



Universidad Autónoma
de Madrid

Biblos-e Archivo
Repositorio Institucional UAM

Repositorio Institucional de la Universidad Autónoma de Madrid

<https://repositorio.uam.es>

Esta es la **versión de autor** del artículo publicado en:
This is an **author produced version** of a paper published in:

International Symposium on Computers in Education (SIIE), Jerez,
Spain, 2018

DOI: <https://doi.org/10.1109/SIIE.2018.8586748>

Copyright: © 2018 IEEE

El acceso a la versión del editor puede requerir la suscripción del recurso

Access to the published version may require subscription

Dropout Detection in MOOCs: An Exploratory Analysis

Cristina Isidro
Computer Science Department
Universidad Autonoma de Madrid
Madrid, Spain
cristina.isidro@estudiante.uam.es

Rosa M. Carro
Computer Science Department
Universidad Autonoma de Madrid
Madrid, Spain
rosa.carro@uam.es

Alvaro Ortigosa
Computer Science Department
Universidad Autonoma de Madrid
Madrid, Spain
alvaro.ortigosa@uam.es

Abstract—The presence of MOOCs has increased exponentially in the context of distance education. However, many students who enroll in this type of courses drop out before completion. Several circumstances may cause dropout at any stage of the course and, in any case, before getting the certificate. Some students are not able to follow the course, others fail in the exams and leave, others only aim at getting a general overview of the course contents, others take all the activities but the final test, because of being interested in gaining knowledge but not in obtaining the certificate, etc. For this reason, it is interesting to analyze and understand the behavior of each student while interacting with the course. In this direction, the goal of this work is to predict whether a student will abandon a MOOC before completing it, so that it is possible to intervene accordingly, by warning the teacher about the dropout risk, notifying the student about this risk, etc. Different machine learning techniques have been tested with real data of a MOOC supported by EdX at UAM. In this article, the results of the work carried out in this direction are presented.

Keywords—Learning analytics, MOOCs, machine learning

I. MOTIVATION AND GOALS

The relevance of MOOCs (Massive Open Online Courses) in the field of education [1] is continuously growing up. Many universities around the world offer free courses in the form of MOOCs, and more and more people are enrolling in them. These courses are usually composed of texts, graphics, videos, individual or collaborative exercises, exams, etc. They can also contain discussion and help forums that provide the possibility of building a community of students and teachers. MOOCs offer new learning opportunities to people without the resources needed to make an economic investment in training.

The fact that MOOCs are open online courses means that students are completely free for their achievement. Each of them can carry out the activities at any time and learn at his own pace. Sometimes the courses are completely open since the beginning and others the lessons are gradually accessible, at the time that teachers consider appropriate. In any case, the student may choose either acquiring knowledge without taking exams or taking exams to obtain the corresponding course certificate.

In MOOCs, student dropouts are massive. It is said that a student abandons a MOOC when he has not finished the activities and does not get the certificate. However, this situation can occur either because the student did not carry out the activities or lost interest or because the student considered it unnecessary to take the exams, even having actually acquired the corresponding knowledge. Therefore, it is desirable to know the actions made by each student along with his evolution to be able to detect and predict the real

risk of failure or dropout. For example, when a student answers an exercise wrongly in a MOOC, it may be difficult to know if he was really interested in it but could not solve it correctly, or if he clicked any option to go on with the next contents. In order to avoid misinterpretations of this type of events, it is convenient to analyze all the users' activity.

This work also explores the possibility that the person in charge of a MOOC can receive feedback on situations of dropout or failure risk, so that it is possible to offer, if desired, some kind of help to these students so that they are retained and even get the certificate. In addition, another goal deals with the possibility of automatically detecting whether there exist factors that favor dropout and at which points of the courses they are, if any.

During the interaction of students with MOOCs, a large amount of information is generated. This information can be analyzed to draw conclusions about the students and their learning processes [2]. There are many platforms hosting MOOCs, such as INTEF [3], Miriada X [4], Udacity [5], Udemy [6], Coursera [7] or edX [8]. The analysis of data produced by different virtual learning platforms can be a powerful tool to improve online teaching and learning processes.

