Investigating How School-Aggregated Data Can Influence in Predicting STEM Careers from Student Usage of an Intelligent Tutoring System

Jihed Makhlouf[1], Tsunenori Mine[1]
[1] Kyushu University, 744 Motooka Nishi-ku, Fukuoka, Japan
makhlouf_jihed@yahoo.fr, mine@ait.kyushu-u.ac.jp

ABSTRACT
For several years the Science, Technology, Engineering and Maths (STEM) fields have been growing and continue to grow rapidly. Recent reports in the United States show that growth in STEM employment has increased to 1.7% per year, while non-STEM positions grew by 0.6% per year during the same period of time [8]. Efforts have been made to encourage people to pursue a career in STEM fields. Nevertheless, the provision of the skills students require to be capable of pursuing a STEM career starts earlier than college. In fact, students need to be trained from middle school, as it is then that students make their choices [14]. Thanks to the increasing adoption of educational software, detecting students' problems with maths and science subjects has become more efficient. And more recently, models using data gathered from an Intelligent Tutoring System have become capable of predicting STEM or non-STEM college major enrollment [14]. In this paper, we focus on predicting STEM field career choice using longitudinal data from an Intelligent Tutoring System. We investigate if comparing student data to that of their peer school-mates can improve the prediction models. We find that aggregating student levels within the same schools improves prediction models and leads to better AUC scores (0.601) compared to models using student data without this school-based transformation.

Keywords
STEM Career choice, Educational Data Mining, Predictive Analytics

1. INTRODUCTION
Science, Technology, Engineering, and Mathematics (STEM) fields are regarded worldwide as the building blocs for a nation's economy. 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 a lot faster than for other non-STEM occupations. Over the last decade, STEM occupations have increased by 24.4% compared to "only" a 4% increase in non-STEM occupations [8]. However, STEM positions require the candidates to have appropriate STEM skills that are acquired in the course of completing a STEM degree or from advanced technical training. Thus, educating pupils in STEM majors and encouraging them to continue their studies are important steps toward filling the need for a STEM workforce which is constantly and rapidly increasing.

Previous research showed concern about student enrollment and retention in STEM fields when they get to college [22]. In fact, this can be explained by the individual choices made during one's academic career, more specifically during high school [14]. Many factors can influence student decisions. For instance, the financial situation of students plays a big role in their future enrollment [9]. Furthermore, quite often, students are influenced by their parents, whether directly or indirectly. That's why the education of parents has been investigated as a factor influencing students' higher education choices and outcomes [12].

External factors can impact personal choices, but stronger effects are more associated with academic success, proficiency in Maths and Science subjects and student's self-assessment of their level [19, 20]. These kinds of factors can be detected early, not only in high school but also in middle school. It is during this period that students acquire the necessary skills to help them prepare for college. Depending on their learning experience, students start to build their self-beliefs, objectives and career aspirations. Throughout their learning journey in middle school, they find themselves more engaged in or disengaged from the learning process at school, either starting to think about academic success and improving grades or becoming more disengaged and deviating from the track of academic success [18, 4].

Since integrating into a STEM career is closely related to graduating with a STEM major [22], the difficulty of responding to the growth of STEM positions is highly sensitive to the numbers of students enrolling in STEM majors. Continuous efforts have been made to increase STEM enrollments. But the promotion of the pursuit of a STEM major has to begin as early as middle school for two reasons. Firstly, the foundation of knowledge required in STEM fields is acquired during the years in middle school and high school. Secondly, very often, student decisions are still easily manageable during middle school, when it is possible to build their confidence in being able to pursue a STEM major [20]. That's why it is necessary to distinguish students who have difficulties and who are most likely to loose interest in STEM fields. These students need more support in order to help them overcome their problems and reignite their interest in STEM fields. Several detectors can indicate which students are most likely to pursue STEM college majors. Factors like family background and financial situation [12, 9] have an influence but they are not easily remediable. While student academic performance is a very strong indicator, it is too late to adjust the student's treatment, and teachers can no longer intervene, by the time a student finishes high school [6]. These detectors rely heavily on student grades and on-field observations. Thus, teachers find it difficult to identify problems and consequently to apply the appropriate type of support.

