Towards Interpretable Automated Machine Learning for STEM Career Prediction

Ruitao Liu
NebuAI
ruitao.liu@rocketcatai.com

Aixin Tan
University of Iowa
aixin-tan@uiowa.edu

In this paper, we describe our solution to predict student STEM career choices during the 2017 ASSISTments Datamining Competition. We built a machine learning system that automatically reformats the data set, generates new features and prunes redundant ones, and performs model and feature selection. We designed the system to automatically find a model that optimizes prediction performance, yet the final model is a simple logistic regression that allows researchers to discover important features and study their effects on STEM career choices. We also compared our method to other methods, which revealed that the key to good prediction is proper feature enrichment in the beginning stage of the data analysis, while feature selection in a later stage allows a simpler final model.

Keywords: STEM careers, automated prediction, penalized logistic regression, forward-backward search algorithm, interpretable machine learning

1. INTRODUCTION

Researchers from Worcester Polytechnic Institute and the University of Pennsylvania organized a data mining competition using educational data from ASSISTments, an online learning platform that supports student learning through the use of scaffolding, hints, immediate feedback, and detailed solutions for middle school mathematics. The aim of the competition was to help educators, researchers, and policymakers understand how students' experiences in middle school mathematics classes are related to eventually choosing a STEM (Science, Technology, Engineering, and Mathematics) career. Indeed, reliable STEM career prediction will help students uncover their STEM interests and further support their academic growth in STEM fields. The data set provided in this competition contains students' interaction information with the platform when they were in middle school. The data analysis challenge is to predict which students pursued careers in STEM fields after they graduated from college.

In the past two decades, we have seen a large number of high-quality works using students' academic performance and learning behavioral data to predict outcome variables, such as standardized test score, dropout from school, college enrollment, and major choice. For example, Feng, Heffernan, and Koedinger (2009) investigated how students' interaction data extracted from the ASSISTments platform can be used to reliably evaluate students' math

proficiency. They were especially interested in building features related to student help seeking behaviors and used the Bayesian Information Criterion (BIC) to compare linear regression models with different groups of predictors. They showed that students' end-of-year exam scores can be better predicted by leveraging the interaction data that reflect assistance requirement, effort, and attendance. Pardos, Baker, San Pedro, Gowda, and Gowda (2014) also studied the ASSISTments system, but they focused on the correspondence between student affect and behavioral engagement and scores on a high-stakes math exam. Using eight machine learning models, they constructed a set of affect and engagement behavior detectors to estimate the probability that a student is in a state of boredom, engaged concentration, confusion, and so on. They further built a model to predict students' math exam scores and showed that the constructed detectors helped the model achieve high prediction accuracies. Baker, Berning, Gowda, Zhang, and Hawn (2019) presented a case study on automatically identifying students that have a high risk of dropping out of high school, using data on students' discipline, attendance, course-taking, and grades. The logistic regression model used in the study helped the authors not only select students at risk, but also found which factors played the largest roles in prediction, which provided information to educators that can be used in individualized interventions. Knowles (2015) described how to create a statewide dropout early warning system that can accurately predict the likelihood of graduation for high school students in the State of Wisconsin. The paper thoroughly demonstrated the workflow of the whole system, from data cleansing to model training and searching. To balance the tradeoff between the correct classification of dropouts and false alarms, the receiver-operating characteristics (ROC) metric is used to identify the best models from a large collection of candidates, from linear logistic regression models to complex nonlinear models, such as support vector machines. This work was also implemented in the open source R package, EWStools (Knowles, 2014).

Instead of using traditional explanatory variables in college enrollment research, such as family background, career aspiration, and assessment scores, San Pedro, Baker, Bowers, and Heffernan (2013) studied how student online learning behaviors observed in middle school related to their college choice. They built a logistic regression model using automatically generated affect and engagement features to achieve decent accuracy at predicting college attendance. Their study was further extended to predicting STEM and Non-STEM college major enrollment by San Pedro, Ocumpaugh, Baker, and Heffernan (2014).

