Process Mining Combined with Expert Feature Engineering to Predict Efficient Use of Time on High-Stakes Assessments

Nathan A. Levin
Teachers College, Columbia University
nal2163@tc.columbia.edu

The Big Data for Education Spoke of the NSF Northeast Big Data Innovation Hub and ETS co-sponsored an educational data mining competition in which contestants were asked to predict efficient time use on the NAEP 8th grade mathematics computer-based assessment, based on the log file of a student's actions on a prior portion of the assessment. In this work, a combined approach of process mining and expert feature engineering was used to build a large set of features that were then trained with an Extreme Gradient Boosting machine learning model to classify students based on whether they would use their time efficiently. Predictions were evaluated throughout the competition on half of a hidden data set and then the final results were based on the second half of the hidden data set. The approach used here earned the top score in the competition. The work presented elaborates on the combined technique for analyzing computer-based assessment log-file data with the hope that this approach will offer valuable insights for future predictive model building in educational data mining.

Keywords: process mining, educational data mining, computer-based assessment, extreme gradient boosting

1. INTRODUCTION

Standardized computer-based testing is rapidly becoming a ubiquitous tool both for summative assessment as well as formative self-evaluation (Pechenizkiy et al., 2009). Up to the moment of the test, a teacher may be the one with the most direct influence on a student's performance; however, the moment the test begins the teacher is removed from the equation and relegated to the sideline. After an exam, a teacher can review the results and reverse engineer what might have happened, but retrospective knowledge is imperfect at best. If a student runs out of time on an exam, a teacher can ask the student which questions took the most time, but understanding why those questions took additional time and where the student might have felt anxiety is much more difficult for a teacher to ascertain. The nature of microlevel clickstream data is well suited to supporting a teacher in providing real-time intervention (Fischer et al., 2020). Teaching time management skills is considered one of the most impactful ways to improve student performance on high-stakes assessments (Gulek, 2003), but the generally accepted best practices involve preparation as opposed to intervention.

With the advent of computer-based assessments, we now have access to real-time student behaviors in the form of clickstream data representing each action taken by a student during an exam (Bannert et al., 2014; Greiff et al., 2015). The immediacy of this information creates a unique window of opportunity for teachers to provide pertinent support to students, which is a key goal of learning analytics (Macfadyen & Dawson, 2010). Specific to this work, the granularity of clickstream data provides insights into ineffective and effective test-taking behaviors.

To push the field of educational data mining forward, the Big Data for Education Spoke of the NSF Northeast Big Data Innovation Hub and Educational Testing Service organized a competition in 2019 to engage leading researchers worldwide to develop metrics for measuring students' test-taking activities and attempt to predict effective and ineffective test-taking time management behavior. Effective time management increases assessment validity because students accurately demonstrate their knowledge without running out of time (Ellis & Ryan, 2003). Accurately predicting time management is an invaluable piece of insight for teachers striving to reduce test anxiety, which may interfere with optimal test performance (Stenlund et al., 2017).

The data set published for the competition consists of the student log data for all actions taken by a group of students in two blocks of a math test, Block A and Block B. The goal of the competition was to predict whether students would spend their time efficiently in Block B, based on the actions taken during Block A. Spending time efficiently was defined by two criteria: 1. Not exceeding the time limit for Block B. 2. Spending a "reasonable amount of time" on each problem, defined as being above the 5th percentile in terms of time taken on each problem.

The input to each predictive model is a set of features derived from the clickstream data log of every action taken by a student in Block A. The output of the model was a binary value for each student indicating whether they would use their time efficiently in Block B (TRUE) or not use their time efficiently in Block B (FALSE).

Different approaches were taken by the top entries in the competition. The third-place approach constructed many features (>4,000) using domain knowledge and automated feature engineering methods. The second-place team constructed features based on process mining, applied a genetic algorithm-based feature selection and modeling technique, and collated the predictions from multiple models into an ensemble model that generated a single prediction.

The model presented in this paper earned first place in the competition by achieving the highest score based on the second half of the hidden data set. However, the model was not the top-scoring model on the first half of the hidden data set; an inconsistency that will be addressed in the further research section of this paper. The features were developed using sequential process mining techniques combined with expert feature engineering conducted manually on the student log data. Hybrid paradigms for feature engineering that utilize data mining techniques as well as expertly engineered features have been shown to achieve improved performance over models built using either feature engineering technique alone (Paquette et al., 2014). The XGBoost Regressor model was applied to the final feature set (~330) to generate the final predictions, which scored an AUC of 0.658 and a Kappa of 0.228.

This work was guided by the following research questions (RQs):

> **RQ1:** Can efficient time use be accurately predicted by the features generated through process mining and expert feature engineering?
> **RQ2:** Are the important features (as measured by weighting in the model) broadly interpretable?

