The NAEP EDM Competition: On the Value of Theory-Driven Psychometrics and Machine Learning for Predictions Based on Log Data

Fabian Zehner
DIPF | Leibniz Institute for Research and Information in Education
fabian.zehner@dipf.de

Scott Harrison
DIPF | Leibniz Institute for Research and Information in Education
harrison@dipf.de

Beate Eichmann
DIPF | Leibniz Institute for Research and Information in Education
beate.eichmann@dipf.de

Tobias Deribo
DIPF | Leibniz Institute for Research and Information in Education
deribo@dipf.de

Daniel Bengs
DIPF | Leibniz Institute for Research and Information in Education
bengs@dipf.de

Nico Andersen
DIPF | Leibniz Institute for Research and Information in Education
andersen.nico@dipf.de

Carolin Hahnel
DIPF | Leibniz Institute for Research and Information in Education
hahnel@dipf.de

ABSTRACT

The *2nd Annual WPI-UMASS-UPENN EDM Data Mining Challenge* required contestants to predict efficient test-taking based on log data. In this paper, we describe our theory-driven and psychometric modeling approach. For feature engineering, we employed the Log-Normal Response Time Model for estimating latent person speed, and the Generalized Partial Credit Model for estimating latent person ability. Additionally, we adopted an n-gram feature approach for event sequences. For training a multi-label classifier, we distinguished inefficient test takers who were going too fast and those who were going too slow, instead of using the provided binary target label. Our best-performing ensemble classifier comprised three sets of low-dimensional classifiers, dominated by test-taker speed. While our classifier reached moderate performance, relative to competition leaderboard, our approach makes two important contributions. First, we show how explainable classifiers could provide meaningful predictions if results can be contextualized to test administrators who wish to intervene or take action. Second, our re-engineering of test scores enabled us to incorporate person ability into the estimation. However, ability was hardly predictive of efficient behavior, leading to the conclusion that the target label's validity needs to be questioned. The paper concludes with tools that are helpful for substantively meaningful log data mining.

Fabian Zehner, Scott Harrison, Beate Eichmann, Tobias Deribo, Daniel Bengs, Nico Andersen and Carolin Hahnel "The NAEP EDM Competition: Theory-Driven Psychometrics and Machine Learning for Predictions Based on Log Data" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 302 - 312

Keywords

Log Files, Psychometrics, Theory-Driven Feature Engineering, Process Data

1. INTRODUCTION

With the *2nd Annual WPI-UMASS-UPENN EDM Data Mining Challenge*,[1] the organizing consortium continued a young series of data competitions featured by the Educational Data Mining Society. The data challenge consisted in predicting students' behavior in a second test part using the log data produced in a first test part. The organizer's goal was to identify students who will act inefficiently by rushing through the second test half or not reaching the end of the test [19]. Another central, and noticeably constraining, secondary goal was that accurate classification should be reached as early as possible during test administration (i.e., with as little log data as possible) [19].

In this paper, we report details on our theory-driven psychometric contribution to the competition.[2] Opposed to data-driven analyses, a theory-driven one is characterized by identifying potential mechanisms at play and an according selection of methods, features, or both. The focus on a theory-driven feature-engineering access rather than some presumably more powerful deep-learning or other black-box methodology traces back to our team's psychometric background with strong experience in log data analysis. We believe that the theoretical understanding of underlying behavioral and cognitive processes that drive characteristics of test-taking behavior such as efficiency is crucial for build-

[1] `http://tiny.cc/CompAIED` [2020-02-29]; also called *Nation's Report Card Data Mining Competition 2019*

[2] Our competition contributions have been submitted under the name *Team TBA* (Centre for Technology-Based Assessment | DIPF).

ing predictive models as requested in the given competition. Otherwise, the risk of integrating spurious associations into productive classifiers is high. Moreover, in the present paper, we provide evidence that the validity of the data challenge's target label needs to be reassessed since we could show that students' ability was hardly associated to the target label. Ability estimation was enabled by the re-engineering of scores from the log data—a unique contribution of the present paper. We suggest potential solutions for identified issues.

Efficiency can be defined as the characteristic of producing desired results without waste [13]. In the context of efficient test taking, this corresponds to successful test taking with minimum effort or time. Obviously, efficiency involves two components, namely goal-reaching and resource-saving. As we elaborate on in detail throughout the following sections, the competition's operationalization of efficiency strongly emphasizes the latter component, but largely neglects the former. This consideration is emblematic and shows the value of a theory-driven and psychometric access to the matter. We regard log data that is captured during test administration as process data, which means it constitutes "empirical information about the cognitive (as well as metacognitive, motivational, and affective) states and related behavior that mediate the effect of the measured construct(s) on the task product" [7]. Thus, log data from assessment contexts is not just a by-product which is nice to have, but it carries relevant information and can be drawn on for purposes such as the one promoted in the competition.

With respect to classification performance, our competition contribution ended up in the top quarter of leaderboard submissions and was ranked eighth within the teams that submitted their code in time [20].

The paper first describes the setup provided by the competition organizers, then focuses on our approach for feature engineering as well as classifier training, and closes with reporting and discussing results on the classifier's performance level as well as single features' predictivity. The Conclusion Section elaborates on the definition of efficient test taking and discusses the state of the art for corresponding operationalizations. Please note that we use the terms *task* and *item* interchangeably here, in accordance with each community's practice.

2. COMPETITION SETUP

2.1 Data

The competition data set [19] comes from the National Assessment of Educational Progress (NAEP), which is a US national assessment conducted across 4th-, 8th-, and 12th-grade students, including tests on a variety of subjects every two years. Specifically, the data set provided for analysis within the competition comes from the 2017 test for 8th-grade students in mathematics. The test comprised two test blocks (Block A and B) that were time limited to 30 min per block.