The above three selected sets of works studied test scores, dropouts, and college choices, respectively, by linking them to student learning behaviors. In comparison, the current competition aims at predicting a longer-term outcome than that studied in any previous work of this nature — to predict STEM career choices after college using middle school learning behaviors. To meet this challenge, we produce a prediction system with the following properties. First, the system should fit existing data well, and make good predictions on new data. Secondly, we would like the system to be automatic, that is, to avoid unnecessary human intervention. Thirdly, the system should help identify a small number of the most influential predictors and allow relatively easy interpretation of the final predictive model. Briefly, our system attains the first property above by selecting models and their parameters using crossvalidation (CV) techniques with respect to a metric determined by the competition organizers. The second property is attained by doing aggregation over records from the original data set, using not just the means, but also additional values including various quantiles, the minimums, and the maximums, and by further forming an extensive collection of transformed variables as well as two-way interactions. With an extra rich pool of candidate features to exploit, we have a better chance of finding a good model. Finally, we attain the

third aforementioned property of the system by adopting a forward-backward strategy (FBS) in variable selection, where the inclusion or exclusion of variables is based on internal CV performances. This is different from traditional variable selection methods and algorithms based on p-values, information criterion, or penalized regression. As a result of adopting FBS, despite a large enriched dataset from which we search for a good model, the final model itself is a rather simple logistic linear regression model that involves only a handful of variables. This makes our system different from the state-of-the-art machine learning methods, in the sense that researchers using our system have a chance to interpret and explore relationships between the selected variables and the STEM career. By comparison, it is harder to demystify machine learning models that crank out black-box predictions.

The rest of the paper is structured as the following. Section 2 introduces the process of data preparation, new feature generation, and simple pruning, which results in an enriched data set for the next step. Section 3 describes the process of model building (including the use of feature selection strategies), with the goal of optimizing model performance in terms of a criterion set by the competition organizers. Section 4 discusses the pros and cons of our final model to that of several others. Section 5 concerns the interpretation of our final model. Finally, Section 6 discusses future research directions.

2. Data Preparation

2.1. Overview of the data set

The competition provided an extensive click-stream data set extracted from the ASSISTments database. It contained user interaction information from 591 students who used the system during their middle school years, as well as whether each of them pursued a career in STEM fields (1) or not (0) after college. The entire data set was divided by the competition organizer into three parts: the training set, the validation set, and the test set. Visible to participants of the competition were user interaction data for all three sets, and career choice data (the target variable) for the training set only. Data on career choice for the validation and the test set were withheld by the organizer for evaluation purposes. Specifically, competition participants used the training set to build models and made predictions on the validation and the test sets. Each day, each team could submit one set of predictions to be scored. The score was a combination of root mean squared error (RMSE) and area under the curve (AUC) based on predictions for the validation set. On each day before the conclusion of the competition, the organizer would post a public leaderboard showing each team's best submission to date and the corresponding evaluation metric values, to help the teams improve their models. Eventually, when the competition concluded, teams were ranked by the performance of their final model over the test set.

Although there are only 591 students, each student has hundreds of interactions with the system. The resulting data set is rather large, with 316,974 records (rows), each with 76 variables (columns). Each record captures one action (such as solving a multiple-choice question related to square root finding) of a student, along with some context information. Examples of context information are: average student knowledge level (according to the Bayesian Knowledge Tracing (BKT) algorithm, Corbett and Anderson, 1995), average student carelessness (San Pedro, Baker, and Rodrigo, 2014), average student boredom effect (Pardos, Baker, San Pedro, Gowda, and Gowda, 2014), and knowledge estimates based on BKT at the previous and the current time step. A detailed description of the variables can be found on the competition webpage: https://sites.google.com/view/assistmentsdatamining/data-mining-

competition-2017. Besides the main data set, the organizers also provided each student's state test score during that year. We included this variable in our analysis, but it was not selected by our model selection procedure in predicting STEM career choice.

2.2. DATA SET REFORMATION

To make predictions of the STEM career choice for each student, we first reformatted the main data set into a tabular data set with 591 rows, and one row for each student. Specifically, for any given variable from the activity information in the original data set, we aggregated the many rows of its value for a single student to a few summaries, as our new variables. Below, we describe the different aggregation methods used for the four different types of variables: single-valued, binary-valued, nominal-valued, and continuous-valued.

First, some variables were already aggregated by the competition organizers. Examples are average student knowledge level ("AveKnow"), total number of student actions in system ("NumActions"), and average student carelessness ("AveCarelessness"). Given any student, each of these columns contains one common value across the multiple rows of this student's actions. And this common value is directly assigned to the corresponding variable in the reformatted data.