We hypothesize that efficient time use (as defined for this competition) can be accurately predicted by the features generated through process mining and expert feature engineering and that the most important features will be broadly interpretable by educators as well as researchers.

2. Related Work

2.1. Microlevel Data Mining

This work falls under the broad tent of research that leverages data mining techniques in education for prediction (Baker & Siemens, 2014). Baker and Siemens (2014) divide predictive work in Educational Data Mining between work that either infers a latent construct or predicts future outcomes. The work presented here is the latter, as we seek to develop a model able to predict efficient time use on both the training data set as well as future computer-based assessment log data. Fischer et al. (2020) categorize big data mining in education as either microlevel, mesolevel, or macrolevel. Clickstream analysis is considered microlevel mining and is often used to "personalize and enhance instruction and learning" (Fischer et al., 2020, p.2).

Considerable research has been conducted to detect and predict comprehension, identify affective states, and understand self-regulated learning strategies (Fischer et al., 2020). Our research most closely aligns with understanding self-regulation, albeit is uniquely applied in the context of a high-stakes assessment. Microlevel data mining of a computer-based assessment offers a way to better understand and design the options available to test-takers and the choices they make as they self-regulate their journey through an assessment (Roll & Winne, 2015). Data mining has been used to understand efficient time use and identify procrastination (Park et al., 2018). However, Park et al.'s (2018) research was conducted using student interactions with an LMS throughout an entire semester.

Self-regulation research tends to be highly context specific. Research has been conducted with promising results in a variety of LMSs and MOOCS from Betty's Brain to Cognitive Tutors (Fischer et al., 2020). However, as Baker points out in Baker Learning Analytics Prize 1, most of these models and their conclusions stay within the confines of their learning system (Baker, 2019). The research presented in this paper has a similar limitation because it was conducted within the specific context of the NAEP computer-based mathematics assessment. Computer-based math assessments are becoming increasingly commonplace and computer-based assessment design tends to remain fairly consistent across platforms (Aksoy et al., 2019).

2.2. PROCESS MINING

Goldhammer and Zehner (2017) suggest that “operationally, process data can be regarded as the empirical data reflecting the course of working on a test item” (p.128). The framing of process data as empirical data supports our hypothesis that process behaviors will reveal students’ more general test-taking efficiency because process mining transmutes log-file data into empirical evidence. Process mining offers a unique lens for uncovering the common navigational patterns across a population of students (Bogarin et al., 2014; Pechenizkiy et al., 2009) as well as potentially creating more interpretable machine learning models because common processes can be extracted from event logs and teachers can readily interpret those processes. Bogarin et al. (2018) describes process mining as “a bridge between data mining (DM) and process modeling and analysis” (p.1). In past work, process mining has been used primarily as a tool to identify “patterns (strategies) repeated with relatively high frequency … that might correspond to … learning sessions of a set of learners” (Nesbit et al., 2007, p.6). Our research takes the lead from Juhaňák et al. (2019) in that it uses process mining to analyze student test-taking behavior, but it presents a unique focus on revealing and predicting effective time management techniques in a large-scale testing environment.

3. METHOD

3.1. DATA

As previously mentioned, this work uses log-file data representing the actions taken by students during the 8th NAEP mathematics assessment in the 2016-2017 academic year. Students completed two blocks of math problems Block A and Block B. Each block contained a set number of problems and students had a 30-minute time limit to complete the problems in each block.

Each row of the data represents a single action taken by a student, such as the student clicking the mouse or typing on the keyboard. Each row of the data contains the following information:

STUDENTID:	A unique identifier for each student
Block:	The block that the action happened in
AccessionNumber:	A unique identification of a problem/item
ItemType:	The type of the item
Observable:	The type of action the student took
ExtendedInfo:	Additional information on the student action
EventTime:	The timestamp of when the action was taken

The goal was to predict efficient use of time on Block B based on log data provided for the first 10, 20, and 30 minutes of actions. To develop a model attuned to each of these data sets, one of the first processing steps was to split the training data into 10, 20, and 30 minutes of actions. This initial clustering was based on the assumption that by separating the data into distinct groups, the model would be more valid and less biased because the relative importance of different features fluctuates depending on the cluster being trained upon (Makhlouf & Mine, 2020). These three data sets were used separately for both training and validation of separate models. A total of 1232 students were present in the training data set, composed of a total of 438,292 rows of actions.

The test data set consisted of hidden data sets for 10, 20, and 30 minutes of actions. The test data sets contained 411, 411, and 410 students, respectively. The test data set target labels were withheld throughout the competition, and subsequently released at the end of the competition. Therefore, the performances of the models are further validated for having been trained with the test data set completely unseen. Although the questions in Block B were different from those in Block A, we can assume that they cover a similar breadth and depth of material.