On the other hand, advances in the field of machine learning are continuously occurring. In particular, deep learning techniques have become very popular recently among the data analysis community [9] [10]. For this reason, it seems interesting to explore the possibility of analyzing the students' activity using this type of algorithms (which, up to our knowledge, has not been explored for the moment in the field of learning analytics) and investigate whether they provide an improvement compared to conventional machine learning algorithms. Deep learning algorithms use logical structures that resemble the organization of the nervous system of mammals, having layers of process units that mimic the functioning of neural networks and specialize in detecting certain characteristics of the perceived objects.

When using deep learning algorithms, it is not possible to know the reasons behind the decisions made; that is, the explanation of why some data is classified in one way or another is not known. This would make difficult to explain why a student is considered to be at risk of dropout or failure. However, if those algorithms offered significantly better results than the more conventional ones, it could be worth using them despite this difficulty.

In this context, the main goal of this paper is to propose a method to provide feedback on student performance in MOOCs as well as clues of potential problems or undesirable situations identified in such courses, using traditional machine learning algorithms and investigating the possibility of improving results using deep learning. With this goal, 1)

the records of the events generated by the users' activity in MOOCs have been processed to extract relevant variables for the analysis; 2) risk factors of dropout or failure have been identified; 3) a proposal has been developed to detect and predict student risk situations; 4) the results obtained using different machine learning algorithms have been compared. This article describes each of these activities, refers to other initiatives in this context and presents the conclusions of the work done as well as some proposals for future work.

II. LEARNING ANALYTICS

The concept of learning analytics refers to the analysis and presentation of student data, including their personal contexts, in order to understand and improve their learning [11]. This analysis can provide interesting information on different issues. From a MOOC-based learning point of view, this information can relate to the sequence of activities carried out during an online session, the time that the student has been active in the MOOC, the results of the activities carried out, the relationships between students of the same course or other data of interest for the analysis.

Within this area, there is a need to use computational methods to perform the data analysis needed to try to understand what happens during each student's learning process, starting from all the data collected about his interactions within the MOOCs. These methods follow different approaches depending on the context to analyze. For example, *network-analytics* methods focus on the relationships between different actors (friendship, professional, information sharing, etc.). These actors do not necessarily have to be students. Another possibility is representing all the course elements and analyzing the relationships between them that can provide interesting information regarding learning [12]. The approach called *Process-Oriented Interaction Analysis* bases on analyzing the student's interactions while taking the course. Its goal is trying to detect student behavioral patterns while learning by analyzing logs [13]. Finally, the *Content Analysis Using Text-Mining* methods base on the analysis of the textual contents generated by the students during the course (e.g., comments they write or doubts they express about a certain topic in the corresponding forums). With this type of analysis, it is possible not only to analyze the nature of the textual content, but also to extract semantic information related to the course topics, being able to classify them or to extract relationships between the words of different topics [14]. These three approaches offer the possibility of analyzing and understanding the activity carried out by the students in the context of distance learning. It is important to emphasize that the possibility of carrying out this type of analysis depends, among other factors, on: i) the type of activities that make up the MOOC and ii) the type of recording that the MOOC management system makes of these activities.

In the eMadrid research network, some work related to learning analytics has already been done. It is compiled and summarized in [15]. For example, a methodology has been proposed for the design and development of MOOCs based on the best practices identified during the analysis of the first MOOCs implemented in edX and MiriadaX. We have also worked on prediction and clustering models; analysis of social interactions while learning; and evaluation of different online educational experiences. Out of eMadrid context,

other works also aim to identify students at risk of not getting the course certificate. In [16] the focus is mainly on finding relevant patterns using association rules.

In this paper, we focus on the analysis of learning processes and data stored in MOOCs available through the edX platform. The goal is to detect those cases in which the students are in risk of either dropout or not obtaining the certificate. With this purpose, several traditional machine learning algorithms are used. In addition, the possibility of using deep learning is explored.

III. DATA ANALYSIS

In this work, we have analyzed the information available for the MOOC *La España del Quijote* of the Universidad Autonoma de Madrid, hosted on the edX platform [17]. In particular, we have analyzed, from all registered users, their demographic data, the final grade obtained, the comments they have written in the forums and every single event recorded, generated by each of their interactions within the course (there exist 21 types of possible events). Each day one file is generated in order to register all the events from the students connected to the course on that day. In this work, 728 files have been analyzed, which represent the activity of the students who have interacted with the course between September 2014 and February 2017.

