An Ensemble Method to Predict Student Performance in an Online Math Learning Environment

Martin Stapel
Department of Computer Science
Humboldt University of Berlin
Berlin, Germany
martin.stapel@hu-berlin.de

Zhilin Zheng
Department of Computer Science
Humboldt University of Berlin
Berlin, Germany
zhilin.zheng@hu-berlin.de

Niels Pinkwart
Department of Computer Science
Humboldt University of Berlin
Berlin, Germany
niels.pinkwart@hu-berlin.de

ABSTRACT

The number of e-learning platforms and blended learning environments is continuously increasing and has sparked a lot of research around improvements of educational processes. Here, the ability to accurately predict student performance plays a vital role. Previous studies commonly focused on the construction of predictors tailored to a formal course. In this paper we relax this constraint, leveraging domain knowledge and combining a knowledge graph representation with activity scopes based on sets of didactically feasible learning objectives. Specialized scope classifiers are then combined to an ensemble to robustly predict student performance on learning objectives independently of the student's individual learning setting. The final ensemble's accuracy trumps any single classifier tested.

Keywords

educational data mining, student performance prediction, ensemble methods, knowledge graph

1. INTRODUCTION

Performance prediction is one cornerstone of a fully personalized learning environment and also an important component of the efforts to deliver quality education. Higher education institutes, for example, are striving to incorporate predictive elements into their educational processes to better support students. Online systems like Massive Open Online Courses, Intelligent Tutoring Systems (ITSs) and increasingly Learning Management Systems (LMSs) also look for methods to compensate the lack of face-to-face interactions with teachers and the resulting problems with student's retention, completion, and graduation rates. Knowledge engineering and Educational Data Mining (EDM) methods and tools have helped to increasingly sharpen the models of student knowledge within these environments.

The foundations for performance prediction and student modeling were introduced more than four decades ago with Knowledge Tracing [1] and have since been constantly refined and extended to build diverse student models [3, 7, 17]. Such models are widely used in ITSs to allow for adaptive and personalized behavior. Technological advancements and innovations enabled the development of more elaborate online learning environments that reduce learning costs [8] and overcome space and time limitations. Through the use of such systems, previously inaccessible data about student's learning behaviors and their activities are now at hand. Analyzing student activities has become an important EDM task [2].

Data mining and machine learning approaches are often employed for the student performance prediction task since classification is one of the most frequently studied challenges by data mining and machine learning researchers. Such analyzes showed the ability to predict student's performance [15, 25] and even their drop out [14] in a broad range of educational technology environments. Usually, such prediction efforts are centered around a rather formal course students have to follow, like a university course or a structured online-only course. In this paper, we focus on a learning technology system that deliberately refrains from such a course structure.

This math learning system – called bettermarks – offers its users, students and teachers alike, guidance without imposing a course on them. The learning platform supports different curricula as well as flexible teacher interventions and leads students to a particular learning objective at their pace. The learning objectives range from introductory knowledge to advanced concepts. For our work in this blended K-12 learning environment where students either work in a traditional school setting or on their own, we opted to focus on performance data for the prediction task. We combine measured performance data with a knowledge graph representation of the platform's learning objectives, without the need for a strict course structure. Pursuing the prediction problem from this angle fully utilizes the math content organization and thereby directly connects extensive domain expertise and machine learning methods. The knowledge graph models how learning objectives are interconnected via pre-knowledge requirements. We use this graph to identify didactically feasible activity scopes. Based on those, special-

ized classifiers are trained and finally combined to predict student performance on a learning objective in an ensemble.

The remainder of this paper is organized as follows. In Section 2, we review how student modeling is approached in traditional ITSs and recent research on student performance prediction in different environments. Section 3 introduces the specific usage scenario of the bettermarks platform, its distinct characteristics, and the dataset. The following Section 4 describes our research method, including the generation of the classifier ensemble. Section 5 presents our findings and Section 6 concludes the paper with a discussion.

2. STATE OF THE ART

For Intelligent Tutoring Systems, student modeling is one major task which has been used for making assumptions about student's latent attributes. It uses observations of student's performance (e.g., correctness of given answers) or student's actions (e.g., the time a student spent on an exercise) to estimate student's hidden attributes, like knowledge, preferences or even motivational state. Which usually cannot be detected directly.