The target variable is a binary classifier indicating whether a student used their time efficiently on Block B of the exam. In the training data, 744 students had a target value of "TRUE," indicating they used their time efficiently on Block B; the remaining students (438) had a target value of FALSE. The imbalance between classes is not enough to justify mitigation through a sampling method, although this may be a course of action for future work if carrying this model forward on similar problems.

3.2. Model Selection

XGBoost is an optimized distributed gradient boosting library, chosen for its speed, flexibility, and open source integration with the scikit-learn machine learning package in Python. Like other gradient boosted tree algorithms, XGBoost relies on optimizing a set (an ensemble) of classification/decision trees by iteratively adding trees ("boosting") in a way that decreases the gradient of a specified loss function. The specific implementation of gradient tree boosting within XGBoost is beyond the scope of this paper, but suffice to say it includes a gradient boosted descent tree regressor called XGBRegressor, which was used as the predictor in this model. The data pre-processing and feature engineering was done in Python using the Pandas data manipulation and analysis package. All code and links to the dataset used in this work are freely available on GitHub.[1]

3.3. Evaluation Metric

To compare all models built for the competition, the evaluation criterion was defined by the metric below combining AUC and Kappa.

$$\mathit{AdjustedAUC} = \begin{cases} 0 & , \mathit{if\ AUC} < 0 \\ 2 * (\mathit{AUC} - 0.5) & , \mathit{otherwise} \end{cases}$$

$$\mathit{AdjustedKappa} = \begin{cases} 0 & , \mathit{if\ kappa} < 0 \\ \text{kappa} & , \mathit{otherwise} \end{cases}$$

$$\mathit{AggregratedScore} = \mathit{AdjustedAUC} + \mathit{AdjustedKappa}$$

[1] https://github.com/nal2163/NAEPDataMining

AUC (Area Under the Curve) compares the false positive rate to the true positive rate of the model. A value of 0.5 would indicate a model performing only as well as random chance would perform, which motivated the use of Adjusted_AUC for the first component of the evaluation metric. Cohen's Kappa, which compares observed accuracy to expected accuracy of a random model, was the foundation of the second component of the evaluation metric. In the case of Kappa, a value below 0 indicates worse than random chance, hence the adjustment above. By combining the two metrics we obtain a balanced evaluation of whether a model is classifying students correctly while avoiding classifying students incorrectly.

In the training of the XGBoost model, the package allows for the use of a custom scoring function during cross-validation. By writing the evaluation metric above as a function, it could be used to evaluate the hyperparameters of the model, as discussed further in section 3.5 below.

3.4. Feature Engineering

The features for the model were generated through multiple iterations of both expert feature engineering and process mining. Due to the robust nature of the XGBoost model, the initial approach was to generate as many features as possible before paring them down based on importance to the model. The complete list of features varied for the 10-, 20-, and 30-minute models with 272, 334, and 368 features, respectively. The increasing number of features was a factor of the additional data in the student log data for 30 minutes of actions, as some of the features sought to capture expanding features such as the number of actions taken each minute. For example, in the 30-minute feature set, there is an action rate for each minute resulting in 30 features as opposed to 10 and 20, as in the other feature sets.

3.4.1. Expert Feature Engineering

Expert feature engineering of the log files generated features that can be categorized into frequency-related and time-related features as suggested by Chen and Cui (2020). Frequency-related features represent the number of times a student performed specific actions on the exam. In this case, the word action is loosely defined to capture a multitude of behaviors from revisiting questions to clicks per minute. The time-related features indicate the average and total amount of time a student spent on specific activities during the assessment. The time-related features are particularly important because they have been found in the past to help distinguish solution-finding behavior from random guessing, which in this case would be classified as the difference between effective and ineffective time management (Juhaňák et al., 2019). The third category of features added to the model were basic statistical descriptors of feature sets, such as standard deviation of time spent on questions and maximum and minimum number of clicks in 1-minute.

3.4.2. Process Mining

Process mining was used to generate additional features in the form of common sequences of actions taken by students. Fluxicon's Disco was used to create a process map for students clustered into two groups based on their classification in the training data. Disco is an application developed by leading academics with years of process mining experience to generate process maps from raw data in a short amount of time, and they provide free access to this tool for academic institutions. Disco facilitates the filtering of process maps based on the duration and frequency of edges between specific events. This allowed for the rapid generation of the reasonably comprehensible process maps seen in Figures 1 and 2. To produce the FALSE process map in Figure 1, an edge limit was set to only include transitions that had occurred >500 times. For the TRUE process map, an edge limit of >1000 was used to account for the larger number of students with the TRUE classification.

A process map was generated for students flagged as FALSE (Figure 1) and students flagged as TRUE (Figure 2). Common processes were then identified in the two process maps and turned into features. This step was based on work done by Van der Aalst et al. (2011), indicating that process models can offer insight into reliable time predictions. The common processes were represented in the form of 2–4 successive actions, and then features were generated based on the frequency and duration of those sequences.

