Application of Tree-based Data Mining Techniques to Examine Log File Data from a 2012 PISA Computer-based Mathematics Item

Qi Qin

Master's Programme
Assessment, Measurement and Evaluation
120 Credits

Centre for Educational Measurement (CEMO)
Faculty of Educational Sciences, University of Oslo

May 13, 2022

Popular abstract

The advancement of technology in the digital world has brought about an extensive transformation in the form of educational assessment, where the tests conducted via screens have gradually gained their ground. Moving away from paper and pen, the computer-based assessments generate not only the scores that a traditional test could offer but also a byproduct of log file data that records students' every single interaction with the task. This type of data is considered to reflect additional illuminations to the understanding of students' cognitive abilities. However, due to its nature of tremendous volume and disorganized structure, how to exploit, model, and interpret the log file data into beneficial information remains interesting. Considered a powerful tool, educational data mining has received noticeable attention for its excellent performance in handling computationally demanding datasets. Applying the most popular tree-based data mining techniques, this study provides implications for the procedural analyses of log file data for a mathematics item as well as specific remarks on the interventions in the computer-based assessment from the Programme for International Student Assessment (PISA).

Acknowledgments

I would like to express my sincerest gratitude to my supervisor, Denise Reis Costa, who has consistently helped me with professional, patient, and responsive support. She is not only the one who provided instructive advice for my study and thesis but also a great mentor who guides me with care and beliefs. Whenever I encountered setbacks, her kind and encouraging words always motivated me to trust myself and keep moving forward.

I would also like to thank all the staff and colleagues at CEMO. Two years of study have not only brought me valuable knowledge and ability, but also an unforgettable learning experience that has added a shining memory to my life.

Many thanks to my parents, who have been providing me with the utmost support. To my best friend Rex, thank you for always being there for me when I was at my most helpless moments. Special thanks to Sognsvann, a peaceful and lovely place that freed me from long days.

Abstract

The development of computer-based assessment (CBA) has generated log file data that reveal considerable insights into students' cognitive processes and educational practice. This study aims to explore the feasibility of applying data mining techniques to analyze the log file data from a 2012 PISA CBA mathematics item. By uncovering the predictive structures, a better understanding of the contribution of the extracted features and their relations to students' item performance is desired. Four steps including feature generation, feature filtering, modeling, and evaluating feature importance were conducted in this study. Specifically, 110 features from the log file data were extracted under both the data-driven and theoretical guidance. Three tree-based data mining techniques: decision tree, random forest, and gradient boosting machine were fitted to classify Australian students' (N=1785) item responses using the filtered 65 features. Feature importance was evaluated via both descriptive Chi-square scores and the importance plot. The results revealed successful model predictions with satisfactory accuracy of .90 for all three methods. Item-specific features such as the clicks on the "GIRL-Highest 5%" checkbox and the clicks on the statement that have contributed the most to the prediction of students' item performance were identified. The concrete steps have showcased a viable process for analyzing the log file data. As the first work to mine the log file data from PISA in the mathematics domain, the findings provide specific action patterns that can lead to students' success in this mathematics item. Meanwhile, it provides insights into the development of items in PISA CBA.

Keywords: computer-based assessment, educational data mining, tree-based models, logfile data, PISA

Application of Tree-based Data Mining Techniques to Examine Log File Data from a 2012 PISA Computer-based Mathematics Item

With the continuous development of technology and the popularity of digitalization in education, the forms of educational activities and assessments have been gradually expanded, where the computer-based assessment (CBA) comes to light. Different from the traditional paper-based assessment, CBA provides various items that enlarge the interactions between the test takers and the task via computer screens. In the process of this transition, analytical techniques have likewise been enhanced accordingly due to the widespread advancement of CBA along with the properties of its derivatives.

For instance, the Programme for International Student Assessment (PISA), a well-known international large-scale assessment that measures 15-year-old students' knowledge, skills, and attitudes in domains such as mathematics, reading, science, problem-solving and financial literacy initially led the way in the implementation of CBA in its 2006 cycle (OECD, 2010). In 2012, PISA placed a focus on the mathematical literacy domain and included a computer-based assessment of mathematics. Following the advantageous flexibility of computer technology, the format of items was designed in a more characterful, engaging, and easily understandable manner that offered different types of mathematical tools such as simulated calculators and operational visualization graphs as assistance for explorations (OECD, 2013).

Consequently, test takers are open to the opportunities to conduct interactive navigations on the screens using the computer devices and the given authentic interfaces. Each of the test takers' clicking actions amid completing the task is identifiable and traceable in chronological order as efficiently recorded in the log files. Defined by Reis Costa and Leoncio Netto (2022), information in any form stored in the digital scripts (named log files) is synthetically considered as the process data in the educational assessment and can be

categorized into raw and semi-processed log file data. In this paper, the log file data is delineated as semi-processed log file data, where the raw log traces are documented with structured events from the start to the end of the task and corresponding timestamps**.** Beyond the scores provided by traditional paper-based assessments, this complex and voluminous data from the log file makes the manifestation of students' interactive work process possible (Goldhammer et al., 2021). These data in sequence or accumulation may reflect students' response strategies in terms of how students engage with the items, how the responses are generated, and how the interactions between students and the items are being produced into a scored result. In this way, more profound understandings of students' performance could be revealed for educational practitioners and different roles involved in education with the analyses and proper interpretation of the results. While offering rich information that may reflect test takers' ability, log file data are massive in volume and deficient in interpretability, requiring procedural data management and computationally intensive methods to analyze such type of data (Reis Costa & Leoncio Netto, 2022).

Data Mining Techniques and Tree-based Models

To address this issue, educational data mining, which exploits developing statistical and machine learning algorithms over different types of educational data originating in an educational context has played an indispensable role (Romero & Ventura, 2020). In contrast with statistical inference, the data mining techniques tend to be data-driven with a better performance in managing a cluttered larger number of variables. Baker and Inventado (2014) summarized the most popular methods in the educational data mining field into four classes: prediction, cluster discovery such as clustering and dimension reduction, relationship mining, and a two-step approach of discovery with models. While other researchers have adapted educational data mining methods into two narrowed categories: supervised learning and unsupervised learning (Qiao & Jiao, 2018).

The unsupervised learning methods focus on clustering and dimension reduction, which is related to the structure discovery of a data set. Without outcome variables, unsupervised learning identifies and divides the groups based on either rows or columns such that it enables the exploratory investigation of the data set and reduces the number of predictor variables. However, without a clear objective (i.e., prediction), the model performance for the unsupervised method is not easily evaluated (Boehmke & Greenwell, 2019). Also, the unsupervised method is relatively more subjective due to the absence of outcome labels compared to other methods such as the supervised learning methods.

While known as the prediction method, supervised learning mainly constructs predictive models where the outcome variables are predicted using the predictor variables given in the same dataset. The aim is to explore and capture the relationships between the outcome and the predictor variables with generalizability, where such relationships can be applied to other circumstances or populations. Depending on the type of the outcome variable, the supervised learning methods are further grouped into regression and classification. With a continuous numeric outcome such as the test scores, the model is referred to as a regression model. When predicting outcomes with discontinuous values, for example, the pass/ fail or 5 level grades, the model is recognized as a classification model.

Among the supervised approaches, the most used in practice are the tree-based models such as decision trees, random forests, and gradient boosting machines (Sinharay, 2016; von Davier et al., 2021). These learning methods narrate the attributes of the objective outcome by growing tree-like models where the feature spaces are recursively bifurcated into homogeneous groups given certain splitting rules. Among these tree-based methods, the decision tree method such as Classification and Regression Tree (CART) forms the most fundamental basis that it grows a single tree at the expense of prediction performance. While as ensembled methods, random forests and gradient boosting machines combine a large

number of decision trees with reduced bias and variance. As von Davier et al. (2021) mentioned, it is not an easy question to articulate which supervised learning methods demonstrate consistently stronger competence. The comparison for supervised learning methods is constrained for different datasets. Among various cases in classification, the support vector machine (SVM) and random forest generally performed the best among others, while the gradient boosting machine also show superior performance and gained popularity. Even though, the basic decision tree method is still sufficient and worth considering in analyzing log file data from large-scale assessments as claimed by Qiao and Jiao (2018).

Predicting Student's Performance from Large-scale Assessments

The application of educational data mining methods using various types of data is popular in the past decades. Among these methods, the prediction of students' knowledge and ability has gained greater importance in both educational measurement and educational data mining topics. Previous studies have investigated the prediction of students' performance with various data mining techniques using survey data from international large-scale assessments but not through log file data. For instance, Gabriel et al. (2018) utilized gradient regression trees to examine the relationship between students' dispositions and mathematical literacy collected from PISA 2012 questionnaires. While Depren et al. (2017) compared the supervised data mining methods such as decision trees, Bayesian networks, logistic regression, and neural networks on their performance in classifying students' achievement in TIMSS mathematics using the factors collected in students' survey. Although the application of data mining methods in education themes is trending, few studies have been found investigating the log file data using the data mining methods in international large-scale assessments.

Instead, various traditional methods are employed to investigate the prediction and understanding of test takers' strategies in large-scale assessments using the log file data. Some studies have opted for intuitive methods such as visualization and frequencies with statistical inferences to investigate the properties of action sequences. For example, Vista et al. (2017) utilized the visualization approach to identify meaningful action sequences of a problem-solving task from the Assessing & Teaching 21st Century Skills project. Greiff et al. (2015) investigated the association between students' exploration strategy performed on a problem-solving item and their item achievement in PISA 2012 cycle using relative frequencies, Pearson's chi-squared test, and correlation analyses. He & von Davier (2015) applied the chi-squared statistics and weighted log-likelihood ratio test to analyze the frequencies of action sequences of test-takers from the Programme for the International Assessment of Adult Competencies (PIAAC). Other studies predicted test takers' navigating behaviors and their personal traits utilizing assorted models. With examples from the PIAAC and PISA the reading items using both time and action features from the log file data, linear mixed models were applied to investigate the relationship between task attributes and the individual characteristics (Goldhammer et al., 2014; Naumann, 2015). Besides, Hahnel et al. (2016) adopted the latent regression and mediation models to investigate the effects of navigation behaviors on test takers' skills using a PISA reading item. Xu et al. (2018) utilized the latent class analysis to explore the process events in a PISA problem-solving item. In addition, the item response theory (IRT) modeling was also applied in the practice to analyze the log file data at both individual and item level (Goldhammer et al., 2017; Liu et al., 2018). Among these studies with traditional analytical methods, the focus has been placed on international large-scale assessments such as PISA and PIAAC within the problem-solving and reading domain by inspecting the action and time features from the log file data.

Although traditional methods are widely used, studies applying data mining methods to predict students' performance using log file data and evaluating the model performance has been flourishing in recent years with the popularity of CBA. Qiao and Jiao (2018) showcased how the supervised and unsupervised data mining methods were applied in an educational context using a PISA 2012 problem-solving item. A random forest algorithm was also applied to investigate the relationship between students' performance and the features generated from the logfile in the PISA 2012 problem-solving item (Han et al., 2019). Using the log data from France's national large-scale assessment in mathematics, Salles et al. (2020) applied both supervised and unsupervised methods to examine how the log data can provide insights into students' performance. The interpretation of students' behaviors that reflect their strategy was also investigated in this endeavor.