The NAEP 2017 mathematics assessment was digitally administered on tablet computers with keyboards [15]. The test items covered several domains such as algebra or geometry and were either presented as pure mathematics tasks or as tasks applied in an everyday context. The items included stimulus material (text and/or figures) and either a list of responses to choose from (multiple choice), drag and drop response elements, or one or more text fields for constructed responses. The students were able to navigate between items within the same test block. For some items, a virtual calculator was available. The test environment also included a virtual drawing tool, which allowed students to freely draw and make handwritten annotations, a text-to-speech feature, which could be used to listen to the tasks' written materials, and a help button offering the students information about the handling of the test environment. Before the first item, a screen with general instructions was presented, called *directions*. After the last item, a review screen was presented indicating the end of the test block. The students could navigate from the review screen back to the items and onward to the next test block. Students' actions in this virtual environment (i.e., clicking on elements or keystrokes when typing a response) were recorded as log data events. The data provided 42 different events (e.g., `Enter Item`, `Click Choice`, `Back`), From this log data, it is possible to reconstruct the course of each student's test completion behavior.

For the competition, the organizers arranged the log data into two sets. The first one was a training set, taken from Block A with log data for $n_{train} = 1232$ students across the whole 30 min of Block A. The data set was accompanied by the target label indicating whether students behaved efficiently in Block B (see next section). The second data set was an evaluation set for which efficiency labels had to be predicted by the contestants. This evaluation set comprised log data stratified into three conditions: i) $n_{10min} = 411$ students with 10 min of log data from the start of Block A; ii) $n_{20min} = 411$ students with 20 min of log data from the start of Block A; iii) $n_{30min} = 410$ students with complete 30 min of Block A. The competition organizers halved the evaluation set so that the leaderboard displayed the teams' prediction accuracy on one half of the evaluation set, and the final evaluation was carried out on the remaining half. The training and evaluation sets consisted of 438,291 and 301,924 event logs, respectively.

2.2 Target Label: Efficiency

The competition organizers categorized the students into two groups. The value `True` indicated that a student completed Block B efficiently, while `False` indicated inefficient student test-taking behavior in Block B. Students were labeled efficient when they met two criteria: "1) being able to complete all problems in Block B, and 2) being able to allocate a reasonable amount of time to solve each problem" [19].

The definition of efficiency captures two key test-taking behaviors: students who go too slow, and as such fail to complete all the items in a block, and students who go too fast through the test, therefore not spending enough time on each question. Students who are inefficient through being too slow can easily be identified due to their failure to complete all tasks. However, for students going too fast, "a reasonable amount of time" can be difficult to operationalize. As such, the organizers chose to impose an arbitrary threshold for which students were evaluated on the total time taken on a task, with "the 5th percentile as the cut-off for the

'reasonable amount of time'" [19]. This operationalization led to labeling 39.6% of the students in the training data as inefficient.

2.3 Evaluation Metrics

The objective of the competition was to develop a classifier model that would predict student efficiency. The prediction was evaluated against two key measures, the adjusted AUC and an adjusted kappa. The AUC stands for Area Under the Curve and comes from ROC analysis [4]. It compares the false positive rate to the true positive rate of the model, measuring how well the model predicts the correct outcome versus an incorrect prediction. A value of AUC $\leq .5$ would indicate a model performing no better than random chance. As such, the competition used an adjusted AUC measure, $\text{AUC}_{adj} = (\text{AUC} - 0.5) * 2$.

The second measure, kappa, also captures classifier performance by comparing how much two raters agree in classifying a given set of data beyond chance. Conceptualized by Cohen [3], it compares the observed accuracy to the expected accuracy between two classifiers. As such, the value of kappa needs to be above zero to indicate performance above random chance. The competition utilized an adjusted kappa value, κ_{adj}, in that they set the lower limit of kappa to 0. For the evaluation of the models within the competition, an aggregated score was made from AUC_{adj} and κ_{adj}.

3. METHODS

In this section, we first describe a data transformation step of splitting the three temporal conditions for feature extraction and training. This turned out to be essential for achieving appropriate classifier generalizability to the test set. Next, we describe our feature engineering as well as restrictive feature selection, and we close the section with outlining how the strings were pulled together for building an ensemble classifier for prediction.

All statistical analyses have been carried out using *R 3.6.1* [16], with the package *mlr 2.17.0* [2] for machine learning, *TAM 3.3-10* [18] for item difficulty and person ability estimation, and *LNIRT 0.4.0* [6] for item time intensity and person speed estimation.

3.1 Improving Generalizability by Separating Conditions

Our early submissions of predictions to the leaderboard revealed that the classifiers' performance—though evaluated by stratified, repeated cross-fold validation—would always decrease substantially when being evaluated on the test set. That is, the generalizability of these classifiers to the test set was low, even when cross-validations testified to stable out-of-sample classification.

The primary reason that we identified was that the training set contained 30 min of log data, whereas the test set was split into three conditions with only the first 10 min, 20 min, or the full 30 min of log data available (see Section 2.1). Obviously, it is reasonable that feature realizations and their indication for one class vary over (testing) time. As an example, the time students take to work on single tasks does not only vary by task characteristics, but is also influenced by the task's position within the test. Another example is the log event of the timeout screen that limits students' time to 30 min. Naturally, this event is reasonably predictive, but while it is available in the 30 min condition, it is not in the 10 min or 20 min condition. Therefore, training sets for each condition were necessary for the classifiers to generalize more properly to the test set.

For this purpose, we created three data sets: (i) the first 10 min of log data from the 10, 20, and 30 min conditions for predicting test set cases with 10 min of log data, (ii) the first 20 min of log data from the 20 and 30 min conditions for test-set cases with 20 min of log data, and (iii) the full 30 min of log data for test-set cases with 30 min of log data. For feature extraction, we combined the respective training and test (sub)sets. This way, we maximized the available information for norm-referenced features and parameter estimation procedures. Since we employed supervised learning methods, the test sets were excluded from classifier training.

The result of splitting the conditions was that we constructed three classifiers for each learning method and set of features. Each case in the test set, however, was classified by only one model, determined by the condition the test case belonged to.

3.2 Feature Engineering

In this section, we describe the selection of engineered features of which some ended up in at least one of the base classifiers that formed the final ensemble bag. We start with the two crucial psychometric models used for estimating students' speed and ability. Then we describe our approach of extracting features from log data and deriving simple indicators that we assumed would indicate efficient or inefficient test behavior, using the software package *LogFSM*. Finally, we describe the concept and operationalization of rapid guessing as well as an adopted technique for representing log event sequences.