For binary-valued variables, we used two methods of aggregation for the rows of each student: summation and relative frequency. For example, the variable "correct" in the original data set takes the value 1 if a student's response to a problem is correct, and 0 otherwise. We summed up all its values for a student to get the total number of the correct answers. Also, we calculated the proportion of the correct answers among all problems attempted.

For nominal variables, we used two methods of aggregation: the number of different values that occurred, and the average number of records per value (the total number of records divided by the number of distinct values that occurred). For example, the variable "problemType" in the original data set describes the type of the current problem the student was worked on. There are a dozen different values possible for this variable, including "textfile question", "radio question", and so on. One student may have worked on 3 types of problems, while another may have encountered all types. We believe the number of types of problems a student attempted reflects the breadth of the students's STEM interests, hence our first aggregation. In addition, the number of problems attempted per type reflects the depth of the effort made by a student for each type he or she chose to work on, hence our second aggregation.

For continuous variables, we calculated the following 13 summary statistics for each of them: the minimum, the maximum, the mean, the standard deviation, and 9 different percentiles (from the 10th to the 90th). In addition to the continuous variables from the original dataset, we formed new ones based on the continuous variables "Ln" and "Ln-1". Here, "Ln" is a measure of the proficiency level for the skill needed for the current problem at the current time, and "Ln-1" is that of the previous time step. The proficiency level is measured by the estimates of a student's math knowledge using the BKT method (Corbett and Anderson, 1995). Also, the cognitive skill needed for the current problem is provided in the nominal variable, "skill", reported in terms of knowledge components (KC). The detailed definition and usage of KC in the ASSISTment system can be found in Razzaq, Heffernan, Feng, and Pardos (2007). Given the above descriptions, it is natural to combine the value of "skill" with that of "Ln" and "Ln-1" to generate potentially useful new variables that reflect students' proficiency level and their improvements per skill. Specifically, we first formed

eight new continuous variables: over the set of records that correspond to a specific skill of a student, we calculate the minimum, the mean, the maximum, and the difference between the maximum and the minimum of "Ln" and "Ln-1", respectively. Then, for each of the eight new variables, we computed the 13 summary statistics mentioned in the beginning of this paragraph. As a result, we have enriched the data set with many variables. Take, for example, a student who practiced on 10 different skills. We can first obtain the maximum "Ln" value within each skill, which reflects the highest proficiency level the student ever achieved on each skill. Then we include new variables based on summaries like the mean, the standard deviation, and the minimum of the 10 maximum "Ln" values, which reflect the average maximum proficiency, the variation in maximum proficiency, and the proficiency of the weakest skill of the student.

After the inclusion of additional variables, the new data set has 591 rows and 717 columns. Unlike in the physical sciences, there is rarely scientific theory in social sciences and in the educational field that analytically relates the target variable to the features. It is possible that some of these 717 features facilitates the prediction of the target variable, STEM career choice, in different linear and non-linear fashions, and they can be impactful by themselves and/or through interactions. To include or approximate the many possible types of relationships among the variables, we generated abundant new features based on the 717 aggregations. Details are described in the next subsection.

2.3. FEATURE GENERATION

2.3.1. Generation of new univariate features

Since all 717 variables are technically non-negative continuous variables, we considered nine mathematical transformations to each of them, including logarithm with the natural base, and power functions with the power of -3, -2, -1, -0.5, 0.5, 1, 2, and 3, respectively. Here, the logarithm transformation helps symmetrize heavily right-skewed distributions. Various power transformations are also common techniques to potentially stabilize the variance of the variables and make their distribution more normal-like. To reduce redundancy, we only kept generated variables that are different enough from existing variables and at the same time are highly correlated to the target. Specifically, we adopted Pearson's correlation coefficient, and for a generated variable to be included, its absolute correlation with the original variable should not exceed 0.7, and its correlation with the target variable should exceed 0.15 and be at least 0.1 more than the correlation between the original and the target variable. In principle, all the aforementioned thresholds can be treated as parameters to be tuned, say, by CV.