In summary, most of the previous works have been focused on analyzing the respondents' behaviors on international large-scale assessment tasks using log file data with different traditional methods, however, little has been investigated utilizing the data mining methods. Other studies have applied the data mining techniques, but the data from surveys with subjective responses were used instead. Among the papers that have investigated students' response indicators using the logfile data from international large-scale assessments such as PISA by applying supervised data mining methods, the focus has been mostly placed on two problem-solving items, while the investigation on mathematics is rarely touched. The lack of literature that explores the application of supervised data mining methods to analyze and extract interpretable information from mathematical logfile data has called for more investigation into this knowledge. On the one hand, as a source that provides more process information than the final results, log file data have the potential to reveal further traits of the test takers (Goldhammer et al., 2014). Various uses for the log file data such as validation of scores and the detection of students' behaviors proved important. On the other hand, the

excessive tendency toward the analyses for the problem-solving items makes it an interesting direction to explore other items. Generalizability issues such as whether the properties of mathematical problems are consistent with problem-solving remain unclear.

In terms of the gap, the present study seeks to apply three popular tree-based data mining techniques to explore the structural relationship between the features and students' performance using the log file generated from students' actions performed and the time spent on the task. Using the 2012 PISA computer-based assessment of mathematics (CBAM) CM038Q03 item, the research questions are:

RQ1) What features can be extracted from the CBAM log file data?

RQ2) Among the tree-based machine learning methods, which one provides better performance measures (e.g., accuracy, precision) in predicting student's outcome for the CBAM item using the extracted features?

RQ3) To what extent do the extracted features contribute to the prediction of students' performance?

By answering the above-mentioned research questions, the present study showcases how the information from students' task-taking behaviors can be extracted from the log file data. Through the analyses, it strives to uncover the predictive structure of the analyzed models for a better understanding of the importance of the extracted features and their relations to students' performance. This study expects to enrich the empirical practice of applying tree-based data mining methods to examine the mathematics item in international large-scale assessments and provide insightful information for various uses in the area of educational assessment.

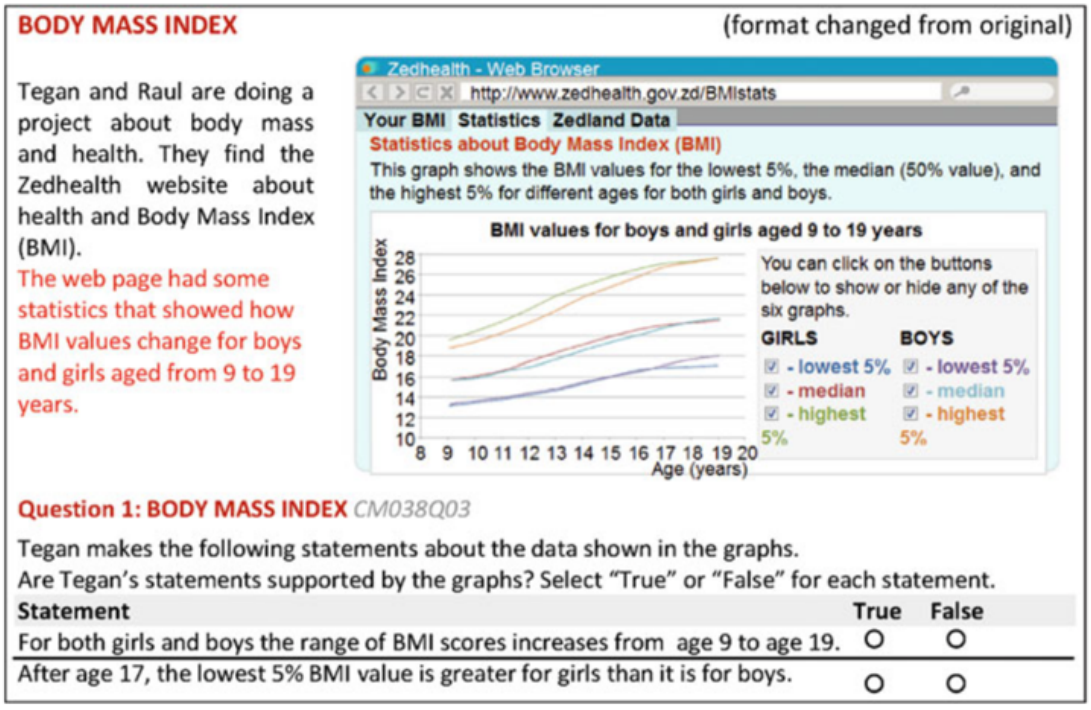
Methods

Material and Procedure

The item used in this study is CM038Q03 Body Mass Index, one of the computer-based mathematics items from the PISA 2012 cycle. In this cycle, PISA administered for the first time an optional computer-based assessment of mathematics in addition to the paper-based tests(OECD, 2013). A rotation design was performed in the assessment, where four-item clusters for CBAM were randomly distributed in 24 forms of booklets. Each of the students was assigned one of the 24 forms. Conducted in a digital format, students were requested to accomplish the task via computer devices such as screens and the mouses. In the meantime, paper and pencil were also provided for facilitation during the responding process.

Figure 1

A Screenshot for the 2012 PISA CM038Q03 Item



Note. Reprinted by permission from Springer Nature Customer Service Centre GmbH: Springer International Publishing. Computer-Based Assessment of Mathematics in PISA 2012. Assessing Mathematical Literacy: The PISA Experience. K. Stacey and R. Turner, eds. 180. Copyright 2022.

As shown in Figure 1, the interface for the 2012 PISA CM038Q03 item consists of mainly two parts, one is an authentically simulated web browser that enables students to navigate through, and the other is the task question with two statements that determine students' task performance. To accomplish this task, students are expected to choose true or false for two statements in terms of the information extracted from the web page. Once the task is started, students are allowed to click anywhere on the screen. However, only when clicking on the six checkbox buttons as instructed, students are able to show or hide the corresponding line graphs which are displayed by default in their initial state on the web browser. There is no limit for the number of navigating steps nor the limit of time on task, despite a 40-minute total response time for computer-based assessment of mathematics and reading tasks (OECD, 2013).

In the framework of the 2012 PISA CBAM, mathematical literacy has been developed in three facets: process, content, and contexts (OECD, 2013). The CM038Q03 item used in the present study was categorized in the interpretation process, uncertainty and data content, and a societal context (Bardini, 2015). Students thus are required to sort and interpret the mathematical statistics into information that facilitates the solving of the task. Specifically, students are expected to compare the line graphs displayed on the web browser using corresponding mathematical knowledge to select each statement. However, it is not an essential requirement for students to click on the checkbox buttons before they select the statements in order to gain credits from this task.

Even though, there are potential strategies that students could apply to determine each statement in terms of their knowledge and understanding of mathematical concepts. For the first statement, the keywords that are decisive for the correct selection are "the range of", "increase", and "both girls and boys". In terms of the definition of the "range", students are expected to display the line graphs that represent the lowest 5% and highest 5% BMI values.

"Both girls and boys" implies that the students may compare the lowest 5% and highest 5% BMI values for girls and boys respectively. While to compare the trend of "increase", students are not required to click on the screen, but to choose the correct option in the first statement.

For the second statement, the keywords are "after age 17", "the lowest 5%" and "greater for girls than it is for boys". Reacting to "the lowest 5%", students may click on "median" and "highest 5%" to show the line graph of the lowest 5% BMI value only. "After age 17" and "greater for girls than it is for boys" require students to compare the lowest 5% BMI value for girls and boys that are older than 17 years old respectively. A correct selection on the second statement could potentially indicate that students properly compared the information.

Data and Sample

Two types of data from the CM038Q03 item were utilized in the present study: the log file data that contain students' log traces and the cognitive item response data that record students' item response scores. Both of them were obtained from the publicly available PISA 2012 CBA database on the OECD website. Students' information contained in the data is anonymous and the GDPR application form can be found in Appendix I. As displayed in Table 1, the computer-based mathematics logfile data set consists of 9 columns including "cnt" (Country code), "nc" (National center 6-digit code), "schoolid" (School ID 7-digit), "StIDStd" (Student ID), "formid" (CBA form ID), "event" (click status), "time" (timestamps), "event number" (order of actions), and "event value" (click actions).

The "event" variable consists of three values: "START_ITEM", "END_ITEM", and "click" that represent the status of actions. While the "event_value" variable contains values that are related to the content of the click action. Except for "NULL" which corresponds to the start and end events, three major types of values are delineated in this task: a) click events

on the web browser in Figure 1 where the test takers can show or hide the line graphs through the checkboxes. They include “girl_radio_low”, “girl_radio_med”, “girl_radio_high”, “boy_radio_low”, “boy_radio_med”, and “boy_radio_high” that correspond to the buttons of GIRLS lowest 5%, GIRLS median, GIRLS highest 5%, BOYS lowest 5%, BOYS median and BOYS highest 5% respectively. b) click events on the statement including “bmiQ3_37”, “bmiQ3_38”, “bmiQ3_39”, and “bmiQ3_40” that represent the True option for the first statement, False option for the first statement, False option for the second statement and False option for the second statement accordingly. c) click events on the other area on the screen such as “girl_radio_highest_set”, “girl_radio_lowest_set”, “boy_radio_lowest_set”, and “ItemQuestionText” listed in Table 1**.** To simplify the data entries in the future steps, these three kinds of events together with the start and end events were abbreviated as “W” (clicks on the web browser), “S” (clicks on the statement), and O “clicks on other regions”, “B” (begin of the task), and “E” (end of the task). Specific event values “girl_radio_low”, “girl_radio_med”, “girl_radio_high”, “boy_radio_low”, “boy_radio_med”, and “boy_radio_high” were shortened to “gl”, “gm”, “gh”, “bl”, “bm” and “bh”, while the “bmiQ3_37”, “bmiQ3_38”, “bmiQ3_39”, and “bmiQ3_40” were coded to “right” and “wrong” for a more concise presentation.

Table 1

CM038Q03 Item Log File Data Set

cnt	nc	schoolid	StIDStd	formid	event	time	event_ number	event_ value
AUS	003600	0000032	00616	48	START_ITEM	978.1	1	NULL
AUS	003600	0000032	00616	48	click	1025.9	2	bmiQ3_37
AUS	003600	0000032	00616	48	click	1044.1	3	bmiQ3_40
AUS	003600	0000032	00616	48	END_ITEM	1046.5	4	NULL
AUS	003600	0000032	00634	60	START_ITEM	1961.1	1	NULL
AUS	003600	0000032	00634	60	click	1986.2	2	girl_radio_high
AUS	003600	0000032	00634	60	click	1989.7	3	girl_radio_low
AUS	003600	0000032	00634	60	click	1990.2	4	boy_radio_high
AUS	003600	0000032	00634	60	click	1990.8	5	boy_radio_low
AUS	003600	0000032	00634	60	click	1992.6	6	girl_label_low

cnt	nc	schoolid	StIDStd	formid	event	time	event_ number	event_ value
AUS	003600	0000032	00634	60	click	1992.6	7	girl_radio_low
AUS	003600	0000032	00634	60	click	1993.3	8	girl_radio_low
AUS	003600	0000032	00634	60	click	1997.5	9	boy_radio_med
AUS	003600	0000032	00634	60	click	2002.8	10	girl_radio_high
AUS	003600	0000032	00634	60	click	2015.3	11	girl_radio_highest_set
AUS	003600	0000032	00634	60	click	2015.7	12	girl_radio_med
AUS	003600	0000032	00634	60	click	2015.9	13	girl_radio_lowest_set
AUS	003600	0000032	00634	60	click	2016.3	14	girl_radio_low
AUS	003600	0000032	00634	60	click	2016.7	15	girl_radio_high
AUS	003600	0000032	00634	60	click	2017.6	16	bmiQ3_38
AUS	003600	0000032	00634	60	click	2017.6	17	bmiQ3_38
AUS	003600	0000032	00634	60	click	2018.5	18	ItemQuestionText
AUS	003600	0000032	00634	60	click	2024.8	19	girl_radio_low
AUS	003600	0000032	00634	60	click	2026	20	boy_radio_lowest_set
AUS	003600	0000032	00634	60	click	2026.4	21	boy_radio_low
AUS	003600	0000032	00634	60	click	2035	22	bmiQ3_39
AUS	003600	0000032	00634	60	END_ITEM	2036.9	23	NULL

Note. Only the first two students' log traces are shown in the table above for illustration.