A. Weekly Model Approach

One of the purposes of this work is to analyze the activity patterns of the students of MOOCs according to the results they obtain, in order to provide feedback to the teacher, which can be useful not only for the next course edition but also for the current one. The aim is to predict the results of each student according to his evolution, so that the interventions considered appropriate can be carried out. For example, it is possible to send an alert to the teacher, to the student or to both of them when risk of failing, being stuck or dropping out is detected.

To this end, the trajectory of each student has been analysed. We have taken one week as the unit of time when analyzing the events that occurred as well as the different paths followed by the students. One prediction model is created every week. In each model, the accumulated activity for a student from the course beginning until that week is considered. By generating and using prediction models weekly, it is possible to calculate the risk of dropout or failure for each student each week, based on all his previous activities up to that moment. Let us notice that the starting time may be different for each student, since the MOOC is open. This means that, e.g., the second week for two students can correspond to different weeks on the calendar, since it is calculated for each of them according to the date they enrolled in the course.

An analysis of the values of all the parameters for each student has been carried out in order to have an overview of the performance of the students enrolled in this MOOC. Figure 1 shows the distribution of students who passed the course according to the number of weeks they needed to complete it. As can be seen, there is a great diversity. It is worth mentioning that this MOOC is completely open and has no time limits either for each lesson or for the whole course. More than 100 users took between 0 and 10 weeks to complete the course. The bulk of the students took between 0

and 42 weeks to do it. There were few students who needed more than 42 weeks to complete it. According to these observations, the number of weeks to be considered for predictive model generation was set to 42. Therefore, 42 predictive models were generated for this MOOC.

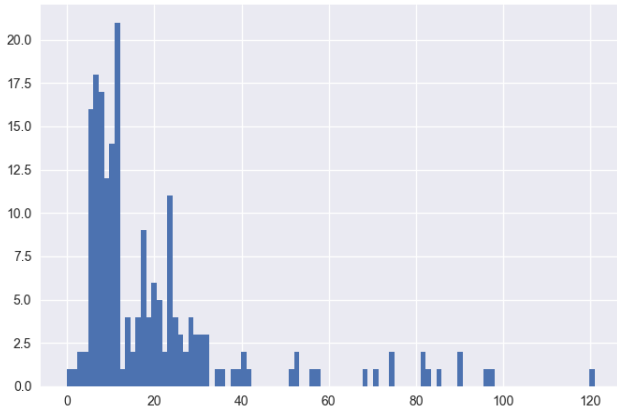


Fig. 1. Distribution of time spent taking the MOOC

B. Attribute Extraction

The first step towards building predictive models deals with the extraction of the variables and attributes to be used in these models. Getting these attributes is a neither simple nor direct procedure. The only source of data available to get all the information needed is the event log generated by edX. These records have not been designed to facilitate a data analysis such as the one intended in this work. Therefore, a first step to carry it out is to evaluate what information can be extracted from the log registry and how. The log is actually a list of events that can be considered low level, since they do not describe semantically significant milestones, but simply timestamps along with atomic events or clicks such as: navigate to a page, start watching a video, forward or backward in a video, send the answer to an exercise, etc. Using different Python libraries, 728 log files have been processed to extract variables with certain relevance for our analysis. The attributes extracted are:

TABLE I. ATTRIBUTES EXTRACTED FOR PREDICTIVE MODELS

n_videos	total number of videos that the student has visited
t_videos	total time in seconds that a user has been watching videos
n_mov_forward	number of forward movements a user has made while watching a video
n_mov_backward	number of backward movements that a user has made while watching a video
n_correct	number of hits of a user in total in the exams and exercises of the course
n_attempts	number of attempts of a user during the realization of the exams and exercises (there may be several attempts per exercise)
n_problems	number of exercises that a user has done
n_comments	number of comments a user has made in the forums
n_events	total number of events that a user has generated during his interaction with the course
n_sessions	total number of sessions in which a user has logged in
approved	indicates whether a user has passed the course; this is the class for supervised classifications

The attribute extraction has been done thinking of analyzing the causes or reasons why a user may pass a course or fail. The approved variable consists of a binary value representing whether the user passed (or is expected to pass) the course: 1 means passing, 0 means failing. Some other attributes can also be extracted, such as gender, age, geographical location or native language, but the stored value for the vast majority of them was NULL.