3.2.1 Latent Test-Taker Speed

Efficient test taking as operationalized in the competition (see Section 2.2) is mainly characterized by test takers' time handling. If a student went relatively quickly through the test (in Block B), they were labeled as inefficient. If a student spent too much time on some tasks (in Block B), they would not be able to complete all tasks and thus be labeled as inefficient, too. Therefore, the most evident feature is test-taker speed.

Test-taker speed can be inferred from the time spent on tasks in a test. However, the time spent on a task is determined by the characteristics of the task and the test-taker. On the one hand, task characteristics, such as complexity, require and evoke a shorter or longer time on task due to the task's inherent *time intensity*. On the other hand, some test takers will have the tendency or skill to move faster through a test than others; this characteristic is called *test-taker speed*. Both time intensity and test-taker speed are not directly observable and can only be estimated as latent variables.

A model that allows the separation of time on task into item and person parameters is the *Lognormal Response Time*

Model [22]:

$$f(t_{ip};\tau_p,\alpha_i,\beta_i) = \frac{\alpha_i}{t_{ip}\sqrt{2\pi}} \exp\left\{-\frac{1}{2}\left[\alpha_i(\ln t_{ip} - (\beta_i - \tau_p))\right]^2\right\} \quad (1)$$

Response time distributions take values in the positive reals and typically have long tails. The log-transformation hence is a sensible way to approximate normality and is expected to lead to better fit than a normal model on the raw response times [22]. The lognormal model takes three parameters into account and is based on the log-transformed time t_{ip} that person p spent at item i. Item time intensity β_i captures item i's tendency to evoke more or less time spent for completing it. Test-taker speed τ_p is a person's tendency and ability to spend more or less time on item completion. Because some items will show more homogeneous time distributions than others, the dispersion parameter α_i estimates an item's discriminatory power.

The parameters of interest are estimated in a Bayesian framework using a Markov Chain Monte Carlo method with a Gibbs sampler [22, 6]. We used expected a posteriori (EAP) estimators of test-taker speed τ_p as features for predictive modeling.

3.2.2 Latent Test-Taker Ability

The provided log data did not include task scores, so scores were re-engineerd based on the log data and information from released items available through the NAEP questions tool [5]. To do this, unique item identifiers were mapped to example items provided in the NAEP questions tool, a public query tool used to showcase NAEP questions. The mapping was verified by text-to-speech contents in the log data. From this, the correct responses to items could be coded for 14 of the 19 items included in the competition data set. Using the 14 scored items, we estimated an intermediate ability score for test takers. By identifying the top 100 test takers across the 14 items, we then used their responses to the remaining 5 unreleased items to identify the most likely correct answer, thus inferring the correct scoring for the data. With this complete set of scores, we applied a Generalized Partial Credit Model [14] for estimating person ability. Theoretically, such ability estimates together with the speed estimates should be reasonably predictive of efficiency as efficiency is defined by a trade-off between performance and effort (see Section 1). The model is represented by the following equation [14]:

$$\mathrm{P}_{jk|k-1,k}(\theta_p) = \frac{\exp\left[a_j\left(\theta_p - b_{jk}\right)\right]}{1 + \exp\left[a_j\left(\theta_p - b_{jk}\right)\right]} \quad (2)$$

The equation models the probability of a person p with the latent ability θ_p to respond to an item j by choosing the kth response category. In this model, subsequent response categories are ordered by their difficulty. The parameter b_{jk} represents the difficulty of an item's response category and a_j constitutes the item discrimination (i.e., the degree to which the item is capable of distinguishing between more or less able test takers). We used Marginal Maximum Likelihood for estimating model parameters. For person ability, Weighted Likelihood Estimators [24] were used. This way, test-taker ability θ_p can be directly used as a feature for predictive modeling.

3.2.3 Simple Indicators of Students' Work Process

The analysis of process indicators is based on the assumption that latent characteristics of a test taker can be inferred from attributes of their work process [7]. However, the creation of indicators is often retrospective, depends on the specific assessment system employed, and is based on plausibility and expert opinion about which indicators might be of potential interest for a particular research question (e.g., time on task, number of page visits, or switching between environments). With the intent to provide a tool to facilitate the creation of process indicators from log data, the software package LogFSM [9] has been developed that can be used in R. Instead of providing a list of generic indicators, LogFSM requires the formulation of one or multiple theoretical models that a test developer or researcher has about the work process in a task. Afterwards, LogFSM reconstructs a given set of log data according to the predefined theoretical model(s). Attributes of the reconstructed work process then serve as process indicators.

The procedure of LogFSM utilizes the concept of finite state machines [10]. The work process is decomposed into a finite number of states which represent sections of the theoretically defined response process. For example, a researcher who wishes to distinguish process components in a math assignment might define the states *Task Reading*, *Task Processing*, *Responding*, and *Reviewing* that could alternatively be collapsed into states of lower granularity like *Stimulus Processing* and *Task Answering*. Practically, states are identified by events that represent test-taker interactions with the assessment platform (i.e., log events). The occurrence of such events can serve as the conditions that must be met in order to change from one state to another one, which is called transition. The interpretation of an event might differ from state to state, which may result in differences as to whether or not a transition is triggered. Depending on the previous state of a test taker, for example, a radio button click event might be interpreted as a first-time response (*Responding*) or an edited response (*Reviewing*). In summary, the interpretation of states and state sequences is constituted by the interplay of visible components of the assessment system (e.g., texts, images), the possibilities for interactions (e.g., buttons, text fields), the contexts in which events take place (e.g., accessing a calculator before or after a response was given), and—most importantly—the predefined assumptions about test-taking behavior and cognitive operations (e.g., reading instructions, reconsidering an answer) [10].

Finally, process indicators can be derived as attributes of the reconstructed states (or the reconstructed sequence of states) from log data that contextualize test-taking behavior according to the theoretically assumed test-taking process. The integration of the characteristics of a task, the available log events, and the theoretical expectations about the test-taking behavior assign a substantive meaning to an indicator [10]. For example, an indicator that reflects how long a student actually spends reviewing and checking a particular response again can be defined as the total time in a state *Reviewing* aggregated over multiple revisits of the task and cleaned for the time in other states such as *Responding*.

