Educational and Information Technologies (2019) 24:3591–3618
https://doi.org/10.1007/s10639-019-09934-y

A predictive approach based on efficient feature selection and learning algorithms' competition: Case of learners' dropout in MOOCs

Mourdi Youssef[1] · Sadgal Mohammed[1] · El Kabtane Hamada[1] · Berrada Fathi Wafaa[1]

Received: 29 March 2019 / Accepted: 16 May 2019 / Published online: 26 June 2019

Abstract

MOOCs are becoming more and more involved in the pedagogical experimentation of universities whose infrastructure does not respond to the growing mass of learners. These universities aim to complete their initial training with distance learning courses. Unfortunately, the efforts made to succeed in this pedagogical model are facing a dropout rate of enrolled learners reaching 90% in some cases. This makes the coaching, the group formation of learners, and the instructor/learner interaction challenging. It is within this context that this research aims to propose a predictive model allowing to classify the MOOCs learners into three classes: the learners at risk of dropping out, those who are likely to fail and those who are on the road to success. An automatic determination of relevant attributes for analysis, classification, interpretation and prediction from MOOC learners data, will allow instructors to streamline interventions for each class. To meet this purpose, we present an approach based on feature selection methods and ensemble machine learning algorithms. The proposed model was tested on a dataset of over 5,500 learners in two Stanford University MOOCs courses. In order to attest its performance (98.6%), a comparison was carried out based on several performance measures.

Keywords Dropout · Distance education · Feature selection · Algorithms competition · Educational datamining · MOOC

1 Introduction

The emergence of technological innovation has given distance learning a new breath in order to respond more and more to the diverse and growing needs of teachers

✉ Mourdi Youssef
mourdiyoussef@gmail.com

Extended author information available on the last page of the article.

and learners. The ultimate goal is to ensure a better learning experience. This development gave birth, in 2008, to a new model of distance learning better known as MOOC.The Massive Open Online Courses are becoming more solicited by universities, not only to offer training to geographically distant learners, but also to complete face-to-face training. In addition, the MOOCs are trying to solve the problem of massive students that the infrastructure no longer supports. This generally applies to developing countries. MOOCs are also used by training companies and private trainers who offer free or paid certifications in a variety of domains via platforms such as Udemy, Cousera, Udacity and many others (Yuan and Powell 2013).

Nowadays, MOOCs offer a recourse for anyone seeking certification, or simply to improve their knowledge in a given field (Sanchez-Gordon and Luján-Mora 2016). They can be especially useful for people in professional activities and having concerns that make it delicate to find regular time to learn. This has led the MOOCs to attract a significant number of enrollees, who sadly give up remarkably as the course progresses. According to Liyanagunawardena et al. (2014), only 10% of registered learners are able to finish classes. For example, a software engineering course offered by the University of MIT and Berkeley received 50,000 subscriptions but just 7% were able to pass the MOOC (Yuan and Powell 2013). Another feedback is from the UK Open University Open Learning Design Studio (Cross 2013), in which the authors studied a course of 2420 enrolled learners and noticed that only 50% of learners consulted at least one course page during the first week of the course. They also noted that no more than 30 students were active learners, and only 22 completed the course, 50% of whom were able to reach the course goal. Onah and Sinclair (2014) cited the experience of Duke University which launched a MOOC of Bioelectricity, a course that received 12175 entries. Despite this huge number of registrations, only 7,761 learners (representing 64% of the total enrolled learners) followed at least one video, 26% answered a quiz and only 2.6% completed the course.

As a result, the dropout rate experienced by MOOCs does not only affect learners, but also means that resources deployed by instructors and course managers are wasted. Thus, this problem can influence the quality of the collective pedagogical activities, such as the projects, that the learners must ensure in teams and thus produce a direct impact of demotivation for the other group members. The impact of this is not only limited to this level;the dropout problem also makes the task of coaching learners a real challenge for MOOC facilitators, which is strongly linked to the number of learners. Determining precisely which learners may leave, succeed or fail will effectively direct all efforts; and thereby streamline the interventions made for each type of learners. In other words, a classification of learners into three distinct classes (class of learners passing the course, class of learners failing and those leaving the MOOC) is a necessity. In addition, the MOOC platforms generate a great deal of data. The analysis of this data can reveal relevant indicators about students' dropout, success or failure, and consequently prediction and classification barometers. This has opened several avenues of research but, according to the literature review, the majority of this research only aims at finding a way to predict learners who are at risk of dropping the MOOC.

MOOC platforms currently store all learners' data, namely connection data, personal information, learner performance data related to a course, and even navigation and interaction data with the proposed teaching resources. Given this large and varied mass of data, and in order to ensure a classification of the MOOCs learners, the choice of the features that model them remains a very critical step that can directly influence the classification quality and accuracy.

Following all these motivations, we present, in this paper, a predictive model based on ensemble learning algorithms in order to classify learners into three classes. This classification has been made by referring to the traces, interactions, performance and personal information of the learners gathered from the MOOCs platforms. On the one hand, we extract the maximum of attributes modeling a learner with experts help. Subsequently, many method of reducing and selecting the most important features was adopted to improve the predictive performance of our model. On the other hand, and in order to validate the proposed model, it was tested on two real datasets of two MOOCs from Stanford University hosting more than 5,500 registered learners. Finally, a comparison was carried out between the results which was obtained by the proposed model and those of the literature review.

The main contributions of this work can be listed as follows:

- Providing an approach to automate the determination of relevant attributes for analysis, classification, interpretation and prediction from the massive data collected about MOOC learners.
- Proposing a predictive model based on ensemble and automatic feature selection methods that ensure the classification of learners.
- Evaluate the importance of feature selection and its impact on the predictive performance of machine learning algorithms.
- Evaluate the performance of the proposed model with those proposed in the literature review.

Our contribution will allow MOOCs' instructors and moderators to have a more refined visibility on the different categories of MOOC learners, by ensuring a weekly prediction. Thus, they can make rational and personalized interventions according to the learners' class.

This paper is divided into three main parts. In the first part, we present a literature review of the different approaches that have been proposed in the same purpose, as well as the categories of features widely used in previous works. The second part illustrates the methodology and materials adopted during this research. The third and last part presents the results obtained and a discussion.

2 Related works

2.1 Dropout prediction in MOOC

According to the literature review, several research projects have been launched with the objective of providing predictive approaches by adopting machine learning techniques. In this section, we present a set of very recent works that we consider interesting.

Vitiello et al. (2018) have focused on MOOCs that are relaunched to predict learners at risk of abandoning the MOOC. In this research, the authors analyze a Curtin University course offered twice successively. They tried to extract the estimated important features to predict learners who may not complete the MOOC. After that, the authors proposed a prediction model based on enhanced decision trees that was tested on the second launch of the same MOOC. By adopting this approach, the authors have achieved an accuracy of 80%.

In the same context, Qiu et al. (2018b) proposed a framework to predict learners at risk of not completing the MOOC. This framework is based on feature selection methods and logistic regression. The idea is to look for the most important features for training and testing the predictive model in order to improve its performance. Trying to answer the same problem, using the data of two different offers of the same MOOC, and based on the social and behavioral characteristics, Gitinabard et al. (2018) combined between decision trees and machine learning techniques for feature selection and prediction. In this research, the prediction also concerned learners likely to obtain certification. The implementation in this research was done on two courses run on the Coursera platform by Columbia University. The authors managed to propose a model that reaches an average accuracy of 93.3%.

Tang et al. (2018) had a different view of the problem, they considered the dropout phenomenon as a time series problem, and therefore proposed a predictive model based on a recurrent neural network (RNN) with short-term memory cells . The validation of this proposal was made on courses of the XuetangX platform. This model reaches 88.1% in terms of AUC.

Qi et al. (2018) have used a supervised classification method for analyzing online learning behavior that evolves over time. The objective was to predict if the learner will be online in the following weeks as well as his final grade. The process has been validated on five courses of the ’icourse163’ platform. The results of the test showed that the mean of the dropout prediction and the final score were 86.3% and 74.8% respectively.

Qiu et al. (2018a) proposed a prediction model based on convolutional neural networks which integrates, in the same framework, both the extraction and the selection of features as well as the classification of learners. After experimenting on several MOOCs of the XuetangX platform, the framework ensures an accuracy of 86.75%. To summarize, the research inspired by these models has highlighted the complexity of the dropout phenomenon in MOOCs and developed prediction approaches based on recent techniques such as data mining methods. However, these methods have focused on a limited number of prediction features that remain relatively similar.

In the following section, we calibrate all of these predictive variables widely used in the literature.