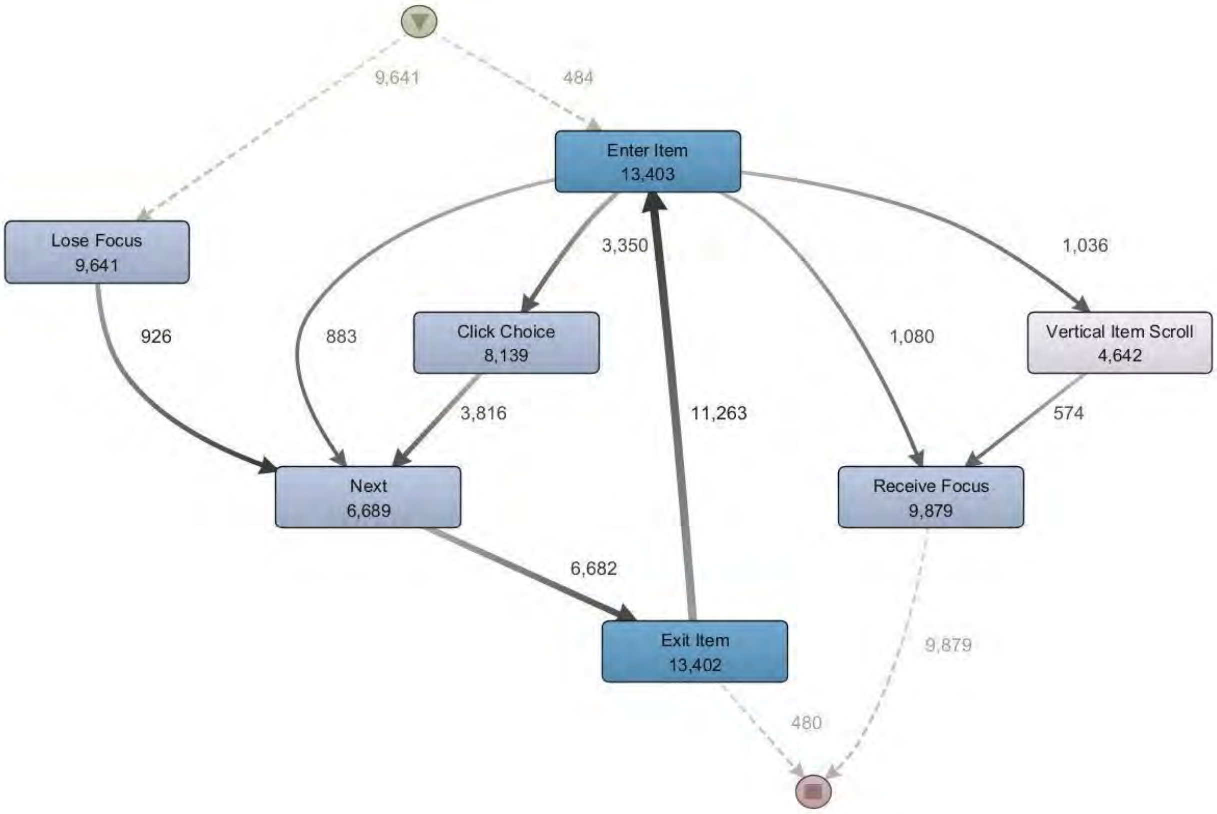

Figure 1: Process map of actions taken by students with FALSE classification.

3.4.3. Feature Importance

A model with >300 features has several notable drawbacks. The first being the significant amount of time it takes to train the model and tune the hyperparameters. The second important reason is the low interpretability of a model with too many features. A teacher attempting to help students reduce test anxiety by practicing better time-management skills would have a difficult time inferring anything useful from a model with >300 features. Finally, with only 1200 students in the training set, 300 features could potentially lead to overfitting and would be less likely to retain validity when applied to unseen data.