For the scored cognitive item response data set, CNT (Country code), NC (National center 6-digit code), SCHOOLID (School ID 7-digit), StIDStd (Student ID), FORMID (CBA form ID), and the item response score variable "CM038Q03T" were extracted. A total of four values were contained in the "CM038Q03T" variable: 1- full score, 0 – no score, 7 – missing value, and 8 – Not reached (OECD, 2013).

To obtain unique identifications for the students, the first 5 columns with contextual information in both data sets were combined respectively for each data set into "NewID" using the LOGAN package (Reis Costa & Leoncio, 2019). Finally, these two data sets were merged in terms of this identification variable "NewID" for further analyses.

With regard to the sample, a total of 1817 students with an average age of 15.93 and an average 10th grade from Australia that have participated in the PISA 2012 BMI task were extracted from the PISA logfile dataset. In 2012 PISA, thirty-two countries were involved computer-based mathematics assessment, where Australia has provided the largest amount sample of students (Reis Costa et al., 2021). It is assumed that the vast amount of data may contain a rich content of responding behaviors that reflect the students' working processes,

which is expected to be sufficient and representative to facilitate the analyses. Thus, the Australian sample was chosen as the main research subject.

Thirteen students were excluded from the sample due to the invalid event values in the log file data set, yielding a sample of 1804 students. Besides, cases were excluded in terms of the item response scores. The current study has excluded the cases that have missing values and that are not reached, resulting in a sample of 1785 students with either 0 or 1 score eventually.

Analytical Framework

Overall, in the field of educational data mining, the analyses of international large-scale assessment log file data in the previous studies consist of three major phases: feature generation, feature selection, and classifier development (Han et al., 2019; Qiao & Jiao, 2018). Inspired by the previous principles, four steps of analysis were performed in the present study to answer the research questions. First, data and theoretical-driven approaches were used for extracting the features from the log files. Second, features were filtered based on the near zero variance measure. Third, three data mining techniques decision trees, random forest, and gradient boosting machines were applied to compare their performance in predicting student outcomes. Finally, feature importance measures were computed.

Feature Generation

Feature generation is considered the most fundamental and critical step in the analysis process of the current study, as it is expected to extract as much information as possible from the log file data set to be used as predictors in the modeling. The log file data including the interaction events, timestamps, and attributes related to the events make respondents' work and cognitive process during the tasks more traceable (Goldhammer et al., 2017, 2021). Specifically, the frequency of sequential patterns related to learning activities, claimed by He et al. (2019), is insightful for understanding, exploring, and interpreting additional

information about individual behaviors. To generate features, empirical and theoretical aspects were taken into consideration on the basis of action sequences and time stamps.

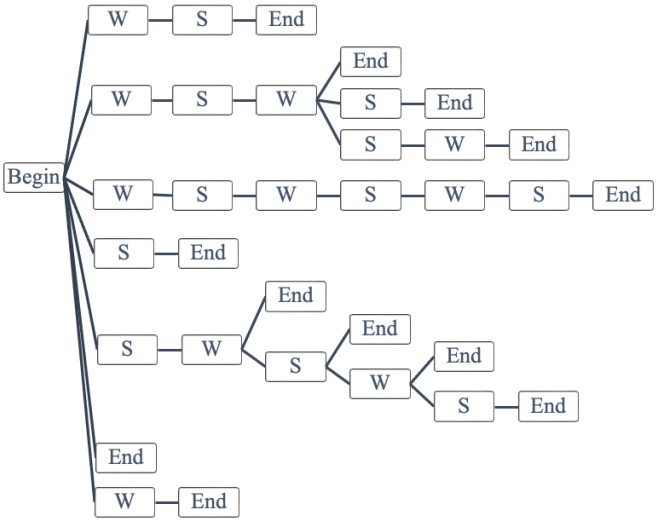
On the one hand, the present study follows the feature types (i.e., n-grams, behavioral indicators, and time-related features) that proved informative in the previous study that has explored the 2012 PISA problem-solving item (Han et al., 2019). Adopted from the natural language processing techniques, n-grams are typically used for the representation of action sequences in a decomposed manner (He & von Davier, 2015). For example, unigrams, which represent the fractions of every single action form the basis of the action sequences. Despite playing an important role in reflecting the fundament of the actions, unigrams provide less information about the process of behavior changes. Bigrams, trigrams, and other n-grams that connect a number of adjacent actions in chronological order, contrarily, are able to capture the delicate shifts. In the present study, action sequences are formed by general clicking actions on three parts of the screen: the web (W), the statement (S), and other areas (O). For example, an action sequence BWSOE stands for that a student starts the task, clicks the checkboxes on the web, clicks on the options in the statement, clicks on the task instruction, and ends the task. Unigrams, bigrams, and trigrams were extracted accordingly from the possible actions (i.e., unigram: W, S, O; bigram: BW, WS, SO, OE; trigram: BWS, WSO, SOE).

Different from the n-grams, behavioral indicators are considered in a more comprehensive aggregated level that reflects beyond ordered events but a combination of actions with higher-order decisions from the test takers (He et al., 2019). They are particularly important for tasks such as problem-solving due to their intrinsic properties that reveal sophisticated interaction with the user interface and high-dimensional thinking towards desirable solutions. During the task of the present study, students are flexible in deciding the navigating paths with certain procedures such as selecting directly on the statement without

other navigations, comparing the line graphs before selecting options in the statement, or clicking on the statement first and then verifying their choices on the website. Other students may choose to go back and forth between the website and the statement to accomplish this task. It is assumed that whether students click on the website or statement first and how many times they have changed the click actions may imply students' thinking processes, decisiveness as well as hesitancy during the task (Han et al., 2019). Consequently, the "WS sequence" that indicates the order and even the hesitation in which a student may interact with the item was extracted.

Figure 2

Example of Behavior Indicators



Type	WS_sequence	n_actions
1	WS_only	4
2	1<=WS<3	5
2	1<=WS<3	6
2	1<=WS<3	7
3	WS>=3	8
4	S_only	3
5	Sequence_Start_from_S	4
5	Sequence_Start_from_S	5
5	Sequence_Start_from_S	6
5	Sequence_Start_from_S	7
6	Incomplete	2
6	Incomplete	3

Figure 2 exemplified the composition of "WS sequence" and the corresponding number of actions for each type. Following Han et al. (2019), when extracting the behavior indicators, successively repeated behaviors were combined into a single action to keep the key action steps, while "O" was excluded from the action sequence since it is less associated with the decisions between W and S. For example, an action sequence BWSSWWWOSE can be shortened to BWSWSE as illustrated in Figure 2 as the second behavioral indicator from Type 2.

While delineating the types for “WS_sequence”, the starting action was considered, that is, whether a student begins with an action that clicks on the web browser(W) or the statement (S). Then, they were further divided into various branches in terms of the number of “WS” actions. For the action sequence starting from W, three types of values were generated: “WS_only” that contains only one “WS”, “1<=WS<3” that contains more than one but less than two “WS” actions, and “WS>=3” that contains more than three “WS” actions(Han et al., 2019). For the action sequence that begins with “S”, two types of values “S_only” and “Sequence_Start_from_S” were generated as illustrated in Figure 2. Besides, a category “Incomplete” with incomplete actions was generated. Other than the “WS sequence”, the total number of actions (n_action) counting from “B” to “E” for each student was also extracted as a behavior indicator.

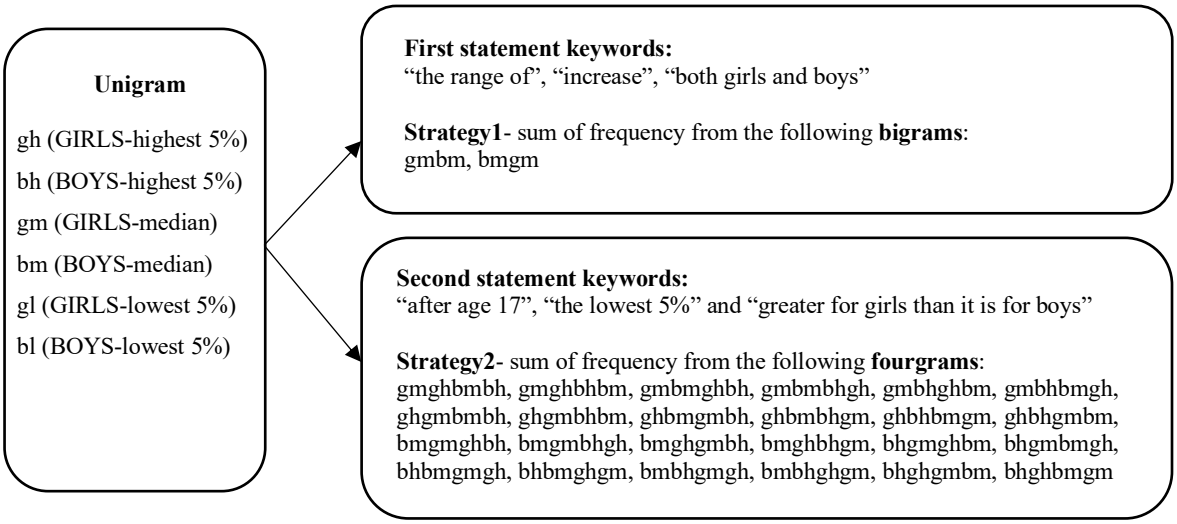
Aside from the features of action sequences, timing data is considered to be insightful for predicting students’ cognitive performance (He et al., 2019). The accumulated time spent on the task is expected to indicate the item difficulty to different students, while the time spent on specific events could reveal the importance of different actions. In this study, three types of the time-related features were generated: total time spent on the task (AUS.TOT), the time spent on each “W”, “S”, “O” action (duration.pos.S, duration.pos.W, duration.pos.O) , and the time spent from the beginning of the task to the first “W”, “S”, or “O” action (durationStart.pos.W, durationStart.pos.S, durationStart.pos.O).

On the other hand, theoretical grounding was considered in terms of students’ action sequences and possible strategies suggested by the framework of PISA CBAM and the point of view of mathematics educators (Bardini, 2015). As mentioned earlier, to accomplish the task, students were expected to apply certain strategies to determine each statement in terms of their knowledge and understanding of mathematical concepts. Although the priority of click actions is not an essential concern for both statements, it is beneficial to extract both the

ordered action sequence and total frequencies regardless of the order to acquire as much information as possible. Therefore, features were extracted in terms of the theoretically expected strategies in the form of n-grams and the sum of combinations as displayed in Figure 3.

