BILINGUAL PUBLISHING CO.
Pioneer of Global Academics Since 1984

Journal of Computer Science Research
http://ojs.bilpublishing.com/index.php/jcsr

ARTICLE

Churn Prediction Task in MOOC

Lisitsyna Liubov* Oreshin S.A.

ITMO University, Kronvrkskiy pr. 49, Saint Petersburg, 197101, Russia

ARTICLE INFO

Article history
Received: 18 February 2018
Accepted: 5 March 2019
Published: 7 March 2019

Keywords:
Machine learning
Data science
Exploratory data analysis
Logistic regression
Gradient boosting on trees
Stacking
Classification; Ranking

ABSTRACT

Churn prediction is a common task for machine learning applications in business. In this paper, this task is adapted for solving problem of low efficiency of massive open online courses (only 5% of all the students finish their course). The approach is presented on course "Methods and algorithms of the graph theory" held on national platform of online education in Russia. This paper includes all the steps to build an intelligent system to predict students who are active during the course, but not likely to finish it. The first part consists of constructing the right sample for prediction, EDA and choosing the most appropriate week of the course to make predictions on. The second part is about choosing the right metric and building models. Also, approach with using ensembles like stacking is proposed to increase the accuracy of predictions. As a result, a general approach to build a churn prediction model for online course is reviewed. This approach can be used for making the process of online education adaptive and intelligent for a separate student.

1. Introduction

The main problem of using Massive Open Online Courses (MOOC) is their low performance (no more than 5%), which is estimated as the proportion of successfully completing the course to the total number of students registered at the start of this course. The low performance analysis of MOOC [1] revealed a number of reasons related to the poor readiness of listeners for e-learning, with low motivation to achieve higher learning outcomes. To solve the problem, approaches [1–4] have been proposed and experimentally confirmed, aimed at situational awareness training of the student when working with electronic forms before learning.

In this paper, we adapt a churn prediction task to predict students' churn in MOOCs. Classical churn prediction task is about building a model which finds a list of clients who are likely to break their contract. This task is solved to predict students' churn in classical higher education [5]. If adapt this task to MOOCs, the formulation is different. Firstly, we need to select the right time period in the course, so we can use the data of students' activity before this point. There may be several such points. Secondary, we consider students a churn if they haven't finished the final exam of the course. Further, we propose an approach to solve this problem demonstrating its effectiveness on

**Corresponding Author:*
Lisitsyna Liubov,
ITMO University, Kronvrkskiy pr. 49, Saint Petersburg, 197101, Russia
Email: lisizina@mail.ifmo.ru

 DOI: https://doi.org/10.30564/jcsr.v1i1.537

online course "Methods and algorithms of graph theory" by IFMO University.

This article proposes a user-based approach to sampling statistical data recorded by the e-learning system during the course to predict the performance of an online course. After the correct sample is collected, the problem is formulated in machine learning terms. Proposed in the paper approach of constructing the correct sample for prediction the performance of online courses and building predictive models is used for the further development of the MOOC platforms with the aim of increasing personalized monitoring of the e-learning process and adaptation of a platform to a student.

2. Exploratory Data Analysis and Data Collection

This section presents the process of collecting pure data from logs of activity in the platform, aggregating this data by every student and choosing the correct time period in the course to build predictions on.

2.1 Course Material Overview

The study used statistical data accumulated on the national open education platform of the Russian Federation during the online course "Methods and algorithms of graph theory" (https://openedu.ru/course/ITMOUniversity/AGRAPH/) for the period from 2016 to 2018. This online course [6-7] is conducted for 10 weeks twice a year (at the beginning of the fall and spring semesters), contains 41 video lectures with surveys and 11 interactive practical exercises. On the 10th week an online exam is held. Table 1 presents practical exercises presented in the course.