C. Analysis of Attribute Values

The analysis of the data extracted from the logs of this MOOC is presented next. The total number of users is 3353. Attribute values are widely dispersed: standard deviations are very large. For the 25th, 50th and 75th percentiles, many of the attributes are set to 0. This means that there is probably a large percentage of users who had very little activity throughout the course. The data collected indicate that only about 6% of the students enrolled in the course finally passed. The average number of videos watched by them is 5.59 (standard deviation 14.43). The student who watched the fewest videos has not accessed any, and the one who has seen the most has visualized 70. Focusing on the data of those who passed, the average number of videos watched per student is 46.78 (standard deviation 22.23). The student who watched fewer videos accessed none, while the one who watched the most viewed 70. 25% of the students watched 31 videos, 50% watched 55 and 75% watched 65.

The correlation between the different attributes and the “approved” class is shown in Table II. The attributes that have the highest correlation with this class are n_correct (0,943) and n_problems (0,926). This fits quite well with our intuition, as the student must answer correctly to different questions and exercises to pass. On the other hand, it can be observed that the variable n_videos also has a high correlation with the “approved” variable (0,716). This reflects that most of the students that passed watched most of the videos and, therefore, indicates that videos play an important role in this course. Two other variables with a similar correlation are n_events (0,748) and n_sesiones (0,653), which directly relates the activity that the student has in the course with the final grade he obtains.

TABLE II. CORRELATION BETWEEN THE PASSING CLASS AND THE DIFFERENT ATTRIBUTES

<i>n_videos</i>	<i>t_videos</i>	<i>n_mov_haciaadelante</i>		<i>n_mov_haciaatras</i>	
0.716989	0.425424	0.207458		0.351697	
<i>n_aciertos</i>	<i>n_intentos</i>	<i>n_problemas</i>		<i>n_comentarios</i>	
0.943520	0.355386	0.926379		0.251459	
<i>usr_mac</i>	<i>usr_windows</i>	<i>usr_linux</i>	<i>n_eventos</i>	<i>n_sesiones</i>	
-0.013628	0.090195	0.083082	0.748162	0.653066	

IV. RESULTS

The comparison between the results obtained when applying different machine learning algorithms (including those for deep learning) is presented next.

A. Naïve Bayes

The matrix of confusion shown in Figure 2 is obtained from the application of the Naïve Bayes algorithm for the classification of students into two classes: those who passed and those who failed.

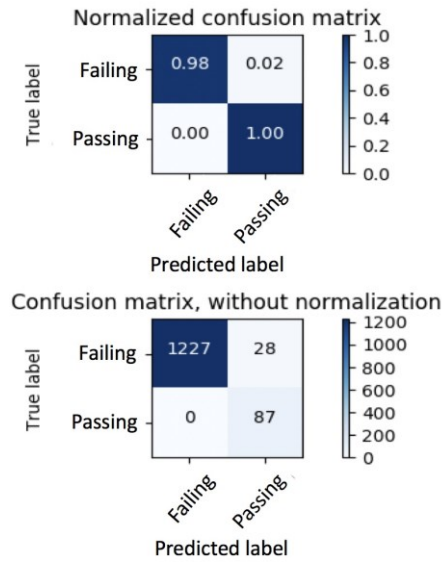


Fig. 2. Naïve Bayes confusion matrix

As can be seen, this algorithm classifies correctly almost all the failing cases and all the passing ones. In addition, when calculating the accuracy and completeness of the failing class, values of 1 and 0.98 are obtained, respectively. This means that, on the one hand, it classifies all the passing cases correctly; on the other hand, 2% of the failing ones are classified as passing. The accuracy and completeness for the passing class are calculated too.