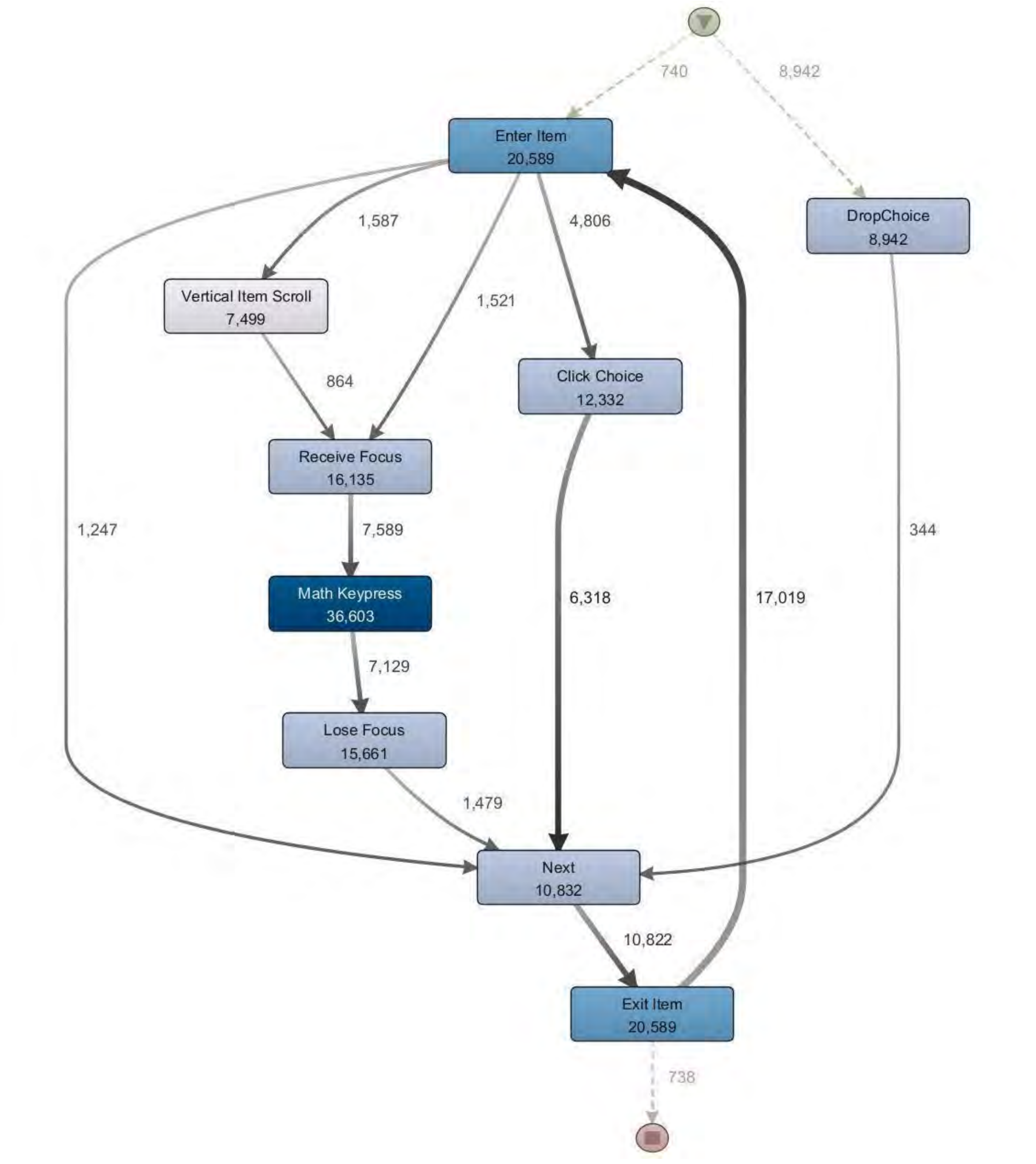

Figure 2: Process map of actions taken by students with TRUE classification.

In this work, a rudimentary iterative feature selection algorithm was used to reduce the number of features. The XGBoost model was trained using a random search for quick hyperparameter tuning, and then the model was evaluated to determine the importance of specific features. In this case, feature importance was determined by the weight of the features as measured by the number of decision trees in which they are present. The top 75% of features were retained and the process was repeated until <30 features remained. This set of features was then used to train the final model for each subset of data.

Table 1: 10-minute model: Feature description

Feature	Description (for first 10 minutes)
actionRate10, actionRate50	actionRate is the rate at which a student is taking actions. e.g., actionRate40 is the amount of time it took a student to go from 30 total actions to 40 total actions.
VH098810, VH098740, VH098808	The total time spent on these specific questions on the exam.
DropChoice;DropChoice_PT_R, Exit Item; Enter Item_PT_R, Enter Item; Click Choice_PT_R	The average time spent performing the sequence of actions specified. e.g., DropChoice;DropChoice. DropChoice is the action of choosing an answer from a dropdown menu twice in a row. PT = Process Time, R = Rate.
Open Calculator_T	The total time spent after taking the "Open Calculator" action and their next action. Intended to capture the lag time between opening the calculator and the next action.

Each of the three models (10, 20, and 30 minutes) generated a different set of top features, indicating that at each duration different features were more predictive of efficient time use. The top ten features for each model, based on their *F* score, are described in Tables 1, 2, and 3. The *F* scores were calculated by summing the number of times a feature occurs in the trees of the model, using the same weight value as in the feature selection process. These top ten feature descriptions offer a glimpse at some of the insights into test-taking behavior offered by this predictive model.

