

Personalized Impact on Students in MOOCs

Lubov Lisitsyna¹, Oreshin Svyatoslav²

¹ITMO University, Kronvrskiy pr. 49, Saint Petersburg, 197101, Russia
lisizina@mail.ifmo.ru

²ITMO University, Kronvrskiy pr. 49, Saint Petersburg, 197101, Russia
Aqice26@gmail.com

Abstract. Churn prediction is a common task for machine learning applications in business. In this paper, we adapt this task to solve a massive open online courses' low efficiency problem which formulates as a very low ratio of students who successfully finish a course. The presented approach is described and tested using course "Methods and algorithms of the graph theory" held on national platform of online education in Russia. This paper includes all the necessary steps in building 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, we overview the outcome of using this approach on real students and discuss the results and further improvements. Our personalized impact showed that the majority of students (70%) perceive such an impact positively and it helps them to pass the hardest tasks considered online course.

Keywords: Machine learning · Data science · Massive Open Online Course · Educational analytics · Learning analytics

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. Different proposes [1–6] were published and reviewed, monitoring situational awareness 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 also applied to predict

students' churn in classical higher education [7]. If adapt this task to MOOCs, the formulation is different. The main different is about the information that we have about a student (client). In MOOCs, we have very limited data about a certain student, thus we need to collect enough data to make predictions in the process of student's learning. We may also use meta information about student's previous performance on courses and his open information in profile if possible.

Firstly, we need to select the correct time period in the course, so we can use data of the whole students' activity in selected course before this point. There may be several such points. Secondly, we need to choose the correct target for predictions. We propose to use a binary target of a fact of successfully passing the final exam on course. This target may differ because of different structures and difficulties on a certain course. Further, we overview the whole process to solve selected problem using machine learning approach and demonstrating its effectiveness on online course "Methods and algorithms of graph theory" by IFMO University. Finally, we discuss the experimental results and further goals of using proposed approach on real students.

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. This approach aims to increasing personalized monitoring of the e-learning process and adaptation of a platform for a certain student.

1 Data mining and Exploratory data analysis

This section presents the process of collecting 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. The process of data mining in MOOCs strongly depends on the structure of a course. That is why we should start with analyzing a certain course material to find the best approach and strategy to make predictions on.

1.1 Course material overview

This research 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 2019. The experiment of personalized impact on students took place in spring's session in 2019. This online course [8-9] 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. As we can see, this course has many practical exercises and lasts 10 weeks of intensive studying. We need to choose the correct time period to make predictions on. By the time we make predictions, we should have enough information about student's performance and we still need to have enough time to impact on a student in such a way to increase this student's motivation thus, increase his or her chances to successfully finish this course. In other words, selecting a correct time period the have

a trade-off between timelessness of impact and having enough information about student's performance.

Table 1. Practical exercises of the course

Algorithm	Typical graph problem	Week number
Lee algorithm	Search shortest route	2
Bellman-Ford algorithm	Search route with minimal weight	2
Roberts-Flores algorithm	Search for Hamilton loops	3
Prim algorithm	Search for minimum spanning tree	4
Kruskal algorithm	Search for minimum spanning tree	4
Magu-Weismann algorithm	Search for largest empty subgraphs	5
Method based on Magu-Weisman algorithm	Minimum vertex coloring of graph	6
Greedy heuristic algorithm	Minimum vertex coloring of graph	6
Hungarian algorithm	Search perfect matching in a bipartite graph	7
Algorithm based on ISD method	Detecting of isomorphism of two graphs	8
Gamma-algorithm	Graph planarization	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 the distribution of an average time that a certain student needs 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 can be an indicator that this task is complicated for some number of students. Also, we can conclude that the average time of solving a practical task is about 20 minutes.

