A Comparison of Machine Learning Algorithms for Predicting Student Performance in an Online Mathematics Game

This paper demonstrated how to apply Machine Learning (ML) techniques to analyze student interaction data collected in an online mathematics game. Using a data-driven approach, we examined 1) how different ML algorithms influenced the precision of middle-school students' (N = 359) performance (i.e., posttest math knowledge scores) prediction and 2) what types of in-game features (i.e., student in-game behaviors, math anxiety, mathematical strategies) were associated with student math knowledge scores. The results indicated that the Random Forest algorithm showed the best performance (i.e., the accuracy of models, error measures) in predicting posttest math knowledge scores among the seven algorithms employed. Out of 37 features included in the model, the validity of the students' first mathematical transformation was the most predictive of their posttest math knowledge scores. Implications for game learning analytics and supporting students' algebraic learning are discussed based on the findings.

Keywords: mathematics learning; online mathematics game; prediction; random forest

Introduction

In recent years, there has been significant growth of digital game-based learning, and in turn, game learning analytics— the application of data analytics to educational game data—has received a lot of research attention (Alonso-Fernandez et al., 2020; Cheng et al., 2019; Emerson et al., 2020; Yang, 2017). Due to their highly interactive environments compared to other types of educational technologies, digital learning games record tremendous quantities of student actions in the form of log files and generate multiple types of data (Serrano-Laguna et al., 2014). By leveraging these data, game learning analytics has provided information and insights into student in-game behaviors, the effectiveness of educational games, as well as improvement and validation of game design for researchers and practitioners (Alonso-Fernandez et al., 2019; Cano et al., 2018; Kerr, 2015).

Along with this growth in game learning analytics, however, many studies in the field have focused on the use of simple aggregations of student actions (e.g., clicks, attempts, interaction times) or correctness of students' answers, and relatively little attention has been paid to qualitative aspects of students' actions or behaviors in the game, such as the productivity of students' problem-solving strategies (Alonso-Fernandez et al., 2019; Asbell-Clarke et al., 2021). Moreover, while it is encouraged to compare various machine learning techniques in order to draw a better prediction or classification result, much of the research (e.g., Chen et al., 2020; Emerson et al., 2020; Hooshyar et al., 2017) tends to use only one or two simple techniques (i.e., linear/logistic regression), rather than choosing the best model after evaluating multiple algorithms or techniques.

Over the past several years, members of our team have developed an online interactive mathematics game based on cognitive science and learning theories, which aims to improve students' algebraic understanding. Previous randomized-control trials have revealed that the game is effective in improving students' mathematical understanding as well as numerical fluency (Authors, 2019; 2022; 2023). However, the best feature sets or algorithms to predict student math performance have not yet been identified, warranting further investigation using data-driven approaches.

Hence, extending our prior work, the paper aims to examine how different student-related features provided for the optimization of machine learning techniques influence the precision of predicting middle-school students' mathematics performance in an online mathematics learning game. In particular, we used both simple aggregations of students' actions and hand-coded data measuring the qualitative aspects of students' actions in the game (e.g., the productivity of students' mathematical

problem-solving, mathematical strategies used). This study addresses the following research questions:

1. Which machine learning algorithm provides the best results for students' mathematics performance (i.e., posttest math knowledge scores) prediction in an online mathematics learning game?
2. What kinds of student behaviors in the game are associated with students' posttest math knowledge scores?

Literature Review

Machine Learning Algorithms for Prediction

Machine learning techniques are roughly categorized into two categories: (1) supervised learning that aims to classify or predict a target variable (e.g., pass/fail, student scores) given a set of input features in training datasets, and (2) unsupervised learning that refers to a data-tagging (or labeling) technique mainly used for identifying underlying structure in unlabeled datasets (e.g., clustering) (Tomasevic et al., 2020). Our review focuses on supervised learning approaches as we attempt to predict student mathematics performance (i.e., posttest scores).

The widely used supervised learning algorithms in the field of game learning analytics are linear/logistic regression, Support Vector Machines (SVM), Random Forest (RF), Multilayer Perceptron (MLP), AdaBoost, Linear Lasso, and Bagging Regressor (Alonso-Fernandez et al., 2019; Pedregosa et al., 2011). Each algorithm has its advantages and disadvantages. Logistic regression is relatively easy to implement and does not require feature scaling; however, it tends to show poor performance on non-linear outcome data (Gupta, 2020). In contrast, Random Forest produces good performance on imbalanced datasets and shows good handling of large datasets, missing

values, or outliers. However, the results of RF (i.e., feature importance) are hard to interpret because it is always relative to features and does not show the statistical relationship between features and outcome variables (Ziegler & König, 2014). SVM is well suited for pattern classification for non-linear data such as images or data with many features (e.g., face detection, image recognition, text classification). However, it may not be suitable for large datasets because it requires a large amount of time for data processing and does not perform well when the number of features exceeds the number of samples in the training dataset (Karamizadeh et al., 2014). MLP uses hidden layers between input and output variables to model the data. Like SVM, it works well on image, audio, and text data; however, it requires a large amount of data for training. AdaBoost is an iterative multiple-classifier that learns from the mistakes of weak classifiers and turns them into improved ones (Shrestha & Solomatine, 2006). It is relatively simple to implement but sensitive to noise or outliers in the data. Linear Lasso regression makes the automatic feature selection to decide features that should be included in a prediction model (Ellis, 2021). However, coefficients from the Lasso model can be biased as they do not present the true magnitude of the relation between the features and the outcome variable. Lastly, Bagging Regressor (also called Bootstrap Aggregating) trains multiple individual models in parallel and then uses an average of the models for overall prediction (Rocca, 2019). It performs well on data with many features, handling missing values well, and reduces model overfitting. However, models might cause underfitting as it ignores the highest and the lowest values and uses average results.

Although many studies in the field tend to use one specific ML algorithm to predict outcome variables (e.g., Chen et al., 2020; Emerson et al., 2020; Hooshyar et al., 2017), the performance of prediction models, such as accuracy and error levels, vary

depending on the algorithm applied in the model. For example, Alonso-Fernández et al. (2020) used nine different ML algorithms to predict the posttest scores (i.e., an increase in bullying awareness) after playing a game to raise students' awareness of cyberbullying. The results showed that Bayesian regression provided the best performance among the nine algorithms applied, followed by gradient boosting and random forests.

Thus, it is important to compare various techniques to identify the algorithm that suits the data as well as to improve the accuracy of prediction models. In this study, we compared the performance of seven supervised machine learning algorithms widely used in data mining (logistic regression, Support Vector Regressor (SVR), Random Forest, Multilayer Perceptron (MLP), AdaBoost, Linear Lasso, and Bagging Regressor; Pedregosa et al., 2011) in predicting student posttest math knowledge scores after playing an online mathematics game. We then examined an algorithm that provides the best results for student performance prediction using evaluation metrics (i.e., the accuracy of models, error measures).