2.2 Predictive features

In order to develop powerful and accurate prediction models, the researchers relied on several features extracted from the studied datasets. These attributes have taken several natures that differ from one study to another. The table (Table 1) presents a

Table 1 Research work and prediction features used

Research	Features category
Fei and Yeung (2018), Al-Shabandar et al. (2017), Xing et al. (2016), Kloft et al. (2014)	Clickstream
Sinha et al. (2014)	Video Clickstream
Crossley et al. (2016)	Clickstream and NLP tools
Yang et al. (2016)	Student behavior in the discussion forum and Social Network Behavior
Chaplot et al. (2015)	Sentiment Analysis

non-exhaustive list of features generally used in the prediction of students at risk of dropping MOOC courses.

According to the Table 1 and some research (Prieto et al. 2017), we note that researchers generally refer to the flow of data generated by learners' navigation clicks and their interactions with the course videos on the platform. In addition, it must be emphasized that the dropout problem requires more predictors and a wide range of concrete variables for more reliable results. Individual, organizational, environmental or attitudinal, these variables, once grouped, can give a rather global theory explaining each aspect of the dropout phenomenon. Yet, the large literature on learner retention in MOOCs indicates that a significant proportion of these features are missing from researchers.

At this stage, the problem that arises initially is mainly related to the features engineering and selection, that is to say, to find good predictors (information-bearing features) to form very strong classifiers, something that many researchers have neglected. Features engineering is, thus, an essential part in the construction of any prediction model. Variables used in data processing directly influence the predictive models used and the results obtained.

In conclusion, dropout does not only involve interactions with the platform or its content but several attributes that influence each other. Predicting learners at risk of quitting MOOCs is therefore anchored to a set of diverse features that we try to present and study in the coming parts.

3 Methodology and materials

3.1 Research process overview

The research presented in this paper is divided into five major phases as shown in Fig. 1. Focusing primarily on feature engineering, the first phase is essential in any classification problem using machine learning algorithms; since it consists in the extraction and selection of relevant, significant and information-rich features from the initial data, this phase required the intervention of the experts namely pedagogues.

The second phase of this research is the extraction of data according to the features selected in phase 1. Following this, these recovered data undergo a cleaning and a

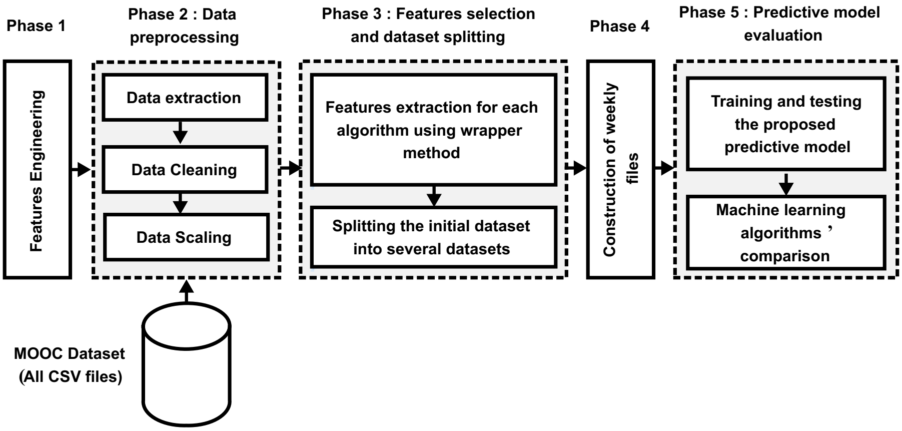

Phase 1
Features Engineering
Phase 2 : Data preprocessing
Data extraction
Data Cleaning
Data Scaling
Phase 3 : Features selection and dataset splitting
Features extraction for each algorithm using wrapper method
Splitting the initial dataset into several datasets
Phase 4
Construction of weekly files
Phase 5 : Predictive model evaluation
Training and testing the proposed predictive model
Machine learning algorithms ' comparison
MOOC Dataset (All CSV files)

Fig. 1 The research process phases

scaling stages to have quality data that is standardized and ready for the training phase.

Subsequently, the third phase is the generation of weekly data for each learner to ensure a prediction for each week of the course. Based on the “wrapping” methods, the fourth phase aims to determine the most predictive features. To do this, we have proceeded, for each machine learning algorithm, to cut the initial dataset into several subsets according to the features that are most relevant to it. These subsets, deduced for each algorithm, will constitute the input of the proposed prediction module.

Finally, and following the previous steps, the last phase is the training of the proposed model and other machine learning algorithms to compare the performance of each one. This comparison will place the predictive model presented in this research work against standard prediction models. In what follows, we focus on each phase in a detailed way by presenting the dataset adopted as well as the different tools used.

3.2 Dataset and feature engineering

3.2.1 Stanford courses dataset

The course studied in this research is a course of ”quantum mechanics” for scientists and engineers provided by Stanford University. It has been divided into two parts: QMSE01 and QMSE02. The first session was proposed in 2016 and lasted 9 weeks, while the second session, which represents the continuation of the first session, was held in 2017 and lasted 10 weeks. In terms of enrollment rates, the first session registered 3585 learners, unlike the second session, which received 1742 learners.

Regarding the educational content, the first session of the course contained 26 quizzes of which 5 are optional and 9 end-of-week exams, of which only one is optional. The MOOC also offers 83 learning objects as videos. In return, the second session of the course offered 94 videos, 83 quizzes of which 3 are optional and 9 end-of-week exams, one is optional. With all these features and number of important videos, we are convinced that this dataset is a good study sample.

Table 2 Dataset files and content

CSV File	Content
Demographics	Contains learner demographic information such as gender, year of birth and academic level. This information may be empty or null.
EventExtract	Contains all the information concerning the users' navigation on the platform. This data includes interaction with videos, transcripts, forum discussions and quizzes sent.
ActivityGrade	Includes learner score data in quizzes, including good and bad answers, answers validated by each learner, date of first submission and date and time of last submission.
Forum	Contains threads of the learner in the forum.
allData	Includes characteristics representing commitment as the number of connections, and the number of events in each session.
weeklyEffort	Contains the effort provided by each learner in a week (in seconds).

The dataset is anonymous and divided into several Comma-Separated Values (CSV) files extracted from the OpenEdx platform. The Table 2 exposes each file used in this search with a brief description of its contents. For more information on these files, visit CAROL Stanford's website[1].

3.2.2 Feature engineering

The raw data of the datasets remain, in the majority of the cases, unusable for the lack of data sometimes, the types of fields and heterogeneous data types. Performing feature engineering modeling a phenomenon is therefore an indispensable step since it directly influences the performance of any machine learning algorithm and therefore the final decision-making. This extraction of features often makes use, initially, of the common sense and the expertise of the people experienced in the studied field.

In the same direction, six pedagogues, having previously conducted several MOOC courses, collaborated in an initial phase. Their intervention allowed the extraction of a maximum of features proving to be interesting and useful for the modelisation. In addition, the features retained are uncorrelated; thus guaranteeing a good functioning of the future predictive model.

Unlike the work done and which is generally limited to the interaction of the learners with the videos or just the traces of their navigations on the platform, the features retained in this present paper are 61, and are grouped under 11 categories. Table 3 details all these features.

[1] https://datastage.stanford.edu/

Table 3 The retained features and their categories

Category	Features
Video Interaction	Number of completed videos in chapter 1, 2, 3, 4, 5, 6, 7, 8 and 9
	Number of times the learner tries to go back in the videos
	Number of times the learner tries to move forward in the videos
	Number of times the learner tries to speed up the video
	Number of times the learner tries to speed down the video
	Number of times the learner tries to play videos
	Number of times the learner tries to pause videos
	Number of times the learner tries to stop videos
Transcript Interaction	Number of the transcript's downloads
	Number of times the learner interact with the transcripts
Quiz Interaction	Number of sent quizzes
	Number of quizzes whose score is greater than 50% of the score defined by the instructor
	Number of quizzes whose score is less than 50% of the score defined by the instructor
	Number of attempts to send quizzes
	Number of 100% correct quizzes
	Average time between two quizzes sent (in minutes)
Effort	Time spent on the platform
	Number of connections
	Average number of days between two connections
	Number of active days of which the learner was logged on the platform
Personnal Information	Academic Level
	Age
	Gender
Performance	Weekly Final Grade
Prerequisites	Number of completed prerequisites videos
Forum	Number of learner's thread response
	Number of created threads
	Number of up votes
	Number of down votes
Navigation	Number of views of course information
	Number of forum access
	Number of visits to the progress page
	Number of accessed chapters
	Number of visited sequential
	Number of reference access
Weekly Final Test	Number of answer in the weekly final Test in week 1, 2, 3, 4, 5, 6, 7, 8 and 9
Supplementary Resources	Number of week 1, 2, 3, 4, 5, 6, 7, 8 and 9 Supplementary resources access

3.3 Data preprocessing and normalisation

It is a module that we developed using the Python programming language and using the Apache Spark tool for processing Big Data. This choice was made by referring to the nature of our dataset which is distributed over several CSV files and also on the number of observations in each file, which make the search and the access to the information a very expensive task in terms of processor and RAM.