Table 1. Practical exercises of the course

Task number	Typical graph problem	Algorithm	Week number
1	Search shortest route	Lee algorithm	2
2	Search route with minimal weight	Bellman-Ford algorithm	2
3	Search for Hamilton loops	Roberts-Flores algorithm	3
4	Search for minimum spanning tree	Prim algorithm	4
5	Search for minimum spanning tree	Kruskal algorithm	4
6	Search for largest empty subgraphs	Magu-Weismann algorithm	5
7	Minimum vertex coloring of graph	Method based on Magu-Weisman algorithm	6
8	Minimum vertex coloring of graph	Greedy heuristic algorithm	6
9	Search perfect matching in a bipartite graph	Hungarian algorithm	7
10	Detecting of isomorphism of two graphs	Algorithm based on ISD method	8
11	Graph planarization	Gamma-algorithm	9

To select the most appropriate time period to build predictions on, analysis of practical exercises in the middle of the course was performed. Figure 1 presents average maximum time needed to a student to complete a practical exercise. As we can see, Magu-Weismann algorithm is held on 5th week (middle points of the course) and has a bimodal distribution, which means that this task is complicated for some number of students.

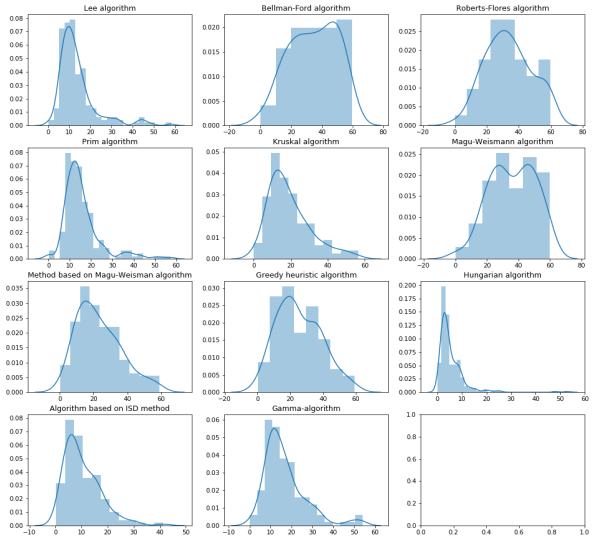

Figure 1. Average maximum time (in minutes) taken to compete exercise

In addition, Figure 2 presents mean amount of tries of students to pass a practical exercise. We can see that task 6 (Magu-Weismann algorithm) has the biggest number of mean amounts of tries. So, the hypothesis that the fact about passing this exercise can be a good feature for a further model and the 5th week is the best time to build predictions on was proposed.

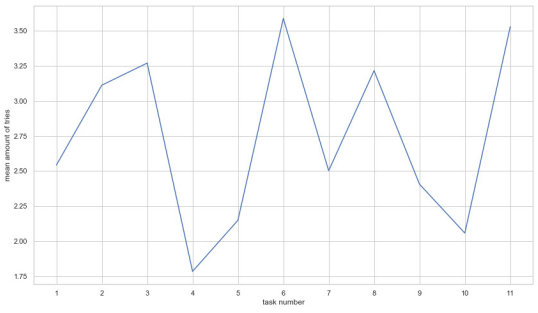

Figure 2. Average number of tries of practical exercises

2.2 Collecting and Analyzing Data-set for Further Predictions

After the point of building predictions was chosen, we

DOI: https://doi.org/10.30564/jcsr.v1i1.537

collected different features of statistical activity of every student of watching lectures, solving practical exercises and quizzes, activity on forum and other. As a result, 50-dimensional feature space was created. As a target, we took a binary feature of a fact of passing a final exam (1 – passed; 0 – not passed). Figure 2 shows to descending trend of overall activity of students for the first 5 weeks.

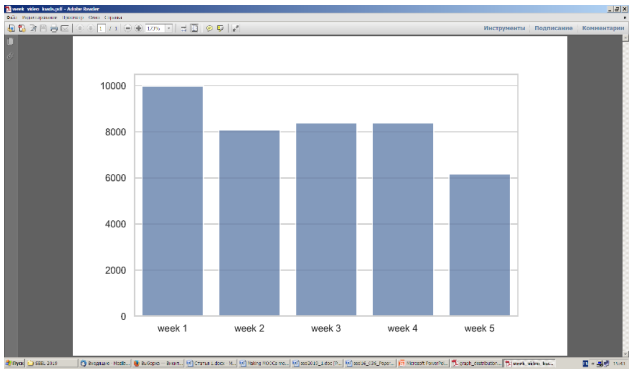

Figure 2. Overall activity of students in the first 5 weeks of the course

To visualize 50-dimensional feature space on a plain, t-SNE algorithm was applied [8, 9] (Figure 3). Students who passed the exam are marked with orange color and other students are marked blue. From the figure, most of the students who passed the exam are grouped in one area in both projections. It means that there is a hyperplane in the original dimension of features that separates the majority of students who pass the exam more likely, so the problem has a solution.