Figure 3

Illustrations for Strategy-related Features



Note. The strategies for two statements require clicks without the order of priority, thus the sums of frequencies were computed.

Unigram features "gh" (GIRLS- highest 5%), "bh" (BOYS- highest 5%), "gm" (GIRLS- median), "bm" (BOYS- median), "gl" (GIRLS- lowest 5%), and "bl" (BOYS- lowest 5%) that composed the basis for the six checkboxes were generated primarily. Then for each statement, corresponding features were generated in terms of the strategies that may lead to students' success in this task. For the first statement, the expected strategy is to click (i.e., unselect) on the "BOYS- median" and "GIRLS- median" without certain priority. Bigrams such as "gmbm" and "bmgm" along with the sum of their frequencies were thus generated. In the spirit of obtaining more exploratory information, bigrams for other possible combinations were also considered. For the second statement, it is anticipated to click on the "GIRLS- highest 5%", "BOYS- highest 5%", "GIRLS- median", and "BOYS- median"

without particular order. As a consequence, fourgrams that relate to the strategy for the second statement were developed.

In summary, features were generated based on two perspectives. Similar to Han et al. (2019), three types of features were considered in the phase of feature generation from a general data-driven perspective of view: a) n-gram features, b) behavioral indicators, and c) time-related features. While from a theoretical perspective of view, focusing on the actions on the simulated web browser, the d) strategy-related n-gram features were generated.

Weighted features. The frequency for each action-related feature was computed for additional information such as the repeat rate of the binary values (i.e., click vs. no click). Han et al. (2019) utilized binning that groups the frequencies into equal percentiles to reduce the data sparsity, but it might face a challenge of information loss. To avoid this issue, weighted frequencies were computed for the action features such as n-grams and strategy-related features to balance the extreme action occurrences that entail little information in the prediction as mentioned in the methods (He & von Davier, 2015). An action that occurs more frequently in all the observations, such as "START_ITEM" that repeats in every observation, will significantly weaken the feature's ability to predict the outcome groups due to high consistency. Also, actions with high frequencies tend to exhibit higher feature importance given their dominant occurrence in all the observations. It is worth noticing and distinguishing actions with a high frequency of occurrence in one observation from the actions with lower frequency in one observation but occurred in a large number of observations. These weighted features instead of features with raw frequencies will be further used in the modeling part.

He and von Davier (2015) described in detail how the weights of the features are calculated. They have adopted the concept from text mining and transformed the inverse document frequency (IDF) into inverse sequence frequency (ISF) to lower the effect of

overly repeated features. Besides, they also combined the methodologies from natural language processing to mitigate the effect of frequent occurrence. An action's term weight thus can be expressed as:

$$weight(i,j) = \begin{cases} \left[1 + \log\left(tf_{ij}\right)\right]\log\left(N/sf_i\right), & if\ tf_{ij} \geq 1 \\ 0 \qquad\qquad\qquad\qquad\qquad\quad , & if\ tf_{ij} = 0 \end{cases}, \tag{1}$$

where tf_{ij} is the frequency that an action i shows in a student's action sequence j, sf_i represents the number of students that the action i has occurred, and N means the total number of students in the sample. The data management and feature generation were conducted in R software using LOGAN package (Reis Costa & Leoncio, 2019).

Feature filtering

Before modeling, a descriptive analysis for the evaluation of the generated features was conducted. Putting a large number of features into the model will not only increase the difficulty of model interpretation but also be time demanding (Boehmke & Greenwell, 2019). Therefore, an analysis of (near) zero variance was applied to filter the features.

Zero or near-zero variance is concerned with the fact that a feature consists of only one single value or few values that this feature may enclose little information in the prediction. The detection of near-zero variance is computed in terms of two rules of thumbs: a) the number of unique values over the sample size is equal to or lower than 10. b) The ratio between the value with the highest frequency and the second-highest frequency is equal to or larger than 20% (Boehmke & Greenwell, 2019; C & RamaSree, 2015). For reproducibility, the R package LOGANTree (Reis Costa and Qin, 2022) was developed to compute these measures.

Modeling

In the current study, three tree-based machine learning techniques including the decision tree, random forest, and stochastic gradient boosting machine were utilized to predict students' performance on the CM038Q03 task. Using LOGANTree with functions

based on caret and caretEnsemble package (Kuhn, 2021; Deane-Mayer & Knowles, 2019), four general steps for each method were conducted collectively for three methods: a) splitting the original data set into training and testing data; b) training the models with the cross-validations on the training data set and tuning the models with different parameters across the resamples; c) comparing the models to find the optimal one with the best parameter; d) fitting the final model into the testing data set.

Training and Testing Sets. With a stratified sampling scheme, 70% of the data set was split into a training set (n = 1250) and 30% into the testing set (n = 535), while making sure that both sets have the same distribution of the levels as the outcome variable "CM038Q03T" (i.e, correct and incorrect answers). For replication of the results, the seed was set to 2022 with an R version4.1.1 (R Core Team, 2021).

Algorithms and hyperparameters. ***Decision Tree.*** For the decision tree method in the present study, the renowned Classification and Regression Tree (CART) algorithm was utilized to build a classification tree (Boehmke & Greenwell, 2019; Breiman et al., 2017). By performing a binary recursive partition, the training data are divided into subgroups with alike outcome values. Each subgroup is considered a node in the decision tree, where for each node the finest feature that maximizes the gaining of impurity is determined for the partition. According to Coppersmith et al. (1999), two mostly applied impurity measures, namely the splitting rules, are Gini Index applied in CART and Entropy used in C4.5. The current study utilized Gini Index to select the outperformed feature in each node, where in each node the dominant observations are from a single class in the outcome. The Gini Index is computed by

$$G = \sum_{k=1}^{n} \hat{p}_{mk}(1 - \hat{p}_{mk}), \tag{2}$$

where $\hat{p}_{mk}$ denotes the proportion that the observations in the training data set from the kth class in the m_{th} region (James et al., 2021). In this study, two classes: "correct" and

"incorrect" were contained in the outcome variable. The region refers to the subgroups that the observations have been assigned. The goal of the partitioning in the classification is to retain as homogenous as possible the final nodes. A smaller value of the Gini Index is preferred as this indicates that the major observations in the region were from the same single class and the total variance across the k classes was minimized.

As the feature space split recursively, the growth of the decision tree faces the problem of overfitting when the tree is too complex that the bias is lowered at the cost of lacking the generalizability to different datasets. To address this issue, the pruning approach was applied to balance the trade-off between the tree complexity and the model fit (Boehmke & Greenwell, 2019). Instead of restricting the tree growth, pruning grows an excessively complex tree first. Then the tree is pruned to a series of subtree T using a range of cost complexity parameters (α) that penalizes the function for the terminal node number |T|:

$$minimize\left\{\sum_{m=1}^{|T|}\sum_{i:x_i\in R_m}\left(y_i-\hat{y}_{R_m}\right)^2+\alpha|T|\right\} \quad (3)$$

where R_m denotes the subset of features in the m_{th} terminal node, and $\hat{y}_{R_m}$ denotes the outcome predicted by features from R_m (James et al., 2021). An optimal subtree is eventually determined with the best complexity parameter α.

Random Forest. The random forest is an ensemble method developed from bagging where a collection of decision trees are aggregated for the prediction (Boehmke & Greenwell, 2019). A single decision tree is easily interpreted but suffers from high variance and low accuracy (Boehmke & Greenwell, 2019). Through bagging, a large number of trees are built with randomness in extended datasets that reduce the variance for the model. However, bagging considers the most influential feature when the tree starts to split. As a result, these trees perform to be highly correlated which impairs the improvement of variance. Random

forest, on the other hand, randomly subsets only a few features in each split to lower the correlation among the trees and thus enhance the model performance.

The general algorithm used for random forest is:

First, determining the number of trees in the forest. Second, resampling from the original data set. Third, growing each of the tree using the resampled data. For each split in each tree, randomly selecting m_{try} features from the entire group of features and determining the best feature for the split. Last, splitting the nodes until the stopping criteria such as the minimum node size has been met.

To train the random forest model, four tuning hyperparameters were thus considered in the current study: the number of trees, the splitting rule, the number of features used in each split (m_{try}), and the minimum terminal node size. The number of trees was settled before the training process starts. According to Boehmke & Greenwell (2019), a larger number of trees is required to reach a stable error rate. Using the R functions developed from the ranger package **(**Wright & Ziegler, 2017), the number of trees for the random forest is fixed to 500 by default. Regarding the m_{try}, commonly the number of features is set to the square root of the total number of features. In the current study, a grid of m_{try} as shown in Table 2 was fitted in developing the classifier to find the best value. In addition to the Gini Index, a computation efficient algorithm, the extremely randomized trees(extratrees), were performed in the ranger package with a grid search for the splitting rules. Different from the Gini Index utilizing all the possible splitting values, the extremely randomized trees select one single value and partition the regions fully at random without referring to the outcome variable (Geurts et al., 2006). By default, the minimum terminal node size was held 1 in the modeling.

Gradient Boosting Machine. Similar to the random forest, the gradient boosting machine is also an ensembled data mining technique. However, rather than generating

abundant complex decision trees parallelly, the gradient boosting machine builds trees sequentially to improve the residuals from the previously grown tree to boost the model performance (Boehmke & Greenwell, 2019; James et al., 2021). The algorithm for gradient boosting starts with a weak learner, that is, a smaller tree with few nodes that the tree performance is nearly equivalent to a random guess. Then, the next tree model is fitted using the residuals from the previous tree as the outcome variable. With the new residuals, another tree is fitted again, and this process continues until the best-tuned parameters have been found in the resampling process. During the training process, the parameters are adjusted iteratively through a stochastic gradient descent algorithm that measures the local gradient of the function for residuals and proceeds to obtain its minimum (Boehmke & Greenwell, 2019). The stochastic gradient descent extracts a random sample from the training data set to train the subsequent trees in order to reach a global minimum that avoids local minimum values resulting from the irregular convex.

To train the gradient boosting machine model, four types of hyperparameters were used: the number of trees, learning rate (shrinkage), tree depth, and the minimum terminal node size. The number of trees refers to the trees that have been built in sequence. A grid of numbers has been attempted in the training as displayed in Table 2. The learning rate is used to determine the size of iterations in the gradient descent, where a smaller value leads to considerable iterations while a larger value may indicate the chance to miss the minimum loss. In the caret package, the learning rate was held constant at 0.1 by default in the model training. As indicated in Table 2, three values for the tree depth were searched in the training, while the minimum terminal node size was held constant at a value of 10. Given the binary classification models in the present study, an argument “distribution = bernoulli” is specified in the function using the caret package.