For the competition's data analysis, we specified five FSMs to represent different attributes of students' work process. The states of theses FSMs represented students' on-screen page (26 states); attempting, processing or reviewing of one of the 14 multiple-choice tasks (46 states) and tasks with other response formats (19 states); students' use of the text-to-speech tool (4 states); and their use of the calculator and the drawing tool (5 states). Figure 1 shows the last mentioned model as an example. We distinguished between having the calculator active (state *CalcOn*), having the drawing tool active (state *textit*), and both tools being inactive (state *textit*). Transitions between states were triggered by the log events described in Section 2.1. For example, the state *CalcOn* was transferred to the state *ToolsOff* when the calculator was closed. That is, when the student pressed the calculator button (*CloseCalculator*), the drawing tool was activated (*ScratchworkModeOn*), or the item was left (*ExitItem*). Vice versa, when the drawing tool was activated, students' could not open the calculator, allowing for the modeling of distinct states. Self-transitions were specified to deal with, for example, double-clicks.

Several simple indicators were then derived as aggregated attributes of the reconstructed states or sequence of states. For example, the number of occurrences of the state *CalcOn* across items reflects how often a student opened the calculator during the assessment. A summary of the derived simple indicators and their descriptions is provided in Table 1.

3.2.4 Rapid Guessing

Compromised effort and persistence have been shown to be identifiable by investigating rapid guessing behavior [25]. The concept of *rapid guessing behavior* is based on the assumption that the amount of time that a test taker spends on a task before responding is not sufficient to perceive the task and develop a serious solution [21]. A rapid guess is therefore defined as a response to a task with a response time below a certain threshold.

For the definition of the thresholds, multiple approaches are possible [26]. Following the competition's operationalization of inefficient test-taking behavior [19], the present work identified task-specific response time thresholds for rapid guesses based on a 5th percentile cut-off value. This implies the assumption that the slowest 5 percent of test takers on each item showed rapid guessing behavior. This was in line with the competition's definition of inefficient test-taking behavior and, thus, necessary for predicting the accordingly constructed target label. However, this is not state of the art and the Discussion Section reviews alternative approaches.

On the basis of the identified rapid guesses, a response matrix X_{pj} was constructed, indicating whether a response to task j by person p was observed and identified as a rapid guess. The entries in this matrix are specified as follows:

$$x_{pj} = \begin{cases} \text{NA} & \text{if no response is observed} \\ 0 & \text{if a response is observed \& a rapid guess} \\ 1 & \text{if a response is observed \& no rapid guess} \end{cases} \tag{3}$$

X_{pj} was then used to extract several rapid guessing indicators. The indicators encompass a dichotomous grouping-

Table 1: Simple Indicators Serving as Features or Used for Derived Feature Modeling

Indicator	Description
Time on Screen	Time a student spent on each task within the test. This included the directions, review, and help screens.
Tasks Attempted	A count of the number of tasks at which a student showed behavior indicating they were attempting to complete the task.
Tasks Completed	A count of the number of tasks a student had completed such that it could be scored.
Tasks Incomplete	A count of the number of tasks which a student attempted, yet left the response area with incomplete information; e.g., only placing 3 out of 4 drag-and-drop boxes into the response area.
Timeout	A binary variable indicating whether a student received the time-out screen, typically indicating that they failed to complete all tasks within the time limit of 30 min.
Reviews	A count variable indicating the number of times a student visited the review screen.
Too Fast	A count variable indicating the number of times a student was in the fastest 5% of test respondents for a given task.
Viewed/No Attempt	A count variable for the number of times a student viewed an item without interacting with the item in any meaningful manner.
Time on Directions	A time variable capturing the total amount of time spent on the directions screen.
Text to Speech	A count variable indicating the number of times a student utilized the text-to-speech feature.
Help	A count variable for the number of times a student opened the help dialogue to seek assistance.
Calculator	A count variable for the number of times a student opened the calculator feature.
Drawing Tool	A count variable for the number of times a student opened the drawing tool.

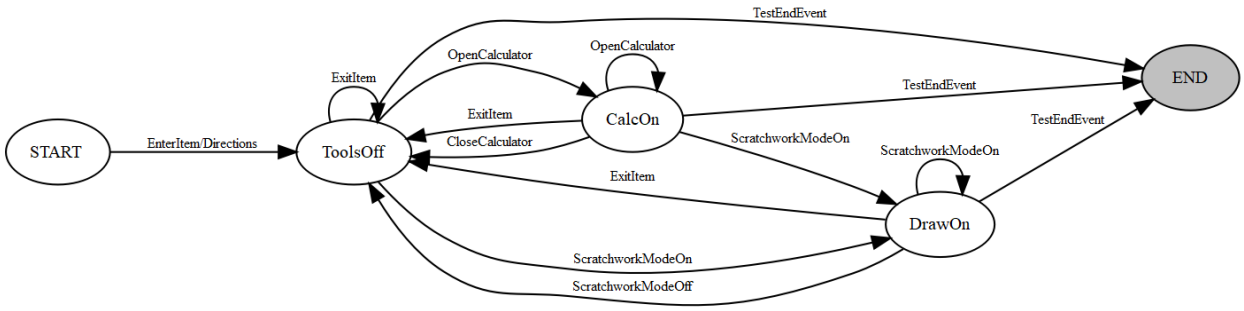

Figure 1: Exemplary Finite State Machine for Reconstructing Information from the NAEP Log Data

variable (whether a person showed at least one rapid guess), the sum of rapid guesses, and an estimation of a latent rapid guessing propensity [11]. For the estimation of the latent rapid guessing propensity, a Rasch model [17] was selected:

$$\mathrm{P}(X_{pj} = 1) = \frac{\exp(\theta_p - \sigma_j)}{1 + \exp(\theta_p - \sigma_j)} \tag{4}$$

The Rasch model is similar to the GPCM presented in Section 3.2.2, just reduced to the dichotomous case and keeping the discrimination parameter constant. While the notation of symbols and indices is generally continued here, σ_j represents an item's difficulty (or propensity to evoke rapid guessing) and X_{pj} denotes the observed response correctness (or rapid guessing behavior), with $x \in \{0, 1\}$. For person parameter estimation, Expected A Posteriori estimates were used.