Factors Affecting Students' Learning in Online Learning Games

There has been extensive research examining features predicting students' learning outcomes in the game learning analytics field. Although there is variability in the most influential predictor of learning outcome (Alonso-Fernandez et al., 2019), some studies have consistently found that students' in-game progress is predictive of their learning outcomes measured after the gameplay in multiple settings.

For instance, one study (Nguyen et al., 2020) investigated factors related to middle school students' decimal understanding in a digital learning game that teaches decimal numbers and operations concepts. Out of 19 features included in the linear regression model, students' pretest scores and two in-game behaviors (i.e., bucket

mastery, sorting mastery) showed significant and positive associations with their posttest scores. Data other than the game logs (e.g., demographic, engagement measured using a survey) were not predictive of posttest scores but game enjoyment. Another study (Shute et al., 2015) examined relationships among middle school students' prior knowledge (measured by pretest), in-game progress, persistence, and their understanding of physics in the digital learning game Physics Playground. Similar to Nguyen et al.'s study (2020), structural equation modeling results indicated that students' pretest scores and in-game progress (i.e., receiving gold trophies in the game) significantly predicted their understanding of physics. As such, although simple aggregations of students' actions or interactions in the game are useful to predict learning outcomes, adding additional information (e.g., prior knowledge) or more complex data (e.g., exploration strategies/failures) may improve prediction accuracy.

Our prior studies using the game logs collected in an online mathematics learning game found that some students' in-game behaviors were positively related to their posttest math knowledge scores (Authors, 2019; 2022a; 2022b). For example, one study (Author, 2019) identified 19 in-game metrics reflecting students' problem-solving processes using the log files within the game and examined which in-game metrics were significantly related to higher math achievement. The results indicated that students' in-game progress (i.e., completing more problems in the game) was positively related to their posttest math scores. Another study (Authors, 2022a) used students' pause time before solving the problems in the game as a proxy measure of students' thinking and planning and examined its influence on their strategy efficiency. The results revealed that students' longer pause time was related to their use of more efficient strategies, suggesting that taking time to plan out mathematical strategies before solving problems leads to higher efficiency in math problem-solving.

However, these studies used either (1) a subset of problems of our interest, or (2) a subset of variables of interest, rather than examining the best set of features using all available in-game metrics to predict students' math performance more precisely. Thus, extending our prior work, this study leverages a data-driven approach and builds models to predict student math performance by using both simple aggregations of students' actions in the game and qualitative, hand-coded data of the students' game exploration (i.e., mathematical problem-solving) strategies. Including various types of data from the game would provide us with a more holistic picture of student in-game behaviors and how they relate to student learning.

Methods

Game Description

The game used in this study was developed to help improve middle-school students' conceptual and procedural understanding of algebra. Developed based on perceptual learning theories, numbers, and mathematical symbols in the game are reified as movable objects so that students can dynamically manipulate and transform numbers or mathematical expressions on the screen using gesture actions. In doing so, students can identify algebraic structures easily, think flexibly, and realize that mathematical transformations are dynamic rather than static recopying of lines.

The game consists of 14 worlds (a total of 252 problems) that cover a variety of mathematical concepts, such as addition, multiplication, division, and distribution, presented in the order of increasing complexity. As shown in Figure 1, each problem consists of two mathematically equivalent but perceptually different mathematical expressions, a start state (e.g., 121×144) and a goal state (e.g., 11×132×12). The goal of the game is to transform the start state into a goal state using permissible gesture

actions, such as moving, tapping, or decomposing numbers or expressions. Rewards (i.e., three clovers) are given if students solve the problem with the minimum required number of steps (also called "best step") to reach the goal state. As an example, the most efficient way to reach the goal state for the problem in Figure 1 is using three steps strategy: [Start state: 121×144] ‑ [Step 1: 11×11×144] ‑ [Step 2: 11×11×12×12] ‑ [Step 3: 11×132×12]. However, the number of clovers is reduced if the student exceeds the fewest steps possible to reach the goal state. Students can also reset the expression to the initial state, request hints, and reattempt the problems as many times as they want. As described in the previous section, our prior work has shown that the game is effective in improving students' algebraic understanding after controlling for their prior knowledge (Authors, 2019; 2022; 2023).

Participants and Research Procedure

The present study used data collected from a randomized control study conducted in Fall 2019, which consisted of 358 sixth and seventh-grade students from six middle schools located in the Southeastern United States. Of the 358 participants (male: 51%, female: 39%, not identified: 10%), the majority of the students were in 6th grade (85%), and the rest of the students were in 7th grade. Before starting the intervention, the students first took a pretest of their understanding of algebra and math anxiety. After that, they played the game individually at their own pace for four 30-minute intervention sessions during the regular math classes. On average, the students solved 97.4 distinct problems (SD = 34.2) across the four sessions. After completing the intervention, they took a posttest measuring understanding of algebra and math anxiety with isomorphic items to those given at the pretest. Both pretest and posttest math knowledge scores were measured using 11 items selected from two validated measures

(Rittle-Johnson et al., 2011; Star et al., 2015). The KR-20 coefficients of these 11 items were 0.69 at pretest and 0.76 at posttest, indicating an acceptable level of reliability.

Data pre-processing

Data construction and exploration. We first extracted log data (i.e., raw data) from the game database that automatically recorded all student mouse, touch, or keyboard actions with timestamps of each student action in the game. Using the raw data, we constructed aggregations of log files that resulted in 52 features for each student and each problem in the game (e.g., completion of the problem, number of steps, use of hints, time spent on each problem). Because the students played the game individually at their own pace, not all problems were attempted by every student, which led to many missing values in the dataset (Note that the students solved 97.4 problems on average across the four intervention sessions). In order to minimize the amount of missing data, we selected the problems that were attempted by at least 150 students, resulting in a subset of 98 problems. In addition, if there is an unattempted problem by the students, the students' behavior of solving that problem was imputed based on their problem-solving behaviors in the past. Specifically, we computed Z-scores (Z-score = actual value – mean / standard deviation) for each preliminary feature. Then, the students' behavior features for the unattempted n^{th} problem were calculated by taking the average of z-scores from the first problem to $(n\text{-}1)^{th}$ problem. In addition to the student behavioral features, we added two features collected through the assessment, pretest math anxiety scores, and posttest math anxiety scores. Students' math anxiety scores were measured using 13 items adapted from previously established measures (Ganley & McGraw, 2016). The inter-item reliabilities (Cronbach's α) of these items were 0.87 at pretest and 0.91 at posttest, showing a satisfactory level of reliabilities.