In a hopeful sign, the adoption of educational software has been expanding within different academic institutions in recent years. The utilization of this kind of software allows educators to gather data about student usage. The recorded data is fine-grained and relative to every student action within the system, opening up possibilities for extensive analysis, and ultimately growing into substantial sub-fields such as Educational Data Mining and Learning Analytics. With a large amount of data at hand, it became feasible to build predictive models capable of detecting student affects across a wide range of constructs such as gaming the system, boredom, carelessness, frustration, and off-task behaviours [2, 1, 17, 11, 16]. These affect detectors were the building blocks for subsequent research work that aimed at predicting learning outcomes [11], college enrollment [18] and more importantly predicting whether or not students will enroll in a STEM major in college [14].

The objectives of this research are two-fold. Firstly, we are building models for longer-term prediction as to whether or not students will pursue a career in STEM fields. For that, we use data gathered from a longitudinal study, over a decade long, featuring click-stream recorded data of middle school student interactions with an Intelligent Tutoring System for mathematics called ASSISTments. Secondly, we are investigating how school-aggregated data of student features can improve the model's correctness. For that, we measure the z-score for each student feature relative to his peer school mates. We call this approach the school-based approach, and we compare it to the normal approach, where no school-based feature transformation is done. We also discuss which features of affects, performance and behaviours are good predictors, at the same time, we introduce the usage of genetic programming to the process of finding the best machine learning pipeline for each approach. Finally we analyse the outcomes and compare the school-based approach to the normal approach.

2. METHODOLOGY

2.1 ASSISTments Tutoring System

In order to proceed with our research, we used a large amount of data coming from the ASSISTments platform. ASSISTments [1] is a web-based Intelligent Tutoring System provided for free by Worcester Polytechnic Institute. It is intended for application to middle school mathematics where teachers can use a predefined set of contents or can create their own. The system provides students with the right assistance while assessing their knowledge. 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 right, 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 to have another try. Then, after correctly answering this problem, they move on to the next one. Questions in the ASSISTments platform are related to specific skills, which makes tracking student performance more precise. On the other hand, teachers get full reports on student activities and their performance. That allows them to identify common mistakes and problems and find out who struggled to solve the problems; all of this can be done even before meeting their students in the classroom [9].

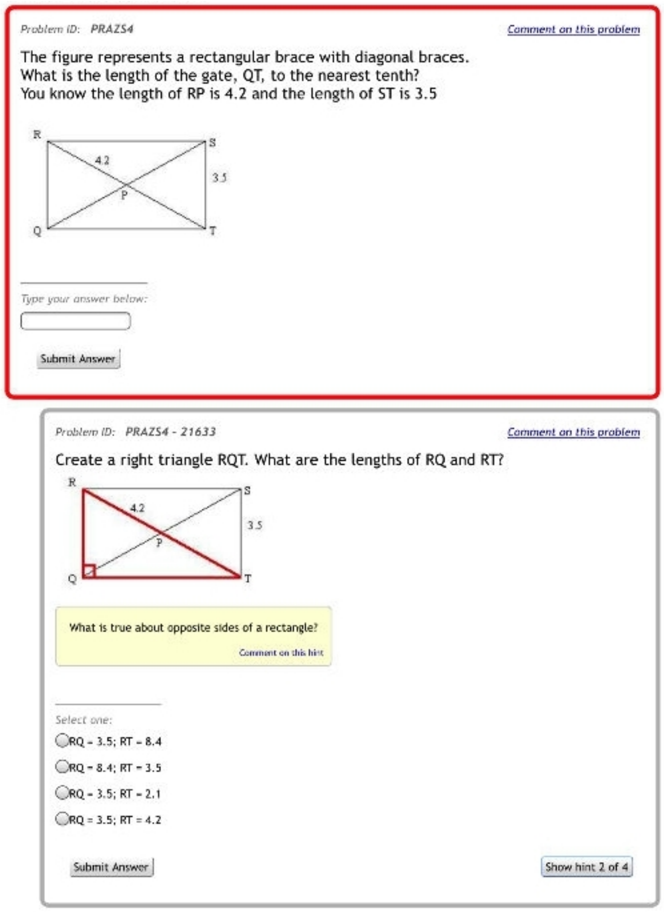

Figure 1: Example of an ASSISTments [1] problem where the student answered incorrectly, and is thus led to solve a scaffolding problem. The student can request a few hints from the system.

