao, "On the prevalence of multiple-account cheating in massive open online learning: a replication study," Delft University of Technology, Netherlands, Tech. Rep., 2017.

[21] J. A. Ruiperez-Valiente, P. J. Munoz-Merino, G. Alexandron, and D. E. Pritchard, "Using machine learning to detect 'multiple-account' cheating and analyze the influence of student and problem features," *IEEE Transactions on Learning Technologies*, vol. 12, no. 1, pp. 112–122, 2019.

[22] G. Alexandron, S. Lee, Z. Chen, and D. E. Pritchard, "Detecting cheaters in MOOCs using item response theory and learning analytics," in *User Modeling, Adaptation and Personalization (UMAP)*, 2016.

[23] J. A. Ruipérez-Valiente, D. Gašević, S. Joksimović, V. Kovanović, P. J. Muñoz-Merino, and C. D. Kloos, "A data-driven method for the detection of close submitters in online learning environments," in *26th International World Wide Web Conference 2017, WWW 2017 Companion*. International World Wide Web Conferences Steering Committee, 2019, pp. 361–368.

[24] G. Alexandron and D. E. Pritchard, "Towards a general purpose anomaly detection method to identify cheaters in massive open online courses," in *Proceedings of the 12th international conference on educational data mining*, 2019, pp. 480–483.

[25] D. J. Hand, "Principles of data mining," in *Drug Safety*, vol. 30, no. 7. Springer, 11 2007, pp. 621–622.

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

[27] ——, "Educational data mining: A survey from 1995 to 2005," *Expert Systems with Applications*, vol. 33, no. 1, pp. 135–146, 7 2007.

[28] A. P. Sanjeev and J. M. Iytkowt, "Discovering Enrollment Knowledge in University Databases," in *First International Conference on Knowledge Discovery and Data Mining*, AAAI, Ed. ACM, 1995, pp. 246–251. [Online]. Available: www.aaai.org

[29] "Creating a Pandas DataFrame - GeeksforGeeks." [Online]. Available: https://www.geeksforgeeks.org/creating-a-pandas-dataframe/

[30] C. M. Bishop, *Pattern Recoginiton and Machine Learning*. Springer-Verlag New York, 2006. [Online]. Available: https://www.springer.com/gp/book/9780387310732http://users.isr.ist.utl.pt/~wurmd/Livros/school/Bishop-PatternRecognitionAndMachineLearning-Springer2006.pdf

[31] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 9 1995.

[32] "Support vector machine - Wikipedia." [Online]. Available: https://en.wikipedia.org/wiki/Support_vector_machine

[33] "Kernel method - Wikipedia." [Online]. Available: https://en.wikipedia.org/wiki/Kernel_method#Mathematics:_the_kernel_trick

[34] S. P. Lloyd, "Least Squares Quantization in PCM," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137, 1982.

[35] Z. Wen, H. Pei, H. Liu, and Z. Yue, "A Sequential Kriging reliability analysis method with characteristics of adaptive sampling regions and parallelizability," *Reliability Engineering and System Safety*, vol. 153, pp. 170–179, 9 2016.

[36] G. C. Cawley and N. L. C. Talbot, "On Over-fitting in Model Selection and Subsequent Selection Bias in Performance Evaluation," Tech. Rep., 2010.

[37] "Cross-validation (statistics) - Wikipedia." [Online]. Available: https://en.wikipedia.org/wiki/Cross-validation_(statistics)

[38] J. A. Ruiperez-Valiente, G. Alexandron, Z. Chen, and D. E. Pritchard, "Using multiple accounts for harvesting solutions in MOOCs," in *L@S 2016 - Proceedings of the 3rd 2016 ACM Conference on Learning at Scale*. New York, New York, USA: Association for Computing Machinery, Inc, 4 2016, pp. 63–70. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2876034.2876037