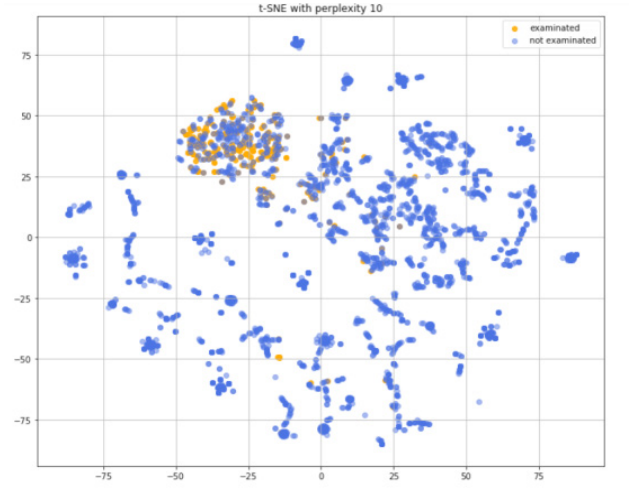

Figure 3. The projection of students on a two-dimensional space using t-SNE algorithm (perplexity equal 50)

Concluding this section, we establish that 5th week is an appropriate time period to build predictions. If we choose a later time, then the number of students that we could try to keep will be less. The choice of this period is confirmed by the Mage-Weisman algorithm, which is the most difficult in the course. Also, the assessment of the solution of the problem is confirmed on the graph of the t-SNE projection.

3. Machine Learning Part

In this section, we formulate churn prediction problem in machine learning terms and build an ensemble model for classification.

3.1 Problem Overview

The purpose of building a model in this task is to rank the participants according to their likelihood of passing the exam. To evaluate the models, the ROC AUC metric [10] was chosen due to the operation of the probabilities of the object belonging to the class with different thresholds. To determine the threshold, the expected number of participants is used based on the historical data of mean number of students who passed the exam in each session. After building a predictive model, participants are ranked according to their likelihood to successfully complete the course. The group of students, which is located below the selected threshold, is a group on which additional effects are required to increase the likelihood of a successful completion of the course, and, accordingly, increase the effectiveness of their learning. The formulation of the problem is a probabilistic binary classification.

3.2 Building Classifier Model

To build a baseline for this classification problem, support vector machine [11], logistic regression[12, 13], random forest [14] and gradient boosting on decision trees (GBDT) [15, 16] were chosen and validated. Due to the small sample size, the evaluation and comparison of the models was implemented through a cross-validation with using different sessions as folds, which allowed to consider the time component in the data. The session, which took place in the Fall of 2018, was chosen as a test set.

Table 2 present the final results of cross-validation for these models of ROC-AUC value and its std. As we can see from the table, GBDT has the best value of the chosen metric. But we have a hypothesis that we can improve out baseline due stacking [17, 18]. For this purpose, we choose one linear model and one tree-based model. We chose logistic regression as a linear model for stacking because SVM has constant probabilities with Radial basis function kernel (RBF) [19], which is not appropriate for ROC-AUC and stacking. SVM with linear kernel is not able to operate with probabilities as other models because it does not apply any sigmoid function. GBDT model was chosen as

 DOI: https://doi.org/10.30564/jcsr.v1i1.537

a tree-based model for further improvement.

Table 2. Results of cross-validation for baseline models

Model	Result of ROC AUC
Logistic Regressor	0.8699 ± 0.0274
Support vector machine	0.8763 ± 0.0258
Random Forest	0.9027 ± 0.0353
Gradient boosting on trees	0.9153 ± 0.0312

After the chosen two models were fitted, they were compared due the chosen metric and the similarity of their prediction was analyzed. The results of comparison models using ROC-AUC is introduced in the table of final scores below. The graph of the similarity of predictions of gradient boosting and logistic regression models on a separate split of cross-validation is shown in Figure 4. Each point of the plot is a separate student. Orange points are the students who passed the exam and blue points are the others. On x and y axis the predicted probability of passing the exam by logistic regression and gradient boosting on decision trees models are shown respectively. If these two models have similar predictions, the points should be located on the blue line. But, from the figure, there are many points far from the line. Therefore, a hypothesis about improving the value of the ROC-AUC metric by using ensembling of the initial models via stacking was approved.