2.2 Data Acquisition

The gathered data consists of action log files representing click-stream interactions of students with the ASSISTments software during the period 2004-2007. We count 942,816 actions stored in the log files coming from different types of student interactions, such as, requesting help, answering a question or even revealing a hint. Each action is specified by a set of recorded information, and those actions were carried out by a group of 591 students from 4 different schools which used ASSISTments. Several other items of information relating to these students were recorded, like their high school course-taking, college enrollment, and first job out of

[1] www.assistments.org

college. This dataset contains no less than 3765 problems related to a complete set of 93 skills.

2.3 Features Exploration

The dataset contains 80 features, some of which were generated following a discovery with models approach, including student knowledge prediction, and student behavioural features and affects. We also used other features that are directly determined by student interactions with the software:

- Number of problems solved within the system
- Time taken to answer a question
- Number of original and scaffolding problems
- Correctness in original problems and scaffolding problems
- Correctness overall
- Number of hints used as well as "bottom hint" usage
- Number of help requests done as first attempts
- School id of the student

2.4 Discovery with models

Several models have already been used to capture some student behaviours or to predict their knowledge. In fact, for many years, predicting student knowledge was an active field of research [5, 13, 15, 7] that has been characterized by the emergence of Bayesian Knowledge Tracing (BKT) [5] as one of the most used models. 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 involved skill. In its classic formulation it has 4 parameters that are recalculated for each new skill: L0, T, G, S where L0 is the probability that the student already knows the skill before even the first try; T is the probability that the student learns the skill when he applies it; G is the probability that the student guesses the correct answer without really knowing it and S is the probability that the student slips and answers incorrectly even if he knows the answer. The BKT model used in this dataset has its parameters fitted using a brute-force grid search [3].

Along with predicting student knowledge, different models were developed in order to estimate student affects and disengaged behaviours. Research such as [11] has produced 4 affective state detectors: Boredom, Engaged Concentration, Confusion, and Frustration. The disengaged behaviours appear in the form of an off-task attitude, gaming the system and carelessness. To build these models, field observations were recorded when students used the ASSISTments software. Then the recorded data was synchronized with the internal log data of the system, resulting in an automated model that can be used to replace the in-field experiments.

2.5 Feature Transformation and Selection

To make the predictions relating to student enrollment in a STEM career, we needed to change the granularity of our data from the interaction level to the student level. To this end, we took the average of the selected features across all actions for each student. Picking the right features was done using univariate feature selection, only keeping features that have a strong relationship with the predicted variable. The results of the selection process are shown in Table 1

Table 1: Univariate Features Selection

	STEM Career	Mean	Std	F-Value
Avg Bored	0	0.252	0.033	2.90e-05
	1	0.252	0.031	p=0.99
Avg Bottom hint	***0***	***0.046***	***0.035***	***10.811***
	1	***0.034***	***0.029***	***p<0.01***
Avg Carelessness	***0***	***0.12***	***0.065***	***18.207***
	1	***0.15***	***0.078***	***p<0.001***
Avg Confused	0	0.106	0.038	0.013
	1	0.105	0.035	p=0.910
Avg Correct Original	***0***	***0.43***	***0.156***	***11.458***
	1	***0.485***	***0.176***	***p<0.001***
Avg Correct Scaffold	***0***	***0.584***	***0.106***	***4.494***
	1	***0.606***	***0.101***	***p<0.05***
Avg Correct	***0***	***0.417***	***0.152***	***16.516***
	1	***0.471***	***0.144***	***p<0.001***
Avg Engaged Concentration	0	0.647	0.03	1.209
	1	0.650	0.026	p=0.271
Avg Frustration	0	0.127	0.047	1.834
	1	0.121	0.052	p=0.176
Avg FirstHelpRequest	0	0.285	0.066	1.126
	1	0.292	0.071	p=0.288
Avg Gaming	***0***	***0.113***	***0.124***	***4.115***
	1	***0.088***	***0.105***	***p<0.05***
Avg Hint	***0***	***0.266***	***0.141***	***14.108***
	1	***0.214***	***0.124***	***p<0.001***
Avg Knowledge	***0***	***0.224***	***0.135***	***16.881***
	1	***0.283***	***0.162***	***p<0.001***
Avg Off-Task	0	0.216	0.082	0.069
	1	0.219	0.074	p=0.792
Avg Original	***0***	***0.298***	***0.125***	***8.904***
	1	***0.337***	***0.139***	***p<0.01***
Avg Scaffold	0	0.418	0.114	0.573
	1	0.426	0.118	p=0.449
Avg Time Original	0	64.38	34.18	0.946
	1	67.82	38.16	p=0.331
Avg Time Scaffold	0	32.51	17.16	0.416
	1	33.64	17.99	p=0.518
Avg Time Taken	0	40.84	21.09	2.445
	1	44.25	23.51	p=0.118
Nb Problems	0	236.3	139.5	1.754
	1	255.1	143.9	p=0.185