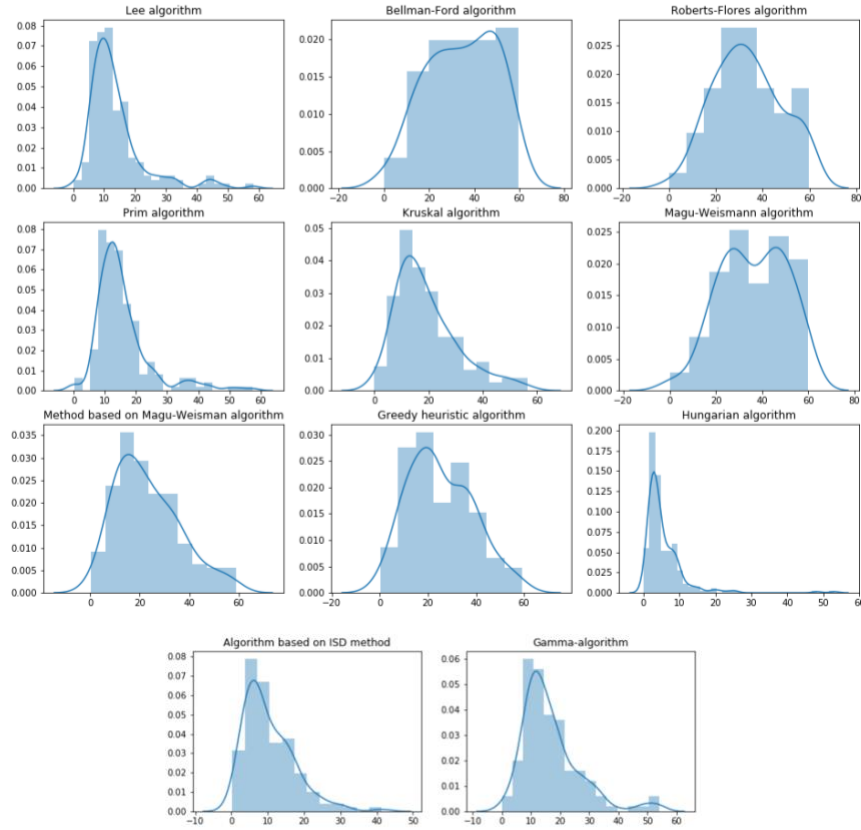


Figure 1. Average time (in minutes) taken to complete practical exercises

We have 10 weeks of course, so it is reasonable to take 5th week as a middle point of a course as a time to make predictions. Also, we can see that the most complicated task is held on 5th week. Completing this week strongly defines completing an overall course. As a conclusion, we take 5th week as a time point of a course on which we will make predictions about students' performance on final exam. As was mentioned, this time point may differ for a certain online course. Choosing this time point, we recommend estimating the ratio of overall information that we can gain about a certain student, complicatedness of problems before and after this point and fullness about already learned material.

2 Task formalization and model fitting

In this section, we formulate churn prediction problem in MOOC in machine learning terms and build a model for this binary classification task. After the model is tested and compared, we propose an ensembling approach to increase overall models' performance.

2.1 Problem overview

We have a problem of binary classification of target that indicates a successful passing the final exam of a course. Having a probability of successful passing, we rank all the students by these probabilities. Then, we need to find out students that we should apply additional impact on. These group of students should have be a group of active students but who have a low chance to successfully finish a course. Thus, the final probability should not be very low and very high. We suggest taking an expected value of students who pass an exam we use it as a higher bound of probability of this group. Then, we can take a various number of students depending on the resources that we have to apply additional impact. If we have an automatized process of applying additional impact, then we may take a big group. In our experiment we did a personalized impact sending emails with analysis certain student's problems and giving him personalized advices depending on his or her case. It means that this type of impact has a high cost of human's time to analyze each student's case. Also, we can combine several types of impact with a different number of students and different threshold of likelihood in passing final exam.

Evaluating classifiers for this task the ROC AUC metric [10] was chosen due to the operation of the probabilities of the object belonging to the class with different thresholds. Also, ROC AUC indicates the quality of ranking, which is the most important subtask to make a correct choice of this group of students to make impact on.

The formulation of the problem is a probabilistic binary classification with further likelihood's ranking.

2.2 Classifiers' fitting and analyzing

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. To evaluate different classifiers, nested cross-validation was used.

Table 2 present the 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. For GBDT algorithm, we used XGBoost and CatBoost implementations which had comparable results of ROC AUC metric.

Table 2. Results of cross-validation for baseline models

Model	ROC AUC
Logistic Regressor	0.8699
Support vector machine (rbf)	0.8763
Random Forest	0.9027
Gradient boosting on trees	0.9153

But we have a hypothesis that we can improve our baseline due to stacking [17, 18]. Figure 2 and Figure 3 present the similarity plot between Logistic regression and GBDT classifiers on different folds on cross-validation. From the plots, we can conclude that the predictions of these classifiers are very different in each certain case, but these classifiers have a high ROC AUC score.

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 because of the need for additional probability calibration, which can be bad in a general case of ensembling models. Support vector machine with a linear kernel is not able to operate with probabilities as other linear models because it does not apply any mapping into probability space. GBDT model was chosen as a tree-based model for further improvement because of the highest ROC AUC value.