Table 2

Grid of Hyperparameters Used in the Modeling

Models	Hyperparameter Values
Decision tree	Complexity parameter = c(0.0116, 0.0221, 0.6838)
Random forest	Number of trees = 500 Split rule = gini Number of variables (mtry) = c(2, 35, 69) Minimum value of the node size = 1
Gradient boosting	Number of trees = c(50, 100, 150) Learning rate = 0.1 Tree depth = c(1, 2, 3) Minimum number of observations in terminal nodes = 10

K-fold Cross-validations. A ten-fold cross-validation resampling method was performed in each of the three tree-based learning techniques to discover the optimal parameter as well as to ensure the generalization performance of the models. In the cross-validation, the training data were divided into ten equal-sized subsamples where nine of them were treated as the training set and the remaining one was assigned as the validation set. The tree models were thus fitted in each of the training sets, while the validation set was used for evaluating the model performance. This process was repeated ten times with each of the ten folds serving as the validation set for each time. By averaging the results, generalization error and a final estimation of hyperparameters were determined.

Model Evaluation

To evaluate the model performance, the predictive accuracy is commonly measured by the loss function that computes the error from the predicted values and the actual values in the data set (Boehmke & Greenwell, 2019). When facing the classification issues, a confusion matrix that tabulates the predicted and actual events is usually presented. Four terms related to the prediction are generated from the confusion matrix: a) true positive, where the predicted events actually happened as predicted; b) false positive, where the predicted events

did not happen; c) true negative, the not happened events are not predicted; d) false negative, the events that are not predicted happened in fact. Through the confusion matrix function from caret package, error metrics such as Accuracy, Kappa, Precision, Sensitivity, and Specificity for binary classifiers were computed for the model evaluation.

Accuracy(Hanley & McNeil, 1982) quantifies the proportion of the correctly predicted observations to the total observations. It is computed as:

$$\text{Accuracy} = \frac{TP+TN}{total\ cases}, \tag{4}$$

where TP stands for true positive, while TN refers to the true negative.

Kappa(Cohen, 1960) is defined as:

$$\text{Kappa} = \frac{Accuracy - Random\ accuracy}{1 - Random\ acuracy}, \tag{5}$$

$$Random\ accuracy = \frac{TP+FN}{total} \cdot \frac{TP+FP}{total} + (1 - \frac{TP+FN}{total})(1 - \frac{TP+FP}{total}), \tag{6}$$

where TP represents the number of true positive cases, FN represents the count of false negative cases and FP is the number of false positive cases. A kappa value larger than .6 is considered acceptable (McHugh, 2012).

Precision (Davis & Goadrich, 2006) conveys the extent to which the machine learning model can predict the events accurately. It can be expressed mathematically as:

$$\text{Precision} = \frac{TP}{TP+FP}, \tag{7}$$

where TP is the number of true positive observations, and FP is the number of false-positive observations. It quantifies the proportion of the truly happened events that have been predicted over the total predicted events.

Sensitivity, also known as recall (Boehmke & Greenwell, 2019), measures the proportion between the number of correctly predicted actual events and the total number of actual events.

$$\text{Sensitivity} = \frac{TP}{TP+FN}, \tag{8}$$

where TP represents the count of true positive observations, and FN represents the count of false negative observations. Through sensitivity, we are able to examine how accurately a machine learning model can classify the actual events.

On the contrary, specificity (Boehmke & Greenwell, 2019) measures the accuracy for classifying actual non-events. This metric computes the ratio between the number of correctly predicted non-events and the total number of actual non-events:

$$\text{Specificity} = \frac{TN}{TN+FP}, \tag{9}$$

where TN is the number of the true negative observations, and FP represents the number of the false-positive observations.

The values of the error metrics mentioned above range from 0 to 1. A higher value of the result reveals a better performance of the model.

Besides, the area under the curve (AUC) was measured in the form of receiver operating characteristics (ROC) curves where the false positive rate is plotted on the x-axis against the true positive rate on the y-axis (Fawcett, 2006). It is considered that the false positives and false negatives are minimized when a classifier has a better performance in the precision and sensitivity (Boehmke & Greenwell, 2019). Thus, a larger area under the curve is more optimal for a model.

Feature Importance

Apart from the modeling, what information can we extract from the model prediction structure remains interesting. To answer the third research question, chi-square statistics and feature importance were applied to characterize the interpretation of predictive features.

Independent of the models themselves, Chi-square statistics is considered a descriptive feature importance method that reflects the robustness of a feature in the

prediction. He and von Davier (2015) adopted it from natural language processing to identify the robust features with higher information for predicting different classes in the outcome. By calculating the chi-square statistics, the extent of independence between the feature occurrence and item response outcomes is determined. In the present study, the chi-square statistics were computed using the LOGANtree package. Firstly, contingency tables for each action feature were established. For each table, two columns "correct" and "incorrect" for each class j of the outcome were crossed by two rows where one was the weighted frequency of the action i, the other is the total weighted action frequency minus the weighted frequency of action i. Then, the chi-square statistic for each contingency table was computed with a null hypothesis that there is no difference between the feature occurrence and the correctness of outcomes:

$$\chi^2 = \frac{M(O_{11}O_{22} - O_{12}O_{21})}{(O_{11} + O_{12})(O_{11} + O_{21})(O_{12} + O_{22})(O_{21} + O_{22})}, \quad (10)$$

where M represents the total number of observations, and O_{ij} is the weighted frequencies in the contingency table. As a measure of association, a higher chi-square statistic for a feature implies a stronger correlation with the outcome (He & von Davier, 2015). Thus, features with higher chi-square scores are deemed as more robust features for prediction in the modeling.

Feature importance is considered a model-based measure that reveals the quantities that features have contributed to the model prediction (Greenwell et al., 2018). In the current study, the feature importance for each tree-based method was computed using the function from caret package, where the importance scores are standardized ranging from 0 to 100 in the feature importance plot. In the decision tree method, features are repetitively selected in each node to split the data into subgroups. Given this nature, the feature importance in the present study was measured by the sum of squared improvement in the AUC for each of the features selected in the tree nodes (Greenwell et al., 2018). For the gradient boosting machine method, the feature importance was computed following the same rule but averaged across

the ensembled trees. With regard to the random forest, a permutation method was incorporated, where the feature importance was computed by the difference between the AUCs without and with randomly shuffling each feature in the validation set. Then, the difference in the AUCs for each feature was averaged across the trees in the ensembled forest. A feature is considered important if the AUC reduces after permuting it in the validation set (Greenwell et al., 2018).

Scores computed from these two methods are useful for filtering the features for further analysis, as they reveal the influential characteristic of the features that contribute to the modeling. Nevertheless, using both model-based and descriptive methods, the main purpose here was to highlight the relationship between the input features and the outcome variable.

Results

In this section, the results are presented in the order of three research questions. First, it provides a description of the descriptive statistics, extracted features, and feature filtering results. Then, the results for model training, tuning, and model evaluation are reported. Finally, models are interpreted by means of the feature importance plot and the Chi-square statistics.

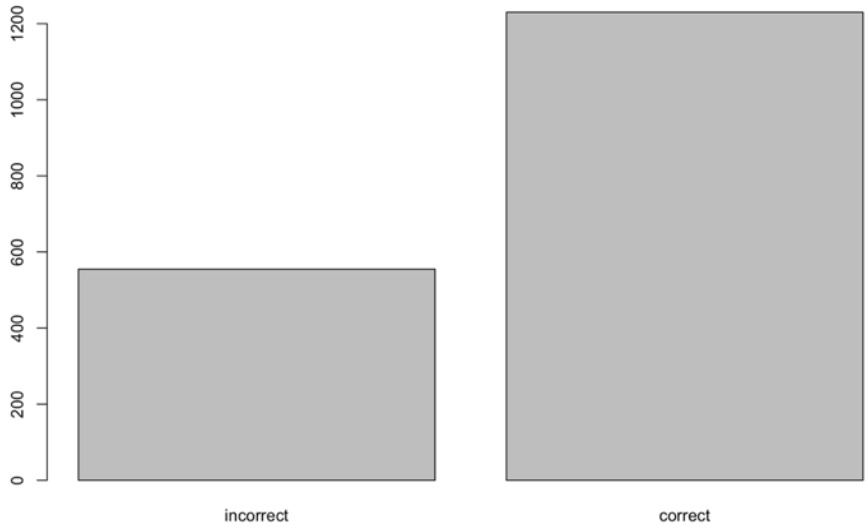
RQ1: Feature generation

Using the logfile data, the working and cognitive process for each student was reflected in an action sequence. In the 2012 PISA log file data set, the number of single-clicking actions per sequence among all 1785 students in the sample ranges from 2 to 67 (mean = 10.26, SD = 8.98, median = 6). Among the students, 1230 of them have correctly accomplished the task, which takes up 69% of the whole Australian sample as displayed in Figure 4. An overview of the summary statistics for students' time spent on the task in minutes and their item response performance can be found in Appendix III. Whilst the

average time spent on task for both item response outcome classes show similar results, students who performed their success in the task spent 0.08 minutes longer than students who did not manage to achieve so.

Figure 4

Distribution of Students' Item Response Outcome



A total of 110 features were generated through both empirically data-driven and theoretical-based methods corresponding to four feature types as exhibited in Table 3: N-grams, behavioral indicators, time-related features, and strategy-related features. Among them, 67 n-gram features including unigrams, bigrams, and trigrams were derived with weighted frequencies. Two behavioral indicators "WS sequence" and "Number of actions" were extracted in their original scale: categories and the numbers respectively. Seven time-related features including the total time spent on the task ("AUS.TOT"), the time spent on each WSO event ("duration.pos.S", "duration.pos.W", "duration.pos.O"), and the time between the start of the task and the first WSO action ("durationStart.pos.S", "durationStart.pos.W", "durationStart.pos.O") were extracted in minutes. Regarding the strategy-related features, 32 n-gram features instructed by the theoretically expected strategies were extracted initially, followed by 2 features that sum up the frequencies for each

strategy corresponding to each statement. Additional descriptive statistics for all the generated features are presented in Appendix III.

Table 3

Features(n=110) Generated for CM038Q03 Item

Feature type		Feature	Scale
a) N-grams (67)	Unigrams	W, S, O	Weighted
	Bigrams	BW, BS, BO, BE, WW, WS, WO, WE, SW, SS, SO, SE, OW, OS, OO, OE	Weighted
	Trigrams	OWW, OOS, BSW, WWS, OOE, WSE, OOO, OSW, BWS, OWE, SOO, SSE, WWW, WOS, OSS, WOO, SWO, SSO, WOW, BWO, SSS, OOW, SOW, SSW, OWS, OSE, WSS, SWE, BSO, OWO, BOS, BSE, OSO, SOS, BWE, WSO, SWW, SWS, BWW, WOE, WWO, WSW, BSS, BOO, BOW, SOE, WWE, BOE	Weighted
b) Behavioral indicators (2)		WS sequence Number of actions (n_action)	Original
c) Time-related features (7)		AUS.TOT, duration.pos.S, duration.pos.W, duration.pos.O, durationStart.pos.S, durationStart.pos.W, durationStart.pos.O	Original
d) Strategy-related features (34)	Unigrams	gm, bm, gl, bl, gh, bh	Weighted
	Bigrams	gmbm, bmgm	Weighted
	Fourgrams	gmghbmbh, gmghbhbm, gmbmghbh, gmbmbhgh, gmbhghbm, gmbhbmgh, ghgmbmbh, ghgmbhbm, ghbmgmbh, ghbmbhgm, ghbhbmgm, ghbhgmbm, bmgmghbh, bmgmbhgh, bmghgmbh, bmghbhgm, bmbhgmgh, bmbhghgm, bhgmghbm, bhgmbmgh, bhbmgmgh, bhbmghgm, bhghgmbm, bhghbmgm	Weighted
	Sum of frequencies	Strategy1, Strategy2	Weighted

Further, the generated features were filtered with respect to the measures of zero variance and near-zero variance as detailed in Table 4. In this process, forty-five features were identified where nine of them with zero variance and thirty-six with near-zero variance. These features were excluded from the tree-based modeling in the next following steps as they contain very little information.

