

Analysis of Click-Stream Data to Predict STEM Careers from Student Usage of an Intelligent Tutoring System

Jihed Makhlouf
Kyushu University
jihed.makhlouf@m.ait.kyushu-u.ac.jp

Tsunenori Mine
Kyushu University
mine@ait.kyushu-u.ac.jp

In recent years, we have seen the continuous and rapid increase of job openings in Science, Technology, Engineering and Math (STEM)-related fields. Unfortunately, these positions are not met with an equal number of workers ready to fill them. Efforts are being made to find durable solutions for this phenomena, and they start by encouraging young students to enroll in STEM college majors. However, enrolling in a STEM major requires specific skills in math and science that are learned in schools. Hopefully, institutions are adopting educational software that collects data from the students' usage. This gathered data will serve to conduct analysis and detect students' behaviors, predict their performances and their eventual college enrollment.

As we will outline in this paper, we used data collected from the students' usage of an Intelligent Tutoring System to predict whether they would pursue a career in STEM-related fields. We conducted different types of analysis called "problem-based approach" and "skill-based approach". The problem-based approach focused on evaluating students' actions based on the problems they solved. Likewise, in the skill-based approach we evaluated their usage based on the skills they had practiced. Furthermore, we investigated whether comparing students' features with those of their peer schoolmates can improve the prediction models in both the skill-based and the problem-based approaches. The experimental results showed that the skill-based approach with school aggregation achieved the best results with regard to a combination of two metrics which are the Area Under the Receiver Operating Characteristic Curve (AUC) and the Root Mean Squared Error (RMSE).

Keywords: STEM career, predictive analytics, educational data mining, intelligent tutoring system

1. INTRODUCTION

Nowadays, Science, Technology, Engineering, and Mathematics (STEM) fields are driving nations' economies. The demand for skilled personnel does continues to grow. Yet for several reasons, the number of open positions does not match the number of workers ready to take these positions. In fact, just in the United States, employment related to STEM occupations has grown considerably faster than for other non-STEM positions. Over the last decade, STEM job openings have increased by 24.4% compared to only a 4% increase in non-STEM jobs (Noonan, 2017). Nevertheless, trying to deal with the shortage in the STEM workforce is an integral part of educating and training the necessary highly skilled personnel.

Previous research has shown concern with student enrollment and retention in STEM fields when they went to college (Whalen and Shelley II, 2010). In fact, this can be explained by the students' personal choices made throughout their academic career. Moreover, students' motivation to enter into STEM careers can be tracked up to middle school (San Pedro et al., 2014). Many factors can influence student decisions. For example, financial situations have a big impact on students' choices regarding their future enrollment (Olenchak and Hbert, 2002). Furthermore, the educational level of parents is also considered an influential factor in students' academic enrollments and outcomes (Pascarella et al., 2004).

Aside from the external factors, stronger effects are more closely associated with academic success. For example, students' abilities in Math and Science subjects and the self-assessment of their academic aptitudes are known to have a strong influence on their decisions (Xueli, 2012; Xueli, 2013). Factors related to academic performance can be detected early, not only in high school but also in middle school, since it is during this period that students acquire the necessary skills to help them prepare for college. It is also at that time that students start to develop their career aspirations and objectives. And depending on their learning experience, they become more engaged in or, unfortunately, disengaged from an effective learning path at school leading to academic success (San Pedro et al., 2013; Balfanz, 2009).

Pursuing a STEM career is closely associated with graduating with a STEM major (Whalen and Shelley II, 2010). Thus, dealing with the growth of STEM positions depends also on increasing the numbers of students enrolling in STEM majors. Continuous efforts have been made to increase STEM enrollments. But encouraging students to continue their study in a STEM major must begin as early as middle school. The reason being that the required knowledge and skills for STEM fields are taught during the middle and high school years. Also, student decisions are still manageable during middle and high school, when it is possible to build confidence in their ability to continue in a STEM major (Xueli, 2013). Therefore, it has become crucial to recognize students who have difficulties and who are most likely to lose interest in STEM fields. Detecting these students makes it easier to provide them with adequate support that helps them overcome their problems and reignite their interest in STEM fields. Previous research proposed several detectors that can indicate which students are most likely to pursue STEM college majors. Factors such as family background and financial situation have an influence, but they can't be addressed easily (Pascarella et al., 2004; Olenchak and Hbert, 2002). Student skills and academic performance are very effective indicators, but since these detectors rely on student grades and in-field observations, it is hard to adjust the students' treatment by the time they have finished high school (San Pedro et al., 2014; Whalen and Shelley II, 2010).

However, thanks to the growing adoption of educational software within different academic institutions, educators are able to gather data about student usage. The recorded data is fine-grained and can describe every student's action within the system. The deployment of such software opens up many possibilities for extensive analysis. With a large amount of data at hand, researchers have been able to build predictive models capable of detecting students' affective states over a wide range of constructs such as gaming-the-system, boredom, carelessness, frustration, and off-task behaviors (Baker et al., 2004; Baker, 2007; San Pedro et al., 2011; Pardos et al., 2013; Sabourin et al., 2011). These detectors have been used, for example, in research that aimed at predicting learning outcomes (Pardos et al., 2013), college enrollment (San Pedro et al., 2013) and more importantly, predicting whether or not students will enroll in a STEM major in college (San Pedro et al., 2014).

Following the previous analysis, the ASSISTments team conducted a longitudinal study in

which they followed up on students that had used the ASSISTments¹ software during middle school between 2004 and 2006, then recorded their college enrollment and first job after college. The team opened the dataset for public use in 2017 and simultaneously organized a data mining competition where several research projects were conducted using that dataset to predict student engagement in STEM jobs. They also held a workshop² at the Educational Data Mining conference in 2018, where participants in the competition presented their work. To compare all models on the same basis, the evaluation criteria were calculated by using a linear combination of the area under the ROC curve (AUC) and the root mean squared error (RMSE).