Spark offers a SparkSQL module which is quite complete and offers a package of features to launch SQL queries and ensures joins between separate files (Meng et al. 2016; Armbrust et al. 2015). Therefore, SparkSql has been an added value for data extraction and construction.

After the extraction phase, the generated files have undergone two necessary operations. First, the data cleansing process of detecting and eliminating incomplete observations. The second operation was the standardization of these data. This phase is very important since in the dataset generated, there is information with different scales (example: age in years, time spent on the platform in seconds, time between two connections in days, etc.). The standardization of the data made it possible to adjust these values to make them comparable. For this, we used the MinMax method. In MinMax, the values of the entities are scaled at the interval [0, 1] as (1):

$$x_i{}^{\text{new}} = \frac{x_i - min(X)}{max(X) - min(X)}. \tag{1}$$

where X is a relevant feature, x_i is a possible value of X in the dataset and $x_i{}^{\text{new}}$ is the normalized value.

The dataset also contains different attributes: quantitative features that cause no problem and qualitative data (nominal with more than 2 modalities) that must be transformed into numerical data so that they can be used during the training phase of the algorithms. For this, we transformed all nominal attributes into dummy attributes.

3.4 Features selection and dataset partitioning

The use of machine learning algorithms with datasets with a very high number of attributes generally gives rise to several problems that significantly affect the performance of these algorithms, cause over-fiting, increase the requirements in terms of calculation and learning time; in addition to the deterioration of the model in the presence of noisy data (Talavera 2005; Salcedo-Sanz et al. 2018).

In order to overcome the problems mentioned above, reducing the dimensionality of datasets is one of the most powerful tools. This power lies in the selection of a subset of features containing the richest information (Alonso-betanzos 2007). Having a dataset with significant features allows to Li et al. (2018):

- Remarkably improve the predictive performance of a machine learning model.
- Decrease the complexity of the model.
- Earn in terms of calculation cost and resources.
- Avoid the algorithms over-fitting.

Although experts in the field can eliminate few irrelevant attributes, selecting the best subset of features usually requires a systematic approach. Currently, there are three families of automatic features selection methods namely:

- The filters: are generally used as a pre-processing step. Feature selection is independent of any machine learning algorithm. Nevertheless, these features are selected on the basis of their scores obtained from various statistical tests (Talavera 2005; Alonso-betanzos 2007).
- The wrappers: The principle is to use a subset of features and to form a model using them. Based on the performance from the previous model, the decision is made to add or remove features from this subset (Karegowda et al. 2010; Jović et al. 2015).
- The embedded: these methods perform the selection of features during the execution of the learning algorithm. These methods are therefore integrated into the learning algorithm in the form of normal or extended functionality (Jović et al. 2015).

Each of these methods has a different and particular selection principle. The filters and despite their speed, remain very limited for several reasons. These methods are independent of the learning model used and therefore the features proposed by these methods may not guarantee interesting performances for machine learning algorithms. Thus, the filters try to measure the relevance of the features by their dependence on a target variable and assign to each feature a calculated information gain, which makes the choice of the number of features to be retained during training, a difficult task.

Unlike filters, wrappers, and considering that they take into account the target machine learning algorithm and its biases, they give a set of features that guarantees the best predictive performance. In other words, they ensure the search for a sub-set with the minimum of attributes guaranteeing the best performance in a systematic way (stop rule of the search).

Referring to the strong points that wrappers offer, we have developed a module presented in Fig. 2. Its objective is to reduce the dimensionality of our dataset and to increase the predictive precision to the best.

This module is divided into two parts: the first part tries, from the initial data, to extract for each chosen algorithm, the list of the most relevant features. In other

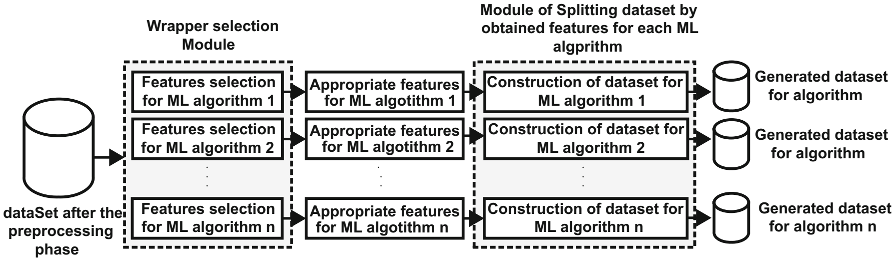

Wrapper selection Module
Module of Splitting dataset by obtained features for each ML algprithm
Features selection for ML algorithm 1
Features selection for ML algorithm 2
Features selection for ML algorithm n
Appropriate features for ML algotithm 1
Appropriate features for ML algotithm 2
Appropriate features for ML algotithm n
Construction of dataset for ML algorithm 1
Construction of dataset for ML algorithm 2
Construction of dataset for ML algorithm n
Generated dataset for algorithm
Generated dataset for algorithm
Generated dataset for algorithm n
dataSet after the preprocessing phase

Fig. 2 Wrapper module

words, those with which the algorithm gives the best precision based on the wrapper method. Once the list of features is retrieved, it is moved to the second part of this module which, in turn, builds on initial data and the list of received features, a new dataset. The latter is used later to train and test the algorithm in concern. In other words, this module partitions the initial dataset by retaining only the columns (features) that provide the best performance for a given algorithm and builds its own dataset.

3.5 Predictive model and evaluation metrics

3.5.1 Ensemble methods

Ensemble methods are techniques that combine several individual machine learning algorithms to improve the performance of predictions generated by singular learning models (Nagi and Bhattacharyya 2013; Sikora and Al-Laymoun 2014). Ensemble methods are known to be strong classes producing more accurate results (bias and variance reduction) than those typically produced when using a separate algorithm (Zitlau et al. 2016).

Several families of ensemble methods exist : Boosting (Sikora and Al-Laymoun 2014; Zhu et al. 2017), Bagging (Choudhury and Bhowal 2015; Kabir et al. 2014) and combining methods. The first two classes of methods work with a single ”weak” algorithm to generate a stronger model. While combining methods combine several algorithms at the same time in order to have a fairly powerful predictive model. These are grouped under several categories: voting, averaging and stacked generalization also known as stacking (Talavera 2005; Zitlau et al. 2016; Alves 2017).

According to the prediction problem to be solved (regression or classification), the principle of combining methods based on the vote or the average consists in training, by means of the same dataset, a set of machine learning algorithms before make predictions (Healey et al. 2018). In this case, each predictive model makes its decision independently of the others. These predictions form an input vector of a module which, based on the combining method adopted (the vote or the average) will make the decision and give the final prediction as shown in the Fig. 3a.

Regarding the combining method based on the Stacking model, the operating mode is different. This difference is localized in the way in which the final prediction is made. In stacking, two levels of prediction exist instead of one as is the

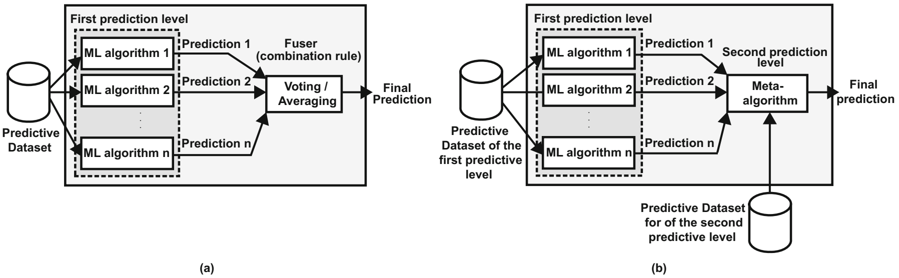


Fig. 3 Ensemble methods based on **a** vote/average and **b** meta-algorithm

case in voting or average methods. The second level is a machine learning algorithm called meta-algorithm having its own dataset. The principle in stacking is to train the first level algorithms with the same dataset and to make predictions that will form the meta-algorithm learning and test dataset (Nagi and Bhattacharyya 2013; Dinakar et al. 2014; Ren et al. 2016). The Fig. 3b illustrates these techniques.

3.5.2 Proposed predictive module

Certainly, the ensemble methods provide more accurate predictions, but this remains dependent on the chosen algorithms as well as the extracted features. In ensemble methods based on stacking, the performance of the set depends strongly on the performance of each of the algorithms used (Nagi and Bhattacharyya 2013). Thus, the selection of attributes makes it possible to remarkably improve the performance of an algorithm.

All of these motivations prompted us to look for ways to improve the performance of stacking models by combining them with wrappers feature selection methods. The idea is to search for each machine learning algorithm the most appropriate characteristics from an initial dataset in order to increase its performance. This search for features is provided by the module presented in Section 3.4. Figure 4 illustrates the new proposed stacking model.