Table 4

Overview of the Filtered Features(n=45) with Zero and Near Zero Variance

	Feature	Zero variance	Near zero variance	Percentage of Unique Values	Frequency Ratio
1	wgt.freq.BE	FALSE	TRUE	0.16	311.50
2	wgt.freq.WE	FALSE	TRUE	0.16	43.64
3	wgt.freq.WSO	FALSE	TRUE	0.24	23.43
4	wgt.freq.SSS	FALSE	TRUE	0.72	22.11
5	wgt.freq.SSW	FALSE	TRUE	0.16	88.29
6	wgt.freq.SWS	TRUE	TRUE	0.08	0.00
7	wgt.freq.SWO	FALSE	TRUE	0.16	39.32
8	wgt.freq.SOO	FALSE	TRUE	0.24	23.47
9	wgt.freq.OSW	FALSE	TRUE	0.16	22.58
10	wgt.freq.OWS	FALSE	TRUE	0.24	27.39
11	wgt.freq.OWO	FALSE	TRUE	0.48	27.14
12	wgt.freq.BWS	TRUE	TRUE	0.08	0.00
13	wgt.freq.BWO	FALSE	TRUE	0.16	137.89
14	wgt.freq.BOW	FALSE	TRUE	0.16	29.49
15	wgt.freq.SWE	FALSE	TRUE	0.16	1249.00
16	wgt.freq.WWE	FALSE	TRUE	0.16	47.08
17	wgt.freq.WOE	FALSE	TRUE	0.16	311.50
18	wgt.freq.OWE	FALSE	TRUE	0.16	1249.00
19	wgt.freq.OOE	FALSE	TRUE	0.16	311.50
20	wgt.freq.BSE	FALSE	TRUE	0.16	155.25
21	wgt.freq.BWE	TRUE	TRUE	0.08	0.00
22	wgt.freq.BOE	FALSE	TRUE	0.16	1249.00
23	wgt.freq.bmgm	FALSE	TRUE	0.24	28.64
24	wgt.freq.gmbmghbh	FALSE	TRUE	0.16	311.50
25	wgt.freq.gmbmbhgh	FALSE	TRUE	0.16	137.89
26	wgt.freq.gmbhghbm	TRUE	TRUE	0.08	0.00
27	wgt.freq.gmbhbmgh	FALSE	TRUE	0.16	1249.00
28	wgt.freq.ghgmbmbh	FALSE	TRUE	0.16	103.17
29	wgt.freq.ghgmbhbm	FALSE	TRUE	0.24	112.55

	Feature	Zero variance	Near zero variance	Percentage of Unique Values	Frequency Ratio
30	wgt.freq.ghbmgmbh	TRUE	TRUE	0.08	0.00
31	wgt.freq.ghbmbhgm	FALSE	TRUE	0.16	311.50
32	wgt.freq.ghbhbmgm	FALSE	TRUE	0.24	415.33
33	wgt.freq.ghbhgmbm	FALSE	TRUE	0.16	624.00
34	wgt.freq.bmgmghbh	FALSE	TRUE	0.16	249.00
35	wgt.freq.bmgmbhgh	TRUE	TRUE	0.08	0.00
36	wgt.freq.bmghgmbh	TRUE	TRUE	0.08	0.00
37	wgt.freq.bmghbhgm	FALSE	TRUE	0.16	1249.00
38	wgt.freq.bmbhgmgh	FALSE	TRUE	0.16	112.64
39	wgt.freq.bmbhghgm	FALSE	TRUE	0.16	61.50
40	wgt.freq.bhgmghbm	FALSE	TRUE	0.16	624.00
41	wgt.freq.bhgmbmgh	TRUE	TRUE	0.08	0.00
42	wgt.freq.bhbmgmgh	FALSE	TRUE	0.16	311.50
43	wgt.freq.bhbmghgm	FALSE	TRUE	0.16	624.00
44	wgt.freq.bhghgmbm	FALSE	TRUE	0.16	624.00
45	wgt.freq.bhghbmgm	TRUE	TRUE	0.08	0.00

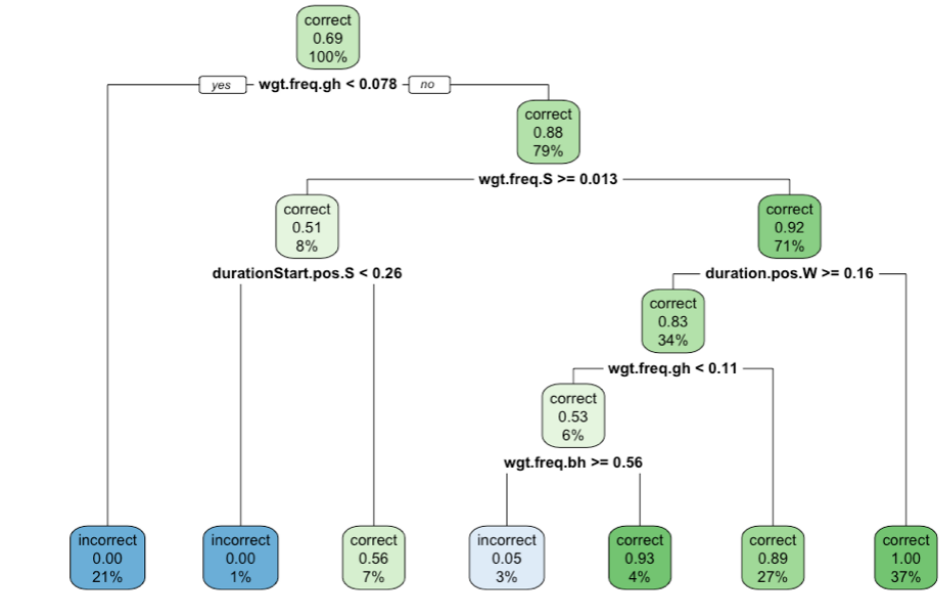
RQ2: Predictive Analysis

For the decision tree method, a tree was built as illustrated in Figure 5. The root node contains full 1250 observations from the training data set, where 69% of them are with an actual outcome of “correct”. This root node was split into two groups, depending on the feature “wgt.freq.gh”, that is, whether the weighted frequency of clicking on the “GIRLS-highest 5%” checkbox is less or larger than 0.078. For the observations with clicking frequencies less than 0.078, they were classified into “incorrect” group that takes up 21% of the whole training sample. For those 79% observations with frequencies higher than 0.078, a dominant 88% of the observations were classified into the “correct” class. This group was further partitioned by the feature “wgt.freq.S” which stands for the weighted frequency of clicking on the statement into two groups that are classified as “correct”. These two groups were further split in terms of two time-related features “durationStart.pos.S” (the time spent between the start of the task and the first click on the statement) and “duration.pos.W” (the time spent on clicking the web browser) respectively as illustrated in the tree plot. For those

who spent no less than 0.16 minutes on the web browser, these observations were again binarily divided into "correct" groups in terms of the feature "wgt.freq.gh" and those with weighted frequencies less than 0.11 were classified eventually by the feature "wgt.freq.bh" (weighted frequency of clicking on "BOYS-highest 5%").

Figure 5

Decision Tree Plot

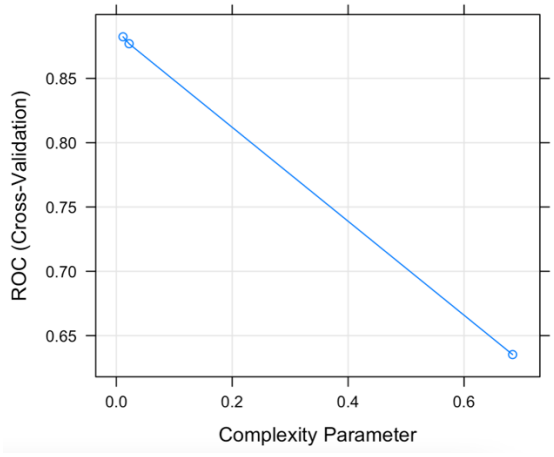


During the process of classifier development, a 10-fold cross-validation resampling was performed for each tree-based method. Figure 6 presents a grid search of hyperparameters in the cross-validation, where the value of parameters on the x-axis is plotted against the cross-validated ROC rate on the y-axis. As indicated in the plots, for the decision tree method, three complexity parameters were fitted in the tuning process, showing a monotonically decreasing trend. For the random forest method, two splitting rules were applied with a grid search for three values of m_{try}. Both of the lines showed a positive relationship between the number of features randomly tried at each split and the cross-validated ROC. Nevertheless, Gini Index performed better before the second $m_{try} = 35$, while the extremely randomized trees (extratrees) showed better ability after the second

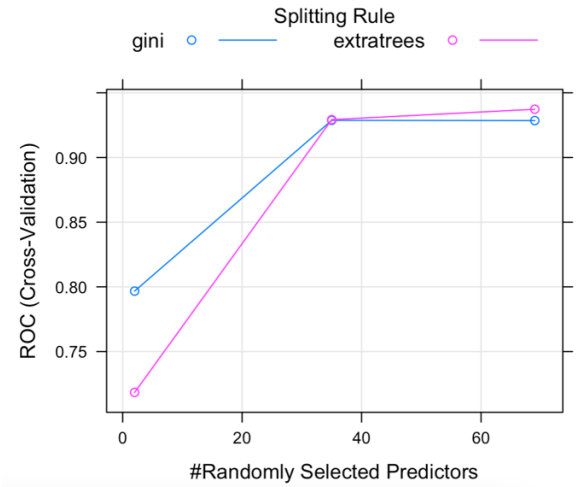
point. For the gradient boosting machine, a grid search for max tree depth and the number of boosting iterations is plotted against the cross-validated ROC. It can be observed from the graph that as the number of boosting iterations increase, the ROC shows an overall increasing trend. A larger maximum tree depth associated with a higher ROC value.

Figure 6

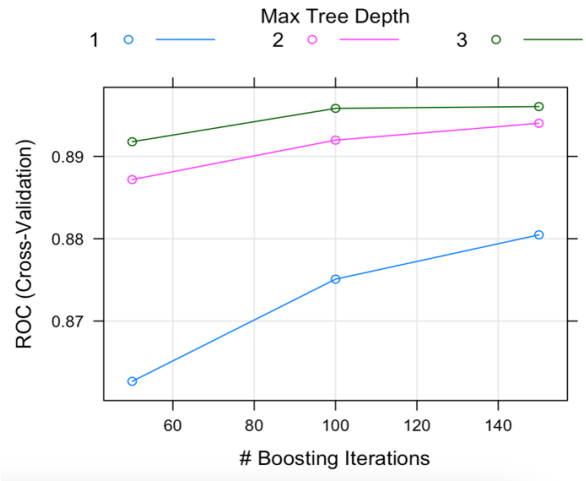
Selection of the Hyperparameters via Cross-Validation by Model



(a) Decision Tree



(b) Random Forest



(c) Gradient Boosting Machine

The optimal parameters used in the final model for each method are summarized in Table 5. As indicated in the table, for the decision tree method, the final value for the complexity parameter is 0.0116. For the random forest modeling, 500 trees were built in the

forest. For each tree, the number of features randomly selected at each split was 69, while the minimum size for each node was 1. For the gradient boosting machine, 150 trees were established in the iteration with a fixed learning rate of 0.1. The minimum node size was determined to be 10, and the depth of trees was selected as 3 in the final model.