AUC was the initial choice as the evaluation criterion, but using only one metric opens the risk of having models overfit to it. Therefore, having a linear combination of two metrics is a more robust way of evaluating models. Moreover, AUC and RMSE grasp different aspects of the model's performance. In fact, AUC is particularly applicable in the case of binary classification problems since it captures the ability of a model to distinguish between the predicted classes, while RMSE is more suitable for comparing two numbers. Using RMSE as a performance metric rewards models that are more certain when they are correct, and punishes models that are uncertain. The final score used to compare models is a linear aggregation of the AUC and RMSE, with the RMSE value being inverted (Thanaporn et al., 2018).

$$Score = AUC + (1 - RMSE)$$

Different approaches were taken to predict the students' enrollment in STEM-related jobs. For example, one team investigated three predictive methods: an ensemble classifier, clustering prior to classification, and a probabilistic classifier. They also compared the classifiers' performance in different scenarios: using the raw data, using the data after removing the outliers, and using the data after resampling. They used features describing students' affective states, knowledge, correctness, carelessness and gaming-the-system behaviors. The best score was achieved using the probabilistic classifier. To build the model, they split the dataset according to the dispersion of the data points relative to the selected features. The split was made using the Median Absolute Deviation (MAD) metric, and three splits were chosen. A split where data points had at least one attribute below two MAD of the corresponding attribute (called Data_05), another split in which data points had at least one attribute above two MAD of the corresponding attribute (called Data_95), and the last split contained data points having all attributes within a range of two MAD of the corresponding attribute (called Data_rest). The best model achieved an RMSE of 0.383 and an AUC of 0.836 using the Data_05 split. However, this split contained only 73 data points which are very few (Effat et al., 2018).

Another contribution to the workshop consisted of building an automatic machine learning system that can reformat the dataset, create new features, proceed to feature selection and build different models with the least possible human intervention. The authors used many summary statistics (e.g., mean, std, 9th percentile) to generate the student-level dataset from the click-stream data. Later, they investigated the interactions between features pair by pair, by measuring many values such as addition, multiplication, feature A divided by feature B, etc. And after a first round of feature elimination based on correlation, they used both forward and backward feature selection strategies. Finally, for the model selection, they used Penalized Logistic Regression and tried many different penalty functions. The best model achieved an AUC of 0.628 and an RMSE of 0.292 (Ruitao and Aixin, 2018).

¹<https://www.assistments.org>

²<https://sites.google.com/view/edm-longitudinal-workshop>

One more participating team used “state-of-the-art” Deep Knowledge Tracing (DKT) models (Piech et al., 2015) and an enhanced version called DKT+ (Chun-Kit and Dit-Yan, 2018). They measured student knowledge tracing using DKT+ and then combined it with other features from the dataset that they grouped together and called “Student Profile”. The Student Profile contained 11 features related to a student’s abilities, correctness, affective states and dis-engaged behavior. The authors used different machine learning methods to train their models. They tried Gradient Boosted Decision Tree, Linear Discriminant Analysis, Logistic Regression and Support Vector Machines. Then they tried a combination of features to train those models. The best result was achieved by training a Logistic Regression model using the combination of the student profile features with the DKT estimation of the knowledge state. The model attained an AUC of 0.694 and an RMSE of 0.414 (Chun-Kit et al., 2018).

In our submission to the workshop, we wanted to investigate if comparing students’ performances to their schoolmates could improve predictions of STEM careers. We used univariate feature selection to choose the best features. The dataset contained clickstream data from four different schools. To compare students to their peer schoolmates, we separated the students per school and applied the z-score function to all features of all students, school by school. We called this approach the “school-based approach”. We compared it to a baseline approach where we did not proceed to any school-based aggregation. Once the final dataset was generated, we used genetic programming to obtain the best machine learning technique and its hyper-parameters. We applied the optimization process to each approach independently. The best result was attained with the school-based approach, achieving an AUC of 0.601 and an RMSE of 0.546 (Makhlouf and Mine, 2018).

Even though our results in the workshop were not the best compared to other submissions, we wanted to push our analysis further. In fact, after finding that aggregating students within their school could lead to better predictions, we wanted to explore the dataset more and examine school-based aggregation from different angles. Therefore, in this paper, we aim to achieve two objectives, the first one of which is to improve the prediction models that identify which students will/won’t pursue a career in STEM fields compared to our submission in the workshop. For that purpose, we used the same dataset provided in the ASSISTments data challenge. Our second objective consisted of investigating different approaches for building the prediction models. The first technique, called “problem-based approach” consisted of measuring student features based on the problems they solved within the system. In the second technique, called “skill-based approach” we carried out the same procedure, but based on the skills that the students practice in the software. Meanwhile, we also proved that the school-based aggregation of the students’ features improved the performance of the model. In the process of building our models, we continued using genetic programming to find the best machine learning pipeline for each approach. The experimental results showed that the skill-based approach outperformed the problem-based approach on the same metrics used in the workshop, which are the AUC and the RMSE.

2. METHODOLOGY

2.1. ASSISTMENTS INTELLIGENT TUTORING SYSTEM

To accomplish our research, we used a large amount of data provided by the ASSISTments team. ASSISTments is a web-based Intelligent Tutoring System provided for free by Worcester Polytechnic Institute. It is used for middle school mathematics where teachers can use a predefined

set of contents or create their own. The system provides students with the right assistance while assessing their knowledge and reporting it back to the teachers. When students use the platform to work on problems assigned to them by their teachers, they receive immediate feedback as to whether their answers are correct or not. If they are correct, they can proceed to the next problem; if not, the system provides them with scaffolding exercises which are sub-components of the original problem to help students master the required skills. Once those skills have been acquired, the student is directed back to the original problem for another try. Then, after correctly answering this problem, they can move on to the next one. Questions in the ASSISTments platform are related to specific skills, which make tracking student performance more precise. At the same time, teachers get full reports on student activities and their performance. The reports help the teachers to identify students' common mistakes and problems and determine who struggled to solve which problems, which can all be done even before they meet their students in the classroom (Olenchak and Hbert, 2002). The ASSISTments team gave us access to the data gathered during a longitudinal study over a decade long.