After the data extraction phase, a dataset is constructed containing the observations about the MOOC learners represented by the initial characteristics adopted by the experts. This dataset will be subjected to dimensionality reduction processing in order to approve the performance of the machine learning algorithms used. For this, the initial dataset is passed to the feature selection module. The latter makes it possible to search, for each algorithm used in the stacking module, for the most relevant features and builds on a new dataset (partitioning of the initial dataset). At the end of this phase, we retain "n" datasets, with "n" the number of the algorithms in the stacking.

The second step involves training each stacking algorithm with the appropriate dataset as follows:

1. Divide the dataset into two parts, one for the training phase and one for the test phase.
2. Train each algorithm of the first level.

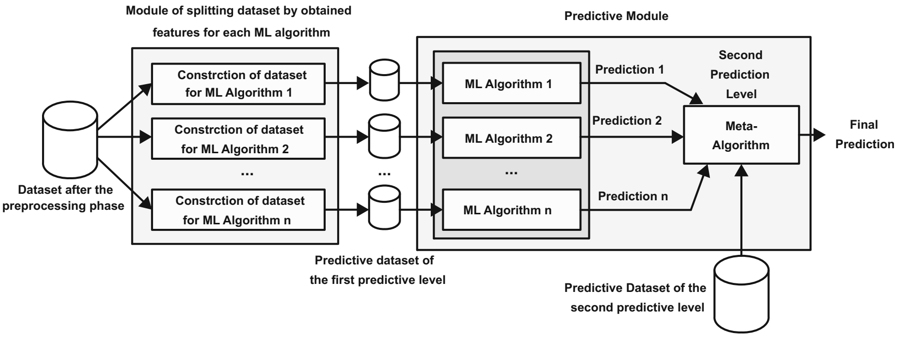

Module of splitting dataset by obtained features for each ML algorithm
Predictive Module
Constrction of dataset for ML Algorithm 1
Constrction of dataset for ML Algorithm 2
...
Constrction of dataset for ML Algorithm n
Dataset after the preprocessing phase
...
ML Algorithm 1
ML Algorithm 2
...
ML Algorithm n
Prediction 1
Prediction 2
Prediction n
Second Prediction Level
Meta-Algorithm
Final Prediction
Predictive dataset of the first predictive level
Predictive Dataset of the second predictive level

Fig. 4 The proposed predictive module

3. Make predictions on the first level algorithms (test).
4. Use the predictions of (3) to create the dataset to drive the meta-algorithm.

The last step was the test of the predictive model as a whole, for which we used the dataset of the second course.

The data generated by MOOC platforms is very large, which negatively influences the learning time of any predictive model based on machine learning. Consequently, the proposed approach may require a high computational power when it comes to classify course learners, and even higher power when analysing the whole platform that contains a multitude of courses with huge number of registred students. Thus, the parallel computing programming model MapReduce was used. It has been chosen due to its capacity to monitor tasks and handle failures in addition to taking into consideration the intra-cluster communication. Many frameworks were created to implement the MapReduce. The best known is Hadoop which was developed by Apache Software Foundation (White 2012). In our case, the MapReduce was implemented with Hadoop and used both to train and test the proposed predictive model and therefore the classification of MOOC learners. This significantly minimized the response time of the proposed model.

4 Results and discussion

4.1 Used ML algorithms, wrapper methods, evaluation metrics and experimentation approach

In order to validate the proposed predictive model, several learning algorithms were used. These are the most adopted in the works frequented in the literature: Decision Trees (Witten 2016), Support Vector Machine (SVM) (Naghibi et al. 2017), Naive Bayes (NB) (Witten 2016), K Nearest Neighbor (KNN) (Martínez-España et al. 2018), Random Forests (RF) (Witten 2016; Naghibi et al. 2017), and Logistic Regression (LR) (Burgos et al. 2018).

The idea is to ensure a comparison between the various indicators of the prediction performance of each model. Thus, and since we propose a model based on the stacking method, this section presents a comparison between the proposed model and the voting, bagging and boosting methods. In this step, we used the Scikit-Learn library developed with the Python programming language. This library offers a very important set of supervised and unsupervised machine learning algorithms, using a coherent task-oriented interface which facilitates the comparison of the methods of a given application (Pedregosa et al. 2012). The fact that Scikit-Learn is based on Python, makes the proposed model easy to integrate into MOOC platforms.

In the experimental phase, the machine learning algorithms were evaluated on the set of characteristics (61 characteristics) at first, then three wrapping methods namely Sequential Forward Selection (SFS) (Jindal and Kumar 2019), Sequential Backward Selection (SBS) (Panthong and Srivihok 2015) and Recursive Feature Elimination (RFE) (Xu et al. 2018) were adopted in order to compare the precision provided by the algorithms on the reduced dataset and the initial one. In the second step, the stacking and models were evaluated on the complete dataset. In the last step, we tested the proposed model on the same dataset.

With respect to the proposed model, the SVM, DT, NB algorithms were used to form the first prediction level and the Logistic Regression (LR) was used as the meta-algorithm. Performance measures and accuracy were based on:

$$Accuracy = \frac{TruePositives + FalseNegatives}{TotalNumberofSample}. \tag{2}$$

$$Precision = \frac{TruePositives}{TruePositives + FalsePositives}. \tag{3}$$

$$Recall = \frac{TruePositives}{TruePositives + FalseNegatives}. \tag{4}$$

The ROC curve represents the rate of true positives (TPR) as a function of the false positive rate (FPR), with:

$$TPR = \frac{TruePositives}{TruePositives + FalseNegatives}. \tag{5}$$

$$FPR = \frac{FalsePositives}{FalsePositives + TrueNegatives}. \tag{6}$$

AUC is the area below the ROC curve and it is calculated as follows:

$$AUC = \int_0^1 ROC(t).dt\,. \tag{7}$$

$$F1Score = \frac{2 \times (Precision \times Recall)}{Precision + Recall}. \tag{8}$$

With:

- TruePositives: YES was predicted and the actual output was also YES.
- TrueNegatives: NO was predicted and the actual output was YES.
- FalsePositives: YES was predicted and the actual output was NO.
- FalseNegatives: NO was predicted and the actual output was also NO.

4.2 Results

This section presents the results obtained for each adopted machine learning algorithm. These results represent the average predictive performance of the three classes of learners on the test set. It is reported that the performance tests were conducted on the nine weeks of the course, but in this section, and for each part, only the four-week results are presented to avoid paper clutter.

4.2.1 Without use of any wrapper method

In this first part, we present the results reflecting the performances of SVM, KNN, DT, NB, LR and RF models. These were tested on the complete dataset (61 attributes)

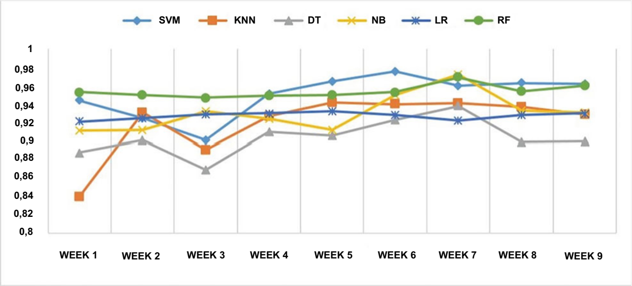


Fig. 5 Accuracy of ML algorithms without use of any wrapper method on the 9 weeks of the course

without resorting to a feature selection by the wrapping methods. Figure 5 shows the accuracy of the models over the 9 weeks of the course.

From Fig. 5, it is clear that the RF model provides interesting average predictions over the other models through most weeks; while the DT model remains the weakest. For the KNN, the performances remain close to the average. Compared to the rest of the models, they are very close together.

Table 4 Learner algorithms performance measures over 4 weeks without features selection methods

Week number	ML algorithm	Accuracy	AUC	Precision	F1 score	Recall
1	SVM	0.944	0.931	0.911	0.944	0.980
	KNN	0.84	0.808	0.813	0.757	0.708
	DT	0.888	0.893	0.875	0.928	0.988
	NB	0.911	0.922	0.901	0.935	0.971
	LR	0.921	0.913	0.911	0.921	0.932
	RF	0.953	0.933	0.937	0.958	0.981
3	SVM	0.901	0.972	0.988	0.984	0.980
	KNN	0.891	0.920	0.969	0.940	0.912
	DT	0.869	0.900	0.981	0.978	0.976
	NB	0.932	0.955	0.922	0.917	0.912
	LR	0.929	0.921	0.919	0.928	0.937
	RF	0.947	0.973	0.931	0.954	0.979
5	SVM	0.965	0.949	0.961	0.970	0.980
	KNN	0.942	0.912	0.957	0.957	0.958
	DT	0.906	0.909	0.980	0.980	0.980
	NB	0.912	0.940	0.951	0.918	0.888
	LR	0.932	0.944	0.915	0.913	0.911
	RF	0.950	0.969	0.930	0.956	0.983
7	SVM	0.96	0.976	0.977	0.976	0.976
	KNN	0.941	0.962	0.961	0.961	0.962
	DT	0.938	0.949	0.978	0.978	0.978
	NB	0.972	0.708	0.803	0.748	0.700
	LR	0.922	0.931	0.926	0.928	0.930
	RF	0.969	0.981	0.938	0.961	0.985

The Table 4 presents the Accuracy, AUC, Precision, F1 Score and RECALL values obtained on predictions made on weeks 1, 3, 5 and 7.