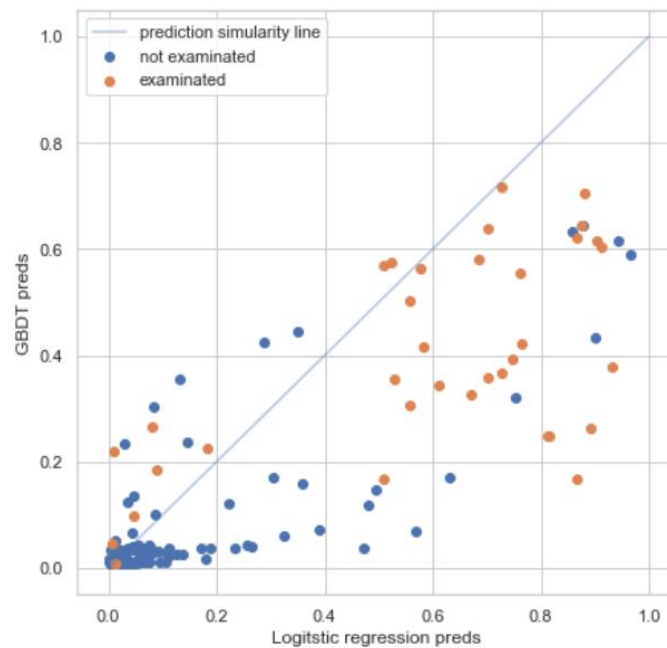


Figure 2. The similarity plot between GBTS and Logistic regression on 4th fold of cross-validation

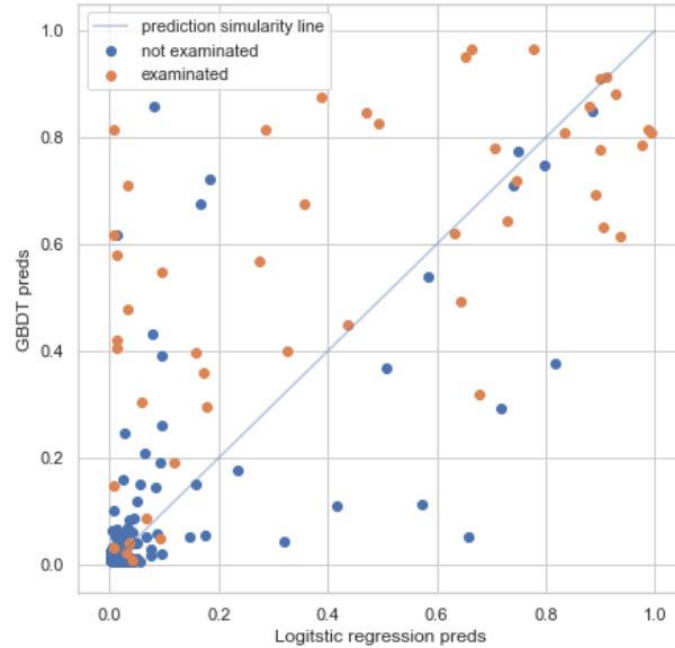


Figure 3. The similarity plot between GBTS and Logistic regression on 5th fold of cross-validation

Stacking was applied using another logistic regression model to build new predictions on predictions of the initial models. We chose the 3rd session of a course as a validation set for meta classifier in ensembling. The results of cross-validation of Logistic Regression, GBDT and meta model in this ensemble is presented in Table 3. We conclude that stacking increased the results of each classifier. We will use stacking as a final model for this problem

Table 3. Results of cross-validation for Logistic regression, GBDT and ensembling of these models

Split	Model	ROC-AUC score
1	Logistic Regression	0.9255
	Gradient Boosting on decision trees	0.9546
	Stacking	0.9767
2	Logistic Regression	0.8702
	Gradient Boosting on decision trees	0.9302
	Stacking	0.9688
3	Logistic Regression	0.9116
	Gradient Boosting on decision trees	0.9780
	Stacking	0.9742
4	Logistic Regression	0.7659
	Gradient Boosting on decision trees	0.9117

	Stacking	0.8876
	Logistic Regression	0.8651
5	Gradient Boosting on decision trees	0.8925
	Stacking	0.9160
<hr/>		
	Logistic Regression	0.8612 ± 0.0531
	Gradient Boosting on decision trees	0.9189 ± 0.0427
	Stacking	0.9304 ± 0.0459

After we fitted final model, we analyzed the results of feature importance of each model in an ensemble and calculated the feature importance for the final model. The results of feature importance for Logistic regression and Gradient Boosting are presented in Table 4 and Table 5 correspondingly. We took 5 the most important features for each base model in ensemble and calculated scalar product of these importance with the importance of meta model in stacking. As a result, we get the results of feature importance for the final model which is presented in Table 6. The most important features for Logistic regression are overall mean number of attempts in interactive tasks and mean number of attempts on a 4th week. The most important features for GBDT are overall activity of a student during and activity of a student on the 2nd week. We can conclude that basically Logistic regression classifier pays attention on features with a statistic related with attempts on interactive tasks and GBDT classifier pays more attention on overall activity statistics. All these features were calculated before the 5th week of the course.