After running the test we observed that only some features have a strong relationship with the predicted variable. In fact, correctness is a strong predictor not only in this study but also in previous research focusing on college enrolment [18, 14]. This is more emphasised when we look at the correctness in the original problems, where the difference in the mean value is higher than the mean correctness in scaffolding problems. This is due to the fact that scaffolding questions aim to help the student acquire the skill and help him/her solve the original problem. In a way, having higher correctness in original problems gives us more insight about the skills of the student. Another strong predictor is the average of original problems, since it is the proportion of original problems over the total number of problems done by the student. A higher proportion of original problems translates to less of a "learning phase" involving scaffolding questions.

One interesting feature is the hint functionality usage. Hints give the student some advice on how to solve a problem while explaining the skill. That's why students with high hint requests are more likely to pursue a non-STEM career. Furthermore, bottom hints explain the problem from its basic notions. They are the lowest level of help, and that's why they are used less often, but the difference between the two groups of students is still significant. Extensive hints usage has been reported as a detector for gaming the system behaviour [1], which is another strong predictor for student enrollment in a STEM career. Students who loose interest in STEM have higher mean values in gamin the system.

Additional features that can be good predictors are carelessness and knowledge estimation. Similarly to STEM major predictions [14], the carelessness of students seems to increase when they are going to continue in a STEM career, which is a non-intuitive finding shared by the two pieces of research. Finally the average knowledge of a student is an estimation of his/her skills and to what extent he mastered the involved skill. It's the most straightforward predictor, since more knowledge means that the student has more aptitude to pursue a STEM career without serious problems.

However, we expected some of the features to be important while in reality they are not. Affects and behaviours such as boredom, confusion, engaged concentration, off-task behaviours and frustration are not good predictors for a STEM career. The same is true when predicting STEM college major enrolment [14]. Surprisingly enough, the difference in the time needed to answer questions is not statistically significant enough to be considered as a predictor. Although we assumed that unsuccessful students tended to use more time in order to give their answers, this was not the case. This can be explained by the fact that successful students might privilege taking their time to answer with certainty over answering quickly but without verification. This is valid for both original or scaffolding problems where students that did not pursue a STEM career answer faster than their counter parts.

2.6 Approaches

Once the useful features were selected, we transformed the dataset to prepare for the first approach, which consists of studying the effect of the school on the student's career outcome. If we put the student's performance in the context of their surroundings, which, in this case, is the school, we might grasp some important information about whether or not the student is willing to pursue a STEM career. So, the first approach, called the school-based approach, is to separate students by their schools, and then measure the z-score of all students' features school by school. This gives us a set of transformed data describing student data relative to their peer school mates. That was straightforward because none of the students in the dataset had changed their school during their usage of ASSISTments. On the other hand, the normal approach is to simply use the features without any similar transformation.

2.7 Optimization and genetic programming

Since we compare two different approaches independently, we want to find the most adequate machine learning method with the best hyper-parameters for each approach. And in order to find this "Pipeline", we use genetic programming as our tool for searching. We do not compare two machine learning methods but rather try to give each approach its best shot.

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 operators called crossover and mutation. Crossover is used to diversify the research in the research space by taking some parts of the parent individuals 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 research in a certain area of the research space. The population is evolving from one generation to another while keeping the fittest individuals in regard to one or many objectives. When using genetic programming for machine learning optimization, we use the pipeline score as the objective function; the pipeline accuracy score is an example of an objective function which has to be maximized.