4.2.2 Using sequential forward selection method (SFS)

In this second part, we present the results reflecting the performances of SVM, KNN, DT, NB, LR and RF models. These were tested on the reduced dataset via the SFS feature selection method. The Table 5 presents the Accuracy, AUC, Precision, F1 Score and RECALL values obtained on predictions made on weeks 1, 3, 5 and 7.

Figure 6 shows the performance of models adopted in terms of accuracy during weekly predictions. In this case, the models are trained and tested on a dataset reduced by the SFS method.

The first remark that can be drawn is that the predictive performance of the KNN algorithm is significantly increased compared to the first test on the complete dataset. Basically, the accuracy of the algorithms are very close together. RF remains the best performer among all algorithms.

Table 5 Learner algorithms performance measures on 4 weeks with SFS methods

Week number	ML algorithm	Accuracy	AUC	Precision	F1 score	Recall
1	SVM	0.957	0.959	0.981	0.978	0.975
	KNN	0.933	0.940	0.975	0.978	0.981
	DT	0.902	0.931	0.962	0.960	0.959
	NB	0.926	0.968	0.981	0.982	0.984
	LR	0.919	0.953	0.941	0.942	0.944
	RF	0.954	0.962	0.989	0.976	0.964
3	SVM	0.921	0.964	0.974	0.974	0.974
	KNN	0.912	0.953	0.961	0.960	0.960
	DT	0.925	0.939	0.922	0.937	0.953
	NB	0.946	0.972	0.949	0.960	0.971
	LR	0.934	0.931	0.939	0.941	0.943
	RF	0.951	0.972	0.979	0.979	0.980
5	SVM	0.971	0.941	0.932	0.921	0.910
	KNN	0.952	0.960	0.969	0.975	0.981
	DT	0.931	0.975	0.949	0.956	0.963
	NB	0.929	0.981	0.959	0.969	0.980
	LR	0.94	0.952	0.95	0.952	0.955
	RF	0.96	0.981	0.986	0.983	0.981
7	SVM	0.974	0.976	0.941	0.940	0.940
	KNN	0.961	0.950	0.972	0.969	0.966
	DT	0.942	0.971	0.952	0.955	0.959
	NB	0.973	0.960	0.911	0.945	0.982
	LR	0.931	0.949	0.953	0.952	0.951
	RF	0.971	0.986	0.983	0.981	0.979

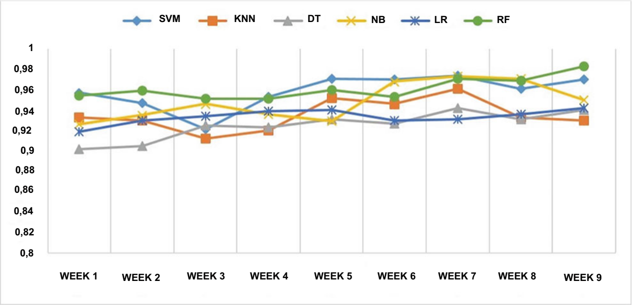


Fig. 6 Accuracy of ML algorithms with the use of SFS method on the 9 weeks of the course

4.2.3 Using sequential backward selection method (SBS)

Instead of the SFS method, this third part presents the performances of the same algorithms but by adopting the SBS feature selection method. The Table 6 presents

Table 6 Learner algorithms performance measures on 4 weeks with SBS methods

Week number	ML algorithm	Accuracy	AUC	Precision	F1 score	Recall
1	SVM	0.951	0.961	0.976	0.980	0.985
	KNN	0.946	0.972	0.961	0.967	0.973
	DT	0.922	0.944	0.938	0.944	0.951
	NB	0.949	0.962	0.970	0.973	0.977
	LR	0.93	0.951	0.963	0.966	0.970
	RF	0.954	0.953	0.962	0.974	0.987
3	SVM	0.942	0.948	0.940	0.945	0.950
	KNN	0.929	0.953	0.961	0.960	0.960
	DT	0.919	0.952	0.939	0.953	0.968
	NB	0.938	0.968	0.937	0.946	0.955
	LR	0.933	0.949	0.96	0.966	0.973
	RF	0.953	0.961	0.972	0.978	0.985
5	SVM	0.965	0.972	0.941	0.957	0.973
	KNN	0.942	0.971	0.964	0.959	0.955
	DT	0.929	0.949	0.936	0.953	0.971
	NB	0.964	0.970	0.961	0.969	0.977
	LR	0.945	0.94	0.944	0.953	0.962
	RF	0.967	0.961	0.97	0.974	0.979
7	SVM	0.978	0.980	0.945	0.951	0.958
	KNN	0.957	0.967	0.970	0.972	0.974
	DT	0.94	0.942	0.900	0.933	0.969
	NB	0.939	0.975	0.975	0.966	0.958
	LR	0.939	0.943	0.941	0.954	0.967
	RF	0.978	0.982	0.983	0.982	0.981

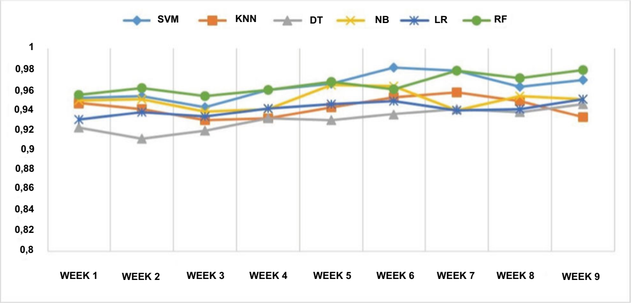


Fig. 7 Accuracy of ML algorithms with the use of SBS method on the 9 weeks of the course

Table 7 Learner algorithms performance measures on 4 weeks with RFE methods

Week Number	ML Algorithm	Accuracy	AUC	Precision	F1 Score	Recall
1	SVM	0.957	0.973	0.979	0.982	0.985
	KNN	0.949	0.942	0.951	0.962	0.973
	DT	0.93	0.939	0.939	0.939	0.939
	NB	0.953	0.964	0.972	0.973	0.974
	LR	0.952	0.970	0.971	0.975	0.979
	RF	0.964	0.977	0.982	0.984	0.986
3	SVM	0.953	0.968	0.961	0.969	0.978
	KNN	0.932	0.943	0.931	0.935	0.939
	DT	0.93	0.921	0.924	0.940	0.957
	NB	0.96	0.968	0.977	0.977	0.977
	LR	0.949	0.958	0.963	0.968	0.973
	RF	0.971	0.978	0.979	0.981	0.983
5	SVM	0.97	0.972	0.971	0.970	0.969
	KNN	0.943	0.951	0.953	0.955	0.957
	DT	0.926	0.937	0.933	0.934	0.935
	NB	0.969	0.976	0.970	0.972	0.974
	LR	0.952	0.966	0.967	0.968	0.970
	RF	0.969	0.983	0.98	0.980	0.980
7	SVM	0.98	0.981	0.976	0.978	0.981
	KNN	0.954	0.959	0.958	0.957	0.957
	DT	0.942	0.944	0.949	0.949	0.95
	NB	0.966	0.975	0.980	0.981	0.982
	LR	0.959	0.945	0.963	0.971	0.979
	RF	0.979	0.985	0.987	0.987	0.987

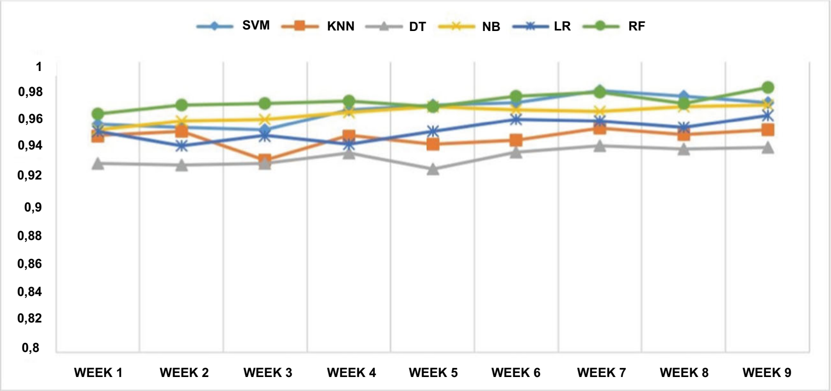


Fig. 8 Accuracy of ML algorithms with the use of RFE method on the 9 weeks of the course

the Accuracy, AUC, Precision, F1 Score and RECALL values obtained on predictions made on weeks 1, 3, 5 and 7.

After using the SBS method, we note that there is no performance degradation compared to the SFS method. Figure 7 shows the predictions of the models used on the 9 weeks of the course.