2.3.2. Interaction features generation

Once the univariate transformations and screenings are done, we further enrich the pool of predictors with seven kinds of pair-wise interactions: multiplication, addition, subtraction, variable A divided by variable B, variable B divided by variable A, and the minimum and the maximum of the two variables. To avoid dividing by zero, the denominators were set to 1 plus the value of the denominator variables in the division operations. We again include a fast screening step to eliminate the interaction variables that either look similar to existing variables or are poorly correlated with the target. The same thresholds were used as that of the univariate screening, except that the absolute correlation between the interaction variable and

target variable is required to exceed 0.01 plus the maximum of the absolute correlations between the two original variables and the target variable. These screening criteria helped retain promising predictors while avoid inter-collinearity problems for later regression analysis.

After the above screening processes, there is still a rich pool of 2217 variables. Also, the total number of students remained 591, with 467 of them in the training set.

2.3.3. Feature elimination

In the early stage of the competition, we used the above enriched data set in the subsequent model building procedure (including variable selection) for data analysis, described in Section 3. However, feedback from the public leaderboard suggested that the models that performed better on our internal CV set tended to do worse on the validation set. This somewhat surprising result prompted us to investigate discrepancies among the training, the validation and the test set. Indeed, we found serious discrepancies in the distributions of several predictors in the three sets. This type of problem is often referred to as the covariate shift problem in machine learning research (Sugiyama, Krauledat, and Müller, 2007). Besides problems that concern the predictors, there is also an imbalance label problem. Indeed, we deduced that the distribution of the target variable is much more imbalanced in the validation set than in the training set. Actually, only 5% of the students in the validation set had chosen a STEM career, compared to 25% in the training set. After all, we decided that features generated in Section 2.3 that suffered from the discrepancy problems were not the most promising predictors for the test set. So, we designed the following additional feature elimination step.

First, we identify the variables for which the distributions in the training set and the test set are the most different based on a measure of discrepancy (MOD) that we now describe. (For the calculation of MOD, we simply combined the validation and test set as one test set.) Given any variable, we obtained ten percentiles (from the 10th to the 100th) of its values in the training set and recorded the percentiles theses values corresponded to in the test set. For example, if the 50th percentile of a variable in the training set was 224.5, and the value 224.5 happened to be the 57th percentile of this variable in the test set, then we recorded an absolute difference of 57-50 = 7 percentage points for this variable. Then, among the 10 absolute differences (one for each of the 10 percentiles inspected), the maximum value was defined to be the MOD. Any variable with MOD greater than a given threshold value will be eliminated. To choose a good threshold value, we considered six integer values, from 4 to 9, which led to six different data sets. For each data set, we performed the analysis of Section 3.2. Among the 90 (15 times 6) combinations of models and data sets. We chose as the final model the one with the best CV result based on the evaluation metric defined by the organizer (see Section 3.1). It turned out that the optimal value for the MOD threshold was 6.

3. MODEL BUILDING

3.1. MODEL EVALUATION AND MODEL SELECTION

The competition organizer used an interesting, nonstandard evaluation metric (EM): the sum of the (1-RMSE) and AUC. As far as we know, no existing statistical or machine learning methods are designed to optimize (that is, to maximize) this EM directly. Recall that we

intended to obtain a final model that allows certain degrees of interpretability. Therefore, we decided not to pursue the state-of-the-art predictive methods such as gradient boosting machine (GBM) (Friedman, 2001), which makes predictions using highly sophisticated combinations of all available features. Instead, we decided to consider logistic regression models built upon different subsets of features and identify the model with the best crossvalidated performance in the aforementioned EM. Note that there was an astronomical number of 2^p different models to consider, where p was the number of features, which was in the thousands for the enriched dataset from Section 2. Hence strategies were needed to find the optimal or close-to-optimal model, which we discuss in the next subsection.

3.2. THE FORWARD-BACKWARD STRATEGY (FBS)

In searching for an optimal subset of variables, a common strategy in the literature is the Forward-Backward Strategy (FBS), which updates the current model by including or excluding one variable in each step. As for which variable to include or exclude, standard practice involves fitting the resulting model using logistic regression and checking if the associated gain or loss in fitting is worthwhile, say, if it improves value of some information criteria like AIC or BIC. Since this competition defined its own EM, we decided to modify the above standard practice by (1) evaluating a model using cross-validated EM instead of popular information criteria, and (2) fitting a model using a slightly more general approach than logistic regression.