The important features identified in Tables 1, 2, and 3 were not always the same depending on the tuning of the hyperparameters. However, they were chosen based on being the top features for the competition-winning versions of the models and their inclusion in the top-performing models across multiple rounds of hyperparameter tuning. In the results section, the meaning of the important features, especially those appearing in multiple models, is explored.

3.5. Hyperparameter Tuning and Model Evaluation

Ten-fold cross-validation was used to tune the hyperparameters instead of holding out a set of data for multiple reasons. Firstly, the data set was only 1233 students; with such a small amount of data, it is difficult to withhold a meaningful number of students for validation without also significantly detracting from the data available for training. Secondly, the final test data would act as the hold-out set, providing a more robust evaluation of the model in the end.

To tune the large range of hyperparameters used in XGBoost, the RandomizedSearchCV from the scikit-learn python package was used to conduct a random search of the hyperparameter space. A grid search could be done of the space to conduct a more thorough survey of the permutations of hyperparameters, but a randomized search is capable of finding models that are almost as good as the best in much less time (Bergstra & Bengio, 2012).

Table 2: 20-minute model: Feature description

Feature	Description (for first 20 minutes)
Receive Focus_T	Total time spent between receiving focus and the subsequent action. The receive focus event occurs on MultipleFillInBlank questions and seems to indicate that the student clicked on a different component within the question.
Scratchwork Mode On_T	Total time spent between turning on scratchwork mode and taking the subsequent action.
MultipleFillInBlank	Total time spent on the MultipleFillInBlank question type.
minQ	The minimum of the set of the times spent on each question.
VH098812, VH098810, VH124387	The total time spent on these specific questions on the exam.
Move Calculator;ReceiveFocus_PT_R	Move Calculator is the action of moving the calculator. Receive Focus indicates clicking on an element of a MultipleFillInBlankquestion. This process indicates moving the calculator and then clicking on a component of the question. "_PT_R" signifies the average time spent on this process, PT = Process Time, R = Rate.
Exit Item;Enter Item;Click Choice	This process indicates leaving one exam item, opening the next item, and choosing an answer. This feature is the total number of times the student has done this process.

4. RESULTS

4.1. MODEL PREDICTION

As expected, the predictive power of the model increased as more information became available to the model. This can be seen in the increased AUC from the 10 to the 20 to the 30-minute model (Table 4). That said, the increase was small, and we can see that the Kappa value remained relatively unchanged, indicating that the longer-duration models were slightly better at avoiding false positives.

Table 3: 30-minute model: Feature description

Feature	Description (for full 30 minutes)
clickRate16	clickRate# indicates the number of actions in the #th minute. Thus, clickRate16 is the number of actions taken between 15 and 16 minutes.
Move Calculator;Receive Focus_PT_R	Move Calculator is the action of moving the calculator. Receive Focus indicates clicking on an element of a MultipleFillInBlankquestion. This process indicates moving the calculator and then clicking on a component of the question. "_PT_R" signifies the average time spent on this process, PT = Process Time, R = Rate.
BlockRev	The total time spent on the BlockRev item, which is a screen allowing students to review and navigate to different items on the exam.
VH098597, VH098522, VH098834	The total time spent on these specific questions on the exam.
Yes_T	The amount of time spent between taking the "Yes" action and the subsequent action. "Yes" is only available toward the end of the exam when students receive a "TimeLeftMessage".
Mean_PT	Average time spent on all processes.
CompositeCR	Total time spent on the composite constructed response questions.
Exit Item;Enter Item;Click Choice_PT_R	This process indicates leaving one exam item, opening the next item, and choosing an answer. "_PT_R" signifies the average time spent on this process, PT = Process Time, R = Rate.

The results below are for the combination of the two halves of the hidden data set. For the competition, the first half was used for a public leaderboard during the competition and the second half was used for the final results of the competition. On the public leaderboard, this model earned a 0.643 AUC; on the final leaderboard this model earned a 0.672 AUC. This difference was enough to position this model in the top 10 on the public leaderboard but moved it to the top position on the final leaderboard. The discrepancy in performance between the two hidden data sets indicates that this model is not significantly more predictive than the models developed by other entrants to the competition. Additionally, testing on data sets from different assessments, years, or students would serve to further validate the predictive power of this model.

Table 4: Performance data for the model predictions

Model	**Validation Set**			**Hidden Test Set**		
	AUC	**Kappa**	**Aggregated Score***	**AUC**	**Kappa**	**Aggregated Score***
10-minute	0.963	0.779	1.705	0.643	0.227	0.513
20-minute	0.883	0.519	1.286	0.658	0.225	0.540
30-minute	0.832	0.485	1.150	0.676	0.215	0.567
Combined	0.893	0.594	1.380	0.658	0.223	0.538

***** Aggregated Score defined in Section 3.3

4.2. FEATURE IMPORTANCE

In all three models, the time spent on specific questions was one of the key predictive features. This leads to the potential conclusion that a few key questions may require the time management skills that demonstrate students' overall ability to manage their time. Without knowing the student performance, or the specific details of the individual questions, further conjecture is impossible. At the very least, however, it is worth considering that students may be revealing their time management techniques during a few key questions on an assessment.