3.2.5 *n-Grams of Log Events*

The occurrence of certain log events can indicate behaviors or unobservable meta-cognitive, cognitive, or affective states of interest. This is also true for combinations of such. In the context of the competition, disengaged behavior might be a precursor or indicator for (later) inefficient test taking. For example, (a) whether a student uses the assessment system's drawing tool in a task that does not require its usage could be indicative of inefficient test taking as could be (b) the playing-around with the text-to-speech feature. For incorporating such predictive features, we adopted an approach by He and von Davier [8] that borrows techniques from natural language processing and information retrieval.

At the core of the procedure [8], a student's log events are considered as n-grams of a sequence. n-grams constitute all possible tuples of subsequent log events within a student's complete sequence of log events. For computational as well as sample size reasons, it is common to limit analyses to uni-, bi-, and trigrams. Hence, a sequence such as `ACAD` (representing four log events) would be decomposed into four unigrams ($2 \times \langle$`A`$\rangle$, $\langle$`C`$\rangle$, $\langle$`D`$\rangle$), three bigrams ($\langle$`AC`$\rangle$, $\langle$`CA`$\rangle$, $\langle$`AD`$\rangle$), and two trigrams ($\langle$`ACA`$\rangle$, $\langle$`CAD`$\rangle$). We decided to make each event task-specific; that is, the event `Draw` was captured together with the task ID, for example, `DrawTask4`. This way, events were contextualized. Varying by the 10, 20, and 30 min conditions, we obtained 7448, 13,482, and 17,553 n-grams, excluding sequences that occurred in less than 15 students' sequences.

Next, the frequency sf_{ij} of each n-gram i is computed for each student j (i.e., sequence frequency). These frequencies are then weighted by inverse sequence frequency (borrowing from the term *inverse document frequency*), $ISF_i = log(N/sf_i)$, with N representing the total number of sequences, and log-normalized; that is $(1 + log(sf_{ij})) * ISF_i$. This way, sequences occurring across many test administrations are scaled down in their importance and vice versa. Also, higher frequencies are dampened by the log-transformation.

The weighted n-gram frequencies can then be checked for their predictivity of, for example, efficiency, using a χ^2-distributed statistic (details at [8, 12]). This revealed 841, 1259, and 1190 significantly predictive n-grams ($\alpha = .05$) for the respective condition.

In a last step, we compressed the selected features in a principal component analysis. Due to the need for a low-dimensional feature space (see Section 3.3), we extracted only a few components, retaining only 5% of the original information. This resulted in 6, 9, and 14 components, respectively, for the three conditions.

3.3 Feature Selection

We applied several different feature selection strategies. First, we used random forests to obtain features' importance for predicting students' efficiency in Block B. Second, we evaluated the accuracy of predictions using different combinations of features. Both strategies showed speed to be the most predictive feature in all conditions. However, the importance of the other features differed depending on the data set and combination of features.

Moreover, we frequently observed that if the addition of a feature improved the classification performance on the training data substantially (evaluated by stratified, repeated ten-fold cross-validation), it reduced the performance on the test data significantly. Thus, low-dimensional models were always to be favored over high-dimensional ones. For our final ensemble bag, the 10 and 20 min classifiers indeed turned out—with one exception—to work best with only one single feature: latent person speed. In the 30 min condition, more features were selected for the final prediction. For a list of the resulting features for all conditions, see the following Section 3.4.2.

Table 2: Three Sets of Base Classifiers

Classifier Set (1): Speed & Test Completion

	ML	Speed	#Complete	#Incomplete	#TooFast	n-grams
10min	SVM	+				
20min	SVM	+	+			
30min	SVM	+	+	+	+	

Classifier Set (2): Multiclass Speed & Test Completion

	ML	Speed	#Complete	#Incomplete	#TooFast	n-grams
10min	mSVM	+				
20min	mSVM	+				
30min	mSVM	+	+	+	+	

Classifier Set (3): Speed, Test Completion, & n-Grams

	ML	Speed	#Complete	#Incomplete	#TooFast	n-grams
10min	JRip	+				+
20min	SVM	+				+
30min	SVM	+	+	+	+	+

3.4 Prediction

3.4.1 Harvesting More Information: Multi-Label Classification

The binary target label split students into efficient and inefficient test takers. However, the competition's definition of *inefficient* behavior mixed two types of test takers: those who are going too fast and those who are going too slow. Since the two types have different feature realizations, the learning algorithms have to optimize towards at least two different conditions for the same class. Most algorithms' optimization works better if they have less conditions to optimize for within each class.

Therefore, we used the latent test-taker speed feature for further splitting the inefficient category into *Going Too Slow* and *Going Too Fast*. This new target label with now three instead of two classes was used for one set of classifiers (see Section 3.4.2). For doing so, the latent speed estimated by the Lognormal Response Time Model (see Section 3.2.1) distinguished between students going too fast and going too slow. An analysis showed that substantial rapid guessing behavior started at a threshold of about $\tau = 0$ and, thus, optimally divided the two inefficient groups. The resulting target label identified about 23% of the test takers as going too fast and about 17% as going too slow, keeping the original share of 60% of efficient test takers.

3.4.2 Three Sets of Base Classifiers

For the final prediction, we created three sets of base classifiers that were to be merged in an ensemble bag. Each set followed a different idea, incorporated different features, and was trained by a different learning algorithm. In turn, each set contained three classifiers, with one of them tailored to the 10, 20, and 30 min condition, respectively. We experimented with different feature sets, learning algorithms (common ones such as support vector machines, AdaBoost, J48, neural nets, and others), and hyperparameters for each base classifier. Table 2 shows which features and learning algorithms were used in which classifier. Which features were included and which learning algorithm was employed was determined by resulting performance with respect to the leaderboard. Due to the unstable performance in the test set, no systematic hyperparameter tuning was carried out.

Our first set of classifiers used support-vector machines with a radial kernel and C-classification for all three conditions (with $C = 1$, $\gamma = 1/n$, $\epsilon = 0.001$, shrinking). In the 10 min condition, only speed was used for the prediction. In the 20 min condition, the number of completed items was added. In the 30 min condition, all features that got through feature selection (except n-grams, on purpose) were incorporated: speed, number of completed items, number of incompleted items, and items completed too fast.

Our second set of classifiers was designed similarly to the first one, but with a multiclass support-vector machine and the multiclass label distinguishing going-too-slow and going-too-fast students (see Section 3.4.1). In the 10 and 20 min conditions, speed was the only predictor of importance according to the feature selection procedure. In the 30 min condition, again, all features (except n-grams) were incorporated.