A well-established method for student modeling that has been used in various fashions for more than 40 years now is called Knowledge Tracing (KT). This technique was pioneered by Atkinson [1] and substantially developed by Corbett and Anderson. Their variant is based on a 2-state dynamic Bayesian network [7]. The observed variable is the student performance, and the student knowledge is the latent one which is estimated. Regarding student performance, there are two additional parameters to account for accidental and careless mistakes (slip) and solving an exercise despite not knowing (guess). The set of parameters is completed with one for any prior knowledge a student might already have and one for her learning rate. This standard KT model is often used for its abilities to provide skill level diagnostics. In recent years, a range of extensions to Knowledge Tracing have been proposed to mitigate some of its shortcomings. A particularly noteworthy one is Baker et al.'s contextual guess and slip model [3]. Recently, Pardos and Heffernan proposed an extension to the standard model to incorporate item-level difficulty [17].

Besides KT, other approaches exist. A comparably new option is called Performance Factor Analysis (PFA) which was proposed by Pavlik et al. [19]. Their student modeling method uses a logistic regression model with a reconfigured version of Learning Factor Analysis [6] whose skill variable is replaced by one parameter per item (e.g., exercise, question, knowledge component) and the student variable is dropped entirely. The model estimates the individual item difficulty as well as effects of prior successes and failures for each skill. It predicts student performance based on item difficulty and prior performances. Comparative analyzes of KT's and PFA's performance showed that either of them appear to be suitable for student modeling [4, 10, 19].

In learning environments without such semantically rich data and a domain model, data mining, and machine learning approaches are often applied for the performance prediction task. The goals here remain mainly the same, with additional emphasis on early warning and drop out prediction. In general, student's prior performances are used to train different machine learning models to predict future test or exam performance, similarly to PFA. However, not all environments provide access to performance data. The steadily growing number of LMSs, for instance, do not always collect such data. In such environments, one has to resort to data about student's activities. Hu et al. developed an early warning system based on student's usage of an LMS utilizing metadata captured while students interact with the system [12]. The studied dataset includes information like login counts, time spent logged in, and metadata concerning homework assignments and was gathered during two semesters of a fully online university course with 300 enrolled students. The course required students to attend online classes and watch videos in specific time periods. To build their early warning system, the authors generated three datasets to create different periods to study (4, 8 and 13 weeks) and applied three often used classification techniques, C4.5, CART, and logistic regression. Additionally, Hu et al. employed AdaBoost to achieve greater prediction accuracy which led to the best performing classifier constructed from AdaBoost and CART. This classifier achieved a prediction accuracy of at least 0.972 on each of the three datasets. A similar scenario, yet more open, was studied by Zacharis who investigated student performance related to online activities in an LMS, which was used as part of a blended learning university course [29]. 134 students were enrolled in this course for one semester. To account for student-teacher and student-student interactions which could not be observed, all of the captured online activities were treated equally while searching for significant correlations with the student's final grades. Out of 29 variables, almost 50% were found to be important. A stepwise regression yielded a model with four variables which were used in a logistic analysis to discriminate between failing and not at risk students. An overall classification accuracy of 81.3% was achieved. Predicting student performance in a timely fashion as done by Koprinska et al. underscores the usefulness of performance data [13]. Their studied dataset included submission sets, assessment information, and engagement data from a discussion forum. All of the data was gathered from different online systems used in a blended university course. Koprinska et al. defined their classification problem as a three class problem and divided the 224 participating students into high-, average- and low-level students based on exam performance at the end of the course. To predict the exam result, they employed a decision tree classifier which achieved an accuracy score of 72.69% using the complete course data. Using just the data from the first half of the course led to an accuracy score of 66.52%. Here, almost half of the used features are performance related.

Our work uses a similar approach to predict student performance in a blended K-12 learning environment. The critical difference between other datasets used in previous research and ours is that students on the bettermarks platform neither attend nor follow a formal course. The system provides teachers and students with "math books" for a term's curriculum. Since the learning platform is often used supplementary to traditional lessons in class, teachers make use of the learning material at their discretion. Likewise, students in a self-regulated learning setting might pick a couple of learning objectives or decide to work through a whole

book on their own. The resulting freedom for students and teachers introduces a huge amount of diversity in the user behavior and poses challenges for performance prediction algorithms. To fully capture student behavior and overcome the problem of fitting a single prediction model based on diverse data sources, Essa and Ayad proposed a domain-specific decomposition of different (online) learning related aspects [9]. The final prediction would consequently consist of an ensemble of classifiers specialized on each aspect's data. Hence, the resulting model should be more generalizable and flexible than models build on single courses. Building on this idea, we focused on learning objectives as the common data underlying every user's interaction and decomposed the math content organization of the platform into different activity scopes. Classifiers trained on those scopes act as base classifiers for the developed ensemble which robustly predicts student performance independently of their learning situation.