B. Decision Trees

The algorithm used to build decision trees was CART (Classification And Regression Trees). In addition, we used decision trees in conjunction with the AdaBoost algorithm to improve the results when predicting. AdaBoost combines in a weighted way the output of other learning algorithms, with the idea of getting a more robust classifier from several classifiers. In this case, multiple decision trees generated from the same training data have been used, but each with differences in their nodes. From all these predictions, the AdaBoost algorithm calculates the most probable result. Figure 3 shows the confusion matrices of this algorithm.

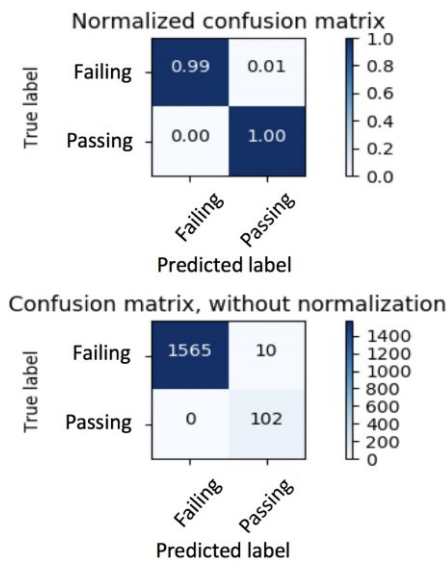


Fig. 3. Confusion Matrix for Decision Trees (entropy)

It can be seen that almost all the cases (passing and failing) are correctly classified. When calculating the accuracy and completeness of the failing class we obtain 1 and 0.9936 respectively. This means that it classifies all the passing cases well, while 0.64% failing cases are wrongly classified as passing ones. The accuracy and completeness of the passing class are 0.9107 and 1 respectively. These results are better than those obtained when using Naïve Bayes.

C. Support Machine Vectors (SVM)

Figure 4 shows the confusion matrix obtained when training and using the SVM algorithm with the available data. The first thing to emphasize is that this algorithm does not classify any pattern as passing. This may be because 94.0650% of the instances belong to the failing class. The values for the accuracy and completeness of the failing class are 0.9419 and 1 respectively. This means that it classifies correctly all the failing cases and considers all the passing cases as failures (it does not classify any case as passing). With this algorithm, the accuracy has not decreased too much, but this is because there is a high percentage of failing cases and it does classify all the cases as failing.

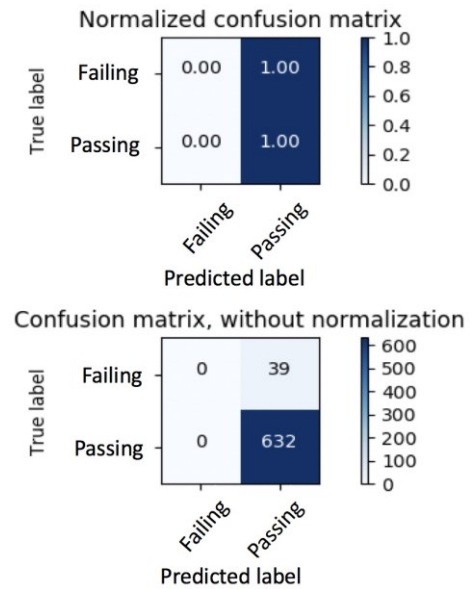


Fig. 4. Confusion matrix for SVM

D. Long Short-Term Memory

Recurrent neural networks (RNN) are the basis of a deep learning approach where the prediction not only bases on the training data, but also uses the output of the neural network itself in previous stages as input in new training phases. As can be seen in the left part of figure 5, the network A receives a data collection X with which generates a prediction h, and the network feeds back with its own intermediate results. When looking into the training loop (right part of the figure), it can be observed that this type of networks have a kind of memory where the previously obtained results are used for further training.

Among the recurrent neural network algorithms, we have chosen LSTM (Long Short Term Memory) for this research, since it has offered very good classification results in different contexts. This network is able to establish

dependencies between elements even when these elements are separated by several training phases (long-term dependencies).