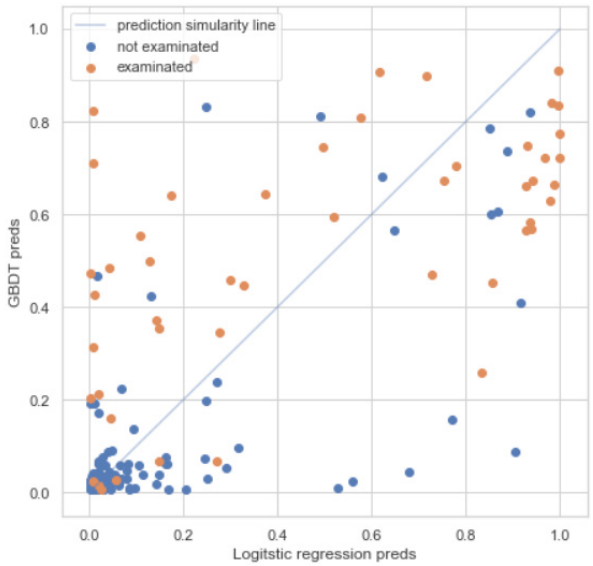

Figure 4. The similarity of the predictions of different models for students of the 3rd session from the beginning of the course via cross-validation

Stacking was applied using another logistic regression model to build new predictions on predictions of the initial models. The same session as in the figure 2 was used as a validation set for stacking due to the large variance of the predictions between two initial models.

Table 3 shows the results of the ROC-AUC score of all the applied models after the final cross-validation run. Table shows that the model of gradient boosting on trees is always preferable to logistic regression. Also, the ensemble of models in most cases shows a better quality than the gradient boosting model. From the mean results of cross-validation, the ensemble of the models gives a significant increase of ROC-AUC value. The possible reason for this is that logistic regression gives a better score using some linear dependencies in the dataset and gradient boosting is better in more complicated cases. In further calculations we will use stacking as a final model.

Table 3. Results of cross-validation

Split	Model	Result of ROC AUC
1	Logistic Regressor	0.9255
	Gradient Boosting on decision trees	0.9546
	Stacking	0.9767
2	Logistic Regressor	0.8702
	Gradient Boosting on decision trees	0.9302
	Stacking	0.9688
4	Logistic Regressor	0.9116
	Gradient Boosting on decision trees	0.9780
	Stacking	0.9742
5	Logistic Regressor	0.7659
	Gradient Boosting on decision trees	0.9117
	Stacking	0.8876
6	Logistic Regressor	0.8651
	Gradient Boosting on decision trees	0.8925
	Stacking	0.9160
Logistic regressor mean and std		0.8612 ± 0.0531
Gradient Boosting on decision trees mean and std		0.9189 ± 0.0427
Stacking mean and std		0.9304 ± 0.0459

3.3 Analysis of the Final Model

After building the final predictions, the model was analyzed. Tables 4, 5, 6 respectively show the 5 most significant features that were used by a separate model in the ensemble to build the final predictions. Table 7 provides a description of these features. The extended description of tasks is introduced in article [1]. It can be seen that different features are used by different models. As an example, for linear model an overall activity in solving interactive tasks in the course is important, but gradient boosting uses many features based on a separate week of the course.