For (2), we considered five different penalized logistic regression methods and let the training data help decide which one eventually yields the best final model. The five different penalties are the least absolute shrinkage and selection operator (LASSO, Tibshirani, 1996), the Ridge penalty (Hoerl and Kennard, 1970), the Elastic Net penalty (Zou and Hastie, 2005), the smoothly clipped absolute deviation penalty (SCAD, Fan and Li, 2001), and the minimax concave penalty (MCP, Zhang, 2010). Each of these penalty functions has its own tuning parameter(s). For simplicity, we allowed the parameters to take values on a pre-defined grid, and eventually experimented with 15 different combinations of penalty methods and their tuning parameter values. The R package ncvreg (Breheny and Huang, 2011) can be used to implement all the above penalized logistic regression methods with efficient coordinate descent algorithms.

Next, we explain the CV-based criteria of including or excluding a variable. Despite the seemingly lengthy description to follow, the procedure is entirely automated by a searching system that we coded in R (R Core Team, 2017). The training set is partitioned into five subsets with roughly equal numbers of students. When evaluating a model in a step, a five-fold CV is performed by holding out one subset as the internal test set, while using the rest for training. A repeat over all five folds generates five values of the EM.

To start, we fix a penalized method, such as the Lasso. In the forward stage, we first find the variable that has the highest absolute correlation with the target variable and call it the best one-variable set. A five-fold CV is conducted with the given penalized method using this variable, which resulted in five EM values that we call the current best CV values. Next, to find the best two-variable set, we enumerated all the remaining variables, paired one at each time to the best one-variable set and use the penalized method in another five-fold CV to get a set of five new values of the EM. If the mean of the new EM values is greater than a small positive threshold value plus the mean of the current best values, and that the minimum of the new EM values is greater than the current best minimum minus a small positive number, then

the best two-variable set and the best CV values are updated accordingly. Once every variable was screened, the best two-variable set was found, and we moved on to find the best three-variable set, and so on. The forward stage stopped when no more variable met the criteria to enter the best set. The above strategy of considering both the mean and the minimum (that is, the worst case) of the CV values came from the intention to attain good average performance while being robust over the five folds.

In the backward stage, we excluded one variable from the current best set each time and compared the model performance with the current best model using the same criteria as in the forward stage. When no more variables can be dropped, the backward stage halted, giving a final set of variables.

Note that we can execute the FBS process using different penalized logistic regression models, and they may lead to different final sets. A comparison among these different final sets can be done, simply by consulting their respective set of five CV EM values, which were part of the output of the FBS process. Our final model is based on the final set that has the highest mean CV EM.

4. COMPARISON OF THE FINAL MODEL AND OTHERS

4.1. OUR FINAL MODEL

The eventual best subset of features was selected by using the MCP logistic method through the FBS and retained 14 features. As is typical in making predictions using methods that include variable selection steps, we refitted the data with the 14 selected features. This time, a Ridge model was used for estimating coefficients because it had the best performance among all regression models experimented on in terms of internal CV, and it turned out to also have the best performance on the validation set used for the public leaderboard standing. A parsimonious model like ours avoids overfitting the observed data and is likely to generate smaller prediction error for future observations.

4.2. COMPARING DIFFERENT PREDICTION METHODS

It is natural to wonder how much our model improved upon simpler ones, and how it compares to other more advanced prediction methods. Recall that two main ideas that lead to our final model are feature enrichment and feature selection. Through comparisons with different methods that use some or none of these ideas, we show that feature enrichment is the step that brought major improvement in prediction for many different follow-up prediction methods, while feature selection using our FBS is the step that led to a parsimonious final model, hence better interpretability.

(1) **Using basic features only.** In the original data set, there were ten aggregated variables, including “AveKnow”, “AveCarelessness”, “AveCorrect”, and so on. Using only these 10 aggregated variables, we implemented the classical logistic regression method, several penalized logistic regression methods, and the sophisticated GBM method. The value of EM of these predictions on the test set is shown in Table 1. For clarity, among the penalized regression methods, only the performance of the Lasso method (with its penalty parameter optimized by internal CV) is included because it performed better than its peers.