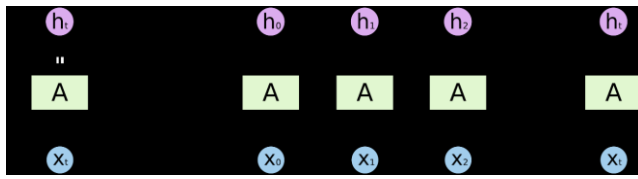


Fig. 5. Training loop of the Recurrent Neural Network

Figure 6 shows the confusion matrix for LSTM. As can be seen, most failing cases are classified correctly, but a few of them are wrongly classified. In the same way, the majority of the passing cases are well classified, but some of them are not. The accuracy and completeness of the failing class are 0.9949 and 0.9984 respectively. This algorithm does not provide better results than those obtained previously by traditional algorithms such as, for example, decision trees. Next section describes the comparison of the results obtained.

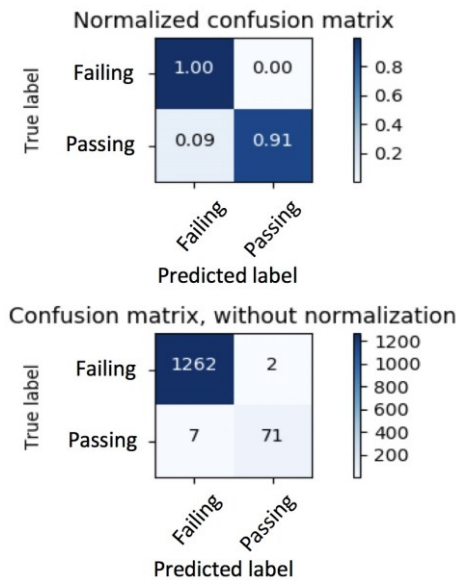


Fig. 6. Confusion matrix for LSTM

E. Comparison of results

Table III shows the success rate of some learning algorithms when training them with all the data available for the MOOC. Table IV shows the success rate of these algorithms when generating weekly models, specifically for the fifth week.

TABLE III. COMPARISON OF HIT RATES (GLOBAL, ALL DATA)

Naive Bayes	Decision Tree Entropy	Decision Tree Gini	SVM	LSTM
97.47 %	99.40 %	99.64 %	93.59 %	94.41 %

TABLE IV. COMPARISON OF HIT RATES (WEEK 5)

Naive Bayes	Decision Tree Entropy	Decision Tree Gini	SVM	LSTM
96.88 %	99.17 %	98.93 %	93.31 %	94.37 %

As can be seen, Decision Trees is the most precise algorithm in this research, both in the overall prediction (with all the data available) and in weekly predictions (with models generated each week), followed by Naive Bayes. Long Short-Term Memory, which implements deep learning, does not provide a clear improvement in this case study. Deep learning algorithms produce good results when the number of variables to be handled is very large. It could be investigated whether it would be possible to extract or calculate more variables from the logs in order to experiment with a greater number of them to check if, in that case, this type of algorithms would lead to better results. Yet for the time being and with the data available, it can be concluded that LSTM does not improve the results obtained when using traditional machine learning algorithms.

Decision Trees have produced the best results among all the algorithms explored. In addition, they provide information about the reasons of its classifications and predictions. Therefore, they have been chosen to generate the weekly predictions described above. Prediction models based on decision trees are generated for every week in which the student is taking the course.

In the first few weeks, the most relevant attributes for the predictions were the number of exercises solved correctly, the number of times the videos are rewinded and the number of attempts to solve each exercise. As the weeks go by, the number of problems performed and the time spent watching the videos are added, both with less weight. In any case, as the time goes by, the attribute that gains more influence in the prediction is the number of exercises correctly solved.

V. CONCLUSIONS AND FUTURE WORK

In this work, we have explored the use of machine learning techniques, including deep learning, for the analysis of data from the interactions of students within the MOOC *La España del Quijote*, delivered by the Universidad Autonoma de Madrid through EdX, with the aim of predicting dropout or failure risk.

A first conclusion is that, even though the logs of EdX were not specifically designed for the analysis of student interactions, the attributes extracted from them are sufficient to make accurate predictions. In any case, after extracting these attributes, we have concluded that it would be very convenient to redesign the log structure to facilitate the extraction of relevant information for learning analytics.