Furthermore, we included three additional features assessing the qualitative aspects of students' mathematical problem-solving in the game: mathematical expressions made by students, mathematical strategies used (e.g., calculating, commuting, decomposing, factoring), and the productivity of the first mathematical transformation (hereafter, productivity), which refers to whether or not the student's action moved them closer to the goal state of the problem. In particular, we hand-coded the mathematical strategies and the productivity using the log data collected in the game. Specifically, students' mathematical strategies on their first transformation were coded into a categorical variable with eight different categories: performing calculations (CALC), commuting a number or a letter (CM), decomposing a number (DC), factoring (FAC), creating an equivalent expression with subtraction (SUB), creating an equivalent expression with division (DIV), making an equivalent expression (EQV), and simplifying the expression (SMP). Productivity refers to whether or not students made a productive mathematical transformation towards the goal state in their first transformation, and a dummy variable was used to code (i.e., productive = 1, non-productive = 0). For example, for the problem with the start state "7+6+b+10" and the goal state "12+11+b", transforming the start state into"7+6+b+5+5" by decomposing 10 into "5+5" was coded as a productive first step because this transformation brought the student closer to the goal state. However, transforming the start state into "13+b+10" by adding 7 and 6 was coded as a non-productive first step because "13" did not bring the student closer to the goal state of the problem. The intra-correlation coefficients of the hand-coded data ranged between 0.91-0.98 for the mathematical strategies and 0.74-0.96 for productivity, indicating satisfactory levels of reliability.

Feature Selection. We conducted a feature selection based on several criteria to improve the performance of prediction models and reduce the computational cost of modeling.

We first examined the relationship between features and the outcome variable (i.e., posttest scores), and the features that have correlation coefficients with posttest scores greater than 0.9 (e.g., pretest math knowledge scores) and the features with a training-set variance lower than 0.05 were eliminated. We also used the Variance Inflation Factor (VIF) to detect multicollinearity between input variables (Chan et al., 2022). This selection process resulted in 32 features of students' behavioral patterns in the game, two features of student assessment, and three features of mathematical strategies. In sum, the total number of features included in prediction models is 37. Table 1 lists all features included in the prediction models as well as descriptions of each feature.

Re-sampling. Because our dataset consists of features with different scales of values, we used a set of techniques, such as Synthetic Minority Oversampling Technique (SMOTE) and outlier detection, to increase the effectiveness of handling the imbalanced dataset. SMOTE algorithm helps produce a more balanced dataset by creating a new dataset by oversampling observations from minority classes in the data. For outlier detections, we used the interquartile range (IQR) approach for continuous variables and histogram-based outlier scores for discrete variables (Barbato et al., 2011) and masked them for further statistical computations.

Evaluation (Split, Training, Tuning, and Testing). Finally, we used 80%-20% train-test data splitting to evaluate prediction models. In other words, 80% of the data was used for training, such as tuning the algorithm and parameters, and 20% was used to evaluate the models. We used a 10-fold cross-validation method for parameter turning. Figure 2 summarizes our data pre-processing and evaluation processes.

Data Analyses

In order to examine the algorithm that best predicts students' mathematics performance (RQ1), we compared seven different state-of-art machine learning algorithms

commonly used in learning analytics: Random Forest Regressor, Multilayer Perceptron (MLP), AdaBoost, Linear Lasso, Logistic Regression, Bagging Regressor, and Support Vector Regression (SVR).

Random Forest Regressor uses multiple decision trees on various sub-samples of the entire dataset, which allows for the use of averaging for improved prediction accuracy. MLP, also called neural networks, is a logistic regression classifier where the input data is transformed using a learned non-linear transformation. AdaBoost is a boosting algorithm used for both classification and regression that combines multiple weak classifiers to create one strong classifier. Linear Lasso creates simple models through the use of shrinkage (i.e., data is shrunk towards a center point). Logistic regression describes the relationship between a dependent binary variable and one or more nominal, ordinal, interval, or ratio-level independent variables. Bagging Regressor fits base regressors on random subsets of the original dataset and then aggregates their individual predictions to form a final prediction. Finally, SVR aims to find an appropriate line that has the least error rate, but it has more flexibility in terms of acceptance of error in comparison with ordinary least squares regression. We used three metrics for the evaluation of the performance of seven algorithms: Mean Square Error (MSE), Mean Absolute Error (MAE), and R^2. All data analyses were performed using seven packages (e.g., numpy, pandas, tqdm, math, seaborn, matplotlib, sklearn) in Python version 3.7.

Results

Correlation Analysis

Before building prediction models, we conducted a Pearson correlation analysis for the exploration of the data. Figure 3 represents a correlation matrix showing correlation

coefficients among the features included in the model, with a darker red indicating a stronger positive coefficient and a darker blue representing a stronger negative coefficient.

Regarding the correlation between extracted features and the posttest math knowledge scores (i.e., math performance), four student behavior features showed positive and higher correlations ($r \geq 0.5$) with posttest scores than other features: the trial of the problem (i.e., whether or not the student tried the problem; labeled "tried" in Figure 3), the number of clovers earned in the first attempt ("clover_first"), the validity of the first mathematical transformation on the first attempt (i.e., whether or not the first step on the first attempt is a mathematically valid step; "interaction_step_first"), and the validity of the first mathematical transformation on the last attempt ("interaction_step_last"). Contrary to these features, the number of resets made by the students ("num_reset" in Figure 3), the use of hints (i.e., whether or not the student used a hint; "use_hint"), the time taken to solve the problem ("time_interaction"), the number of errors made ("error") and the pretest math anxiety scores were negatively related to the posttest scores ($r \geq -0.4$).

RQ1: Comparison of Machine Learning Algorithms for Student Posttest Math Knowledge Scores Prediction

We examined how different features provided for the optimization of seven machine learning techniques influenced the precision of students' posttest scores prediction. We built prediction models using the 37 extracted features listed in Table 1. Table 2 presents the results of evaluation metrics (i.e., error measures and the accuracy) of models predicting student posttest scores.

In order to get a better perspective of the performance results, we also created bar graphs with two performance metrics, MSE, and R^2 (See Figure 4). As shown in Table 2 and Figure 4, RF showed the lowest values for MSE (2.858) and MAE (1.354), while MLP had the highest values of MSE (20.289) and MAE (2.405) among seven ML algorithms. Regarding prediction accuracy (R^2 scores), the model with the RF algorithm showed the highest accuracy score among seven models ($R^2 = 0.408$), which explained 40.8% of the variance in the posttest scores. Similar to the results of error metrics, MLP indicated the lowest accuracy score ($R^2 = -3.197$). Note that R-squared is negative when the model does not follow the trend of the data, which indicates that it fits worse than a horizontal line. We further examined whether there were statistical differences in evaluation metrics among the seven ML algorithms using Statistical Tests for Algorithms Comparison (STAC) (Rodríguez-Fdez et al., 2015). As the data did not meet the normality assumption and the number of datasets was larger than four, we performed a Friedman test following the recommendation of the developers of the tool. The results of the Friedman test indicated that the means of the two or more algorithms were statistically different ($\chi^2 = 11.26$, $p < .001$). We ran Holm post hoc analyses to determine which algorithms significantly differed from each other. The results showed that the RF algorithm showed significantly higher accuracy than linear lasso ($p < .001$) and logistic regression ($p < .001$). AdaBoost ($p = .018$) and SVR ($p = .048$) also

significantly outperformed linear lasso. In sum, RF showed the best performance in predicting student posttest math knowledge scores, while MLP showed the lowest accuracy and highest error values for our dataset.