Table 5

Summary of Model Parameters

Decision Tree	
Method	CART by rpart
Split rule	gini
Number of resampling iterations	10
Final value for complexity parameter	0.0116
Random Forest	
Method	Random Forest by ranger
Split rule	extratrees
Number of resampling iterations	10
Number of variables tried at each split (mtry)	69
Minimum value of the node size	1
Number of trees	500
Gradient Boosting Machine	
Method	Stochastic Gradient Boosting by gbm
Distribution	bernoulli
Number of resampling iterations	10
Learning rate	0.1
Minimum value of the node size	10
Tree depth	3
Number of trees	150

Table 6 displays the results of the model evaluation computed using the confusion matrices in Appendix III and the equations (4) to (9) mentioned in the methods. Overall, all three tree-based methods showed a satisfying accuracy with values larger than 0.9. This indicates that the classifiers could correctly predict the outcome most of the time. Among them, the decision tree method appeared to have the highest prediction accuracy, while the random forest showed slightly inferior. Kappa presented a similar picture, where the decision

tree showed the best performance (Kappa=0.81) while the random forest performed slightly lower (Kappa=0.80). With Kappa values larger than 0.6, all three methods again showed an acceptable performance in the prediction (Cohen, 1960). Besides, all the tree-based methods had achieved an excessively high value on the sensitivity, indicating that the actual correct cases were classified with high accuracy. For specificity and precision, random forest performed the best among these three tree-based methods. Additionally, the ROC curves presented in Figure 7 indicate good performance of three models, given the large areas under the curves. All three methods demonstrated a similar prediction performance, yet the random forest generally outperformed other methods.

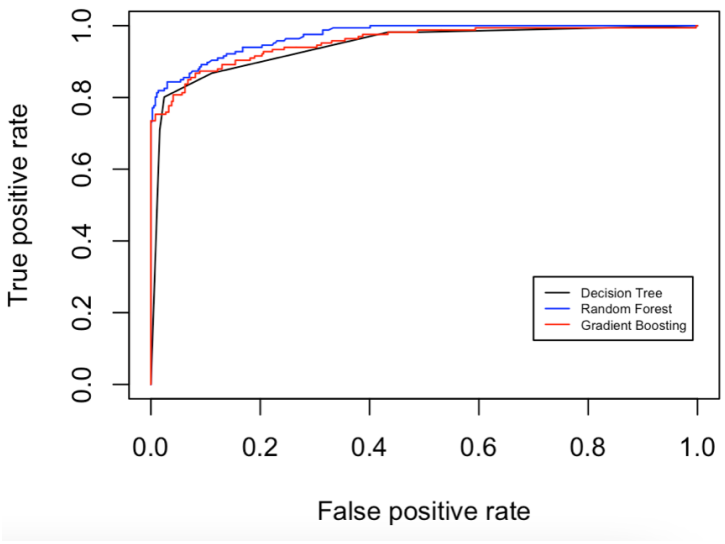
Table 6

Performance of Three Tree-based Models

	Decision Tree	Random Forest	Gradient Boosting
Accuracy	0.92	0.92	0.91
Kappa	0.81	0.80	0.78
Sensitivity	0.98	0.95	0.96
Specificity	0.80	0.85	0.81
Precision	0.92	0.93	0.92

Figure 7

ROC Plot for Three Tree-based Models

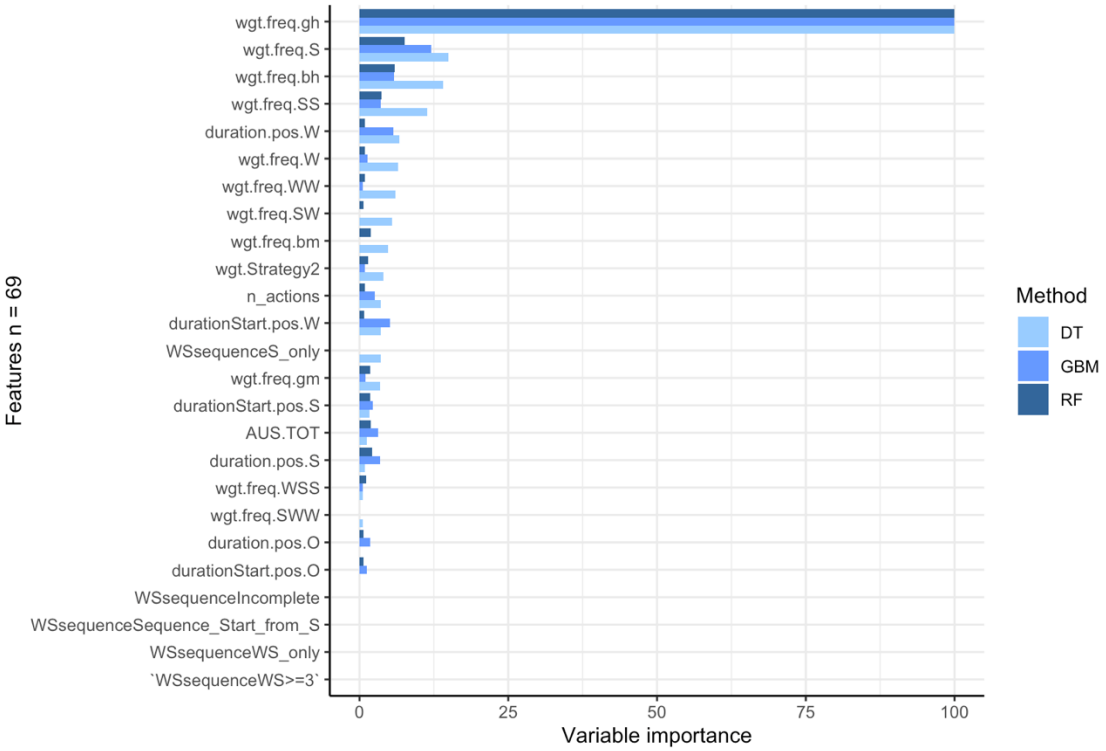


RQ3: Feature Importance

The scaled feature importance for the decision tree, random forest, and gradient boosting machine method are plotted collectively for comparison as shown in Figure 8. The plot depicts the relevant feature importance with standardized values ranging from 0 to 100 which corresponds to low to high importance. A total of 65 features were put into each tree-based model, while only a selection of the top-ranked 25 features were shown in the plot given the rest of the features provide insufficient importance.

Figure 8

Plot for Top-ranked 25 Variable Importance of Three Tree-based Models



Note. A total of 65 features were put into the tree-based models. Feature n = 69 is shown in the figure due to a decomposition of the categorical variable "WS_sequence".

In general, feature "wgt.freq.gh" demonstrated significantly the highest importance in predicting students' item performance among all three methods. This indicates that whether students click on the "GIRLS-highest 5%" button played a critical role in determining students' item response results. Apart from that, the feature "wgt.freq.S" and "wgt.freq.SS"

that relates to the single and successive click action on the statement also showed their importance in the modeling, which conform to the fact that the options in the statement directly determines the “correct” and “incorrect” outcome. Besides, the feature “wgt.freq.bh” ranks top 3 among all the models, revealing that students’ click action on “BOYS-highest 5%” has a major contribution in predicting their item response outcome. Regarding the time-related features, the time spent on the web browser “duration.pos.W” showed a visible contribution to the model prediction. It is worth noticing that the strategy-related feature “wgt.Strategy2” which sums up the frequency that students have applied the expected strategy for the second statement, showed its importance in predicting the outcome. Even though, it is clear from Figure 8 that there is an extreme drop in feature importance after the highest ranked feature “wgt.freq.gh”, making it the most decisive feature that outperforms others in the prediction.

Chi-square scores ranked in a decreased order are summarized in Appendix III. The most informative 25 features listed along with their Chi-square scores are presented below in Table 7. It can be observed from the table that the feature “wgt.freq.SS” has ranked at the top among all the weighted action features. Consistent with the result in the feature importance, the continuous click action on the statement(“SS”) has a strong association with the prediction of two outcome classes and performed to be the most robust classifier. Besides, the feature “wgt.freq.BSS”, “wgt.freq.OSO”, “wgt.freq.SSO”, “wgt.freq.SOS” also showed a higher score among other features, revealing their abilities in distinguishing the classes in the outcome. Unsurprisingly, these features with higher Chi-square scores are all in close relation to the action on the statement(“S”) given the statement is straightforwardly linked to the outcome. Other than that, the mini action sequence that reveals the clicking order on different areas of the screen such as “wgt.freq.WSS” (clicking on the web and select twice on the statement) and “wgt.freq.WSW” (clicking on the web first, then choosing the option in

statement and going back to the web) were identified as more robust features providing greater information. Different from the top ranked "wgt.freq.gh" in feature importance, single action "wgt.freq.gl" (the clicking frequency on "GIRLS-lowest 5%") was identified informative in distinguishing the different classes in the outcome variable with the measure of Chi-square scores. What is identical is that the strategy feature related to the second statement "wgt.Strategy2" demonstrated importance in both measures.

Table 7

Chi-square Score Table for Top-ranked 25 Features

Rank	Feature	Chi-square scores
1	wgt.freq.SS	42.4646
2	wgt.freq.BSS	38.0689
3	wgt.freq.OSO	28.5641
4	wgt.freq.SSO	27.8082
5	wgt.freq.SOS	27.6951
6	wgt.freq.WSS	26.9919
7	wgt.freq.SSE	26.8527
8	wgt.freq.BS	21.6156
9	wgt.freq.WSW	21.2305
10	wgt.freq.SO	19.4242
11	wgt.freq.gl	12.5188
12	wgt.freq.BOS	11.8082
13	wgt.freq.WWW	11.2769
14	wgt.Strategy2	10.979
15	wgt.freq.OS	10.6149
16	wgt.freq.SW	10.3338
17	wgt.freq.SWW	10.1266
18	wgt.freq.OE	9.8785
19	wgt.freq.WSE	9.5144
20	wgt.freq.gmghbhbm	8.8726
21	wgt.freq.OSE	8.6091
22	wgt.freq.bl	8.0338
23	wgt.freq.WWS	7.2076
24	wgt.freq.OSS	6.9846
25	wgt.freq.WW	5.9291

Discussion

In the present study, three tree-based data mining methods: decision tree, random forest, and gradient boosting machine were applied to predict students' item performance in 2012 PISA computer-based mathematics using the log file data that explicit students' navigating behaviors and their cognitive process during accomplishing the task. While showcasing how the tree-based data mining techniques can be applied in the international large-scale assessment using the log file data from the mathematics domain, three major goals were introduced for this paper. First, to identify and extract features from the given 2012 PISA log-file data of the CM038Q03 item. Second, to compare which tree-based data mining methods provide better performance in predicting students' item response scores by evaluating the error metrics. Third, to explore the interpretability of given tree-based methods by investigating feature contributions to the prediction of students' item response performance.