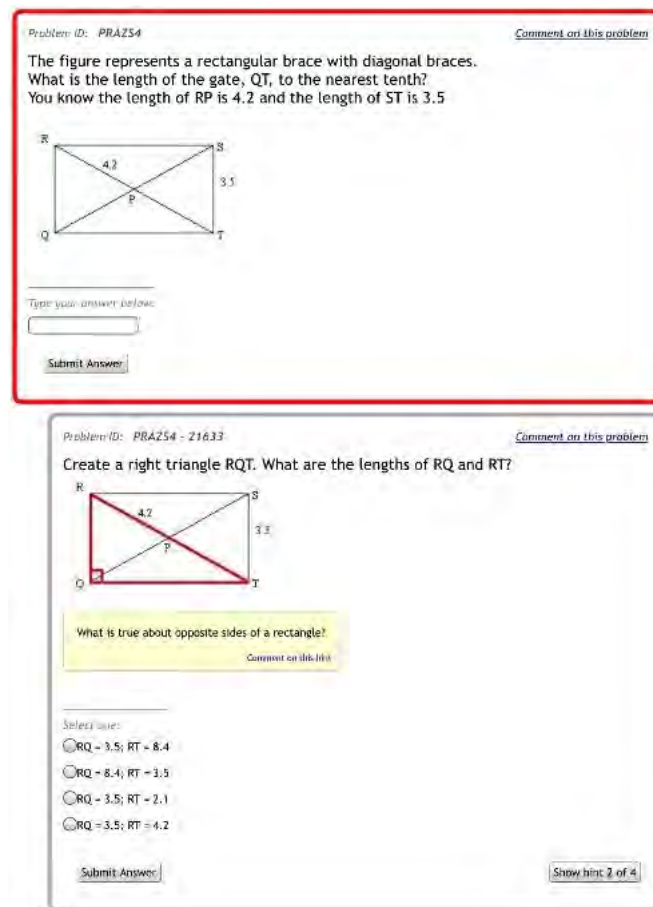


Figure 1: Example of an ASSISTments problem where the student answered incorrectly and is led to solve a scaffolding problem.⁴

⁴<https://www.assistments.org/>

2.2. DATA ACQUISITION

The dataset consisted of click-stream log files describing students' interactions with the ASSISTments software. The dataset contained students' actions spanning from 2004 to 2006. We counted 942,816 actions stored in the log files related to different types of student interactions, such as answering a question or requesting help. Each row identified an action within the system, and each action was described by a set of recorded information. Those actions were carried out by a group of 1,709 students enrolled in four different schools that used ASSISTments. The dataset also contained several other kinds of information related to these students, such as their MCAS (Massachusetts Comprehensive Assessment System) score, their anonymized ID, and whether their first job out of college was STEM-related or not. This dataset contained no less than 3,765 problems related to a complete set of 102 skills. Some data was duplicated and, at the same time, some students did not have their first job out of college registered in the dataset. So after proceeding to an initial cleanup of the dataset, we ended up having an overall total of 316,947 actions completed by 591 unique students. The number of problems the students practiced dropped to 3,162, same as the skills involved as we kept 93 out of the 102 unique skills. We also had access to the students' STEM job enrollment files, which ultimately was the predicted variable.

2.3. FEATURE EXPLORATION

The dataset consisted of a list of actions recorded when students used the ASSISTments system. It contained 82 features. These features described different aspects of the usage of the ASSISTments system. Some features were related to the context of the usage, such as the school ID and the academic year. Features such as the Student ID, the Inferred Gender and the MCAS test score were related to the student who used the system. Another subset of features was associated with the action performed. In this subset of features, we obtained time-related information, such as the time taken to answer the question, or the detected long pauses after a correct answer. We also had access to features relevant to the correctness of the answers given by students and features that described the type of the answer, whether it was a fill-in or chosen answer (e.g., Multiple choice). The dataset also described some functionalities of the ASSISTments system. In fact, information about the hint and help request usage was registered. Moreover, ASSISTments provided problems at different levels: original problems and scaffolding problems. In this dataset, many features related to the original or scaffolding problems were available. Finally, there is a subset of features related to models assessing students' knowledge, behaviors, and affective states such as boredom, engaged concentration, confusion, frustration, off-task and gaming-the-system behaviors.

2.4. DISCOVERY WITH MODELS

Some of the dataset features were generated using models based on previous research on student behaviors, affective state or latent knowledge. In fact, tracking student knowledge is an active field of research that has been characterized by the emergence of Bayesian Knowledge Tracing (BKT) as one of the most-used models (Corbett and Anderson, 1995; Joel and Kurt, 1995; Reye, 2004; Pavlik et al., 2009). Indeed, BKT is able to estimate a student's latent knowledge of a specific skill given previous observable performances. It runs continuously and, for each student's attempt, it measures the probability that the student knows the skill involved.

Table 1: Features chosen by Univariate Feature Selection.

Feature Name	F-Score	P-Value
AveKnow	16.88	0.000045 ($p < 0.001$)
AveCarelessness	18.20	0.000023 ($p < 0.001$)
hintCount	11.11	0.000908 ($p < 0.001$)
hintTotal	10.05	0.001601 ($p < 0.05$)
attemptCount	7.19	0.007514 ($p < 0.05$)
frPast5HelpRequest	8.58	0.003520 ($p < 0.05$)
frPast8HelpRequest	5.86	0.015705 ($p < 0.05$)
past8BottomOut	7.18	0.007538 ($p < 0.05$)
timeSinceSkill	10.54	0.001234 ($p < 0.05$)
totalTimeByPercentCorrectForSkill	5.37	0.020812 ($p < 0.05$)
res_gaming	4.11	0.042891 ($p < 0.05$)
Ln-1	16.10	0.000068 ($p < 0.001$)
Ln	16.89	0.000045 ($p < 0.001$)
correct	16.56	0.000053 ($p < 0.001$)
original	8.95	0.002884 ($p < 0.05$)
hint	14.12	0.000188 ($p < 0.001$)
bottomHint	10.82	0.001062 ($p < 0.05$)
frIsHelpRequestScaffolding	5.97	0.014831 ($p < 0.05$)
timeGreater10SecAndNextActionRight	16.46	0.000056 ($p < 0.001$)
manywrong	15.97	0.000072 ($p < 0.001$)

Along with predicting student knowledge, different detectors were able to estimate students' affective state and disengaged behavior. Research conducted by (Pardos et al., 2013) identified different affective states such as boredom, engaged concentration, confusion, and frustration. They also detected students' disengagement, such as the off-task and gaming-the-system behaviors. They performed field observations to track the students' behaviors while using the ASSISTments software, and then synchronized these observations with the log data. They then created automated individual models for each affective state and disengagement behavior. All these detectors were used in ASSISTments and for each action in the dataset we had records of the students' affective state, disengagement as well as their knowledge estimation.