Table 1: Summary of EM

Model_#features	EM	Improvement relative to logistic_10
Based on 10 basic features		
logistic_10	0.994	0
lasso_10	0.982	-1.2%
gbm_10	1.03	3.6%
Based on 2217 enriched features		
lasso_22	1.033	3.9%
gbm_2217	1.086	**9.3%**
Based on 14 enriched features selected by FBS		
ridge(FBS)_14 (final)	1.048	**5.5%**
gbm(FBS)_14	1.067	**7.4%**

(2) **Using enriched features, but without FBS.** There are 2217 features in our enriched dataset after the feature generation and screening steps described in Section 2. Since there are more features than the number of subjects, the classical logistic regression method is not applicable, but penalized methods are. Many penalized methods select variables automatically and possess various theoretic properties. In brief, under certain conditions, the model resulting from certain penalized methods approaches the "true" model when the number of students increases. Therefore, we implemented several penalized methods on our enriched data set directly (without FBS). For example, the Lasso method (with its penalty parameter optimized by CV) generated a model that retained 22 features.

(3) **Using features selected by the FBS.** Recall that 14 variables remain after feature enrichment and selection using FBS. Our final model is a Ridge regression model. We also constructed GBM based on the same 14 variables.

4.2.1. Implications of Table 1

First of all, we mention that the test set is the set used by the competition organizers to rank the participating teams. Since prediction performance in EM will change for a different test

set, the numbers reported in Table 1 and the implied rankings of the different methods involve uncertainty. Nevertheless, some insights can be drawn. One can see that all prediction methods based on enriched features performed better than all methods based on the original 10 features, indicating the importance of generating more features.

Based on the enriched features, the two GBM methods perform the best, followed by our method (ridge(FBS)_14). The GBM based on 2217 features had the best prediction performance, but it does not explicitly show how each feature affects the target variable and does not help researchers interpret the most influential features.

In terms of feature selection, the FBS we used is a key step to reduce the number of features to only 14, a variable set for which both the GBM (gbm(FBS)_14) and our more interpretable penalized regression method (ridge(FBS)_14) perform reasonably well. Note that for feature selection, one could have used a penalized regression method that is less costly than the FBS, but the latter seems to have an advantage in selecting the most useful features: recall that the Lasso method selected 22 variables, while our regression model based on FBS used only 14 features and improved the EM value on the test set by 15% compared to the Lasso.

An explicit formula for prediction using our model is provided in the next section, which provides data analysts and domain experts a chance to study and explain it. The same type of inference is hard to do with GBM.

5. INTERPRETATIONS OF THE FINAL MODEL

The final logistic regression model with estimated coefficients for each predictor is:

$$\begin{aligned} log\left(\frac{P(Y=1|X)}{1-P(Y=1|X)}\right) = \quad & -1.154 + 0.085X_1 + 0.061X_2 - 0.122X_3 \\ & +0.105X_4X_5 - 0.051X_5X_6 + 0.113X_7 - 0.079X_8 \\ & -0.157X_9 - 0.101\frac{X_{10}}{X_4+1} + 0.087\frac{X_{12}}{X_{11}+1} \\ & -0.168\frac{X_6}{X_{13}+1} + 0.047\frac{X_{15}}{X_{14}+1} + 0.113\frac{X_{16}}{X_{14}+1} \\ & +0.126\frac{X_{18}}{X_{17}+1} \end{aligned}$$

Here, $P(Y=1|X)$ represents the probability of choosing a STEM career given a set of values of the predictors, X. Table 2 below lists the variables used in equation (1). There were 14 predictors that were formed by 18 variables. Each of the 14 predictors has been standardized to have mean 0 and standard deviation 1, so that the regression coefficients are comparable in size.

We now give a couple of examples to show what the coefficients in our model may imply. The variable X_1 is described in Table 2. It measures the variability in the number of times scaffolding hints had been accessed among the different skills the student had practiced. In model (1), the coefficient of X_1 is 0.085, which means that students with larger values of X_1 have a higher tendency to choose STEM careers. Specifically, by holding the value of other variables unchanged, increasing the value of X_1 by one standard deviation increases the odds

of choosing a STEM career for a student by about 9% (exp(0.085)-1). Note that an association between X_1 and the target does not imply causation. However, seeing the association from the model allows researchers to further explore the data set or the literature to see what other variables are highly correlated with X_1 that may have an impact on the STEM career choice. We found that the students who practiced many skills but mainly focused on a handful of them tended to have large values of X_1. It could be that these students had decent interest in learning mathematics and also made efforts to improve their weaknesses; students who have such an attitude and approach for math learning are more likely to choose STEM careers.