The most important features for final model are the composition of feature importance from Table 4 and Table 5 with meta model's coefficients. As we can conclude, the most important feature for the final model is activity of a student on the 2nd week of the course. Features as an overall number of attempts on a course and overall activity are also important.

Table 4. Top 5 the most important features for Logistic regression

Feature	Importance
Mean number of attempts on interactive tasks	20.16%
Number of attempts of the 4 th interactive task	11.94%
Mean grade score	11.89%
Number of attempts of the 1 st interactive task	6.32%
Number of attempts of the 6 th interactive task	4.61%

Table 5. Top 5 the most important features for Gradient Boosting

Feature	Importance
Overall activity	18.38%
Activity on the 2 nd week	10.43%
Number of attempts of the 2 nd interactive task	5.46%
Number of attempts of the 6 th interactive task	5.28%
Mean number of attempts on interactive tasks	5.17%

Table 6. Top 5 the most important features for the final model

Feature	Importance
Activity on the 2 nd week	8.25%
Number of attempts of interactive tasks	7.87%
Overall activity	7.64%
Activity on the 1 st week	5.89%
Mean number of attempts of interactive tasks	5.72%

In this section we discussed the formulation of the problem in machine learning terms, fitted and compared different classifiers and analyzed its feature importance. Finally, we applied ensembling strategy to increase overall performance of Logistic regression and Gradient boosting classifiers and analyzed its feature importance. Now, we can use fitted models to predict a boundary group of students to make a personalized impact on each group. In the next chapter we demonstrate the results of our experiment of making a personalized impact on students in Spring of 2019 session of the course.

3 Results of the experiment

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 7 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 8). 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: there are only 2 of students who actually passed the final exam in the table with ranked students below selected threshold. Table 8 was used to create a group of students to make personalized impact on. We took 10 students for our experiment and made a personalized impact on this group of students.

Table 7. Students with highest probability of examination

Student	Probability of examination	Examined
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 8. Students below the threshold of examinations

Student	Probability of examination	Examined
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

After the boundary group of students was revealed, we made a personalized impact on that group. Our impact was sending email with personalized analysis of each student performance with hints and advices. This kind of impact is useful in this particular course because of the following reasons:

- 1) Student feels personalized treatment from the author of the course, and it increases his or her involvement in learning process.
- 2) Student can see his or her mistakes in solving practical exercises, so it increases his or her chances on further success in solving his or her problematic task.
- 3) Student gets personalized advices about particular topics that he or she should pay more attention at.
- 4) Student can give his or her feedback about the course and ask questions about incomprehensible topics.

The methodology of making personalized impact may be different in each case. In MOOCs there are many parts of quizzes and practical exercises that we can add personalized feedback on if we correctly indicate the boundary group of students. It is important to give personalized feedback that is suitable for each particular student because in some cases it can decrease student's motivation because of apparent simplicity. In other case, it can be not relevant to waste additional resource to make feedback on students who already gave up or didn't plan to successfully finish a course.

The results of using proposed personalized impact are presented in Figure 4. 70% of students from the target group positively responded to our mailings. 50% of students accepted our suggestions of helping, and 30% of students successfully passed the hardest task of the course. Such results indicate that students perceive such an impact positively and it helps students to pass the tasks that they feel problems with.



Figure 4. The results of personalized impact on students in Spring's 2019 session

5 Conclusion

In this research we indicated a problem of MOOCs' low efficiency and proposed an approach to solve this problem using machine learning algorithms. We used online course "Methods and algorithms of graph theory" to show all the steps in building such a solution. We compared different classifiers and proposed an approach to increase overall quality using stacking. 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 according to their probabilities of passing a final exam. 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 or she 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 MOOCs' 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. Finally, we present the results of our experiment of using personalized impact on found boundary group of students. This personalized impact was positively perceived by students and helped them to solve the hardest practical problem of a course.

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. Oreshin S.A., Lisitsyna L.S., Machine learning approach of predicting learning outcomes of MOOCs to increase its performance//Smart Innovation, Systems and Technologies, IET - 2019, Vol. 144, pp. 107-115
6. Oreshin S.A., Lisitsyna L.S., Sampling and analyzing statistical data to predict the performance of MOOC//Smart Innovation, Systems and Technologies, IET - 2019, Vol. 144, pp. 77-85
7. 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.
8. 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.
9. 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.
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.
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.