In this specific research, the use of an advanced algorithm for deep learning (Long-Term Memory Short) did not produce better results than those obtained when using other traditional machine learning algorithms. If the opposite had happened, it would have been necessary to justify the use of deep learning in spite of the fact that it does not provide information about the reasons why the subjects are classified in one way or another.

Nevertheless, it would be possible to explore the possibility of extracting or calculating a greater number of attributes from the events stored in the logs, in order to check whether deep learning produces better results in this case, since it seems to perform better with large sets of variables.

On the other hand, the conclusions in this type of studies can closely relate to the type of MOOC considered. In the MOOC analysed in this research, watching videos seems to

be relevant and the predictors refer to this type of activity. In MOOCs with other types of activities, the predictor variables may be others, and the accuracy achieved by the corresponding predictive models could vary.

For this reason, it would be interesting to do this research with other MOOCs on the same platform and observe the results. In this way, it would be possible to analyse the effectiveness of the predictions based on weekly models in other courses and with other students, regardless the specific types of contents deployed. It would also be very interesting to analyse the impact of communicating predictions to students and teachers weekly on both the student retention and the results obtained. In fact, we are currently analysing the impact of both dropout risk prediction and intervention in the context of a fully online university.

ACKNOWLEDGEMENTS

This work has been partially funded by the regional project from Comunidad Autonoma de Madrid eMadrid-CM (S2013/ICE-2715). We want to thank Unidad de Tecnologías para la Educación from Universidad Autonoma de Madrid for the data of the MOOC *La España del Quijote*.

REFERENCIAS

- [1] C. J. Bonk, M. M. Lee, T. C. Reeves, y T. H. Reynolds. MOOCs and Open Education Around the World. Routledge, 2015.
- [2] R. S. Baker y P. S. Inventado. Educational Data Mining and Learning Analytics. Learning Analytics, Springer, New York, NY, 2014, pp. 61-75.
- [3] «INTEF - educaLAB». Disponible en: <http://educalab.es/intef>
- [4] Miriada X, Disponible en: <https://miriadax.net/home>.
- [5] Udacity - Free Online Classes & Nanodegrees. Disponible en: /.
- [6] Cursos online: aprende de todo y a tu propio ritmo | Udem. Disponible en: <https://www.udemy.com/>
- [7] Coursera | Online Courses From Top Universities. Join for Free. Disponible en: <https://es.coursera.org/>
- [8] edX.. Disponible en: <https://www.edx.org>
- [9] Y. LeCun, Y. Bengio, y G. Hinton. Deep learning. Nature, vol. 521, n.o 7553, 2015, pp. 436-444.
- [10] Deng, L., & Yu, D. Deep learning: methods and applications. Foundations and Trends in Signal Processing, 7(3-4), 2014, pp 197-387.
- [11] C. Lang, G. Siemens, A. Wise, and D. Gasevic. Handbook of Learning Analytics – First edition. : Society for Learning Analytics Research. 2017, pp 1-356
- [12] Ferguson, R., & Shum, S. B. Social learning analytics: five approaches. In Proceedings of the 2nd international conference on learning analytics and knowledge. ACM. 2012, pp. 23-33
- [13] Bannert, M., Reimann, P., & Sonnenberg, C. Process mining techniques for analysing patterns and strategies in students' self-regulated learning. Metacognition and learning, 9(2), 2014, pp 161-185.
- [14] Daems, O., Erkens, M., Malzahn, N., & Hoppe, H. U. Using content analysis and domain ontologies to check learners' understanding of science concepts. Journal of computers in education, 1(2-3), 2014, pp 113-131.
- [15] Kloos, C. D., Alario-Hoyos, C., Fernández-Panadero et al. eMadrid project: MOOCs and learning analytics. In Computers in Education (SIIE), 2016 International Symposium on. IEEE . 2016, pp. 1-5.
- [16] Srilekshmi, M., Sindhumol, S., Chatterjee, S., & Bijlani, K. Learning Analytics to Identify Students At-risk in MOOCs. In IEEE Eighth International Conference on Technology for Education. 2016, pp. 194-199.
- [17] Course Quijote501x edX. En: <https://courses.edx.org/courses/course-v1:UAMx+Quijote501x+3T2017/course/>.