For another example, we look at the interaction term that has X_{10} in the numerator, and X_4 in the denominator (and recall from Section 2 that the plus one in the denominator is just for technical reasons). Here, X_{10} is the total number of hints used by a student, and X_4 is the number of sessions, that is, the number of logins to the ASSISTments system. Thus, their ratio reflects the number of hints a student accessed per session. The regression coefficient is -0.101, suggesting that students who depend more on hints are less likely to pursue STEM careers. For example, take two students A and B who have the same records except that the number of hints per session A needed is one standard deviation higher than B. In this case, the odds that student A chooses a STEM career is predicted to be 90% (exp(-0.101)) that of student B. Note that the above is a naive attempt to interpret the effect of a predictor in our regression model. It is most likely that two students who need very different numbers of hints per session will have different learning behaviors that result in different values for many other predictors as well. More comprehensive ways to interpret regression models and the real impact of different predictors are available and are under continuous development. We will not go over similar interpretations of the effect of each predictor due to space limitations.

Table 2: Predictors and their descriptions.

Predictor Name	**Symbol**	**Description of the predictor, or how its value is obtained for each student**
fsca_oppo.sd	X_1	The standard deviation of the number of times a student accessed the scaffolding hints among all skills the student has worked on
sumt3.0.	X_2	The minimum value of "sumTime3SDwhen3RowRight"
ln_diff_mean.0.	X_3	First, take Ln and Ln-1, variables described in Section 2.2, which measure the proficiency level for the skill needed for the current problem at the current and the previous time, respectively. Then, the mean of their differences reflects the average instantaneous speed of improvement in proficiency for a particular skill. Finally, we take the minimum of this speed across all skills, which reflects the speed of improvement for the skill that was the least improved upon.
num_session	X_4	The number of sessions

perc_ogi	X_5	The proportion of non-scaffolding problems the student practiced
5help.70.	X_6	First find the sums of the number of helps requested for the past 5 problems at each time, and then obtain the 70th percentile of the sums
perc_sca	X_7	The proportion of scaffolding problems the student practiced
hint.20.	X_8	The number of hints used for the first one fifth of problems
hint.40.	X_9	The number of hints used for the first two fifths of problems
hint.100.	X_{10}	The total number of hints used
8help.80.	X_{11}	First find the sums of the number of helps requested for the past 8 problems at each time, and then obtain the 80th percentile of the sums
num_sca	X_{12}	The number of scaffolding problems the student practiced
num_prob_type	X_{13}	The number of problem types the student practiced
perc_prob_type	X_{14}	Total number of activities divided by num_prob_type
8help.50.	X_{15}	First find the sums of the number of helps requested for the past 8 problems at each time, and then obtain the 50th percentile of the sums
8help.90.	X_{16}	First find the sums of the number of helps requested for the past 8 problems at each time, and then obtain the 90th percentile of the sums
5help.40.	X_{17}	First find the sums of the number of helps requested for the past 5 problems at each time, and then obtain the 40th percentile of the sums
num_ogi	X_{18}	The number of non-scaffolding problems the student practiced

6. SUMMARY

For the competition, we built a machine learning pipeline to automate predictions for students' choice of STEM career. The pipeline consists of feature extraction, feature generation and basic screening, feature selection using FBS, and automatic model selection based on internal CV. The end product of the pipeline is a logistic regression model that involves both original features and generated ones, including two-way interactions. We showed how to interpret the effects of some of the predictors in the final model.

We also compared our model to several others that use different features and different model structures. Based on the comparison results, we believe that the key for good prediction of students' STEM career choice is to form a good set of basic summaries of their learning behavior (Section 2.2) and generate a rich enough set of features and interactions based on the basic summaries (Section 2.3), before further modeling and feature selection steps. Despite the advance in automatic machine learning tools, these initial steps of forming meaningful features and interactions are best done by domain experts and data analysts together. Only after this initial step of material collection and generation, can one expect to use machine learning techniques to harness the power of data for prediction. While highly sophisticated nonlinear and/or multi-level machine learning methods such as neural network, support vector machine and GBM might produce good predictions, models with relatively simple structures such as our regression model can also perform well, and provide more insights to researchers for current and future studies.