Table 4. Feature importance for GBDT with small depth

Feature	Importance in %
Activity_sum	18.3777
Week_2_activity	10.4238
Task_1_amount_of_tries	5.4629
Task_5_amount_of_tries	5.2787
Mean_attempts	5.1692

DOI: https://doi.org/10.30564/jcsr.v1i1.537

Table 5. Feature importance for GBDT with big depth

Feature	Importance in %
Problems_solved	12.9358
Week_1_activity	11.4598
Week_2_activity	11.2761
Task_5_amount_of_tries	6.6758
Week_5_video_loads	4.0969

Table 6. Feature importance for logistic regressor

Feature	importance in %
Mean_attempts	20.1610
Task_3_amount_of_tries	11.9389
Grade_mean_rate	11.8862
Task_0_amount_of_tries	6.3207
Task_5_amount_of_tries	4.6111

Table 7. Feature meaning

Feature	Meaning
Task_1_amount_of_tries	Number of attempts of a student of solving a task with Lee algorithm
Task_2_amount_of_tries	Number of attempts of a student of solving a task with Bellman-Ford algorithm
Task_4_amount_of_tries	Number of attempts of a student of solving a task with Prim algorithm
Task_6_amount_of_tries	Number of attempts of a student of solving a task with Magu-Weismann algorithm
Mean_attempts	Mean number of attempts of a student during solving an interactive task
Problems_solved	Total number of polls solved by a student
Grade_mean_rate	Rate of the correct answers of a student
Week_1_activity	Overall activity of a student in the first week
Week_2_activity	Overall activity of a student in the second week
Week_5_video_loads	Number of viewed video by a student in the fifth week
Activity_sum	Overall activity of a student in the first 5 weeks of the course

To obtain the results of the final feature importance, the values of the calculated feature importance of each separate model were multiplied by the corresponding meta-model coefficients (0.612 for GBDT and 0.388 for logistic regression). The final feature importance is shown in Table 7. From the table it concluded that the most important features for the final model are features based on activity of students during separated weeks of the course and their overall amounts of attempts in interactive tasks and solving quizzes.

Table 8. Final feature importance

Feature	importance in %
Week_2_activity	8.2561
Mean_attempts	7.8725
Activity_sum	7.6408
Week_1_activity	5.8877
Grade_mean_rate	5.7178

4. Results and Discussions

To select a correct threshold value, we take the percentage of students who passed the exam in previous sessions (5.6%) multiplied by the number of students in the current session. Table 9 presents the results of ranking students in the test set on their likelihood to complete the course, starting with the highest probability. After applying calculated threshold, we get a list of students who need to have an additional impact to increase the effectiveness of their learning (Table 10). The last column in the tables shows whether the participant has actually passed the exam (1 for yes, 0 for no). The resulting tables show that the model correctly ranks the students of the course according to their likelihood to pass the exam in general. The resulting tables can be used to further impact a particular group of students of the course.

Table 9. Students with highest probability of examination

User	Probability of examination	Examinated
Student 11	0.8661	1
Student 12	0.8616	1
Student 13	0.8542	1
Student 14	0.8221	0
Student 15	0.8217	0
Student 16	0.8162	1
Student 17	0.8038	1
Student 18	0.7765	0
Student 19	0.7719	1
Student 110	0.7666	0

Table 10. Students below the threshold of examinations

User	Probability of examination	Examinated
Student 21	0.4741	0
Student 22	0.4453	0
Student 23	0.4352	0
Student 24	0.4232	0
Student 25	0.4216	1
Student 26	0.4163	0
Student 27	0.4015	0
Student 28	0.3793	1
Student 29	0.3771	0
Student 210	0.3348	1

 DOI: https://doi.org/10.30564/jcsr.v1i1.537

5. Conclusion

Accumulated statistics on the activity of MOOC's students allow to predict their future behavior and learning outcomes. This paper overviews all the process of building such a model: from EDA of course materials to constructing a strong classifier to predict a fact of passing an exam by a student using his activity in the first half of the course. To solve this problem, various machine learning approaches and models have been proposed. According to the results, the most significant features were obtained for assessing the fact that the exam was passed by the students. As a result of model's prediction, a list of participants was received. This approach can be used as to increase the efficiency of learning of separated students and to improve course materials in general. Also, this problem can be interpreted as a churn prediction problem. After the final list of students is received, it can be used to make the course more personal for this group of students. As an example, we suggest giving some hints and additional bonuses for the student if he will continue learning or increasing deadlines. Results of the final model analysis can be used for exploring aspects of the course that are important for a separate group of students. Thus, this article proposes a general approach for assessing and identifying MOOC students during the course, on which additional impact is required to improve the performance of e-learning using MOOC. Using this approach in MOOCs can increase effectiveness of online courses and make e-learning more self-organized and adaptive for a separate student.