RQ2: Students' Math Performance (Posttest scores) Prediction

For the second research question, we examined the relations between student features and their posttest scores using the prediction model with RF as it outperformed the other six ML algorithms. Figure 5 presents the feature importance in RF regressor for student posttest scores in ascending order. The x-axis represents relative feature importance.

As shown in Figure 5, the most influential feature in predicting the posttest scores was the "interaction_step_first," which indicates whether or not a student made a mathematically valid transformation on their first problem-solving. The second most influential feature was "use_hint", which refers to whether or not the students requested a hint. Although the use of hints was one of the important features in predicting the posttest scores, the direction of the association was negative. In other words, the students who requested hints more frequently tended to achieve lower posttest math knowledge scores. The third most influential feature was "interaction_step_last" indicating the validity of the first mathematical transformation on the last attempt. Lastly, while two student assessment features (pre-math anxiety scores, post math anxiety scores) showed relatively higher importance values, three features of math strategies (e.g., mathematical strategies used, the productivity of solution strategies) had relatively lower importance values than other features in predicting the posttest scores.

Discussion

In this paper, we investigated how different ML algorithms influenced the precision of predicting middle-school students' math knowledge scores in an online interactive

mathematics game. While many of the studies in the field of game learning analytics tend to use a simple aggregation of log files, we used aggregations of student actions logged within the database (e.g., number of attempts, time taken to solve the problem), and hand-coded, qualitative aspects of students' problem-solving strategies (e.g., mathematical strategies used, the productivity of the mathematical transformations), as well as assessment (e.g., math anxiety) to examine the relationship between student in-game metrics and their math performance.

For the first research question, we compared the performance of widely used seven supervised ML algorithms in predicting students' posttest math knowledge scores and identified the ML algorithm that produced the best performance. The results revealed that Random Forest (RF) outperformed the other six algorithms (MLP, AdaBoost, Linear Lasso, Logistic Regression, Bagging Regressor, SVR) with the highest accuracy scores and the lowest error metrics for our dataset, while MLP showed the lowest accuracy and highest error values. Our results confirm that the RF algorithm performs well on the imbalanced dataset and shows good handling of outliers (Gupta, 2020), as many of the features included in our prediction models were positively skewed. A possible explanation for the worst performance of MLP may be due to the relatively small amount of data as it requires a large amount of data for training. Although many studies in the field tend to use one specific ML algorithm to predict outcome variables (e.g., Chen et al., 2020; Emerson et al., 2020; Hooshyar et al., 2017), these varying performances of prediction models depending on ML algorithms show the importance of identifying the algorithm that suits the data well to improve the accuracy of prediction models (Alonso-Fernández et al., 2020). Indeed, our results indicated the effectiveness of the prediction model in estimating student posttest scores, even after

excluding the pretest math knowledge scores that were highly correlated with the outcome variable.

For the second research question, using a data-driven approach, we investigated what features (i.e., student in-game behaviors, mathematical strategies, math anxiety scores) were associated with student posttest math knowledge scores. Our earlier work using variables of interest showed that students' in-game progress (Author, 2019; 2022b) was significantly and positively related to their posttest math knowledge scores. The current study revealed that making mathematically valid actions on their first problem-solving was the most influential predictor for math performance out of 37 features in the prediction model. In other words, if a student made a mathematically valid or accurate transformation without making any errors (e.g., adding before multiplying in 3+4*5) on their first attempt, the student was more likely to receive a higher posttest math knowledge score.

In order to make mathematically valid actions on their first attempts, students need to take the time to recognize the underlying structure of the equation and apply the correct procedure, rather than rushing into problem-solving. For example, in order to transform the start state “4*6*c*24*16” to the goal state “96*96*c” in the game, students must notify the multiplicative relations between “4*24” and “96”, and “6*16” and “96”. Our results seem consistent with our earlier work, which found that students’ pre-solving pause time to think about the problem was positively associated with higher strategy efficiency (Authors, 2022a). Together, our finding implies that noticing the underlying mathematical structure of the equations prior to problem-solving may play an important role in students' mathematical understanding. Thus, teachers and instructional designers may consider instructional strategies that encourage students to

identify underlying patterns or structures of the equations before solving math problems rather than rushing into problem-solving.

The second most influential feature in predicting students' mathematical understanding was requesting hints in the game. Specifically, we found that the students who requested hints more frequently tended to have lower posttest scores. Similarly, one study (McLaren et al., 2022) investigated the effects of on-request hints in a digital mathematical learning game and found that students in the no-hints condition performed significantly better than those in the on-request hints condition. Their follow-up analysis of the student learning curve revealed that the students in the no-hint condition performed worse than those in the hint condition at the beginning. However, the students in the no-hint condition gradually reduced their errors by constructing their own understanding, which led to more robust learning than the students in the hints condition in the posttest. Together, these results imply that if students acquire the necessary knowledge or skills, fading hints may be more helpful to their learning than providing hints for all problems in the game. Thus, a further study with more focus on the usage of hints in the game is suggested.

Limitations and Directions for Future Work

Finally, a number of limitations need to be considered. While we included three different types of features in the prediction models (i.e., student in-game behaviors, student assessment, mathematical strategies), the features other than in-game behaviors (i.e., student math anxiety scores, mathematical strategies) had relatively low importance in predicting posttest scores (i.e., students' mathematical understanding), which seems to be consistent with other research that found data other than game logs were not predictive of posttest scores (Nguyen et al., 2020; Shute et al., 2015). Although

they had low associations with the cognitive learning outcome, they might be related to other outcome variables, for example, enjoyment (Nguyen et al., 2020). In addition, we did not include student demographics and problem-related features (e.g., difficulty of each problem) in the prediction models. Thus, further research should be done to investigate the relationship between these features and other outcome variables. Although the RF algorithms identified the feature importance in predicting the posttest scores, it is relative to features and does not show the statistical relationship between features and outcome variables (Ziegler & König, 2014). Lastly, many of the features in our prediction models are game-specific, so the results should be interpreted cautiously (Serrano-Laguna et al., 2014). Further work is required to replicate these analysis methods to a different or larger dataset in order to validate our findings.