4.2.4 Using recursive feature elimination method (RFE)

This fourth part deals with the performance results obtained by adopting the selection method on the same algorithms. The Table 7 presents the Accuracy, AUC, Precision, F1 Score and RECALL values obtained on predictions made on weeks 1, 3, 5 and 7.

According to Fig. 8, we can conclude that the accuracy of the different models has been increased in comparison with the accuracies found with the other methods of selection of characteristics (SFS and SBS).

4.2.5 Comparison with other ensemble methods

The proposed model is based on the stacking method, hence the importance of evaluating its performance compared to other kinds of ensemble methods. For this, we first evaluated the classical VOTING and STACKING methods on the same (complete) dataset. In a second step, feature selection methods (SFS, SBS and RFE) were introduced. The returned results are displayed in the Table 8.

It is important to report that the voting adopted with the complete dataset was based on the classical architecture. On the other hand, when using the feature selection methods, the proposed architecture for stacking as the voting method was used and for each first-level prediction algorithm, the dataset generated by the Wrapper module was introduced.

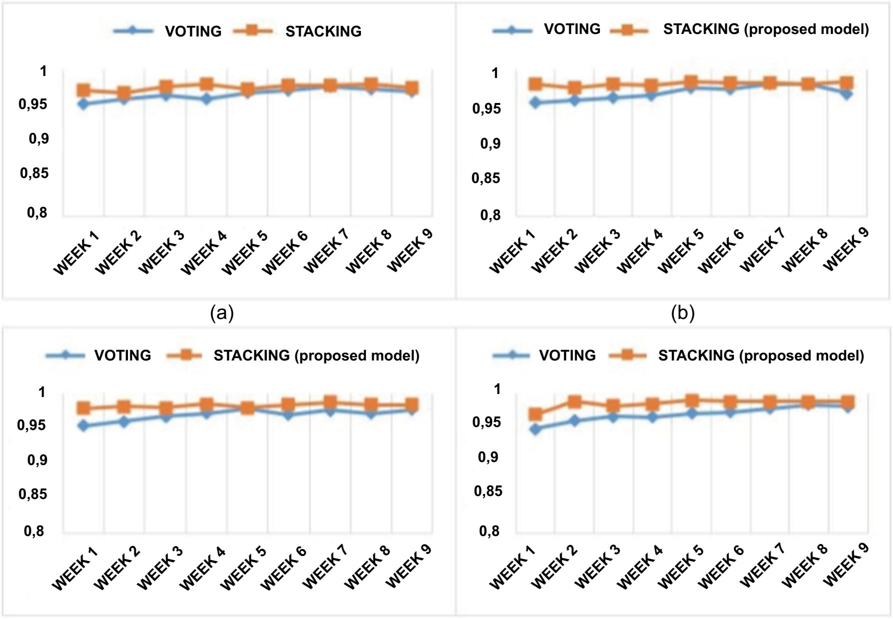
Referring to Fig. 9, we can notice that the stacking model provides more interesting predictions in all cases (with or without reduction). Thus, we can conclude that,

Table 8 Ensemble algorithms performance measures on 4 weeks without and with features selection methods

Features selection method	Week	Algorithm	ACCURACY	AUC	PRECISON	F1SCORE	RECALL
Without any feature selection method	1	Voting	0.953	0.955	0.959	0.958	0.958
		Stacking	0.971	0.97	0.973	0.971	0.969
	3	Voting	0.964	0.959	0.959	0.959	0.959
		Stacking	0.976	0.978	0.982	0.984	0.986
	5	Voting	0.968	0.971	0.972	0.971	0.970
		Stacking	0.973	0.98	0.984	0.982	0.981
	7	Voting	0.976	0.979	0.981	0.980	0.979
		Stacking	0.978	0.982	0.987	0.986	0.986
SFS	1	Voting	0.955	0.957	0.961	0.964	0.968
		Stacking (Proposed model)	0.979	0.975	0.979	0.980	0.981
	3	Voting	0.961	0.963	0.969	0.969	0.969
		Stacking (Proposed model)	0.98	0.982	0.986	0.987	0.988
	5	Voting	0.975	0.978	0.976	0.977	0.978
		Stacking (Proposed model)	0.983	0.989	0.990	0.989	0.989
	7	Voting	0.979	0.979	0.981	0.980	0.979
		Stacking (Proposed model)	0.981	0.992	0.980	0.983	0.987
SBS	1	Voting	0.954	0.950	0.959	0.964	0.970
		Stacking (Proposed model)	0.979	0.981	0.979	0.983	0.987
	3	Voting	0.967	0.965	0.971	0.972	0.973
		Stacking (Proposed model)	0.98	0.984	0.989	0.989	0.990
	5	Voting	0.979	0.980	0.983	0.982	0.981
		Stacking (Proposed model)	0.98	0.990	0.991	0.991	0.991
	7	Voting	0.976	0.978	0.983	0.981	0.980
		Stacking (Proposed model)	0.987	0.993	0.981	0.984	0.988

Table 8 (continued)

Features selection method	Week	Algorithm	ACCURACY	AUC	PRECISON	F1SCORE	RECALL
RFE	1	Voting	0.95	0.951	0.950	0.959	0.969
		Stacking (Proposed model)	0.97	0.98	0.983	0.986	0.989
	3	Voting	0.967	0.965	0.974	0.974	0.974
		Stacking (Proposed model)	0.982	0.986	0.99	0.990	0.990
	5	Voting	0.972	0.981	0.982	0.982	0.982
		Stacking (Proposed model)	0.99	0.990	0.991	0.991	0.991
	7	Voting	0.979	0.975	0.987	0.987	0.987
		Stacking (Proposed model)	0.988	0.994	0.986	0.987	0.989



(a)

(b)

(c)

(d)

Fig. 9 Accuracy of combining method: **a** without features selection and using **b** the SFS method, **c** the SBS method, **d** and the RFE method on the 9 weeks of the course

by using the RFE method, the proposed model ensures very close predictions to that provided without reduction but with more precision.

4.3 Discussion

4.3.1 The performmance analysis of the predictive model

In the first phase of the experiment, we evaluated the performances of the most used basic algorithms in the literature on all the attributes of the dataset (61 characteristics). The Fig. 10 shows that the RF model provides more accurate predictions than

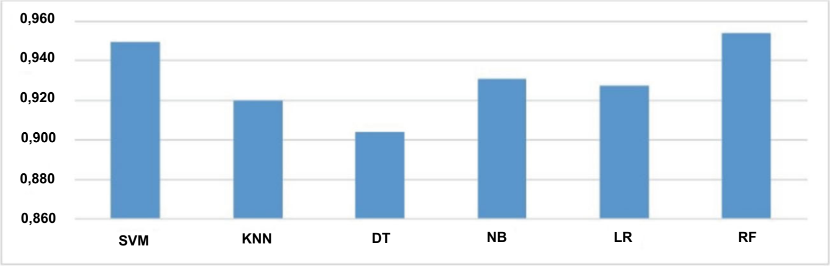


Fig. 10 ML algortihms accuracy average

Table 9 ML algortihms accuracy average comparison with the use of different feature selection methods

ML algorithm	Without wrapper	SFS	SBS	RFE
SVM	0.950	0.958	0.962	0.967
KNN	0.920	0.935	0.942	0.948
DT	0.904	0.925	0.930	0.935
NB	0.931	0.948	0.950	0.964
LR	0.927	0.933	0.940	0.953
RF	0.954	0.961	0.965	0.973

the rest of the models by proposing an average accuracy of 95.4%, followed by the SVM model, which guarantees an average accuracy of 95%. Just after, the NB with an accuracy of 93.1% followed by the LR model that ensures an accuracy of 92.7% to finally find, KNN and DT which do not exceed 92% and 90.4% respectively.

The second phase of the experiment was about testing these models by calling three different methods of feature selection (SBS, SFS and RFE methods). The performance of the models in this phase has increased significantly depending on the algorithm. Table 9 shows the differences between the performance of predictive models with and without feature selection methods.

Roughly, the use of feature selection methods positively influenced the performance of machine learning algorithms. For the SVM, note that the accuracy increased by 0.8%, 1.2% and 1.7% using respectively the SFS, the SBS and the RFE methods. Regarding the RF model, it is found that the accuracy was increased by 0.7% while adopting the SFS, 1.1% for SBS and 1.9% when using RFE. For the KNN model, its accuracy is improved by 1.5% using the FSF, 2.2% with the adoption of SBS and 2.8% with the RFE method. Compared to the DT algorithm which remains the algorithm with the weakest performance, it proposes more accurate predictions of 2.5% by combining it with the SFS method, 3% with SBS and 3.5% by using it with the RFE method. For the NB model, it also experienced an increase in performance using the feature selection methods, its accuracy is increased by 1.7% using the SFS method, 1.9% with SBS and 3.3% using the RFE method. Finally, the LR model performed 0.6% in SFS, 1.3% with the SBS method and 2.6% using the RFE method. On average and by adopting these dimensionality reduction methods, the accuracy of our model could be increased by 1.2% with a selection by SFS, by 1.7% by adopting the SBS method and by 2.5% using the RFE method.