In our case, we used genetic programming by searching through a multitude of machine learning techniques and their respective hyper-parameters to find out which combination gives the best results. To achieve our goals we used the python library TPOT [10]. However, in order to use genetic programming there are several hyper-parameters that we need to initialize.

Table 2: Genetic Programming Hyper-parameters

Generations count	Population size	Offspring size	Scoring
200	150	100	ROC AUC
Mutation rate	**Crossover rate**	**Internal Cross Validation**	
0.8	0.2	5-fold	

Table 2 explores the principal hyper-parameters that we have to initialize. The Generations count is the number of iterations of the whole optimization process. A bigger number gives better results but also takes more time to finish. We also can fix a maximum amount of time for the whole process. The Population size is the number of individuals which will evolve in each iteration. The offspring size is the number of individuals that are supposed to be generated from the previous population using the genetic algorithm operators. After executing the operators and generating the offspring, the individuals from the population and the offspring compete to survive and be part of the next population. When the individuals compete against each other, we only keep the fittest ones, meaning the individuals with the best score. The method used to measure the score is defined in the scoring hyper-parameters. We used the Area Under the Receiver Operating Characteristic Curve (ROC AUC) as our scoring method. That means we only keep the individuals (thus the pipelines) which have highest ROC AUC values. 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 80% chance of having a mutation against a 20% of having a crossover operation. Finally, the TPOT tool gives us the possibility to cross-validate our pipelines internally, therefore we set the number of folds to 5.

We ran two separate processes for each approach but with the same hyper-parameters and at the end of the optimization process we ended up with two different machine learning techniques in two different pipelines. For the school-based approach we use a Gaussian Naive Bayes classifier and for the normal approach we found that a Random Forest Classifier was the most efficient. Before running the optimization phase we did split the data into training and validation data that we hold for the final validation at the end of the process. This split was stratified using the label (STEM Career) and the school, in order to respect the proportions of school diversity and STEM outcome. Furthermore, within the process of searching with genetic programming we have an internal cross-validation mechanism. Finally, once we had trained our models, we tested them on the validation data that had been kept out of the optimization process.

As shown in Table 3, the approach in which we z-scored the student features within their respective schools gave us a statistically significant better result than the normal approach with an AUC of 0.604 while the result of the normal approach is about 0.494. But in the case of RMSE, the normal approach had a better score of 0.425 compared to 0.476 achieved by the school-based approach.

Table 3: Validation score of both pipelines

	School-based	Normal approach
ROC AUC	0.604	0.494
RMSE	0.476	0.425

Once we know which methods and parameters to use we proceed to train a cross-validated model using them. We use the same features as for the genetic programming, and we conducted a 10-fold stratified cross-validation training. As previously, the folds were stratified, where we respect the proportions of the label (STEM Career) and the difference in schools in each fold.

Table 4 shows the mean of the cross-validated values for both models. This time the school-aggregated model showed an increase of RMSE to over 0.54 compared to its counter part. On the other hand, the normal approach attained 0.521 in ROC AUC score, but was still lower than the score of the school model (0.601).

Table 4: Cross-validated scores for both approaches

	School-based	Normal approach
ROC AUC	0.601	0.521
RMSE	0.546	0.45

Figure 2 shows more about the cross-validated ROC AUC scores. The values of the normal approach are spread from the minimum of 0.36 to the maximum of 0.65 with 25% of the values exceeding 0.63 and another 25% are being less than 0.44. On the other hand, the school-based approach is less diverse, since its minimum is 0.47 and maximum is 0.70. Half of the values exceed 0.59 and 25% of them are above 0.67.

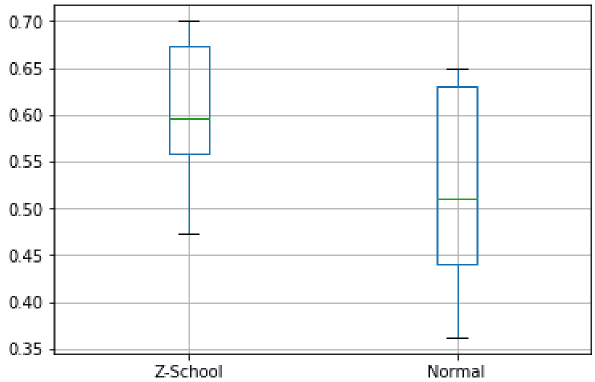

Figure 2: Cross-validated scores of ROC AUC for both approaches.