The particularly chosen focus on exercises (or learning objectives, for that matter) in our research is a crucial distinction to prior ensemble-based prediction works. Student performance within an ITS as well as on a paper post-test was predicted by Baker et al. utilizing ensembles of different student models (including the previously discussed BKT and PFA). The achieved results let the authors conclude that ensembling appeared to be only slightly better [4]. Looking further into the previous results and concentrating exclusively on post-test predictions did not yield better prediction results over the best individual models [18]. Again, different student modeling approaches were combined to ensembles. Gowda et al. found that ensembles build on large enough datasets (about 15 times more data than used in the previous two studies) can very well yield superior prediction performance, even with similar models as a base [11].

3. THE USAGE SCENARIO

The bettermarks system is an online math learning platform with more than 100k interactive exercises, covering K-12 math curricula (grades 4-10) in English, Spanish, German and Dutch language. It is designed to be used in math classes at school without implying a formal course structure. Teachers can decide to teach math entirely with the system, supplement their lessons with related bettermarks content right in class, or assign exercises as homework. At any time, teachers can be aware of their student's progress through detailed reports which present high-level performance aggregation as well as every single solution attempt. The system can also support and guide students working on their own in a self-regulated learning setting without additional teacher interventions. Each month, more than 100k students across Europe and America use bettermarks.

Besides offering detailed textbook-like explanations of math topics, the primary means of learning math on the bettermarks platform are math exercises. Exercises are grouped into exercise series. Each series helps students achieve a well defined and fine-grained learning objective. Examples of such learning objectives are "Calculate the surface area of a prism given the edge lengths and the height" or "Find the zeros of linear and quadratic functions." These series are arranged into digital books based on curricular themes and didactical concepts without imposing any curriculum

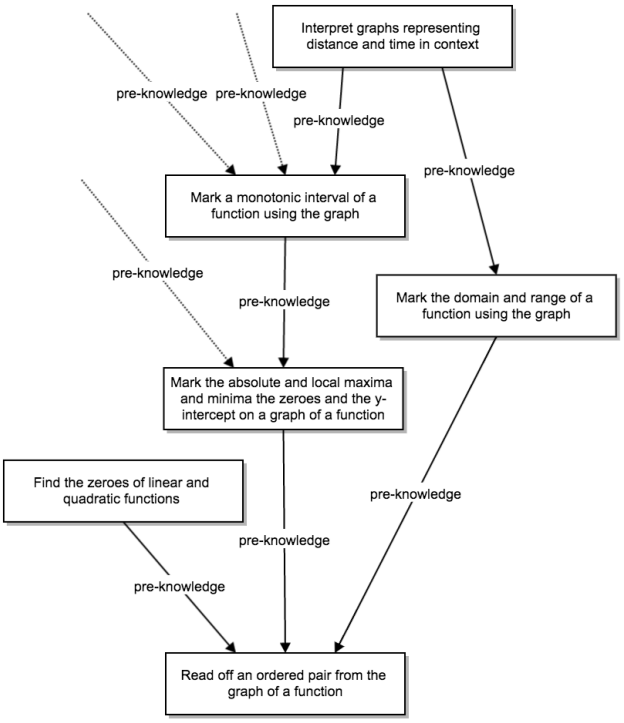

Figure 1: Small section of the entire knowledge graph spanning more than 1,500 vertices

structure on the user. Each book is organized similarly to a printed math book with chapters and series of exercises within these chapters. Behind those books that are visible to teachers and students lies a knowledge graph (not visible to users). This graph describes how learning objectives relate to each other regarding required prior knowledge.

3.1 Knowledge Graph

The idea of a concept map was first introduced in the 1970s by Novak. In his later work, he used this framework to organize and connect already acquired knowledge with new knowledge [16]. The usefulness of maps related to the original ideas for learning and assessment in technology-based learning environments has already been shown [24, 27]. Building on these concepts, the underlying structure of the bettermarks content is called a knowledge graph. This graph is built by connecting nodes concerning their pre-knowledge requirements. Each of the graph nodes represents a learning objective – a particular skill a student reaches once she successfully finishes a series of exercises designed especially for this skill. These objectives include introductory/elementary skills as well as core knowledge and advanced skills. The direction of an edge indicates which node is defined as the required pre-knowledge for another node. A particular node might have more than one pre-knowledge node. The entire bettermarks knowledge graph contains more than 1,500 learning objectives in total. A small subset of them is shown in Figure 1. A digital math book on the bettermarks platform includes a number of these learning objectives. Usually, not all of them are directly (or indirectly) related.