One of the most exciting commonalities was the presence of at least one process feature in the top ten for each model. In all three cases, we see that a feature based on the average amount of time spent on a process was important to the model. This finding validates the use of process mining to identify time-management behavior in the log data generated by students in a large-scale testing environment.

5. DISCUSSION

Our results indicate that efficient time use can be accurately predicted using a combination of process mined and expert-engineered features (RQ1). Our hypothesis is supported by both the statistical score indicating greater-than-chance predictive power, as well as the first-place position of this model on the final leaderboard. RQ2 hypothesized that the features generated by this model would be broadly interpretable. Initial evaluation of this claim is favorable, as supported by the concise explanation of the features highlighted in the feature importance section above.

5.1. LIMITATIONS AND FURTHER RESEARCH

For the ends to justify the means, the results of any educational data mining must be in service of teachers and even more importantly students who can make use of the research. As Bogarin et al. (2014) offered, "we want to improve both the performance and comprehensibility of the models obtained" (p.1). Admittedly this is a mandate somewhat beyond the scope of this work, but one that hopefully will be mitigated in the extension of these findings that offer an entry point into further analysis of, and applications for, student log data from a computer-based assessment.

The work performed here demonstrates a small example of combining process mining with expert feature engineering, but predicting efficient use of time is just one possible application. In the future, this same feature extraction technique could be applied to predicting student performance in terms of both learning and test-taking strategies. By understanding the minutiae of the processes successful test takers use, we may be able to gain additional insight into how to design tests with better validity while coaching students to achieve their highest potential.

This data set also offers the potential to investigate the efficacy and appropriateness of extended time accommodations. By predicting efficient use of time, we may be able to better understand how and why some students benefit from extra time on assessments and how those accommodations can be made effectively.

Finally, it is hoped that this work will be a starting point for refining this model of predicting efficient time use. During the hyperparameter tuning and training processes, the stochastic nature of machine learning algorithms resulted in varied results at times; depending on the day this model could have been outperformed by the other models in the top swath of the competition. None of the top 3 models were able to consistently achieve an AUC score above 0.7. In future work, through collaboration, breaking the 0.7 barrier is all but certain.

5.2. CONCLUSION

The utility of a test is diminished by the multitude of extraneous factors beyond the students' knowledge and skills, which can impact their final score (Stenlund et al., 2017). To this end, the goal of this study was to use educational data mining to open a small window into test-taking behavior, by investigating the actions a student is taking during an exam and how those actions can predict the student's efficient use of time.

The results of this work are encouraging. They present the possibility that we can predict with greater-than-chance accuracy whether a student will use their time efficiently on an exam. Practically this model could be adapted to the classroom testing environment and enable teachers to further understand and support the test-taking strategies being used by their students. If a student's performance on an assessment is being impacted by time pressure, this model could notify a teacher in the first 10 minutes of an assessment that a student may need to review the strategies they are using to manage their time.

In terms of generalizability, the model showed strong performance in both validation as well as on the unseen test set. The performance was improved through the process of feature selection, which follows the work done by Chen and Cui (2020) where they found that fewer features led to less variance and a sacrifice in training performance increased testing performance. This work demonstrates that finding the most accurate predictive models can be a filtering process during which the initial attempt of the 'kitchen sink' approach (throwing in all the features) offers a promising starting point from which to winnow down to the best features.

The most influential features in the machine learning model used in this work indicate that significant insight can be gained from delving deeper into the way students are spending their time on specific exam items. The processes that students repeat while taking an online assessment are manifestations of cognitive processes that students are undertaking and are likely influenced by their affective state.

For the field of Educational Data Mining, this work puts forth a combination of process mining and expert feature engineering for generating insights from the log data of a computer-based assessment. Process mining was developed for business operations, but it can yield unique features when applied to educational log data. This work demonstrates that at the micro-level, process mining can offer additional insights into the test-taking behaviors students are demonstrating.

6. Acknowledgments

The author would like to thank the Teachers College Data Science in Education Association, Karen Zhou, Alexander Audet, Benjamin Finkelstein, and Dr. Charles Lang for their invaluable conversations and support

References

Aksoy, A., Ledet, J. W., and Gunay, M. 2019. Design considerations of a flexible computer based assessment system. *4th International Conference on Computer Science and Engineering,* 1–6. doi: 10.1109/UBMK.2019.8907044.

Baker, R. S. 2019. Challenges for the future of educational data mining: The Baker learning analytics prizes. *Journal of Educational Data Mining, 11(1)*, 1–17.