2.5. INITIAL FEATURE SELECTION

The first thing we notice about the dataset is that each row described a single action, while the predictions were being made for each student. Thus, to predict which students would pursue a STEM career, we had to change the granularity of our data from the action level to the student level. The first step in this research was to identify which features were useful and that we could investigate in our different approaches. So we began by transforming the dataset to the student level. In fact, for each student, we took the average of the numerical features and the frequency of 1 (True) in binary features. Then we proceeded to Univariate Feature Selection using the ANOVA F-Score. We only kept predictors that had a statistically significant relation to the predicted variable: $p_value < 0.05$. The resulting selected features are shown in Table 1.

Table 2: Feature set to be used.

Feature Name	Meaning
correct	Answer is correct
timeTaken	Time spent on the current step
bottomHint	Bottom-out hint is used
frIsHelpRequestScaffolding	First response is a help request Scaffolding
timeSinceSkill	Time since the current Knowledge Component (KC) was last seen.
hint	Action is a hint request
attemptCount	Total problems attempted in the tutor so far.
manywrong	Many wrong answers given
Ln	Bayesian Knowledge Tracing's knowledge estimate at the time step (Corbett and Anderson, 1995)
res_gaming	Rescaled of the confidence of the student affective state's prediction: gaming-the-system
frPast5HelpRequest	Number of last 5 First responses that included a help request
totalTimeByPercentCorrect Forskill	Total time spent on this KC across all problems divided by percent correct for the same KC
timeGreater10SecAnd NextActionRight	Long pause after correct answer

Some of the features shown in Table 1 were highly correlated. For example, the “hintCount” and “hintTotal” features were notably correlated, as were the “frPast5HelpRequest” and the “frPast8HelpRequest” features. “AveKnow”, “Ln” and “Ln-1” were measured using the BKT formulas and thus expressed the same aspect and their correlation was high as well. Also, we were going to use the “original” feature to measure and generate new features, therefore we did not use it as a predictor. Accordingly, we reduced the feature set and we added “timeTaken” as we were curious to see how it would perform in our approaches. The final set of features we used is presented in Table 2

2.6. MODEL BUILDING APPROACHES

Once we selected the features to investigate, we proceeded to build different prediction models. For each approach, we followed different feature transformation and selection procedures. In the baseline model, we continued to use the normal average for numerical features and the frequency of 1 in binary features, for every student. The first approach consisted of measuring the students' performance on the selected features based on the problems they solved. And in the second approach we evaluated students' performance based on the skills they practiced. We wanted to investigate whether using the information about the problems or the skills could improve our predictions. In fact, problems might involve more than one skill in order to find the correct answer. Therefore, skills are more fine-grained in analyzing students' aptitudes. Moreover, skills represent a single aptitude or knowledge about a specific concept such as addition, multiplication, measuring the surface of a square, etc. So far in the dataset we had records of

3,162 unique problems related to 93 unique skills.

2.6.1. Problem-based Approach

In this approach, we did not take the simple average values (for numerical features) and the frequency of 1 (for binary features) across all actions. Instead, for each student, we took the average/frequency problem by problem. The features used are those selected in Table 2, that we called `{selected_features}`. In the process of measuring the average/frequency, we distinguished between actions done in an original problem and actions done in a non-original problem. In fact, in ASSISTments, there are two types of problems. Original and scaffolding. While using the ASSISTments software, students are asked to solve original problems. If they fail to answer correctly, they are redirected to scaffolding problems which are sub-components of the original problem, where the aim is to help students learn the required skills. We considered the difference between the two types of problems and we added the suffix `_o` and `_no` accordingly. Thus, we had `{selected_features}_o` which are the measured features when the problem is an original problem (original = 1). And `{selected_features}_no` are measured when the problem is not original (original = 0). Thus we generated 13×2 features. By the end of the calculation, we had a dataset in which each row represented a student's performance in a specific problem. Finally, to change the granularity to the student level where each row represents only one student, we took the average values for all 26 generated features. Figure 2 shows how the transformation is done.

2.6.2. Skill-based Approach

In the skill-based approach, we followed the same steps as in the problem-based approach to transform the dataset. The only difference was that we used the skill, not the problem, when taking the average/frequency of values. In fact, we measured students' performances skill by skill, while differentiating between actions done in original problems and actions done in non-original problems.

2.7. MODEL FEATURE SELECTION

After generating the problem-based and skill-based datasets, we proceeded to another round of feature selection. For each approach, we separately used a combination of stepwise forward feature selection and backward feature elimination. Then, we took the union of the feature sets that resulted from each feature selection method.

For the problem-based approach, the selection gave us the following features listed in Table 3. The selected features set was quite small and contained predictors related to hint usage in original and non-original problems. Likewise, the behavior of gaming-the-system in non-original problems was detected as a strong predictor. The correctness, the longtime pauses after a correct answer and the number of the five last first responses that included a help request, all in non-original problems, were also selected as strong features. Finally, the average time since the skill has been seen across original problems was the last strong predictor in the features set.

We ran the same selection process in the skill-based approach and we found different features. Table 4 shows the list of selected features for the skill-based approach. The selected feature set for the skill-based approach was larger than that for the problem-based approach. Again, we found the behavior of gaming-the-system in non-original problems to be a strong predictor. The average BKT estimate and the average carelessness both in the original problems