3.2 Data

The analysis in this paper focuses on the particularly well-frequented book "Calculating Percents" from the German version of the bettermarks system. From this book's learning objectives, we chose one with a relatively large amount of required pre-knowledge as a classification target. It is called "Calculate decreased and increased base values in context" and located close to the end of the book. The data was gathered during the entire year of 2015 and includes student's activities on the bettermarks platform 40 days before their first attempt on the classification target. The 40 day period allows students in a school setting to reasonably work their way to this objective. In total, the dataset includes performance measurements of 566 students on 903 different learning objectives which are the results of 10,363 solution attempts by 6th - 10th-grade students from all over Germany. A student is free to repeat an exercise series as often as she wants. Since the system presents the student's best solution attempt to a teacher first, we also used this result for each student and learning objective. Table 1 shows a randomly chosen sample of the entire dataset with results on three learning objectives (represented by identifiers). The results correspond to the ratio of correctly solved exercises in a series. It is evident that not all learning objectives have been addressed by the same amount of attempts. The last column shows the highest success rate on the classification target achieved by a student within 3 hours of starting the exercise series for the first time. We noticed that students employed different strategies involving repetitions while solving exercise series which makes the success rate achieved in the first attempt a bad indicator for the final result a student settles on by continuing with the next series. Therefore, the 3 hours allow students some time to repeat the exercise series and also account for the fact that students might have reached the classification target during their math lesson at school and want to repeat the exercise series again at home. These collected performance measurements are used as possible features in our classification models.

4. RESEARCH METHODOLOGY

Over the course of the following section, our research method is discussed in detail, we were guided by a two-fold research focus: (1) Can an ensemble of classifiers based on the decomposed math content organization accurately predict student performance? (2) Given the usage scenario, is this approach suitable for an "early prediction" setting? Since the bettermarks system offers its users lots of flexibility, an early prediction task is different from a formal course's early prediction task. In our case, the early prediction challenge is not transferable to a subset of the course's allocated time and exercises. Instead, we looked into students showing low usage rates over the examined period. In our case (and in contrast to online-only environments), a lack of activity does not imply that students did not attend a regular math lesson and progressed in school.

In a first step, the math content was decomposed into activity scopes relating to the classification target. A following pre-processing step used different aggregations to gain better insights into the available dataset. The primary concerns that governed this step refer to how much of the data is missing and if the classifiers can learn from roughly balanced classes created by the class split. The first question is also relevant regarding the number of actually achieved learning objectives by students within the different scopes since those directly translate into the initial feature sets. Afterwards, six different algorithms were evaluated on each scope as base classifiers for the ensemble. The process is described in the Ensemble Construction section which also discusses the imputation and standardization strategies we employed. Following the final model selection, the ensemble's weights were optimized. This step also concluded the generation of the entire ensemble.

4.1 Activity Scopes

To reflect the flexibility the learning system offers its users, we defined three activity scopes and constructed specialized classifiers for them. All scopes center around a particular subset of the knowledge graph's vertices and thus decompose the graph into relevant groups related to the classification target. The subgraph spun by the classification target's vertice via the pre-knowledge relation serves as the binding element between the three scopes.

The first scope includes all learning objectives that are part of the classification target's pre-knowledge in the knowledge graph. These are all vertices connected directly or indirectly to the classification target through pre-knowledge relation edges. In total, those are 35 different learning objectives for our chosen classification target "Calculate decreased and increased base values in context."

The classification target is located in the math book "Calculating Percents". This book with all of its learning objectives creates the second activity scope, the math book scope. Excluding the classification target itself, the set of potential features for this scope contains 24 learning objectives. Since the book was created with didactical considerations in mind, the math book's learning objectives are arranged similarly to the knowledge graphs vertices. Still, this scope and the pre-knowledge scope share only five learning objectives.

The final scope includes student's activities on learning objectives that are not part of the math book's scope. All of these learning objectives are part of the knowledge graph as well, but those are located in other math books. Nevertheless, the resulting set was not partitioned any further by their books. This scope could share up to 30 learning objectives with the first scope but does not include any from the book's scope. Those would be the learning objectives the pre-knowledge scope does not share with the book's scope. The actual number depends entirely on the student's activities during the examined period. With these defined scopes we attempted to model the different paths teachers and students might have taken to approach the classification target.

4.2 Pre-processing

In Germany, the bettermarks system is often used in math classes to supplement regular lessons. Therefore, it is not expected that students solve a vast amount of exercise series over the chosen 40 days. Figure 2 shows that the median of different exercise series per student is at 14.5 series with the 0.75 percentile at 23 series.