Our third set of classifiers differed from the other two sets in that it incorporated one principal component of the n-grams of event sequences (see Section 3.2.5). Apart from that, the same set of features were used like in the second classifier set. The 10 min condition made use of a propositional rule learner instead of the otherwise employed support-vector machine. The rule learner's parameters were set to $F = 3$ folds, $N = 2$ as the minimal weight, maximum error rate of included rules $\geq .5$, and pruning was used.

3.4.3 Ensemble Bag

The three described sets of classifiers were combined in a final ensemble classifier. We used the bagging approach by averaging probabilities of a condition's three base classifiers, but favoring inefficient classifications. We chose to favor inefficient classification since our base classifiers produced not enough inefficient classifications. Therefore, we ended up with one ensemble bag of classifiers for the 10, 20, and 30 min condition each.

4. RESULTS

The final evaluation of our prediction resulted in $\text{AUC}_{adj} = 0.27$ and $\kappa_{adj} = .19$. In the leaderboard with all 82 competitors, this corresponded to rank 25, with several teams having submitted multiple results. In the final table, which only included 13 teams that submitted their code in time, our contribution was ranked eighth. The winner achieved $\text{AUC}_{adj} = 0.34$ and $\kappa_{adj} = .22$. The rather low performance values, even for the winners, were accompanied with corresponding differences between the test and evaluation set, resulting in substantial changes in the ranking and indicating rather unstable models being prone to changes in the evaluation data. This is in line with the wavering performance during testing we observed.

With respect to single features, two of them draw particular interest: test-taker speed and ability. Figure 2 shows their ROC curves. Obviously, the latent speed feature taken alone predicts efficient test taking noticeably well ($\text{AUC}_{adj} = 0.36$, $\kappa = .30$ in a single-feature support-vector machine[3]). In contrast, students' ability does not capture a lot of relevant information for predicting efficient test taking ($\text{AUC}_{adj} = 0.16$, $\kappa = .07$ in a single-feature support-vector machine[3]).

[3]based on the 30 min training data and a stratified 10-times tenfold cross-validation

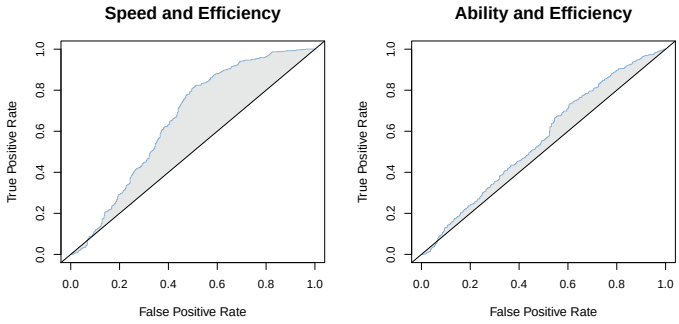

Figure 2: ROC Curves of Two Features: Speed (left) and Ability (right)

The large overlap of distributions between efficient and inefficient test takers for the ability feature further shows that the efficiency label does not contain much information about test takers' ability (right part of Figure 3). There is a small difference in that inefficient students have lower ability values on average ($\Delta = -0.08$, Cohen's $d = -0.20$). While the overlap of distributions appears somewhat similar for the speed feature (left part of Figure 3), the long right tail and prominence of faster inefficient test takers makes the feature space more easily separable. The effect size of the subgroups' difference is remarkably higher ($\Delta = 0.12$, Cohen's $d = 0.57$).

Finally, the feature space which is formed by ability and speed is plotted in Figure 4. The large majority of test takers builds an indistinguishable cloud. The other main message of this plot is, first, that very fast and less able test takers were consequently classified as inefficient in Block B. More surprisingly, second, a few test takers who were relatively fast, but answered correctly (and were thus estimated as relatively able) were classified as *in*efficient in Block B. It is possible that these students changed their behavior in the second test block. The other possibility is that the efficiency label classifies these instances erroneously as inefficient.

5. DISCUSSION

In this paper, we present a theory-driven psychometric modeling approach to predicting efficient test taking behavior in the context of the NAEP Data Mining Competition for 2019. The paper makes two important contributions, one to our understanding of the data, another to the structure of the competition.

The first major contribution is the value of theory-driven psychometric modeling for feature engineering. Referring back to Merriam Webster's bipartite definition of efficiency as the characteristic of producing desired results without waste [13], it is interesting how task success is not incorporated into the competition's conceptual specification of test takers. The data patterns mirror the lack of the desired results in the competition's operationalization of the target label, demonstrating the prominence of speed as the sole determinant for the classification as efficient test taking. Remarkably, the outstanding speed feature serves as the only feature in some classifiers of our final ensemble bag that only falls short of the winning contribution by $\Delta\text{AUC}_{adj} = .07$ and $\Delta\kappa = .03$. Empirically, ability did not provide any incremental increase in kappa or AUC beyond the speed feature. As a result, the ability feature was not included in any of the base classifiers after feature selection. It has to be noted that, at the theoretical level, the definition of efficiency only incorporates ability indirectly. That is the case because students who do not reach the end of the test cannot solve the corresponding items. Students who are going too fast are likely to fail as well. The resulting ability estimates, which are based on item success, hence, are indirectly incorporated in the efficiency label that is actually based on speeding criteria exclusively. Nevertheless, this indirect impact was not large enough for granting substantial predictivity to students' ability for inferring their test-taking efficiency as specified by the competition.

It is apparent that the presented predictive modeling's performance does not exceed a moderate level, if at all. This is similarly true for the competition winners. While behavioral predictions with temporal delay can always be excepted to be weak, there seem to be multiple reasons inherent to the provided data set and challenge behind the moderate predictive classification performance. From our point of view, there are three major points that are worth following-up on in discussions. The most prominent one is the data reduction to twenty and ten minutes of log data for two thirds of the test data. The resulting leaderboard data evaluation was dominated by the secondary goal of predictions with less data. Also, since the different conditions shape the data and derived features quite differently, the training of classifiers had to be tailored to those.