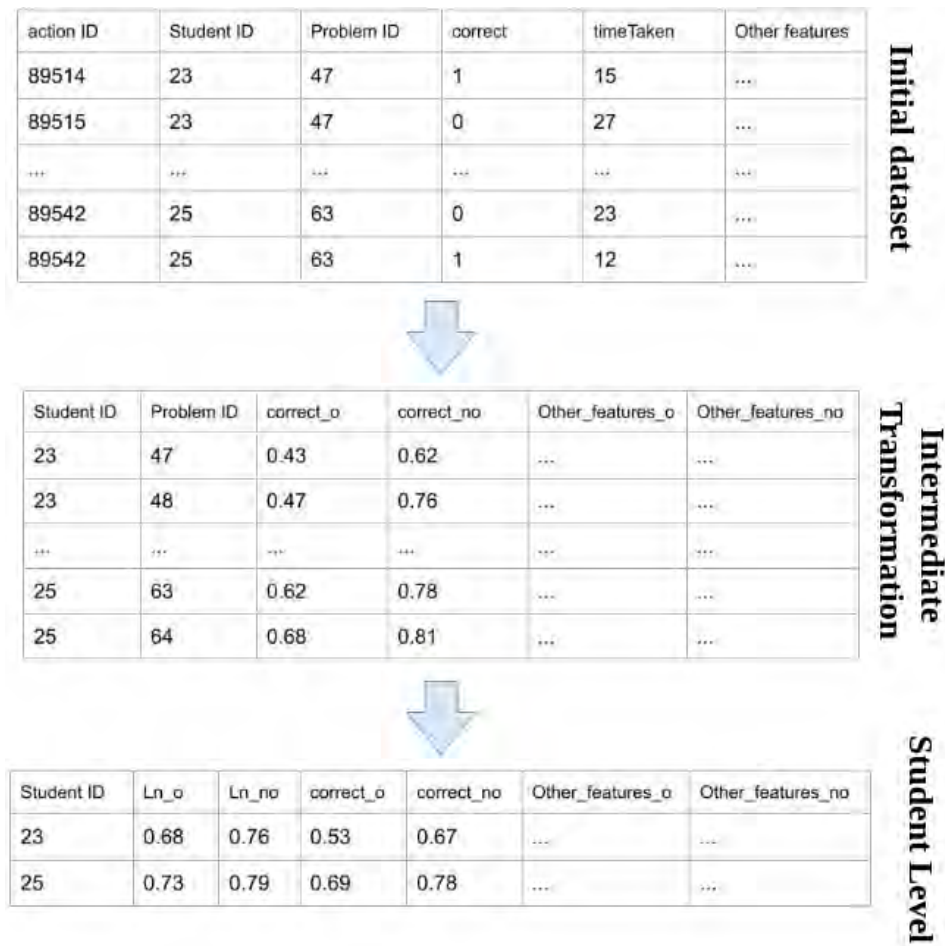


Figure 2: Feature transformation in the problem-based approach.

were selected this time. Surprisingly enough, they were not selected in the problem-based approach. We also found that the average time since the skill was seen in non-original problems was a good predictor, as well as the average correctness in both original and non-original problems. Likewise, the hint and the bottom hint usage in original problems were detected as strong predictors. Similarly to the problem-based approach, the longtime pauses after a correct answer and the number of the five last first responses that included a help request, both in non-original problems, were also selected as strong features.

2.8. AGGREGATING DATA BASED ON SCHOOLS

Along with these two different approaches in constructing the features, we wanted to investigate whether comparing students' performances with their peer schoolmates could improve the predictions and lead to better performances. Accordingly, for each approach, we built two models separately. In the first model, we did not make any change in the data, while in the second model we compared students with their peer schoolmates by measuring the z-score of each feature for all students, school by school. Figure 3 explains how the school-based aggregation was done.

Table 3: Final feature set to be used in the problem-based approach.

Features measured in original problems	Features measured in non-original problems
avg_hint_per_problem_o	frPast5HelpRequest_per_problem_no
timeSinceSkill_per_problem_o	res_gaming_per_problem_no
	avg_correct_per_problem_no
	avg_hint_per_problem_no
	avg_timeGreater10SecAndNext ActionRight_per_problem_no

Table 4: Final feature set to be used in the skill-based approach.

Features measured in original problems	Features measured in non-original problems
Ln_per_skill_o	res_gaming_per_skill_no
AveCarelessness_per_skill_o	timeSinceSkill_per_skill_no
avg_correct_per_skill_o	frPast5HelpRequest_per_skill_no
avg_hint_per_skill_o	avg_correct_per_skill_no
avg_bottomHint_per_skill_o	avg_manywrong_per_skill_no
	avg_timeGreater10SecAndNext ActionRight_per_skill_no

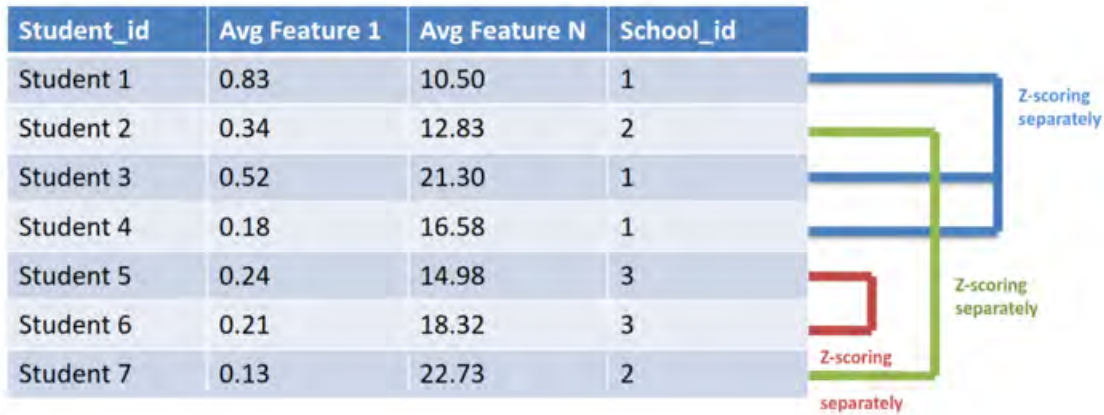


Figure 3: Example of z-scoring within schools.

2.9. OPTIMIZATION AND GENETIC PROGRAMMING

Since we compared different approaches independently, we wanted to find the most adequate machine learning method with the best hyper-parameters for each approach. To this end, we used genetic programming as our tool for searching.

Briefly, genetic programming is a technique derived from genetic algorithms in which instructions are encoded into a population of genes. The goal is to evolve this population using genetic algorithm operators to constantly update the population until a predefined condition is met. The most common ways of updating the population are to use two famous genetic op-

erators called crossover and mutation. Crossover is used to diversify the search in the research space by taking some parts of the parents and mixing them into the offspring. On the other hand, mutation is the process of updating only some part of an individual and it is used to maintain the actual diversity, in other words, intensify the search in a certain area of the research space. The population is evolving from one generation to another while keeping the individuals that are fittest in regard to one or many objectives. When using genetic programming for machine learning optimization, we used the pipeline score as the objective function; the pipeline accuracy score is an example of an objective function which must be maximized.