Conclusion

Although there has been extensive research in the field of game learning analytics, many studies tend to focus on using simple aggregations of student actions or applying a few simple techniques rather than choosing the best model after evaluating multiple algorithms. The present study addresses this research gap by using the best prediction model after evaluating multiple algorithms and encompassing both quantitative and qualitative aspects of students' behaviors in the game. The study can serve as guidance for researchers on how to compare and evaluate prediction models using game log data. In terms of instructional practice, our results suggest that teachers should consider instructional strategies for making students notice the pattern or structure of the problems rather than rushing into problem-solving, to improve students' mathematical understanding.

Acknowledgments

Blinded for Review.

References

Authors (2019).

Authors (2022a).

Authors (2022b).

Authors (2023).

Alonso-Fernandez, C., Calvo-Morata, A., Freire, M., Martinez-Ortiz, I., & Fernández-Manjón, B. (2019). Applications of data science to game learning analytics data: A systematic literature review. *Computers & Education, 141*, 103612. https://doi.org/10.1016/j.compedu.2019.103612

Alonso-Fernández, C., Calvo-Morata, A., Freire, M., Martínez-Ortiz, I., & Fernández-Manjón, B. (2020). Evidence-based evaluation of a serious game to increase bullying awareness. *Interactive Learning Environments*, 1-11. https://doi.org/10.1080/10494820.2020.1799031

Asbell-Clarke, J., Rowe, E., Almeda, V., Edwards, T., Bardar, E., Gasca, S., ... & Scruggs, R. (2021). The development of students' computational thinking practices in elementary-and middle-school classes using the learning game, Zoombinis. *Computers in Human Behavior, 115*, 106587. https://doi.org/10.1016/j.chb.2020.106587

Barbato, G., Barini, E. M., Genta, G., & Levi, R. (2011). Features and performance of some outlier detection methods. *Journal of Applied Statistics, 38*(10), 2133-2149. https://doi.org/10.1080/02664763.2010.545119

Cano, A. R., Fernández-Manjón, B., & García-Tejedor, Á. J. (2018). Using game learning analytics for validating the design of a learning game for adults with

intellectual disabilities. *British Journal of Educational Technology, 49*(4), 659-672. https://doi.org/10.1111/bjet.12632

Chan, J. Y. L., Leow, S. M. H., Bea, K. T., Cheng, W. K., Phoong, S. W., Hong, Z. W., & Chen, Y. L. (2022). Mitigating the multicollinearity problem and its machine learning approach: a review. *Mathematics, 10*(8), 1283.

Chen, F., Cui, Y., & Chu, M. W. (2020). Utilizing Game Analytics to Inform and Validate Digital Game-based Assessment with Evidence-centered Game Design: A Case Study. *International Journal of Artificial Intelligence in Education, 30*(3), 481-503. https://doi.org/10.1007/s40593-020-00202-6

Cheng, Y. W., Wang, Y., Cheng, I. L., & Chen, N. S. (2019). An in-depth analysis of the interaction transitions in a collaborative augmented reality-based mathematic game. *Interactive Learning Environments*, *27*(5-6), 782-796. https://doi.org/10.1080/10494820.2019.1610448

Ellis, C. (2021). *When to use LASSO*. Crunching the Data. https://crunchingthedata.com/when-to-use-lasso/

Emerson, A., Cloude, E. B., Azevedo, R., & Lester, J. (2020). Multimodal learning analytics for game-based learning. *British Journal of Educational Technology, 51*(5), 1505-1526. https://doi.org/10.1111/bjet.12992

Ganley, C. M., & McGraw, A. L. (2016). The development and validation of a revised version of the math anxiety scale for young children. *Frontiers in Psychology*, 7, 1181. https://doi.org/10.3389/fpsyg.2016.01181

Gupta, S. (2020, February 28). *Pros and cons of various machine learning algorithms*. Towards Data Science. https://towardsdatascience.com/pros-and-cons-of-various-classification-ml-algorithms-3b5bfb3c87d6

Hooshyar, D., Yousefi, M., Wang, M., & Lim, H. (2018). A data-driven procedural-content-generation approach for educational games. *Journal of Computer Assisted Learning, 34*(6), 731-739. https://doi.org/10.1111/jcal.12280

Karamizadeh, S., Abdullah, S. M., Halimi, M., Shayan, J., & javad Rajabi, M. (2014, September). Advantage and drawback of support vector machine functionality. In *2014 International Conference on Computer, Communications, and Control Technology* (I4CT) (pp. 63-65). IEEE.

Kerr, D. (2015). Using data mining results to improve educational video game design. *Journal of Educational Data Mining, 7*(3), 1-17.

McLaren, B. M., Richey, J. E., Nguyen, H., & Hou, X. (2022). How instructional context can impact learning with educational technology: Lessons from a study with a digital learning game. *Computers & Education*, *178*, 104366. https://doi.org/10.1016/j.compedu.2021.104366

Nguyen, H.A., Hou, X., Stamper, J., & McLaren, B. M. (2020). Moving beyond test scores: Analyzing the effectiveness of a digital learning game through learning analytics. In A. N. Rafferty, J. Whitehill, C. Romero, & V. Cavalli-Sforza (Eds.), *Proceedings of the 13th International Conference on Educational Data Mining* (pp.487-495*)*. ACM.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., & Vanderplas, J. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research, 12*, 2825-2830.

Rittle-Johnson, B., Matthews, P. G., Taylor, R. S., & McEldoon, K. L. (2011). Assessing knowledge of mathematical equivalence: A construct-modeling

approach. *Journal of Educational Psychology*, *103*(1), 85. https://doi.org/10.1037/a0021334

Rocca, J. (2019, April 22). *Ensemble methods: bagging, boosting and stacking*. Towards Data Science. https://towardsdatascience.com/ensemble-methods-bagging-boosting-and-stacking-c9214a10a205

Rodríguez-Fdez, I., Canosa, A., Mucientes, M., & Bugarín, A. (2015, August). STAC: a web platform for the comparison of algorithms using statistical tests. In *2015 IEEE international conference on fuzzy systems (FUZZ-IEEE)* (pp. 1-8). IEEE.

Serrano-Laguna, Á., Torrente, J., Moreno-Ger, P., & Fernández-Manjón, B. (2014). Application of learning analytics in educational videogames. *Entertainment Computing*, *5*(4), 313-322. https://doi.org/10.1016/j.entcom.2014.02.003

Shrestha, D. L., & Solomatine, D. P. (2006). Experiments with AdaBoost. RT, an improved boosting scheme for regression. *Neural computation, 18*(7), 1678-1710.