Now when comparing the RMSE scores of the two approaches, we clearly see in Figure 3 that the normal approach is almost perfectly distributed around the minimum of 0.45 and the maximum of 0.47. While the school-based approach is spread from the minimum of 0.47 to the maximum of 0.6. 25% of its values are under 0.51. Half of the data is above 0.555 and 25% of it is superior to 0.585.

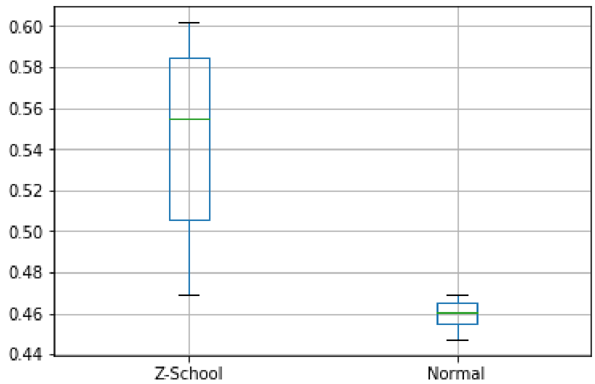

Figure 3: Cross-validated scores of RMSE for both approaches.

Even if the difference between the two approaches is statistically significant ($p<0.01$), the school-based approach has better AUC, while the normal approach has a lower RMSE, thus we cannot clearly confirm that the school-based approach has radically better results. The gain in terms of AUC is significant, but it suffers from a relatively high RMSE.

3. DISCUSSION AND CONCLUSION

In this paper, we aimed at longer term predictions of whether or not students will pursue a STEM career. We used student knowledge estimates, affects and disengaged behaviours, mixed with features from the dataset. Our approach was to take student performance and detector values and put them in context relative to their peer school mates. We wanted to investigate if the data of school mate can improve the model's predictions. As we looked to compare two different approaches rather than machine learning methods, we used genetic programming to find the best machine learning pipeline for both the school-based approach and the normal approach, resulting in a Random Forest Classifier for the normal approach and a Gaussian Naive Bayes classifier for the school-based approach.

The univariate features selection process resulted in the same assertions found before when predicting STEM major college enrollment, further strengthening the features choice. Affects like Boredom, Confusion, Engaged concentration and Frustration are not good predictors in either model. Similarly, disengaged behaviours such as off-task behaviour are not a good predictor either. Furthermore, Hint usage is a good detector of students having trouble with the assignments and loosing interest in the subject, which ultimately results in pursuing a non-STEM career. We have separated the Hint usage into bottom hint which is the lowest level of hint and the regular hint, and both have a strong relationship with the predicted variable. Previous study showed that gaming the system behaviour is correlated to turning away from STEM college enrollment [14]. Average correctness, whether for scaffolding or original problems, shows a statistically significant difference between students that are in STEM careers and those who are not. Showing also that it's better to have good scores in original problems, as they are the principal exercises necessary to complete the task, while scaffolding problems are meant to explain the skill gradually. Finally, student knowledge modelled by Bayesian Knowledge Tracing is a very good predictor, since being successful in solving problems is a trait that is well-known among the STEM field workforce.

Aggregating within the school gave us better ROC AUC scores, but suffered from a high RMSE, suggesting that the improvement is not so big between both approaches. Perhaps, thinking about how well a student performs compared to his peers in the same school may not have an impact on the student's final decision to pursue a STEM field career. But if we push the analysis further to aggregate student performances within their own classroom or maybe compared to students that have the same teacher we can grasp some useful information as to whether a teacher had an impact in reigniting the student's passion for STEM. In fact, schools are just institutions where teachers do their job, and in the field it's the professors who are in contact with the students, and every professor has his/her own pedagogy and way of teaching. It would be interesting to compare student performances within the finer-grained entity which is the classroom. There the comparison might be more fair as the students share the same small environment, exercises and are tough in the same way. Previous research showed that aggregating students data within the classroom can improve the knowledge modelling [21], so it might be a promising research area.

4. ACKNOWLEDGMENTS

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

5. REFERENCES

[1] R. S. Baker. 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, pages 1059–1068, New York, NY, USA, 2007. ACM.