In our case, we used genetic programming to search through a multitude of machine learning techniques and their respective hyper-parameters to determine which combination gives the best results. In fact, genetic programming can be compared to a heuristics-based grid search. Instead of searching every possible combination of parameters, it only investigates combinations that are more likely to improve the end result. Thanks to the concept of evolving the population individuals, it avoids searching in areas that are less likely to give good results and thus makes the process faster by reducing the iteration/instruction count. When using genetic programming to optimize the machine learning pipelines, we encoded the machine learning techniques and their hyper-parameters in an individual. The optimization process then evolves a population of individuals (therefore a population of machine learning pipelines) and only keeps individuals with the best scores. By means of the crossover and mutation operators, it explores heuristically the research space (the possible values of the hyper-parameters) and tries to find the best combination. However, to use genetic programming, there are several hyper-parameters that we need to initialize wisely. Even if the use of genetic programming is faster than a grid search, the quality of the result depends on how much time is given to the process. Therefore, it still requires a lot of time. In our work, we used a genetic programming python library called TPOT (Olson et al., 2016).

Table 5: Genetic Programming Hyper-parameters

Generation count	Population size	Offspring size	Scoring	Mutation rate	Crossover rate	Internal Cross Validation
200	150	150	AUC	0.9	0.1	5-fold

Table 5 explores the principal hyper-parameters that we had to initialize to run a genetic programming experiment. The Generation count is the number of iterations of the whole optimization process. A bigger number has a greater chance of leading to better results but also takes more time to finish. We can also fix a maximum amount of time instead of the Generation count. The Population size is the number of individuals involved in the optimization process. The Offspring size is the number of individuals that are supposed to be generated from the previous population using the genetic algorithm operators. For each generation (iteration), the parents and offspring compete to survive and be part of the next generation's population. When the individuals compete against each other, we only keep the fittest ones, meaning the individuals with the best score.

To determine how we choose which individual is the fittest, we must declare the scoring method in the hyper-parameters. We used the AUC as our scoring method. That means we only keep the individuals (thus the pipelines) that have the highest AUC values. The Mutation and

Crossover rates are the probabilities of having respectively a Mutation or a Crossover operation to evolve one or more individuals. We set them to be a 90% chance of having a mutation against a 10% chance of having a crossover operation. TPOT evaluates individuals using an internal cross-validation, thus we can declare how many folds we are willing to use. We set this parameter to five-fold cross-validation.

Since we were comparing different approaches, we independently ran different optimization processes, but with the same hyper-parameters. For the problem-based approach, we ran two optimization processes, one for the normal problem-based approach, and another one where we used the school-aggregated z-scores. The same process was done with the skill-based approach as we ran two optimization processes, one for the normal skill-based approach and another for the z-scored dataset based on schools. Overall, we had five optimization processes: one for the baseline where we only took the average value of each action for every student, and two for each approach.

2.10. EXPERIMENTAL RESULTS

Before conducting the experiments, we split the data into training data and testing data to validate the results, after finding the best pipeline for each approach. Due to the unbalanced proportions of the label (isSTEM), the split was stratified on the label so the proportions between STEM and non-STEM were preserved in the test set. Table 6 shows which methods were chosen for each approach. For the baseline model, in which we just took the average values across all actions for each student, the optimization process generated a pipeline having Randomized Decision Trees as the prediction method. In the normal problem-based approach, the resulting pipeline contained a stacking technique using a Naive Bayes classifier combined with Logistic Regression. For the problem-based approach with school aggregation, we found that the Extreme Gradient Boosting algorithm had the best results. Similarly, for the normal skill-based approach, a Gradient Boosting Classifier was chosen. Finally, for the skill-based approach with school aggregation the best pipeline used a Decision Trees Classifier.

Table 6: Results of the optimization process.

Approach	Best pipeline
Baseline	Randomized Decision Trees
Problem-based	Logistic Regression
Problem-based, school-aggregated	XGBClassifier
Skill-based	Gradient Boosting Classifier
Skill-based, school-aggregated	Decision Trees

Once the optimization process was complete, we proceeded to train all chosen models using a five-fold cross-validation. The predicted variable consisted of a binary value related to whether the student pursued a STEM-related career after college. The distribution of the STEM-related career values was not balanced, as the dataset contained more students that enrolled in STEM-related careers than students who did not. To deal with this issue, we applied the cross validation in a stratified way, which implies respecting the proportions of the STEM career outcome in each fold. When we had to apply the school-based aggregation, we also had to deal with unbalanced proportions in terms of school data. In fact, the number of students from each school varied.

Therefore, to consider this aspect of the dataset, we applied the stratified cross-validation in regard to the STEM career outcome and also the school of the student when we trained our models involving the school-based aggregation.

To evaluate the quality of our models, we measured the same metrics that were used during the ASSISTments data mining challenge⁵ and workshop. The metrics used are AUC and RMSE. The combined score was calculated as follows:

$$Score = (1 - RMSE) + AUC$$

As explained in the preface of the workshop (Thanaporn et al., 2018), each metric measures different aspects of the performance of our models. Using two performance metrics led to more robust evaluations.

Table 7: Cross-validated scores of all approaches.

Model	AUC	RMSE	Combined Score
Baseline	0.521	0.466	1.055
Problem-based	0.629	0.482	1.146
Problem-based, school-aggregated	0.610	0.474	1.135
Skill-based	0.621	0.461	1.160
Skill-based, school-aggregated	0.682	0.513	1.169

Table 7 shows the scores of the cross-validated models. The best scores for each measure are shown in boldface. The baseline model had the worst results in AUC and in the combined score, suggesting that simply taking the average values across all students' actions was not an effective concept. The problem-based approach had better results in AUC, attaining 0.629, but a worse RMSE of 0.482. Its combined score reached 1.146 which is better than the baseline score. Against our expectations, the aggregation of the features' values within schools did not improve the predictions in the problem-based approach. In fact, the school-aggregated model had a lower AUC, but better RMSE. However, the combined score was worse than the normal problem-based approach. Compared to the problem-based approach, the skill-based approach had a significant improvement in terms of RMSE, dropping to 0.461, which is the best RMSE score among all the models. With a combined score of 1.160, the normal skill-based approach had a better result than the normal problem-based approach and the school aggregated problem-based approach. The best AUC score was achieved by the skill-based approach with school aggregation, which showed a significant improvement, attaining 0.682 in AUC. However, its RMSE was the highest among all the models, reaching 0.513. Despite the high RMSE, this model had the best combined score of all the models considered.