Shute, V.J., D'Mello, S., Baker, R., Cho, K., Bosch, N., Ocumpaugh, J., Ventura, M., & Almeda, V., 2015. Modeling how incoming knowledge, persistence, affective states, and in-game progress influence student learning from an educational game. *Computers & Education*, *86*, 224-235. https://doi.org/10.1016/j.compedu.2015.08.001

Star, J. R., Pollack, C., Durkin, K., Rittle-Johnson, B., Lynch, K., Newton, K., & Gogolen, C. (2015). Learning from comparison in algebra. *Contemporary Educational Psychology*, *40*, 41-54 https://doi.org/10.1016/j.cedpsych.2014.05.005

Tomasevic, N., Gvozdenovic, N., & Vranes, S. (2020). An overview and comparison of supervised data mining techniques for student exam performance prediction. *Computers & Education*, *143*, 103676. https://doi.org/10.1016/j.compedu.2019.103676

Yang, K. H. (2017). Learning behavior and achievement analysis of a digital game-based learning approach integrating mastery learning theory and different feedback models. *Interactive Learning Environments*, *25*(2), 235-248. https://doi.org/10.1080/10494820.2017.1286099

Ziegler, A., & König, I. R. (2014). Mining data with random forests: current options for real-world applications. Wiley Interdisciplinary Reviews: *Data Mining and Knowledge Discovery, 4*(1), 55-63.

Table 1

Features Included in the Final Prediction Models

Features	Descriptions
Student in-game behavior (extracted from the log data; 32 features)	
tried	trial of the problems (i.e., whether or not the student tried the problem)
completions	completion of the problem (i.e., whether or not the student completed the problem)
num_visit	number of visits to the problem
reset	whether or not the student reset the problem (the user clicks reset to restart the problem)
num_reset	number of resets
num_attempts	number of attempts
num_gobacks	number of reattempts (the user goes back and re-completes the problem)
use_hint	hint usage (i.e., whether or not the student used the hint)

clover_first	number of clovers earned in the first attempt
clover_last	number of clovers earned in the last attempt
time_interaction	time taken to solve the problem
avg_time_per_step	average amount of time to take one step
time_first	total amount of time the user spends on their first attempt
time_last	total amount of time the user spends on their last attempt
time_interaction_first_percent	percent of time the user spends on pausing during the first attempt
time_interaction_last_percent	percent of time the user spends on pausing during the last attempt
interaction_step_first	the validity of the first mathematical transformation on the first attempt (i.e., whether or not the first step on the first attempt is a valid step)
interaction_step_last	the validity of the first mathematical transformation on the last attempt
num_steps	number of steps made
user_first_step	total number of steps (i.e., mathematical transformations) on the first attempt
user_last_step	total number of steps on the last attempt
first_efficiency	step efficiency on the first attempt (calculated by the ratio of the "best step" and the "user_first_step")
last_efficiency	step efficiency on the last attempt (calculated by the ratio of the "best step" and the "user_last_step")
first_more_step	the number of steps exceeded the best step (calculated by the difference between "best step" and "user_first_step")
last_more_step	the number of steps exceeded the best step (calculated by the difference between "best step" and "user_first_step")
error	whether or not the student made the error
total_error	total number of errors
first_error	total number of errors on the first attempt
last_error	total number of errors on the last attempt
keypad_error	total number of keypad errors
shaking_error	total number of shaking errors
snapping_error	total number of snapping errors

Student assessment (2 features)	
pre_math_anxiety	pretest math knowledge scores
post_math_anxiety	posttest math anxiety scores
Mathematical strategies (3 features)	
math_expression	mathematical expressions made by a student
math_strategies	math strategies used to solve the problem (e.g., calculating, decomposing, commuting);
productivity	productivity of mathematical transformation (i.e., whether or not the student's action moved them closer to the goal state of the problem)

Table 2

Prediction Error Measures and Accuracy of Models Predicting Student Math Knowledge Scores

Algorithms	MSE	MAE	R^2
RF	2.858	1.354	0.408
Bagging Regressor	2.968	1.366	0.385
AdaBoost	3.174	1.438	0.343
SVM	3.466	1.475	0.282
Linear Lasso	3.606	1.517	0.253
Logistic	4.937	1.770	-0.021
MLP	20.289	2.405	-3.197

Note. R-squared is negative when the model does not follow the trend of the data, which indicates that it fits worse than a horizontal line.