7. ACKNOWLEDGMENTS

We thank the organizers of the ASSISTments Data Mining Competition 2017 for allowing us to access this very well collected and structured longitudinal data set. We also thank the referees and editors that help us improve our manuscript greatly.

REFERENCES

BAKER, R., BERNING, A.W., GOWDA, S. M., ZHANG, S. and HAWN, A. 2019. Predicting K-12 Dropout. *Journal of Education for Students Placed at Risk (JESPAR),* 25 (1), 28-54, DOI: 10.1080/10824669.2019.1670065

BREHENY, P. and HUANG, J. 2011. Coordinate Descent Algorithms for Nonconvex Penalized Regression, with Applications to Biological Feature Selection. *Annals of Applied Statistics*, 5, 232-253.

CORBETT, A. T. and ANDERSON, J. R. 1995. Knowledge Tracing: Modeling the Acquisition of Procedural Knowledge. *User Modeling and User-Adapted Interaction*, 4 (4), 253-278.

FAN, J. and LI, R. 2001. Variable Selection via Nonconcave Penalized Likelihood and its Oracle Properties. *Journal of the American Statistical Association,* 96, 1348-1360.

FENG, M., HEFFERNAN, N. and KOEDINGER, K. 2009. Addressing the Assessment Challenge with an Online System That Tutors as it Assesses. *User Modeling and User-Adapted Interaction: The Journal of Personalization Research,* 19 (3), 243-266.

FRIEDMAN, J. 2001. Greedy Function Approximation: A Gradient Boosting Machine. *The Annals of Statistics*, 29 (5), 1189-1232.

HOERL, A.E., and KENNARD, R.W. 1970. Ridge Regression: Biased Estimation for Nonorthogonal Problems. *Technometrics*, 12, 55-67.

KNOWLES, J. E. 2014. EWStools: Tools for Automating the Testing and Evaluation of Education Early Warning System Models. R package version 0.1.

KNOWLES, J. E. 2015. Of Needles and Haystacks: Building an Accurate Statewide Dropout Early Warning System in Wisconsin. *Journal of Educational Data Mining,* 7 (3), 18-67.

PARDOS, Z.A., BAKER, R.S., SAN PEDRO, M.O.C.Z., GOWDA, Sujith M. and GOWDA, Supreeth M. 2014. Affective States and State tests: Investigating How Affect and Engagement during the School Year Predict End-of-Year Learning Outcomes. *Journal of Learning Analytics*, 1 (1), 107-128.

RAZZAQ, L., HEFFERNAN, N.T., FENG, M., and PARDOS, Z.A. 2007. Developing Fine-Grained Transfer Models in the ASSISTment System. *Journal of Technology, Instruction, Cognition, and Learning*, 5 (3). Old City Publishing, Philadelphia, PA. 2007. 289-304.

R Core Team. 2017. R: A Language and Environment for Statistical Computing. *R Foundation for Statistical Computing*, Vienna, Austria. URL: https://www.R-project.org/.

SAN PEDRO, M.O.C.Z., BAKER, R. S., BOWERS, A., and HEFFERNAN, N. 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.C.Z., BAKER, R. S., and RODRIGO, M. M. T. 2014. Carelessness and Affect in an Intelligent Tutoring System for Mathematics. *International Journal of Artificial Intelligence in Education*, 24(2), 189-210.

SAN PEDRO, M.O.C.Z., OCUMPOUGH, J. L., BAKER, R. S., HEFFERNAN, N. 2014. Predicting STEM and non-STEM College Major Enrollment from Middle School Interaction with Mathematics Educational Software. *Proceedings of the 7th International Conference on Educational Data Mining,* 276-279.

SUGIYAMA, M., KRAULEDAT, M., and MÜLLER, K.-R. (2007). Covariate Shift Adaptation by Importance Weighted Cross Validation. *Journal of Machine Learning Research, 8,* 985-1005.

TIBSHIRANI, R. 1996. Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society. Series B (Methodological)* 58 (1), 267-288.

ZHANG, C. 2010. Nearly Unbiased Variable Selection under Minimax Concave Penalty. *The Annals of Statistics*, 38 (2), 894-942.

ZOU, H. and HASTIE, T. 2005. Regularization and Variable Selection via the Elastic Net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 67 (2), 301-320.