Table 10 Ensemble algortihms accuracy average comparison with the use of different feature selection methods

ML algorithm	Without wrapper	SFS	SBS	RFE
Voting	0.966	0.968	0.970	0.970
Stacking	0.975	0.980	0.982	0.986

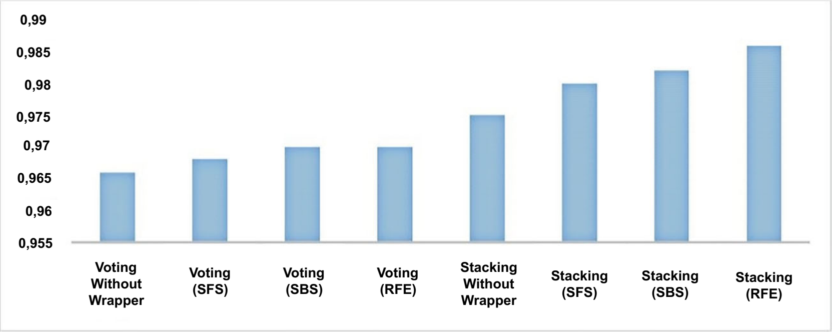


Fig. 11 Ensemble algortihms accuracy average comparison with the use of different feature selection methods

With regard to the proposed model, one can note from the Table 10 and the Fig. 11 that it guarantees the best predictions, independently of the adopted selection method. The results look promising as this model can achieve up to 98.6% accuracy by adopting the RFE method.

Thus, adopting attributes selection methods to perform the ensemble models is efficient, and especially with the stacking model. The performance has increased remarkably by 0.35%, 0.55%, 0.75% by integrating respectively SFS, SBS and RFE methods. The proposed model gives the best precision compared to other algorithms. This performance in terms of accuracy included not only the accuracy of the prediction in identifying students at risk of dropping out, but also its ability to generate more accurate individual dropout probabilities for personalization and prioritization of interventions for all three classes of learners (Fig. 12).

4.3.2 The benefits of the proposed predictive model for MOOCs instructors

The particularity of MOOCs in general remains the large mass of registered learners, which presents to the instructors several challenges related to the learners coaching,

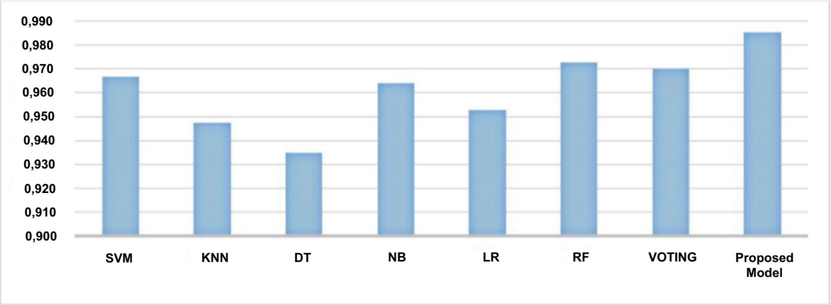


Fig. 12 Ensemble algortihms accuracy average comparison with the use of different feature selection methods

the instructor/learner communication, learners groups forming for group activities and proposing upgrades or prerequisites resources.

Predicting and identifying learners at risk of dropping out the courses is a necessity within the MOOC framework, allowing instructors to guide and streamline their interventions to understand the causes of course overload and to conduct interventions to retain as many learners as possible. Thus, the classification and the prediction of learners at risk of failing their courses will give the MOOC facilitators the opportunity to offer support resources which will improve the learner's self-confidence and increase the rate of instructor/learner communication. Furthermore, knowing the learners on the way to success will give the instructors the opportunity to expand their courses by offering other chapters for example. In other words, predicting the learners' class in MOOCs will allow instructors to customize their interventions according to each learner's profile.

In this research, the proposed model is a relatively better predictor compared to the other machine learning algorithms used. At this stage, this model can be easily integrated into the MOOC platforms that instructors can use as a dashboard to track their learners and intervene for each case.

The prediction in this research was done by adopting the attributes data of each learner per week. Therefore, our prediction model ensures a weekly classification and will allow MOOCs to have a classification of learners into 3 classes at the end of each week. This makes our approach more distinguished from those of the literature, not only in terms of predictions accuracy but also in terms of attributes initially chosen by experts (pedagogues). All axes that could be subject to the interaction of a learner in a MOOC, including its performance and commitment, were taken into consideration.

The proposed module will browse the learner database one by one and display, to the instructors, the learner with his class and the prediction score (the probability that the learner belongs to the predicted class). At this point, it is up to the instructors to make the appropriate decision and intervene in the current week or wait for the next week's predictions. In other words, a learner who has been classified as "at risk of dropping out" with a score of 80% requires an urgent intervention from the instructors; unlike another who was classified in the same class with a score of 50% or less. In this last case, instructors cannot make a decision and must wait for the next week's predictions.

5 Conclusion and future works

In this paper, an approach based on machine learning algorithms allowing not only a classification of the learners of a MOOC, but also the prediction of their dropout, failure or success has been proposed. The objective of using the artificial intelligence, especially machine learning, is to give the possibility to determine in advance the learners at risk of leaving the MOOC, those likely to fail and also the learners who have a good chance to succeed and obtain a certification. These predictions give the opportunity for MOOC instructors and trainers to ensure rational, targeted and effective interventions to ensure the smooth running of the MOOC. Knowing, as soon as possible, learners with a high probability of failure, will allow facilitators to offer support classes, offer help, offer additional resources, or other assistance specific to this class of learners.

To ensure predictions, we proposed a module based on features selection wrapping methods as well as the stacking ensemble method which guarantees weekly predictions in a MOOC with an accuracy of 98.6%. This predictive module was developed by adopting a parallel architecture implemented by Map Reduce. The proposed model has been the subject of several evaluations and comparisons with other models and results found during the literature review. The results obtained by our model are very promising and go far beyond the literature models in terms of prediction accuracy and performance.

Predicting learners at risk of dropping or failing the MOOC is very important, but knowing why people do not finish or fail is a necessity. This is the target of our future investigation; we will try to determine the causes of the dropout while seeking a way to automate the intervention to retain back this type of learners.

Acknowledgements This research was done through Stanford University's Advanced Research Center on Online Learning (CAROL), which we thank immensely for all the facilities they provided for us. We also wish to express our full gratitude to Ms. Kathy Mirzaei for her responsiveness as well as her collaboration. We wish to warmly thank Mr. Mitchell Stevens, Director of Digital Research and Planning, as well as all the CAROL commission for the trust they have given us.

References

Al-Shabandar, R., Hussain, A., Laws, A., Keight, R., Lunn, J., Radi, N. (2017). Machine learning approaches to predict learning outcomes in Massive open online courses. *Int. Jt. Conf. Neural Networks* (pp. 713—720).

Alonso-betanzos, A. (2007). Filter methods for feature selection. A comparative study. *Proc. International Conference on Intelligent Data Engineering and Automated Learning* (pp. 178—187). UK, Birmingham.

Alves, A. (2017). Stacking machine learning classifiers to identify Higgs bosons at the LHC. *Journal of Instrumentation*, *12*, 1–19.

Armbrust, M., Xin, R.S., Lian, C., Huai, Y., Liu, D., Bradley, J.K., Meng, X., Kaftan, T., Franklin, M.J., Ghodsi, A., et al. (2015). Spark SQL: Relational Data Processing in Spark. *Proceedings of International Conference Management Data* (pp. 1383—1394). Australia, Melbourne.

Burgos, C., Campanario, M.L., de la Pena, D., Lara, J.A., Lizcano, D., Martinez, M.A. (2018). Data mining for modeling students' performance: A tutoring action plan to prevent academic dropout. *Computer Electrical Engineering*, *66*, 541–556.

Chaplot, D.S., Rhim, E., Kim, J. (2015). Predicting student attrition in MOOCs using sentiment analysis and neural networks. *Proc. CEUR Workshop*, *1432*, 7–12.

Choudhury, S., & Bhowal, A. (2015). Comparative analysis of machine learning algorithms along with classifiers for network intrusion detection. *Proceedings of International Conference in Smart Technology of Management Computer Communication Controlling Energy Material* (pp. 89—95). India, Chennai.

Cross, S. (2013). Evaluation of the OLDS MOOC curriculum design course: participant perspectives expectations and experiences. *OLDS MOOC Proj.*

Crossley, S., Paquette, L., Dascalu, M., McNamara, D.S., Baker, R.S. (2016). Combining click-stream data with NLP tools to better understand MOOC completion. *Proc. Sixth Int. Conf. Learn. Anal. Knowl.* (pp. 6—14). UK, Edinburgh.