Figure 1

A Sample Problem and Students' Actions in the Game

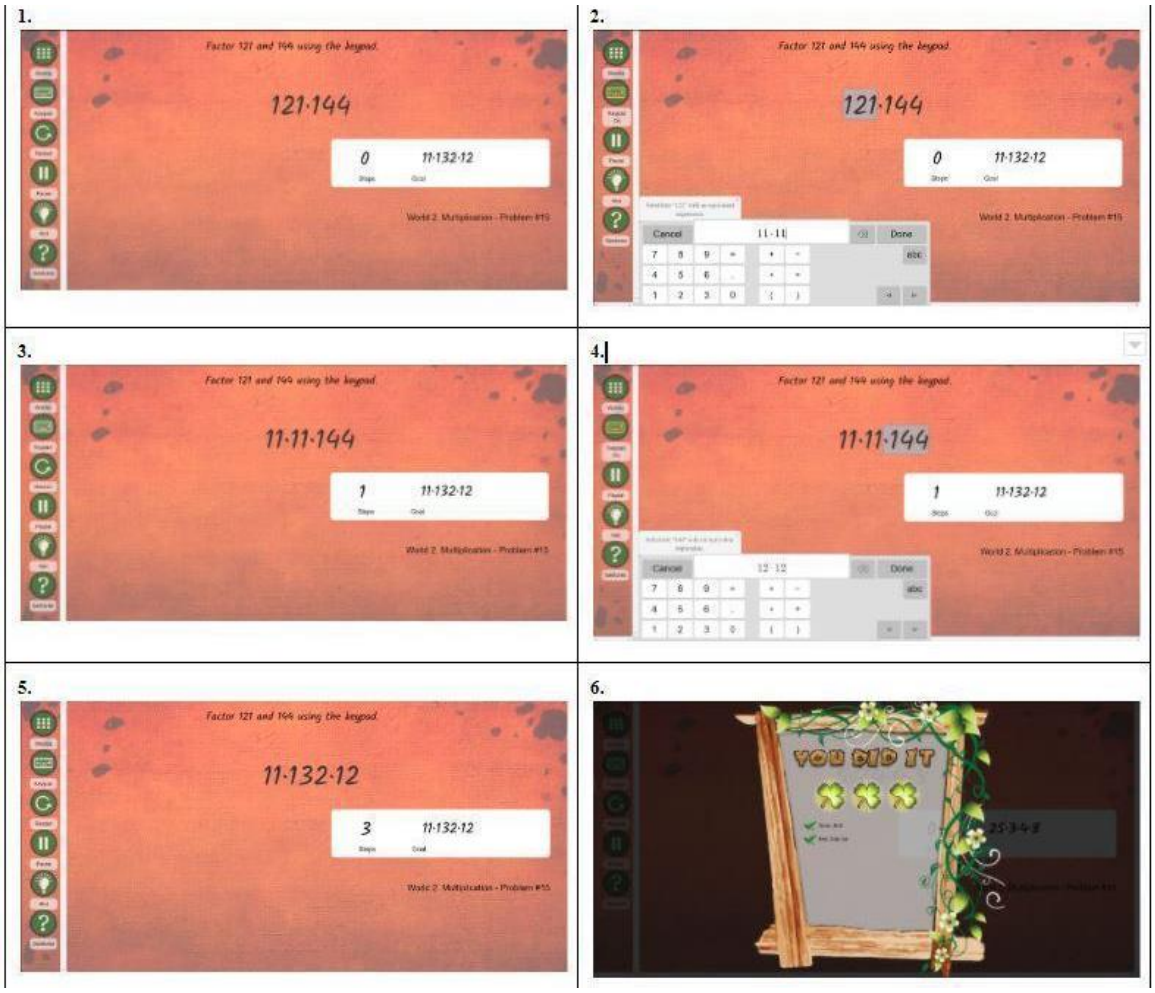


Figure 2

Data Preprocessing and Evaluation Process

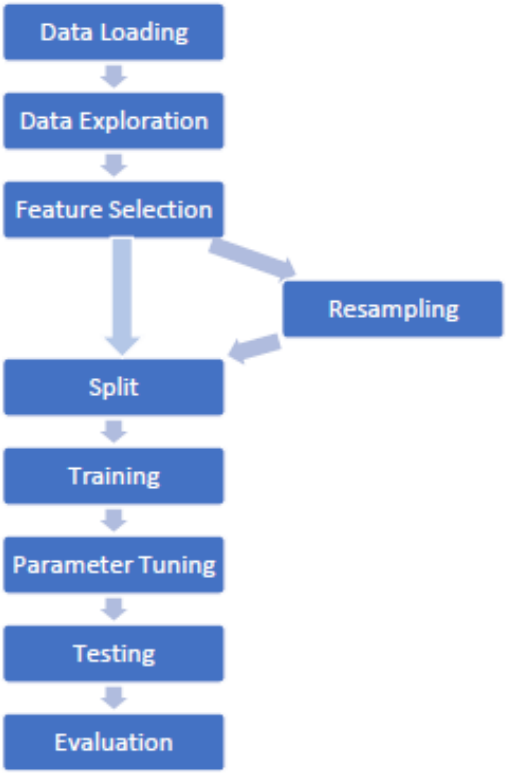
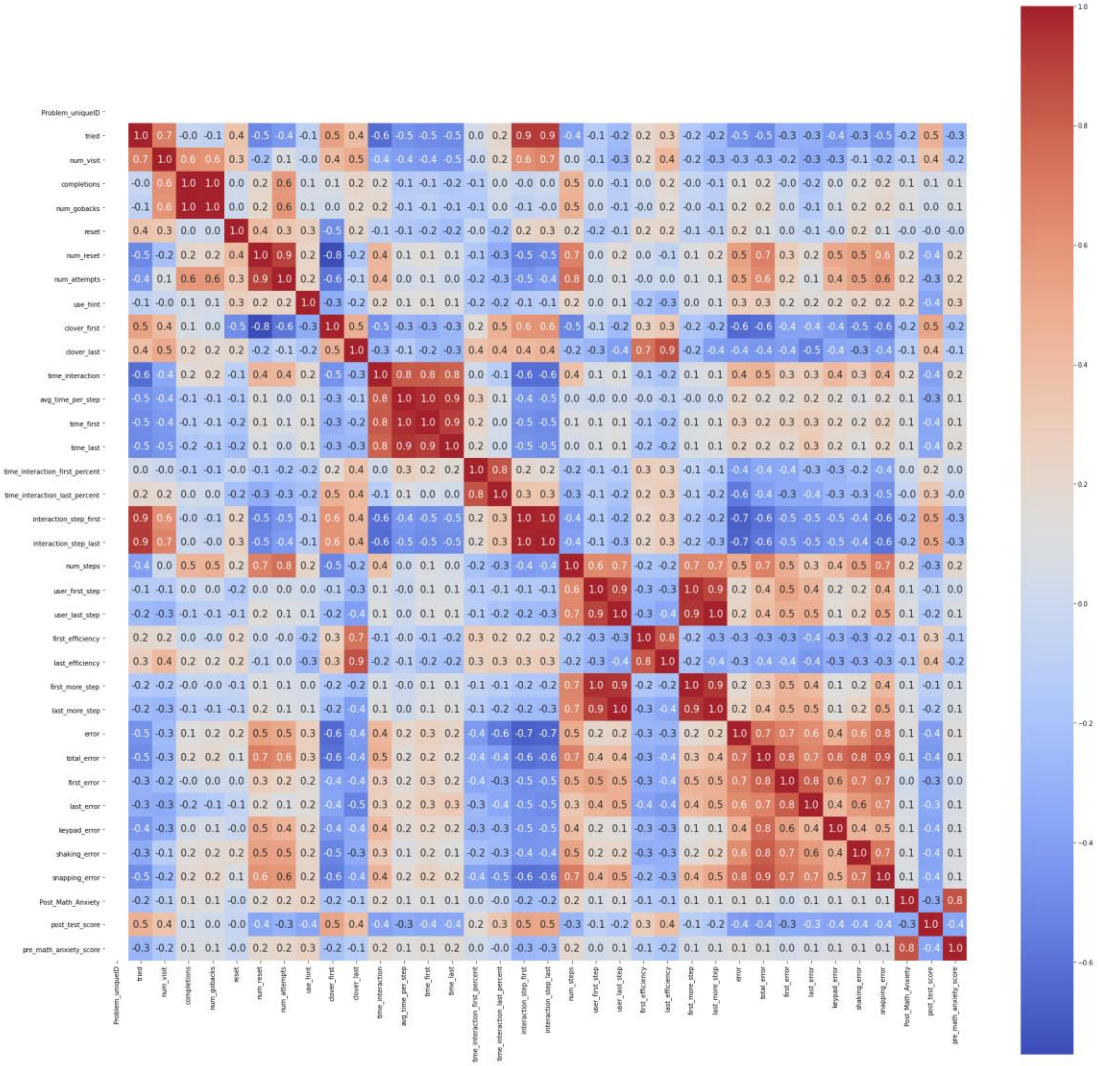