This result suggests that the amount of gathered performance measures per learning objective could be rather sparse

Table 1: Sample of user IDs with success rates on different learning objectives

user_id	Learning objectives				classification_target
	PruZiPruZiRFo.LOB04	PruZiPruZiRDr.LOB06	ZUZUProp.LOB01	...	
369947		0.333		...	0.675
92083	0.708	0.333		...	0.921
5625246	0.708	0.333	0.429	...	0.447
347284	0.208	0.500		...	0.475
361389	0.417	0.333		...	0.675

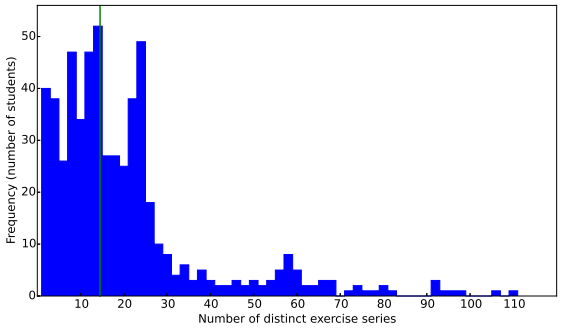

Figure 2: Students solve a rather small number of different series with the median at 14.5 series (indicated as green vertical line)

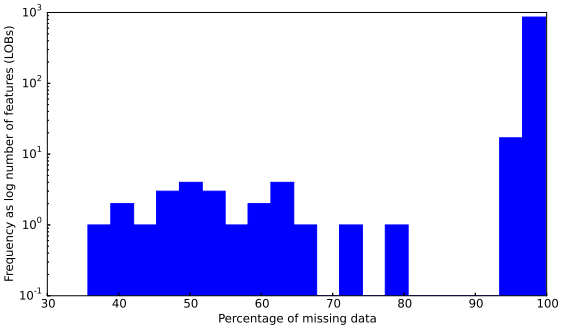

Figure 3: Data Sparsity

for the majority of series. In fact, 566 students worked on 903 different learning objectives with an average of almost 20 different series per student. Further examination reveals that only 22 learning objectives had up to 70% of the data missing. The data sparsity is illustrated in Figure 3. It is important to employ a suitable data imputation strategy and apply feature selection means during the construction of the different classifiers later to cope with this sparse dataset.

We decided to split the classes at a success rate of 0.75. One class is composed of students with success rates lower than 0.75, whereas the second one contains students with success rates of at least 0.75 which would translate to a separation of top performing students from all other students. This class split has the benefit of dealing with quite balanced classes. Figure 4 shows the median success rate at 0.76 (red) and our class split slightly left to it at 0.75 (green). The resulting spread is 45.6% to 54.4% between both classes.

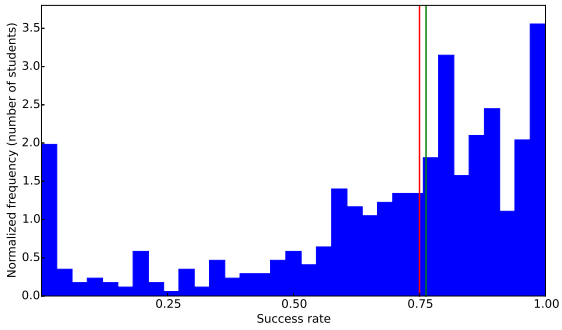

Figure 4: Measured success rates at the classification target. The red line indicates class split at 0.75 and the green one the median success rate at 0.76

The dataset does not contain the entire set of pre-knowledge learning objectives. Out of 35 possible learning objectives, only data for 16 is present. One possible explanation is that pre-knowledge learning objectives are not always part of a single term's curriculum (but available for teachers to choose from). Hence, it is not expected that students work their way through the entire pre-knowledge of a particular learning objective during a short period. All of the expected 24 book scope's objectives are present in the dataset.

4.3 Ensemble Construction

An ensemble of classifiers blends predictions from multiple models with a two-fold goal: The first intent is to boost the overall prediction accuracy compared to a single classifier. The second benefit is a better generalizability due to different specialized classifiers. As a result, an ensemble can find solutions where a single prediction model would have difficulties. A key rationale is that an ensemble can select a set of hypotheses out of a much larger hypothesis space and combine their predictions into one [22].

For our purposes, we started with a set of well-known classification algorithms and used nested cross-validation to determine their performance. The algorithm with the highest average accuracy score in each scope is afterwards chosen for final model selection. The performance of the best model was evaluated on a hold-out dataset (30% of the entire data). Once the model selection took place, the weights for the ensemble were adjusted, again, with cross-validation and the final ensemble's performance evaluated on the hold-out dataset. The following sections describe the whole process in detail.