Baker, R. S., & Siemens, G. 2014. Educational data mining and learning analytics. In R. K. Sawyer (Ed.)*, Cambridge Handbook of the Learning Sciences,* 253–274.

Bannert, M., Reimann, P., and Sonnenberg, C. 2014. Process mining techniques for analysing patterns and strategies in students' self-regulated learning. *Metacognition and Learning, 9*(2), 161–185.

Bergstra, J. and Bengio, Y. 2012. Random search for hyper-parameter optimization. *The Journal of Machine Learning Research*, *13*(1), 281–305.

Bogarín, A., Cerezo, R., and Romero, C. 2018. A survey on educational process mining, *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 8*(1), p. e1230.

Bogarín, A., Romero, C., Cerezo, R., and Sánchez-Santillán, M. 2014. Clustering for improving educational process mining. In *Proceedings of the Fourth International Conference on Learning Analytics and Knowledge*, 11-15.

Chen, F. and Cui, Y. 2020. Utilizing student time series behaviour in learning management systems for early prediction of course performance. *Journal of Learning Analytics, 7*(2), 1–17.

Ellis, A.P. and Ryan, A.M. 2003. Race and cognitive-ability test performance: The mediating effects of test preparation, test-taking strategy use and self-efficacy. *Journal of Applied Social Psychology, 33*(12), 2607–2629.

Fischer, C., Pardos, Z.A., Baker, R.S., Williams, J.J., Smyth, P., Yu, R., Slater, S., Baker, R. and Warschauer, M. 2020. Mining big data in education: Affordances and challenges. *Review of Research in Education, 44*(1), 130–160.

Greiff, S., Wüstenberg, S., and Avvisati, F. 2015. Computer-generated log-file analyses as a window into students' minds? A showcase study based on the PISA 2012 assessment of problem solving. *Computers & Education, 91*, 92–105.

Goldhammer, F. and Zehner, F. 2017. What to make of and how to interpret process data. *Measurement: Interdisciplinary Research and Perspectives, 15*(3–4), 128–132.

Gulek, C. 2003. Preparing for high-stakes testing. *Theory into Practice, 42*(1), 42–50.

JUHAŇÁK, L., ZOUNEK, J., AND ROHLÍKOVÁ, L. 2019. Using process mining to analyze students' quiz-taking behavior patterns in a learning management system. *Computers in Human Behavior, 92,* 496–506.

MACFADYEN, L.P. AND DAWSON, S. 2010. Mining LMS data to develop an "early warning system" for educators: A proof of concept. *Computers & Education, 54*(2), 588–599.

MAKHLOUF, J. AND MINE, T. 2020. Analysis of click-stream data to predict STEM careers from student usage of an intelligent tutoring system. *Journal of Educational Data Mining, 12*(2), 1–18.

NESBIT, J.C., ZHOU, M., XU, Y., AND WINNE, P.H. 2007. Advancing log analysis of student interactions with cognitive tools. In *12th Biennial Conference of the European Association for Research on Learning and Instruction,* 2–20.

PARK, J., YU, R., RODRIGUEZ, F., BAKER, R., SMYTH, P., & WARSCHAUER, M. 2018. Understanding student procrastination via mixture models. In *Proceedings of the 11th International Conference on Educational Data Mining,* K.E. Boyer and M. Yudelson, Eds. International Educational Data Mining Society, 187-197.

PAQUETTE, L., DE CARVALHO, A. M. J. A., BAKER, R. S., & OCUMPAUGH, J. 2014. Reengineering the feature distillation process: A case study in the detection of gaming the system. In *Proceedings of the 7th International Conference on Educational Data Mining,* J.C. Stamper, Z.A. Pardos, M. Mavrikis and B.M. McLaren, Eds. International Educational Data Mining Society, 284–287.

PECHENIZKIY, M., TRCKA, N., VASILYEVA, E., VAN DER AALST, W. AND DE BRA, P. 2009. Process mining online assessment data. In *Proceedings of the 2nd International Conference on Educational Data Mining,* T. Barnes, M.C. Desmarais, C. Romero and S. Ventura, Eds. 279–288.

ROLL, I., & WINNE, P. H. 2015. Understanding, evaluating, and supporting self-regulated learning using learning analytics. *Journal of Learning Analytics, 2*(1), 7–12. https://doi.org/10.18608/jla.2015.21.2

STENLUND, T., EKLÖF, H., AND LYRÉN, P.-E. 2017. Group differences in test-taking behaviour: an example from a high-stakes testing program. *Assessment in Education: Principles, Policy & Practice, 24*(1), 4–20.

VAN DER AALST, W.M.P., SCHONENBERG, M.H., AND SONG, M. 2011. Time prediction based on process mining. *Information Systems, 36*(2), 450–475.