Figure 3

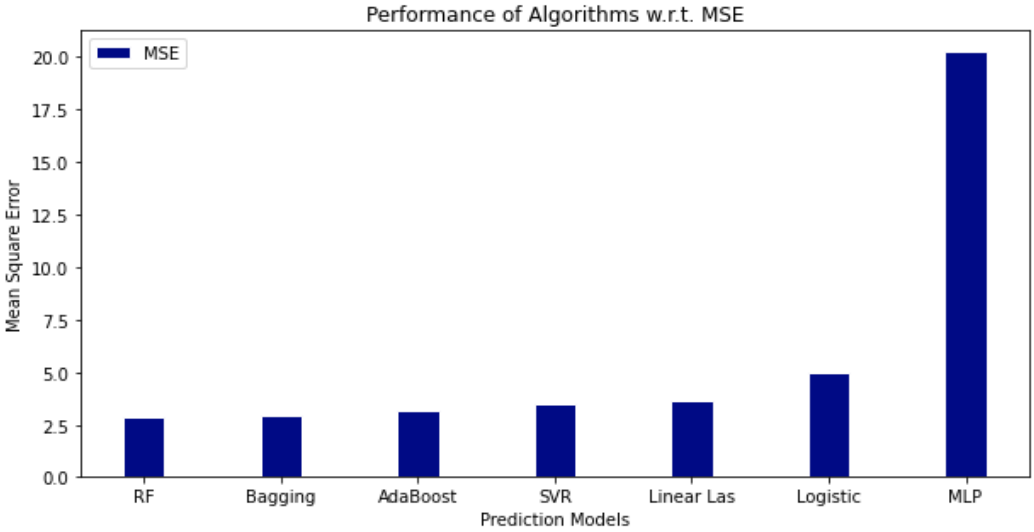
Correlations among Students' Behavioral Features, Assessment features, and Posttest Math Knowledge Scores (for full image: https://tinyurl.com/237z4lfc)



Note. In the correlation matrix, a darker red indicates a stronger positive coefficient, and a darker blue represents a stronger negative coefficient.

Figure 4

MSE and R-squared Score for Each Regression Algorithm



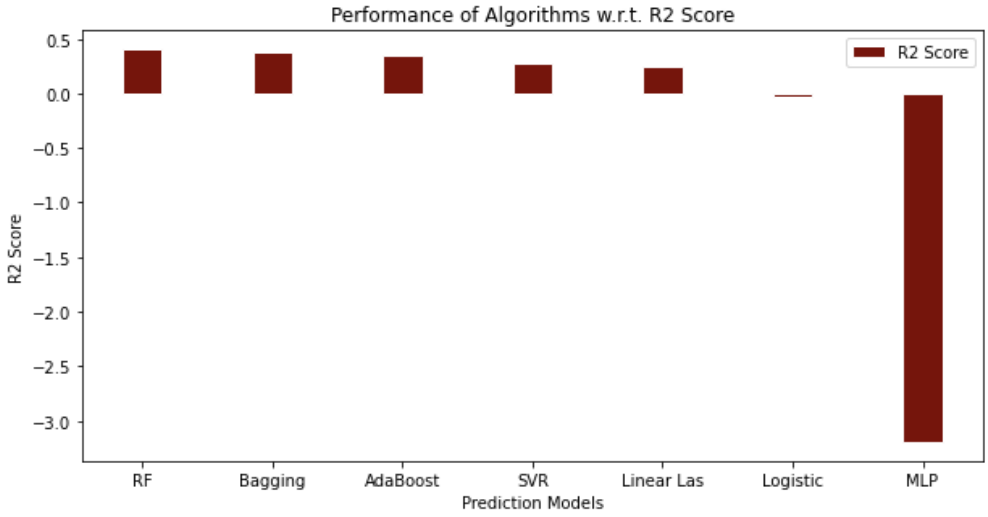
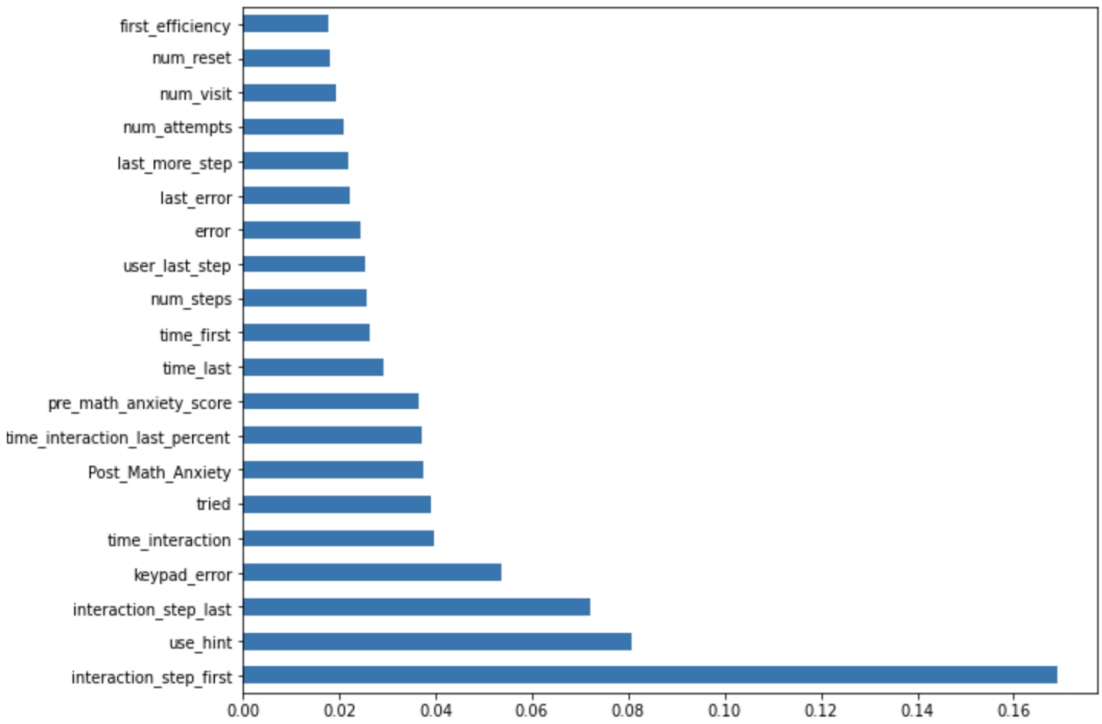


Figure 5

The Results of the RF Prediction Model (Feature Importance)


first_efficiency
num_reset
num_visit
num_attempts
last_more_step
last_error
error
user_last_step
num_steps
time_first
time_last
pre_math_anxiety_score
time_interaction_last_percent
Post_Math_Anxiety
tried
time_interaction
keypad_error
interaction_step_last
use_hint
interaction_step_first
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14
0.16