Table 2: Average accuracy achieved in nested cross-validation for each tested algorithm and scope

Algorithm	Book	Pre-knowledge	Outside
Decision Tree with AdaBoost	**0.715**	0.634	0.525
k-Nearest Neighbors	0.629	0.609	0.546
Logistic Regression	0.682	**0.659**	0.538
Naïve Bayes	0.654	0.636	0.467
Random Forest	0.679	0.652	**0.550**
Stochastic Gradient Descent	0.624	0.594	0.525

4.3.1 Selecting Algorithms

A set of six commonly used classification algorithms were chosen as potential base models. The set consists of Random Forest, Decision Tree with AdaBoost, Logistic Regression, k-Nearest Neighbors, Stochastic Gradient Descent and a Naïve Bayes implementation. For each scope, a classification pipeline was created.[1] To impute missing data we opted for filling missing values with the mean success rate of the particular feature. Tests with the median and the mode did not significantly influence later on achieved classification results. The data was robustly standardized by removing the median and scaling the data according to the Interquartile Range (IQR)[2]. Each pipeline used a scope-specific variance threshold on the imputed data as feature selection mechanism. The actual threshold is determined during model selection (0-60% of the feature's variance). The purpose is to remove features that do not meet the set threshold. This applies to features with low variance due to rather uniform student activities as well as to features with large amounts of imputed data.

To get a conservative and thus fairly unbiased base estimate of each classifiers performance [26], we used nested stratified cross-validation with 10 folds on the outside and 5 folds on the inside with randomized search [5] over the parameter space. Depending on the algorithm, the search space was limited to reasonable values such as restricting the number of trees in a forest. The search included 100 sets of candidate parameters. Table 2 shows the results for each classification algorithm and scope. The best performing algorithm is highlighted in each column.

4.3.2 Model selection and Ensemble construction

AdaBoost on Decision Tree for the math book scope, Logistic Regression for the pre-knowledge scope and Random Forest for the outside scope were picked for the final model selection. It was done by 10-fold cross-validation and a random search over 750 sets of candidate parameters. The best performing model of each scope was afterwards chosen and re-trained on the entire training set for the ensemble.

[1]The pipeline facility, as well as the used algorithms' implementations are part of scikit-learn [20].
[2]The IQR is the range between the 1st quartile (0.25 percentile) and the 3rd quartile (0.75 percentile)

Table 3: Prediction accuracy on the test set

Classifier	Prediction accuracy
Baseline	0.594
Pre-knowledge scope	0.682
Book scope	0.705
Outside	0.647
Ensemble	**0.735**

To construct the ensemble we opted for a soft voting strategy rather than using hard voting. A soft voting strategy has the significant advantage of weighing the three scopes differently. The alternative would be to use a majority decision among the three classifiers where each classifier's vote weights equally. Instead, the ensemble uses soft voting to classify students based on the argmax of the sums of each classifier's predicted probabilities. To determine the weights to be associated with each classifier, we used random search with 10-fold cross-validation on 3k parameter sets. The emerged ensemble with tuned weights was then tested on the hold-out part of the dataset.

5. RESULTS

To assess the performance of each classifier as well as of the entire ensemble more thoroughly we also added a baseline classifier. This simple classifier always predicts the majority class. Table 3 shows each classifier's prediction accuracy on the hold-out dataset.

As before with the nested cross-validation results, the accuracy ranking over the three scopes stayed the same – the book scope's classifier performed best (0.705) followed by the pre-knowledge scope's classifier (0.682). With a prediction accuracy of 0.594, the baseline classifier scores below all other approaches. The constructed ensemble achieved the best prediction accuracy with 0.735.

Since the ensemble showed an improved accuracy on the test set, we investigated the remaining classification errors further. Table 4 displays the confusion matrix for the book scope's classifier which is the best single-scope classifier. As a comparison, Table 5 shows the confusion matrix for the final ensemble. Out of the two, the latter made slightly more errors of type I. This is especially unfortunate because in our case, false positive errors translate to students incorrectly classified as top performers even though they could not reach the required success rate threshold. In our setting, errors of this type are arguably more expensive than classification errors of type II where a student would be wrongly classified as a low scoring student. If our prediction method would be used to trigger human interventions a teacher might determine rather quickly if a student is able to pass a test or not. However, if the system fails to notify the teacher in the first place, she might not at all be aware of a potential problem with the student's performance. Thus, the problem would be revealed after the student has aleardy failed.