3. DISCUSSION AND CONCLUSION

In this paper, we aim to achieve long-term predictions regarding which students would pursue a career in STEM-related fields. To achieve this objective, we made a comparative study of two approaches to building the prediction models. These two approaches were also compared to a

⁵<https://sites.google.com/view/assistmentsdatamining/data-mining-competition-2017>

baseline, in which we took the average value of numerical features and the frequency of 1 (True) in binary features across the actions of each student. This baseline model did not have good results compared to the other models. The problem-based approach, whether it was school-aggregated or not, improved the AUC score, but also worsened the RMSE, compared to the baseline scores. On the other hand, the skill-based approach outperformed the problem-based approach and the baseline. The best RMSE score was achieved by the normal skill-based model, and the best AUC was attained by the school-aggregated skill-based approach. Since we wanted to compare the different approaches and not just compare different machine learning methods, we gave each approach its best try using genetic programming. In that way, we searched for the best machine learning pipeline which is appropriate to each approach. This resulted in different methods being applied to predict students' STEM careers.

In a direct confrontation between the skill-based and the problem-based approaches, we found that the normal skill-based model outperformed the normal problem-based model, and the same was true for school-aggregation. These results can be explained by the fact that building features around skills gives stronger predictors than the problem-based model. And that's because skills are more fine-grained than problems and they better encapsulate the ability of students to master the subject. Moreover, problems can be related to one or many skills at the same time and that's probably why they are not as effective as the skills in terms of describing the failing students and the successful ones. Since problems can be related to different skills, when students master a skill, they are more likely to be successful applying it in different problems that involve that skill. However, the reverse is not always true, as you can't generalize from the problem viewpoint to the skill viewpoint. In other words, mastering one single problem does not mean mastering all the skills involved in that problem. Moreover, when we investigated the effect of comparing students' performances with their peer schoolmates, we found that such aggregation improved our models. Our aim was not to compare which school was the best or had the best students. Our objective was to verify whether students that had the best performance relative to their schoolmates were more likely to enroll in STEM-related fields.

The feature selection schema was also effective in picking the right features that were strong predictors. We took the union of the features picked by stepwise forward feature selection and backward feature elimination. It is also worth noting that the feature creation process was effective. For the same feature set, we separated the measurement of its values based on whether the action happened in an original or non-original problem. That helped us investigate more thoroughly the difference between students. Actions happening in original problems and scaffolding problems did not have the same significance, as original problems are the principal exercises necessary to complete the task, while scaffolding problems are meant to explain the skills gradually. When we look at the features measured in original problems and features measured in non-original problems, we noticed they were different. This suggests that separating the measures in regard to the type of the problem can extract more meaningful information related to students' usage. For example, in the skill-based approach, carelessness was selected as a strong feature when it was measured in original problems but not in non-original problems.

In our first contribution to the ASSISTments data challenge workshop, we found that the school-based aggregation performed better than a baseline model. In this paper, we not only confirmed the same findings about school-based aggregation, but we also improved our prediction models. However, we did not achieve the best results in terms of RMSE or AUC, compared to the other workshop submissions. Nonetheless, we focused on exploring several aspects of the dataset that resulted in very interesting findings that might help the tailoring of better intelligent

tutoring systems. Firstly, we discovered that the performance of the students regarding skills is more significant than their performance in problems. That means that students' performances in the designated skills are stronger predictors than their performance in problems even when they involve the same skill. Secondly, we found that distinguishing between original and scaffolding problems when investigating students' performances, behaviors and affective states grasps more information. Finally, comparing students to their peer schoolmates has a significant impact on the prediction models.

All these findings might improve the design of digital learning and assessment platforms. In fact, the scaffolding concept is not only useful because it helps students learn the required skills, but also because it can be used for more accurate predictions and reporting of students' behaviors and performances. Furthermore, skills should get a greater focus of attention since performances with regard to the required skills have better prediction ability than performances in problems that might involve those skills.

Overall, the ASSISTments dataset was very rich, and exploring it was instructive. The usage of genetic programming helped us automate the search for the best machine learning method with the best hyper-parameters. However, it is still slow even if it is faster than a grid search method. One solution to overcoming this issue is to limit the research space by specifying the hyper-parameter value range and how many different machine learning methods the optimization process is expected to search. In our case, we did not use the automatic dimensionality reduction mechanism (e.g., PCA) not only to reduce the execution time but also to keep our feature exploration and generation methods untouched. Finally, we still have different ideas that we want to explore within the dataset. In fact, several features can be used and investigated. It will be particularly interesting to further explore what kind of influence the type of the problem can have on students. For example, answering a Fill-In question is different from picking the correct answer from the list of potential correct answers.

4. ACKNOWLEDGMENTS

We would like to thank the ASSISTments team for giving us the opportunity to work with their dataset.

This work is partially supported by JSPS KAKENHI No.JP16H02926, JP17H01843 and JP18K18656.

REFERENCES

- BAKER, R. S. 2007. Modeling and understanding students' off-task behavior in intelligent tutoring systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '07. ACM, New York, NY, USA, 1059–1068.
- BAKER, R. S., CORBETT, A. T., AND KOEDINGER, K. R. 2004. Detecting student misuse of intelligent tutoring systems. In *Intelligent Tutoring Systems*. Springer Berlin Heidelberg, Berlin, Heidelberg, 531–540.
- BALFANZ, R. 2009. *Putting Middle Grades Students on the Graduation Path A Policy and Practice Brief*. National Middle School Association, Westerville, OH.
- CHUN-KIT, Y. AND DIT-YAN, Y. 2018. Addressing two problems in deep knowledge tracing via prediction-consistent regularization. In *Proceedings of the Fifth Annual ACM Conference on Learning at Scale*. ACM, New York, NY, USA, 5:1–5:10.