The second important contribution is that the paper provides evidence that questions the target label's validity. Using additional data sources from outside the information provided by the competition, we were able to re-engineer scores for estimating test taker ability. Importantly, feature selection led to excluding the ability feature, as it failed to be predictive of the efficiency label. This was a strong indicator for the suboptimal operationalization of efficiency.

This especially relates to the labeling of students as going too fast. To identify test takers spending a reasonable amount of time on a task, the competition organizers chose the 5th percentile of response times within an item as the threshold. Such a norm-oriented classification leads to labeling a fixed number of test takers as inefficient at each item, even when there are none or substantially less than 5 percent. Instead, criterion-based classification would be worthwhile. However, if corresponding criteria are not available, norm-oriented approaches would need to be combined with a dynamic threshold to be determined for each item, as the response time distributions of items typically differ considerably. The high ratio of 40 percent of students labeled as inefficient, which seems unreasonably high, is probably the result of this purely norm-based decision.

One option for identifying an appropriate threshold constitutes the visual inspection of distributions if little information about items are available. Often, response time distributions are bimodal. The first, very early peak is then typically associated with rapid guessing, while the second

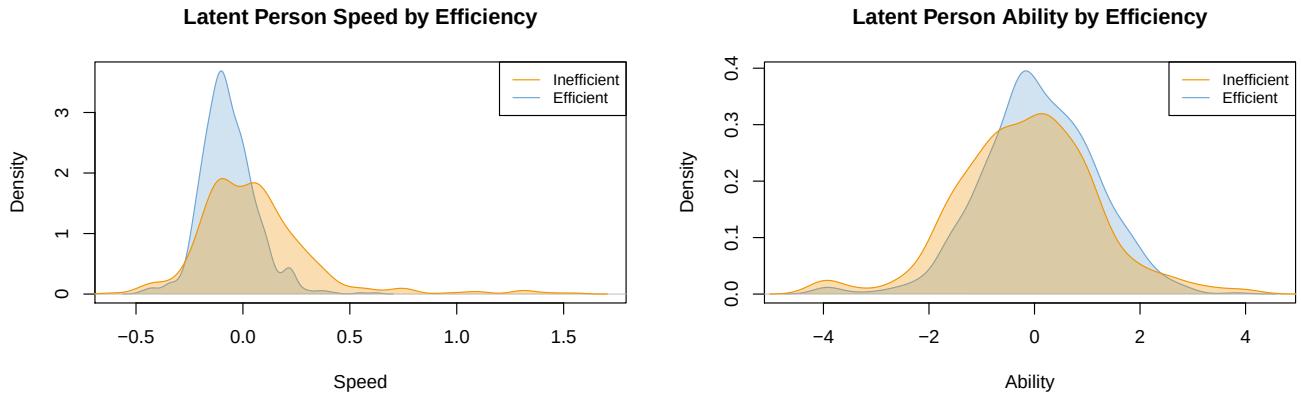

Figure 3: Distributions of Speed (left) and Ability (right), Separated by Efficiency

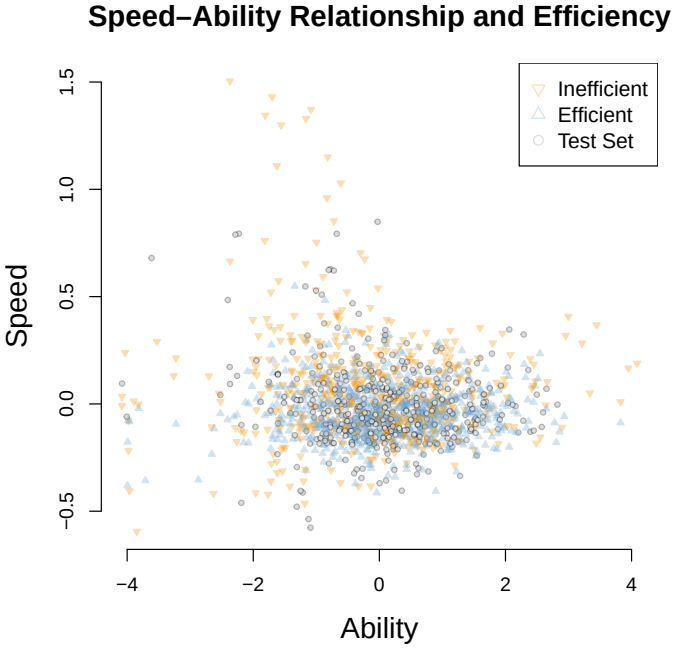

Figure 4: Scatter Plot of Speed and Ability, Distinguishing Efficient and Inefficient Test Takers

peak corresponds to the actual response time mean of those test takers who did not exhibit rapid guessing behavior. The threshold would be set after the obvious extinction of the first peak [27]. For setting an actually accurate threshold, methods that combine response time, item information, and response accuracy are considered state of the art. For an overview see for example [26]. Further, the identification of thresholds should be guided by contextual considerations, judging for example whether false-positives or false-negatives are more acceptable in the context of the test.

An additional area of interest was that the binary label for efficiency mixes two types of students within its inefficient value: going too fast and going too slow. This has implications for the learning algorithms that have to optimize their parameters towards two different conditions within one class. Moreover, from a substantive perspective, this mixes at least two types of students: those who are disengaged—thus, either rushing or meandering pointlessly through the test—and those who are too thoroughly working, poorly monitoring their progress, or who are just less able.

6. CONCLUSION

One of the central messages of the competition is that predictions of test-taking efficiency are highly dependent on the definition, measurement, and evaluation of efficiency itself. That is true for the presented approach, as well as for other competition entrants, as seen through the leaderboard test set evaluation phase. In such a case, and if classifiers are meant to be put into productive usage, it is even more important from our point of view to have comprehensible models. Imagine a hypothetical situation when a teacher sees a student being flagged on a dashboard after 20 min of testing. The flag indicates the risk for inefficient test taking later on, but we know that the flag's accuracy is fairly low. It is vital that the teacher is informed about the basis of the flag's decision criteria. As we have shown, the competition's target label classified some of the most able students as inefficient who by ability are reasonably quick in completing the tasks. The consequences of a teacher going to a successful, engaged student and telling them they should aim at being more engaged or efficient in their test taking, would be reasonably disruptive. It can be assumed that such an

invasive and intrusive test administrator behavior would be counterproductive and decrease, rather than improve data quality. If however, the included features for predictions are transparent, known, and understandable, the teacher could communicate those and contextualize the flag accordingly. A risk of more powerful black-box deep-learning classifiers is that a small to medium share of more accurately classified cases does not necessarily outweigh the resulting obscurity of classification mechanisms. More generally, the effects of the invasive disruption of a test administrator proactively trying to motivate test takers on the standardization of the assessment setting need to be studied. Moreover, before using such a measure, classifiers would need to be checked for biases towards certain subgroups in order to still adhere to standards of standardized assessments [1]. Overall, we would recommend to refrain from using such predictions with low to moderate accuracy in productive assessments as long as the effects of changes in the test administration are unknown.