References

[1] Lisitsyna L.S., Efimchik E.A., "Making MOOCs more effective and adaptive on the basis of SAT and game mechanics", Smart Education and e-Learning, 2018, 75, 56-66.

[2] Liubov S. Lisitsyna, Andrey V. Lyamin, Ivan A. Martynikhin, Elena N. Cherepovskaya, "Situation Awareness Training in E-Learning", Smart Education and Smart e-Learning, 2015, 41, 273-285.

[3] Liubov S. Lisitsyna, Andrey V. Lyamin, Ivan A. Martynikhin, Elena N. Cherepovskaya, "Cognitive Trainings Can Improve Intercommunication with e-Learning System", 6th IEEE international conference series on Cognitive Infocommunications, 2015, 39-44.

[4] Lisitsyna L.S., Pershin A.A., Kazakov M.A., "Game Mechanics Used for Achieving Better Results of Massive Online", Smart Education and Smart e-Learning, 2015, 183-193.

[5] A. Pradeep, S. Das, J. J. Kizhekkethottam, A. Cheah et al., "Students dropout factor prediction using EDM techniques", Proc. IEEE Int. Conf. Soft-Computing Netw. Secur. ICSNS 2015, pp. 544-547, 2015.

[6] Lisitsyna L.S., Efimchik E.A., "Designing and application of MOOC «Methods and algorithms of graph theory» on National Platform of Open Education of Russian Federation", Smart Education and e-Learning, 2016, 59, 145-154.

[7] Lisitsyna L.S., Efimchik E.A., "An Approach to Development of Practical Exercises of MOOCs based on Standard Design Forms and Technologies", Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, 2017, 180, 28-35.

[8] Laurens van der Maaten, Geoffrey Hinton, "Visualizing Data using t-SNE", Journal of Machine Learning Research, 2008, 9, 2579-2605.

[9] Laurens van der Maaten, "Accelerating t-SNE using Tree-Based Algorithms", Journal of Machine Learning Research, 2014, 15, 3221-3245.

[10] Tom Fawcett, An introduction to ROC analysis, In: Pattern Recognition Letters, 27 (2006), 861–874.

[11] D. Michie, D. J. Spiegelhalter, C. C. Taylor, and J. Campbell, editors. Machine learning, neural and statistical classification. Ellis Horwood, Upper Saddle River, NJ, USA, 1994. ISBN 0-13-106360-X.

[12] Chao-Ying Joanne, Peng Kuk Lida Lee, Gary M. Ingersoll, "An Introduction to Logistic Regression Analysis and Reporting", The Journal of Educational Research, 2002, 96, 3-14.

[13] Marijana Zekić-Sušac, Nataša Šarlija, Adela Has, Ana Bilandžić, "Predicting company growth using logistic regression and neural networks", Croatian Operational Research Review, 2016, 7, 229–248.

[14] Boulesteix, A.-L., Janitza, S., Kruppa, J. and König, I.R. (2012). Overview of random forest methodology and practical guidance with emphasis on computational biology and bioinformatics. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, vol. 2, no. 6, pp. 493–507.

[15] Jerome H. Friedman, Greedy Function Approximation: A Gradient Boosting Machine, In: The Annals of Statistics, 2001, 5, 1189-1232.

[16] Friedman, Jerome H., "Stochastic gradient boosting", Computational Statistics and Data Analysis, 2002, 38(4), 367–378.

[17] Smyth, P. and Wolpert, D. H., "Linearly Combining Density Estimators via stacking", Machine Learning Journal, 1999, 36, 59-83.

[18] David Opitz, Richard Maclin, Popular Ensemble Methods: An Empirical Study, Journal of Artificial Intelligence Research, 1999, 11, 169-198.

DOI: https://doi.org/10.30564/jcsr.v1i1.537

[19] Chang, Yin-Wen; Hsieh, Cho-Jui; Chang, Kai-Wei; Ringgaard, Michael; Lin, Chih-Jen (2010). "Training and testing low-degree polynomial data mappings via linear SVM". Journal of Machine Learning Research. 11: 1471–1490.

 DOI: https://doi.org/10.30564/jcsr.v1i1.537