Table 4: Book scope classifiers's confusion matrix

	Other students	Top performers
Other students	51	18
Top performers	32	69

Table 5: Ensemble's confusion matrix

	Other students	Top performers
Other students	50	19
Top performers	26	75

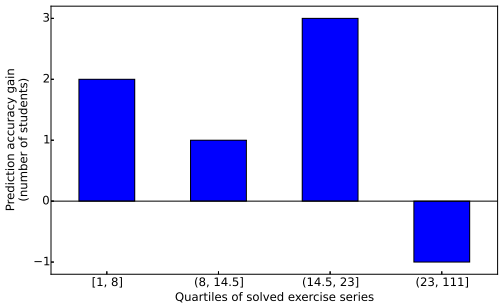

Figure 5: Ensemble's accuracy gain over book scope's classifier per quartile

Lastly, to assess the ensemble's ability to accurately predict student performance in an early prediction task, the accuracy of the best single-scope classifier and the ensemble was compared based on quartiles of student's number of solved exercise series. As described above, 50% of the students in our dataset solved up to 14.5 different exercise series in the examined period. To be used effectively in an early prediction setting, a suitable classifier needs to be able to accurately predict the right class with few data points. Figure 5 shows the accuracy difference between the book scope's classifier and the entire ensemble for each quartile. In the first three quartiles the ensemble predicts more students correctly than the book scope's classifier. These results lead to the conclusion that our approach has the potential be used in an early prediction setting.

6. DISCUSSION AND OUTLOOK

We investigated an approach that decomposes the math content structure underlying an online math learning platform, trains specialized classifiers on the resulting activity scopes and uses those classifiers in an ensemble to predict student performance on learning objectives. Students using this particular math learning platform achieve learning objectives without a formal course imposed on them which is quite different from course-centered online-only or blended learning environments. We showed that looking closer at the math exercises helped us build a robust classification model that can cope with student's notably diverse behavior due to the lack of a strict course framework. Using the knowledge graph to decompose the content domain enabled the individual prediction models to better grasp nuances of student's activities.

In general, the results suggest that our approach yields a robust performance prediction setup that can correctly classify 73.5% of the students in the dataset. This is an improvement over every other classification approach we tested in our study. Further examinations revealed that the ensemble also outperforms the best single-scope classifier in an early prediction or early warning setting. Students with lower levels of activity would benefit the most from our ensemble approach since it clearly improves the prediction accuracy for those students, as we have shown. However, the increased prediction accuracy came with a price: a slight increase in false positives where students are wrongly classified as top performing students. Especially in our area of research, false positive errors like this should be reduced as much as possible if we want to improve educational processes and make a lasting impact on every stakeholder.

Looking closer at the classification errors, we found that in 12 cases the three scope classifiers unanimously attributed the wrong class to a student. Hence, the ensemble was not able to predict the class for these students correctly either. The reason is a shortcoming of the ensemble's soft voting strategy which cannot overturn matching predictions among its base classifiers. Rather than using a simple weighted ensemble, it is possible to use stacking and thus introduce a second stage classifier. This classifier takes the prediction results of the ensemble's base classifiers and employs them as features to predict the final class. The whole concept is known as stacked generalization and exists in different flavors [28]. Gowda et al. have already shown the significant benefits of more sophisticated ensemble methods in a prediction task [11]. Additionally, a number of different ensemble generation methods can be utilized to achieve better diversity within the base classifiers [21]. Besides extending the final ensemble with stacking and exploring the resulting benefits, our future work will include more performance related data, like the number of attempts or the total time a student has spent on a particular exercise series. These efforts will go hand in hand with additional feature selection strategies, and dimensionality reduction means to capture more scope-related nuances of student's performances.

We also plan to investigate whether student's diverse sequences of learning objectives can be used to improve feature extraction and selection. Scheiter and Gerjets' results regarding the order of presented problems and performance improvements point to a possible connection [23].

While some of the discussed extensions seem obvious, the most important challenge is to develop our approach into a strategy suitable for any learning objective in this scenario. Our current approach uses a narrow set of learning objectives and a specifically tailored ensemble. These constraints reduce the cold start problem but require a good strategy to cope with missing data, as we have described. Nevertheless, the ensemble cannot easily be repurposed at scale. Hence, investigating different strategies leading to a broadly applicable solution will be our primary focus.

References

[1] R. C. Atkinson. Ingredients for a theory of instruction. *American Psychologist*, 27(10):921, 1972.

[2] R. S. J. d. Baker. *Data Mining*, pages 112–118. Elsevier, 2010.

[3] R. S. J. d. Baker, A. T. Corbett, and V. Aleven. *More Accurate Student Modeling through Contextual Estimation of Slip and Guess Probabilities in Bayesian Knowl-*

edge Tracing, pages 406–415. Springer Berlin Heidelberg, 2008.

[4] R. S. J. d. Baker, Z. A. Pardos, S. M. Gowda, B. B. Nooraei, and N. T. Heffernan. *Ensembling Predictions of Student Knowledge within Intelligent Tutoring Systems*, pages 13–24. Springer Berlin Heidelberg, 2011.

[5] J. Bergstra and Y. Bengio. Random Search for Hyper-Parameter Optimization. *Journal of Machine Learning Research*, 13(1):281–305, 2012.