Instead, the discussion section gives some insights into what could improve the setup of a more proper training data set for predictions. Mainly, a more representative definition of efficiency might be necessary, one that reflects the current scientific state of the art which factors in students' ability. Furthermore, the described psychometric and theory-driven perspective, together with the referenced tools, can be helpful for mining log data from assessments at the large scale while retaining the individual perspective. With the illustrated software package LogFSM, for example, we were able to identify test takers who clearly showed consistent inefficient behavior, but were labeled as efficient, and vice versa. These observations are constrained by the fact that the log data of Block B was not available, yet served as the basis for the evaluation of the efficiency label. However, we think that the number of these cases is too large for being an effect of temporal instability only. We believe that these analyses combined with more innovative machine learning designs that the educational data mining community can provide are promising for further improving the predictions of test-taking efficiency.

7. LIMITATIONS

The paper already highlighted the presented study's limitations over the course of the different sections. On top of the challenges inherent to the data competition, this study's main limitation constitutes the employment of baseline machine learning. Moreover, speed and ability have been estimated separately, whereas a simultaneous estimation might have been possible as well [23]. The selection of feature sets and learning algorithms was optimized towards the test set which turned out to provide rather unstable evaluations. The conclusion of this paper is that the NAEP Data Mining Competition for 2019 provided an important opportunity to further develop complex conversations about how educational data mining and psychometric modeling can support data quality of assessments by identifying disengaged test taking behavior.

8. REFERENCES

[1] AERA/APA/NCME. *Standards for educational and psychological testing*. American Educational Research Association, Washington, DC, 2014.

[2] B. Bischl, M. Lang, L. Kotthoff, J. Schiffner, J. Richter, E. Studerus, G. Casalicchio, and Z. M. Jones. mlr: Machine learning in R. *Journal of Machine Learning Research*, 17(170):1–5, 2016.

[3] J. Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46, 1960.

[4] T. Fawcett. An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8):861–874, June 2006.

[5] N. C. for Educational Statistics. Naep questions tool.

[6] J.-P. Fox, K. Klotzke, and R. K. Entink. *LNIRT: LogNormal Response Time Item Response Theory Models*, 2019. R package version 0.4.0.

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

[8] Q. He and M. von Davier. Analyzing process data from problem-solving items with n-grams: Insights from a computer-based large-scale assessment. In Y. Rosen, S. Ferrara, and M. Mosharraf, editors, *Handbook of Research on Technology Tools for Real-World Skill Development*, pages 750–777. IGI Global, Hershey, PA, 2016.

[9] U. Kroehne. LogFSM: Analyzing log data from educational assessments using finite state machines. `http://logfsm.com/index.html`, 2019.

[10] U. Kroehne and F. Goldhammer. How to conceptualize, represent, and analyze log data from technology-based assessments? A generic framework and an application to questionnaire items. *Behaviormetrika*, 45(2):527–563, Aug. 2018.

[11] Y. Liu, Z. Li, H. Liu, and F. Luo. Modeling test-taking non-effort in mirt models. *Frontiers in Psychology*, 10:145, 2019.

[12] C. D. Manning, C. D. Manning, and H. Schütze. *Foundations of statistical natural language processing*. MIT press, 1999.

[13] Merriam-Webster. Efficiency. In *Merriam-Webster.com dictionary*. n.d.

[14] E. Muraki. A generalized partial credit model: Application of an em algorithm. *Applied Psychological Measurement*, 16(2):159–176, 1992.

[15] National Assessment Governing Board. *Mathematics Framework for the 2017 National Assessment of Educational Progress*. National Assessment Governing Board, Washington, DC, 2017.

[16] R Core Team. *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria, 2019.

[17] G. Rasch. *Probabilistic models for some intelligence and attainment tests*. University of Chicago Press, Chicago, IL, 1960/1980.

[18] A. Robitzsch, T. Kiefer, and M. Wu. *TAM: Test analysis modules*, 2019. R package version 3.3-10.

[19] Ryan Baker, Beverly Woolf, Irvin Katz, Carol Forsyth, and Jaclyn Ocumpaugh. Nation's report card data mining competition 2019. `https://sites.google.com/view/dataminingcompetition2019/home`, 2019.

[20] Ryan Baker, Beverly Woolf, Irvin Katz, Carol Forsyth, and Jaclyn Ocumpaugh. Press release: 2019

naep educational data mining competition results announced. https://sites.google.com/view/dataminingcompetition2019/winners, 2020.

[21] D. L. Schnipke and D. J. Scrams. Modeling item response times with a two-state mixture model: A new method of measuring speededness. *Journal of Educational Measurement*, 34(3):213–232, 1997.

[22] W. J. van der Linden. A lognormal model for response times on test items. *Journal of Educational and Behavioral Statistics*, 31(2):181–204, 2006.

[23] W. J. van der Linden. A hierarchical framework for modeling speed and accuracy on test items. *Psychometrika*, 72(3):287, 2007.

[24] T. A. Warm. Weighted likelihood estimation of ability in item response theory. *Psychometrika*, 54(3):427–450, 1989.

[25] S. L. Wise. Rapid-guessing behavior: Its identification, interpretation, and implications. *Educational Measurement: Issues and Practice*, 36(4):52–61, 2017.

[26] S. L. Wise. An information-based approach to identifying rapid-guessing thresholds. *Applied Measurement in Education*, 32(4):325–336, 2019.

[27] S. L. Wise and X. Kong. Response time effort: A new measure of examinee motivation in computer-based tests. *Applied Measurement in Education*, 18(2):163–183, 2005.