Dinakar, K., Weinstein, E., Lieberman, H., Selman, R. (2014). Stacked Generalization Learning to Analyze Teenage Distress. *Proceedings of the Eighth International AAAI Conference on Weblogs and Social Media* (pp. 81—90). USA, Michigan.

Fei, M., & Yeung, D.-Y. (2018). Temporal Models for Predicting Student Dropout in Massive Open Online Courses. *IEEE International Conference on Data Mining Working* (pp. 256—263). Singapore.

Gitinabard, N., Khoshnevisan, F., Lynch, C.F., Wang, E.Y. (2018). Your Actions or Your Associates? Predicting Certification and Dropout in MOOCs with Behavioral and Social Features. *Proc. 11th International Conference on Educational Data Mining*. Buffalo NY: In Press.

Healey, S.P., Cohen, W.B., Yang, Z., Brewer, C.K., Brooks, E.B., Gorelick, N., Hernandez, A.J., Huang, C., Hughes, M.J., Kennedy, R.E., et al. (2018). MApping forest change using stacked generalization: An ensemble approach. *Remote Sensing Environment*, *204*, 717–728.

Jindal, P., & Kumar, D. (2019). A Review on Dimensionality Reduction Techniques, *International Journal Pattern Recognition of Artificial Intelligence*. In Press.

Jović, A., Brkić, K., Bogunović, N. (2015). A review of feature selection methods with applications *Proceedings of 38th International Convenience of Information Communication Technology Electronic Microelectronics* (pp. 1200—1205). Croatia, Opatija.

Kabir, A., Ruiz, C., Alvarez, S.A. (2014). Regression, Classification and Ensemble Machine Learning Approaches to Forecasting Clinical Outcomes in Ischemic Stroke. *Biomedical Engineering Systems and Technologies*, *452*, 376–402.

Karegowda, A.G., Manjunath, A.S., Jayaram, M.A. (2010). Feature Subset Selection Problem using Wrapper Approach in Supervised Learning. *International of Journal Computer Application*, *1*, 13–17.

Kloft, M., Stiehler, F., Zheng, Z., Pinkwart, N. (2014). Predicting MOOC Dropout over Weeks Using Machine Learning Methods. *Proc. Conf. Empir. Methods Nat. Lang. Process.* (pp. 60—65).

Li, J., Cheng, K., Wang, S., Morstatter, F., Trevino, R.P., Tang, J., Liu, H. (2018). Feature selection: a data perspective, *ACM Computer Survey, 50*.

Liyanagunawardena, T.R., Parslow, P., Williams, S.A. (2014). Dropout: MOOC participants' perspective. *Proceedings of European MOOC Stakehold* (pp. 95–100). Switzerland: Summit.

Martínez-España, R., Bueno-Crespo, A., Timón, I., Soto, J., Muñoz, A., Cecilia, J.M. (2018). Air-pollution prediction in smart cities through machine learning methods: A case of study in Murcia. *Spain, Journal University of Computer Science*, *24*, 261–276.

Meng, X., Bradley, J., Yavuz, B., Sparks, E., Venkataraman, S., Liu, D., Freeman, J., Tsai, D.B., Amde, M., Owen, S. (2016). Others MLlib: Machine Learning in Apache Spark. *Journal of Machine Learning Research*, *17*, 1235–1241.

Naghibi, S.A., Ahmadi, K., Daneshi, A. (2017). Application of Support Vector Machine, Random Forest, and Genetic Algorithm Optimized Random Forest Models in Groundwater Potential Mapping. *Water Resources Management*, *31*, 2761–2775.

Nagi, S., & Bhattacharyya, D.K. (2013). Classification of microarray cancer data using ensemble approach. *Network Modelling Analysis of Health Informatics Bioinforma*, *2*, 159–173.

Onah, D.F., & Sinclair, J. (2014). Boyatt Dropout Rates of Massive Open Online Courses: Behavioural Patterns MOOC Dropout and Completion: Existing Evaluations, *Proceedings of 6th International Conference on Education* (pp. 1–10). Spain: New Learn. Technol.

Panthong, R., & Srivihok, A. (2015). Wrapper Feature Subset Selection for Dimension Reduction Based on Ensemble Learning Algorithm. *Procedia Computer Science*, *72*, 162–169.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2012). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Prieto, L.P., Rodríguez-Triana, M.J., Kusmin, M., Laanpere, M. (2017). Smart school multimodal dataset and challenges. *Proceedings of CEUR Workshop*, *1828*, 53–59.

Qi, Q., Liu, Y., Wu, F., Yan Xi., Wu, N. (2018). Temporal Models for Personalized Grade Prediction in Massive Open Online Courses. *Proceedings of ACM Turing Celebration Conference* (pp. 67—72).

Qiu, L., Liu, Y., Hu, Q., Liu, Y. (2018a). Student dropout prediction in massive open online courses by convolutional neural networks. *bSoft Computer*, *22*, 1–15.

Qiu, L., Liu, Y., Liu, Y. (2018b). An integrated framework with feature selection for dropout prediction in massive open online courses. *IEEE Access*, *6*, 71474–71484.

Ren, Y., Zhang, L., Suganthan, P.N. (2016). Ensemble Classification and Regression-Recent Developments, Applications and Future Directions. *IEEE Computer of Intelligence Magazine*, *11*, 41–53.

Salcedo-Sanz, S., Cornejo-Bueno, L., Prieto, L., Paredes, D., García-Herrera, R. (2018). Feature selection in machine learning prediction systems for renewable energy applications. *Renewable and Sustainable Energy Reviews*, *90*, 728–741.

Sanchez-Gordon, S., & Luján-Mora, S. (2016). How could MOOCs become accessible? The case of edX and the future of inclusive online learning. *Journal University of Computer Science*, *22*, 55–81.

Sikora, R., & Al-Laymoun, O. (2014). A Modified Stacking Ensemble Machine Learning Algorithm Using Genetic Algorithms. *Handbook of Research on Organizational Transformations through Big Data Analytics*, *23*, 43–53.

Sinha, T., Jermann, P., Li, N., Dillenbourg, P. (2014). Your click decides your fate: Inferring Information Processing and Attrition Behavior from MOOC Video Clickstream Interactions. *Proceedings of Conference Empirial Methods Nat. Lang. Process.* (pp. 6—14).

Talavera, L. (2005). An Evaluation of Filter and Wrapper Methods for Feature Selection in Categorical Clustering. *Proceedings of International Symposium on Intelligent Data Analysis* (pp. 440—451). Spain, Madrid.

Tang, C., Ouyang, Y., Rong, W., Zhang, J., Xiong, Z. (2018). Time series model for predicting dropout in massive open online courses, *Proc. International conference on artificial intelligence in education* (pp. 353–357). UK.

Vitiello, M., Walk, S., Helic, D., Chang, V., Gütl, C. (2018). User behavioral patterns and early dropouts detection: Improved users profiling through analysis of successive offering of MOOC. *Journal University of Computer Science*, *24*, 1131–1150.

White, T. (2012). *Hadoop: The definitive guide*. USA: O'Reilly Media, Inc.

Witten, I. (2016). *Data mining: Practical machine learning tools and techniques*. Burlington: MorganKaufmann.

Xing, W., Chen, X., Stein, J., Marcinkowski, M. (2016). Temporal predication of dropouts in MOOCs: Reaching the low hanging fruit through stacking generalization. *Comput. Human Behav.*, *58*, 119–129.

Xu, S., Lu, B., Baldea, M., Edgar, T.F., Nixon, M. (2018). An improved variable selection method for support vector regression in NIR spectral modeling. *Journal Process Control*, *67*, 83–93.

Yang, D., Sinha, T., Adamson, D. (2016). 'Turn on, Tune in, Drop out': Anticipating student dropouts in Massive Open Online Courses. *Proc. NIPS Work. Data Driven Educ.* (pp. 1—8).

Yuan, L., & Powell, S. (2013). MOOCS and disruptive innovation: Implications for higher education. *In-depth eLearning Papers*, *33*, 1–7.

Zhu, Y., Xie, C., Wang, G.J., Yan, X.G. (2017). Comparison of individual, ensemble and integrated ensemble machine learning methods to predict China's SME credit risk in supply chain finance. *Neural Computer Applications*, *28*, 41–50.

Zitlau, R., Hoyle, B., Paech, K., Weller, J., Rau, M.M., Seitz, S. (2016). Stacking for machine learning redshifts applied to SDSS galaxies. *Monthly Not. R. Astron. Soc.*, *460*, 3152–3162.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Affiliations

Mourdi Youssef[1] · Sadgal Mohammed[1] · El Kabtane Hamada[1] · Berrada Fathi Wafaa[1]

Sadgal Mohammed
sadgal@hotmail.com

El Kabtane Hamada
elkabtanehamada@gmail.com

Berrada Fathi Wafaa
w.f.berrada@gmail.com

[1] Computer Science Departement, CADI AYYAD University, Marrakech, Morocco