- CHUN-KIT, Y., ZIZHENG, L., KAI, Y., AND DIT-YAN, Y. 2018. Incorporating features learned by an enhanced deep knowledge tracing model for stem/non-stem job prediction. <https://drive.google.com/file/d/1OSKlK51XUHFGEPKfsbUhB3BRHvU-vMkk/view>. Workshop on the Scientific Findings from the ASSISTments Longitudinal Data Competition in the 11th International Conference on Educational Data Mining, Buffalo, NY, USA.
- CORBETT, A. T. AND ANDERSON, J. R. 1995. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction* 4, 4 (Dec), 253–278.
- EFFAT, F., MAAZ, S. K., WENJIA, C., AND COLLIN, F. L. 2018. Predicting post-college stem career entrance from middle school clickstream data. <https://drive.google.com/file/d/1dLOYDtQs11C1kC1KdbPrPvEr8eZ2kjik/view>. Workshop on the Scientific Findings from the ASSISTments Longitudinal Data Competition in the 11th International Conference on Educational Data Mining, Buffalo, NY, USA.
- JOEL, M. AND KURT, V. 1995. Student assessment using bayesian nets. *International Journal of Human Computer Studies* 42, 6 (6), 575–591.
- MAKHLOUF, J. AND MINE, T. 2018. Investigating how school-aggregated data can influence in predicting stem careers from student usage of an intelligent tutoring system. https://drive.google.com/file/d/1XS1spxOdbFkFfRsJTao-r0_hTTe6mkTO/view. Workshop on the Scientific Findings from the ASSISTments Longitudinal Data Competition in the 11th International Conference on Educational Data Mining, Buffalo, NY, USA.
- NOONAN, R. 2017. Stem jobs: 2017 update. Office of the Chief Economist, Economics and Statistics Administration, U.S. Department of Commerce(ESA Issue Brief 02-17).
- OLENCHAK, F. AND HBERT, T. 2002. Endangered academic talent: Lessons learned from gifted first-generation college males. *Journal of College Student Development* 43, 2 (03), 195–212.
- OLSON, R. S., BARTLEY, N., URBANOWICZ, R. J., AND MOORE, J. H. 2016. Evaluation of a tree-based pipeline optimization tool for automating data science. In *Proceedings of the Genetic and Evolutionary Computation Conference 2016*. ACM, New York, NY, USA, 485–492.
- PARDOS, Z. A., BAKER, R. S., SAN PEDRO, M. O., GOWDA, S. M., AND GOWDA, S. M. 2013. Affective states and state tests: Investigating how affect throughout the school year predicts end of year learning outcomes. In *Proceedings of the Third International Conference on Learning Analytics and Knowledge*. ACM, New York, NY, USA, 117–124.
- PASCARELLA, E. T., PIERSON, C. T., WOLNIAK, G. C., AND TERENCEZINI, P. T. 2004. First-generation college students: Additional evidence on college experiences and outcomes. *Journal of Higher Education* 75, 3 (5), 249–284.
- PAVLIK, P. I., CEN, H., AND KOEDINGER, K. R. 2009. 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*. IOS Press, Amsterdam, The Netherlands, 531–538.
- PIECH, C., BASSEN, J., HUANG, J., GANGULI, S., SAHAMI, M., GUIBAS, L. J., AND SOHL-DICKSTEIN, J. 2015. Deep knowledge tracing. In *Advances in Neural Information Processing Systems* 28, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. Curran Associates, Inc., 505–513.
- REYE, J. 2004. Student modelling based on belief networks. *International Journal of Artificial Intelligence in Education* 14, 1 (jan), 63–96.
- RUITAO, L. AND AIXIN, T. 2018. Stem career prediction using an automatic machine learning approach. https://drive.google.com/file/d/1ps_LX8mDSdnyY79FqCcZlb8igqgX1JPA/

- [view](#). Workshop on the Scientific Findings from the ASSISTments Longitudinal Data Competition in the 11th International Conference on Educational Data Mining, Buffalo, NY, USA.
- SABOURIN, J., MOTT, B., AND LESTER, J. C. 2011. Modeling learner affect with theoretically grounded dynamic bayesian networks. In *Affective Computing and Intelligent Interaction*. Springer Berlin Heidelberg, Berlin, Heidelberg, 286–295.
- SAN PEDRO, M. O., BAKER, R. S., BOWERS, A., AND HEFFERNAN, N. T. 2013. Predicting college enrollment from student interaction with an intelligent tutoring system in middle school. In *Proceedings of the 6th International Conference on Educational Data Mining*. 177–184.
- SAN PEDRO, M. O., BAKER, R. S., AND RODRIGO, M. M. T. 2011. Detecting carelessness through contextual estimation of slip probabilities among students using an intelligent tutor for mathematics. In *Proceedings of the 15th International Conference on Artificial Intelligence in Education*. Springer Berlin Heidelberg, Berlin, Heidelberg, 304–311.
- SAN PEDRO, M. O., OCUMPAUGH, J., BAKER, R. S., AND HEFFERNAN, N. T. 2014. Predicting stem and non-stem college major enrollment from middle school interaction with mathematics educational software. In *Proceedings of the 7th International Conference on Educational Data Mining*. 276–279.
- THANAPORN, P., HEFFERNAN, N. T., AND BAKER, R. S. 2018. Assistments longitudinal data mining competition 2017: A preface. <https://drive.google.com/file/d/1Dt6xhFHTqpqvJp9rOcfc2X-3l2l9gG1J/view>. Workshop on the Scientific Findings from the ASSISTments Longitudinal Data Competition in the 11th International Conference on Educational Data Mining, Buffalo, NY, USA.
- WHALEN, D. F. AND SHELLEY II, M. C. 2010. Academic success for stem and non-stem. *Journal of STEM Education: Innovations and Research* 11, 1, 45 – 60.
- XUELI, W. 2012. Modeling student choice of stem fields of study: Testing a conceptual framework of motivation, high school learning, and postsecondary context of support. WISCAPE Working Paper. Wisconsin Center for the Advancement of Postsecondary Education.
- XUELI, W. 2013. Why students choose stem majors: Motivation, high school learning, and postsecondary context of support. *American Educational Research Journal* 50, 5, 1081–1121.