[6] H. Cen, K. Koedinger, and B. Junker. *Learning Factors Analysis – A General Method for Cognitive Model Evaluation and Improvement*, pages 164–175. Springer Berlin Heidelberg, 2006.

[7] A. T. Corbett and J. R. Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction*, 4(4):253–278.

[8] D. J. Deming, C. Goldin, L. F. Katz, and N. Yuchtman. Can online learning bend the higher education cost curve? *American Economic Review*, 105(5):496–501, 2015.

[9] A. Essa and H. Ayad. Student Success System: Risk Analytics and Data Visualization Using Ensembles of Predictive Models. In *Proceedings of the 2nd International Conference on Learning Analytics and Knowledge*, pages 158–161. ACM Press, 2012.

[10] Y. Gong, J. E. Beck, and N. T. Heffernan. *Comparing Knowledge Tracing and Performance Factor Analysis by Using Multiple Model Fitting Procedures*, pages 35–44. Springer Berlin Heidelberg, 2010.

[11] S. M. Gowda, R. S. J. d. Baker, P. Zachary A, and N. T. Heffernan. The sum is greater than the parts: ensembling models of student knowledge in educational software. *SIGKDD Explor. Newsl.*, 13(2):37–44, 2012.

[12] Y.-H. Hu, C.-L. Lo, and S.-P. Shih. Developing early warning systems to predict students' online learning performance. *Computers in Human Behavior*, 36:469–478, 2014.

[13] I. Koprinska, J. Stretton, and K. Yacef. Predicting Student Performance from Multiple Data Sources. In *Artificial Intelligence in Education: 17th International Conference, AIED 2015, Madrid, Spain, June 22-26, 2015. Proceedings*, pages 678–681. Springer International Publishing, 2015.

[14] I. Lykourentzou, I. Giannoukos, V. Nikolopoulos, G. Mpardis, and V. Loumos. Dropout prediction in e-learning courses through the combination of machine learning techniques. *Computers & Education*, 53(3): 950–965, 2009.

[15] L. P. Macfadyen and S. Dawson. Mining lms data to develop an "early warning system" for educators: A proof of concept. *Computers & Education*, 54(2):588–599, 2010.

[16] J. Novak. Clarify with concept maps. *The Science Teacher*, 58(7):44, 1991.

[17] Z. A. Pardos and N. T. Heffernan. *KT-IDEM: Introducing Item Difficulty to the Knowledge Tracing Model*, pages 243–254. Springer Berlin Heidelberg, 2011.

[18] Z. A. Pardos, S. M. Gowda, R. S. J. d. Baker, and N. T. Heffernan. Ensembling predictions of student post-test scores for an intelligent tutoring system. In *EDM*, pages 189–198, 2011.

[19] P. I. Pavlik, H. Cen, and K. R. Koedinger. Performance factors analysis – a new alternative to knowledge tracing. In *Proceedings of the 2009 Conference on Artificial Intelligence in Education: Building Learning Systems That Care: From Knowledge Representation to Affective Modelling*, pages 531–538. IOS Press, 2009.

[20] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[21] A. Rahman and S. Tasnim. Ensemble classifiers and their applications: A review. *International Journal of Computer Trends and Technology*, 10(1):31–35, 2014.

[22] L. Rokach. Ensemble-based classifiers. *Artificial Intelligence Review*, 33(1-2):1–39, 2009.

[23] K. Scheiter and P. Gerjets. The impact of problem order: Sequencing problems as a strategy for improving one's performance. In *24th Annual Conference of the Cognitive Science Society*, pages 798–803. Erlbaum, 2002.

[24] D. L. Trumpower, M. Filiz, and G. S. Sarwar. Assessment for Learning Using Digital Knowledge Maps. In *Digital Knowledge Maps in Education: Technology-Enhanced Support for Teachers and Learners*, pages 221–237. Springer New York, 2014.

[25] S. Valsamidis, S. Kontogiannis, I. Kazanidis, T. Theodosiou, and A. Karakos. A Clustering Methodology of Web Log Data for Learning Management Systems. *Educational Technology & Society*, 15(2):154–167, 2012.

[26] S. Varma and R. Simon. Bias in error estimation when using cross-validation for model selection. *BMC Bioinformatics*, 7:91, 2006.

[27] K. Weinerth, V. Koenig, M. Brunner, and R. Martin. Concept maps: A useful and usable tool for computer-based knowledge assessment? a literature review with a focus on usability. *Computers & Education*, 78:201–209, 2014.

[28] D. H. Wolpert. Stacked generalization. *Neural Networks*, 5(2):241–259, 1992.

[29] N. Z. Zacharis. A multivariate approach to predicting student outcomes in web-enabled blended learning courses. *The Internet and Higher Education*, 27:44–53, 2015.