[2] R. S. Baker, A. T. Corbett, and K. R. Koedinger. Detecting student misuse of intelligent tutoring systems. In J. C. Lester, R. M. Vicari, and F. Paraguaçu, editors, *Intelligent Tutoring Systems*, pages 531–540, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[3] R. S. J. d. Baker, A. T. Corbett, S. M. Gowda, A. Z. Wagner, B. A. MacLaren, L. R. Kauffman, A. P. Mitchell, and S. Giguere. Contextual slip and prediction of student performance after use of an intelligent tutor. In P. De Bra, A. Kobsa, and D. Chin, editors, *User Modeling, Adaptation, and Personalization*, pages 52–63, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[4] R. Balfanz. Putting middle grades students on the graduation path a policy and practice brief. 01 2009.

[5] 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, Dec 1994.

[6] D. F Whalen and M. Shelley. Academic success for stem and non-stem. 11, 01 2010.

[7] J. Martin and K. VanLehn. Student assessment using bayesian nets. *International Journal of Human Computer Studies*, 42(6):575–591, 6 1995.

[8] R. Noonan. Stem jobs: 2017 update. Office of the Chief Economist, Economics and Statistics Administration, U.S. Department of Commerce(ESA Issue Brief 02-17), March 30 2017.

[9] F. Olenchak and T. HÃľbert. Endangered academic talent: Lessons learned from gifted first-generation college males. 43:195–212, 03 2002.

[10] R. S. Olson, R. J. Urbanowicz, P. C. Andrews, N. A. Lavender, L. C. Kidd, and J. H. Moore. *Applications of Evolutionary Computation: 19th European Conference, EvoApplications 2016, Porto, Portugal, March 30 – April 1, 2016, Proceedings, Part I*, chapter Automating Biomedical Data Science Through Tree-Based Pipeline Optimization, pages 123–137. Springer International Publishing, 2016.

[11] Z. A. Pardos, R. S. J. D. Baker, M. O. C. Z. San Pedro, S. M. Gowda, and S. M. Gowda. 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*, LAK '13, pages 117–124, New York, NY, USA, 2013. ACM.

[12] E. Pascarella, C. Pierson, G. Wolniak, and P. Terenzini. First-generation college students: Additional evidence on college experiences and outcomes. *Journal of Higher Education*, 75(3):249–284, 5 2004.

[13] 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, Amsterdam, The Netherlands, The Netherlands, 2009. IOS Press.

[14] M. O. S. Pedro, J. Ocumpaugh, R. S. J. de Baker, and N. T. Heffernan. Predicting stem and non-stem college major enrollment from middle school interaction with mathematics educational software. In *EDM*, 2014.

[15] J. Reye. Student modelling based on belief networks. *Int. J. Artif. Intell. Ed.*, 14(1):63–96, Jan. 2004.

[16] J. Sabourin, B. Mott, and J. C. Lester. Modeling learner affect with theoretically grounded dynamic bayesian networks. In S. D'Mello, A. Graesser, B. Schuller, and J.-C. Martin, editors, *Affective Computing and Intelligent Interaction*, pages 286–295, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[17] M. O. C. Z. San Pedro, R. S. J. d. Baker, and M. M. T. Rodrigo. Detecting carelessness through contextual estimation of slip probabilities among students using an intelligent tutor for mathematics. In G. Biswas, S. Bull, J. Kay, and A. Mitrovic, editors, *Artificial Intelligence in Education*, pages 304–311, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[18] S. San Pedro, R. Baker, A. Bowers, and N. Heffernan. Predicting college enrollment from student interaction with an intelligent tutoring system in middle school. In *EDM*, pages 177–184, 01 2013.

[19] X. Wang. 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, 2012.

[20] X. Wang. Why students choose stem majors: Motivation, high school learning, and postsecondary context of support. *American Educational Research Journal*, 50(5):1081–1121, 2013.

[21] Y. Wang and J. Beck. Class vs. student in a bayesian network student model. In H. C. Lane, K. Yacef, J. Mostow, and P. Pavlik, editors, *Artificial Intelligence in Education*, pages 151–160, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[22] D. F. Whalen and M. C. Shelley. "academic success for stem and non-stem majors". *Journal of STEM Education*, 11, 2010.