RQ1: Feature generation

It is known from the literature that feature generation plays a vital role in the analyses of log file data, as the classification results largely depend on the generated features' ability in distinguishing the different classes of outcome variable (Qiao & Jiao, 2018). Nevertheless, feature filtering is considered to possess the same priority claimed (Han et al., 2019). In the phase of feature generation, a total of 110 features were extracted with four types in two categories: a) empirically from a holistic view of the item: n-grams, behavioral indicators, and time-related features; b) theoretically focusing on actions that investigate the web browser: strategy-related features. Among the extracted features, forty-five were identified with zero and near-zero variance that was less informative to the prediction. These filtered features were further excluded from the tree-based modeling.

To generate favorable features with meaningful findings, as Salles et al. (2020) claimed, both data-driven and theory-driven methods are crucial. In line with previous literature, the present study adopted feature generation methods from Han et al. (2019), where the procedural behavior patterns and time-related characteristics related to the whole picture of the item were extracted guided by data and proposed hypotheses. Up to the present, there is no existing research that translates this feature generation procedure aside from the problem-solving item in PISA. Using a novel material CM038Q03 mathematics item, this study demonstrated the feasibility of adopting the feature generation procedure from the previous practice to generate features from the PISA log file data in the mathematics domain.

Different from the holistic procedure in Han et al. (2019), a targeted theory-driven approach was devised to extract features in terms of the expected strategy that educators have suggested (Bardini, 2015). The focus has been placed on investigating actions related to the simulated web browser, specifically, the behaviors that compare the line graphs in the web browser according to the descriptions in each statement. As indicated in the results, feature "wgt.freq.gh" (weighted frequency for clicking "GIRLS-highest 5%") generated from the targeted theory-driven approach outperformed in the prediction, yet the n-grams and behavioral indicators extracted from the previous method did not stand out as equivalent. This discovery has revealed that the feature generation procedure is chiefly item-specific, where various item designs in different domains determine the extraction of features. In spite of that, the theoretical hypotheses guided by the construct and rubrics of the item ought not to be underestimated in the feature extraction.

RQ2: Predictive Analysis

Applying decision tree, random forest, and the gradient boosting machine, this study examined item-level response accuracy for these the tree-based data mining methods. Consistent with the results from the previous study(Qiao & Jiao, 2018), all three models

showed ideal performance in classifying students' item response using the features generated from log file data. Specifically, the decision tree model CART showed an undeniable ability in classification and interpretation, which is in a good agreement with what Qiao and Jiao (2018) claimed that CART is sufficient for analyzing similar log-file data sets.

However, unlike their results where the gradient boosting machine performed better in Accuracy and Kappa, the decision tree and random forest methods in the present study showed higher values, while the gradient boosting did not perform comparably well. Overall, the random forest method demonstrated a better ability in the model prediction according to a larger area under the ROC curves.

These findings indicate the feasibility of applying tree-based data mining methods to classify students' response performance using the log file data with extracted features. In the meantime, educational data mining analyses for the PISA log file data have been expanded from the problem-solving to the mathematics domain. From a model perspective, the remarkable model performance for all three tree-based data mining methods proved their utilities in predicting item-level responses using the log-file data. Among them, the decision tree method by CART confirms its advantages in classifying the students' responses with comparable accuracy and additional interpretability. With a slightly superior performance, the ensemble method random forest is considered an adequate approach to predict students' outcome using log file data from PISA CBAM.

RQ3: Feature Importance

In this study, both the model-based feature importance plot and the description-oriented Chi-square score table were presented to identify which features generated from log file data served a more critical role in the prediction.

Prior research has concluded the remarkable importance of the features that reflect the students' strategies and those that can discern different classes of outcome variables in the

model prediction using the importance measures such as feature importance plot (Qiao & Jiao, 2018; Salles et al., 2020). Inconsistent with the previous findings, the feature importance plot in the current study suggested that the unigram feature "wgt.freq.gh" (weighted frequency for click action on "GIRLS-highest 5%" checkbox) contributed extraordinarily higher than other features. Yet the consecutive action sequences that relate to the strategies for two statements did not receive equal emphasis in the plot, although the feature "wgt.Strategy2" showed its relevant lower importance. This unexpected discovery indicates that students' particular click action on the "GIRLS-highest 5%" button is more critical to the classification of students' item response in the algorithm for all three models compared to the strategies applied in this mathematics item. On the other hand, this also indicates a lack of a clear goal for some students to analyze the item, thus they performed actions without strategies.

What confirmed the above-mentioned studies was that the time-related features did not show distinct contributions in the classification. However, the feature "duration.pos.W" performed minor but undeniable importance in the decision tree and gradient boosting methods, indicating that the time spent on comparing the line graphs on the simulated web browser relates to the students' success in this item.

Regarding the "W", "S", "O" n-gram features generated from a holistic picture of the item, the feature "wgt.freq.S" and "wgt.freq.SS" ranked highly in the feature importance plot. Similar results have been found in the chi-square table, where the feature "wgt.freq.SS" ranks the highest. This has revealed that the consecutive click actions on the statement is a more robust feature that can distinguish "correct" and "incorrect" classes in the outcome. Whilst students may apply certain strategies in solving the mathematics item, whether they have clicked on the statement is more decisive to the response.

Two insights have been reflected concerning this issue. One is that the statement itself directly determines the outcome, as each option served as a component of the correct or incorrect answer. This nature thus led to the importance and information it has obtained. The other is that the action that relates to the statement is dominant among all the actions that students have performed. Students may tend not to click on the checkboxes on the simulated web browser, rather, they select straightly on the statement with a quick glance at the graph. The clear exhibition does not require further actions to compare the line graphs given they are displayed originally by default. Descriptive statistics of the "S_only", representing that the action sequence contains actions that click on the statement only, have further confirmed this speculation. Although the behavioral indicators did not show sizable importance in the classification, the "S_only", one of the values from the behavioral indicator "WS sequence", takes up the prevailing portion of the sample with a number of 1034 out of the total 1785.

Limitations

This study has its own limitations. First, in terms of the data cleaning, invalid cases based on action, time, and item performance score variables were excluded from the analyses. As can be noticed in Appendix III, the observation with New ID "AUS-0000013-00243" detected to have a negative duration time on feature "duration.pos.O" were kept in the dataset, which may impact the data analyses of the time-related features. Second, although taking into consideration of both empirical and theoretical aspects in the feature generation, the current study may not include all the features related to the item in the analyses. Even for the features utilized in this study, there is room for improvement. The behaviors such as "click other" remain interesting to discover their relevance to students' responses. Third, in the phase of modeling, only tree-based methods such as decision tree, random forest, and gradient boosting were applied to analyze the log file data provided in the PISA CBMA item. Although tree-based methods are sufficient for analyzing PISA log file data, other more

complex and popular supervised methods such as support vector machines and neural networks could be applied to investigate further results (Huang & Khan, 2021). Fourth, the feature interpretation used in this study focused on a global measure. Further investigations into the interpretability of features could be expanded using measures such as local interpretations, which describe the contribution of a feature to the outcome for a specific observation (Saarela & Jauhiainen, 2021). Further, the interpretation of the features remains to be validated. Lastly, the current study takes 2012 PISA CBAM log file data as an example, while the application of data mining methods can be generalized to log file data from other international large-scale assessments such as Trends in International Mathematics and Science Study (TIMSS). In addition, it would be beneficial to explore a set of items that may imply certain patterns and insights of students' cognitive behaviors in general instead of one specific item. Future investigations of the analyses could be extended to comparisons among countries.

Conclusion

To conclude, this study has didactically demonstrated the feasibility of applying tree-based data mining techniques to examine the log-file data from a computer-based mathematics item in PISA. Solid steps including feature generation, feature filtering, predictive analyses, and feature interpretation were implemented. An approach in the feature generation phase was devised to extract features successfully based on both empirical data-driven and theoretical guidance from a holistic procedural view as well as a strategy-focused perspective. Using the filtered features to predict students' binary response performance at an item level, decision tree, random forest, and gradient boosting machine methods all showed exceptional model performance. Investigating further into the compelling outcome, both model-based and description-oriented measures were evaluated for feature interpretation.

Regarding the findings, considerable insights have been gained in two aspects. One is the illumination to the analyses of log-file data, where the conducted steps demonstrated potential for general applicability to PISA log file data sets in different domains. However, the minor importance of the features generated using previous approaches from the problem-solving item has revealed a need for specific adaption to different item settings when extracting features from log file data. The excellent performance of the tree-based methods confirmed their ability as a powerful tool in binary classification using the PISA log file data set.

The other aspect is concerned with the interventions for the educational practice and computer-based assessment. Features identified with the highest importance in this paper such as clicks on the "GIRLS-highest 5%" checkbox and clicks on the statement revealed students' specific behavioral patterns that were associated the most with their success on the CM038Q03 item. The concrete features not only provide additional information for the understanding of students' knowledge and skills to students and the educational practitioners but also the knowledge of the item design itself to the test developers. The importance of clicks on the statement instead of an emphasis on the strategies indicate that students tend not to compare the line graphs. This suggests a further inspection of the refined item design and the available computer-based tools that could facilitate students' interactive task-solving process during the assessment. For example, abandon showing all the line graphs in their initial state, leaving more room for students to conduct navigations.

This study contributed to the practice in the analyses of log file data from international large-scale assessments using data mining techniques. Most notably, to the author's knowledge, this is the first study to investigate the application of data mining techniques in examining the log file data using the PISA CBA material in the mathematics domain.

References

Baker, R. S., & Inventado, P. S. (2014). Educational Data Mining and Learning Analytics. In J. A. Larusson & B. White (Eds.), *Learning Analytics* (pp. 61–75). Springer New York. https://doi.org/10.1007/978-1-4614-3305-7_4

Bardini, C. (2015). Computer-Based Assessment of Mathematics in PISA 2012. In K. Stacey & R. Turner (Eds.), *Assessing Mathematical Literacy: The PISA Experience* (pp. 173–188). Springer International Publishing. https://doi.org/10.1007/978-3-319-10121-7_8

Boehmke, B., & Greenwell, B. (2019). *Hands-On Machine Learning with R*. Chapman and Hall/CRC. 13-36,175-190, 203-218,221-234. https://doi.org/10.1201/9780367816377

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (2017). *Classification And Regression Trees*. Routledge. https://doi.org/10.1201/9781315139470

C, S. K., & RamaSree, R. J. (2015). Dimensionality reduction in automated evaluation of descriptive answers through zero variance, near zero variance and non frequent words techniques—A comparison. *2015 IEEE 9th International Conference on Intelligent Systems and Control (ISCO)*, 1–6. https://doi.org/10.1109/ISCO.2015.7282351

Cohen, J. (1960). A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement*, *20*(1), 37–46. https://doi.org/10.1177/001316446002000104

Coppersmith, D., Hong, S. J., & Hosking, J. R. M. (1999). Partitioning Nominal Attributes in Decision Trees. *Data Mining and Knowledge Discovery*, *3*(2), 197–217. https://doi.org/10.1023/A:1009869804967

Davis, J., & Goadrich, M. (2006). The relationship between Precision-Recall and ROC curves. *Proceedings of the 23rd International Conference on Machine Learning*, 233–240. https://doi.org/10.1145/1143844.1143874

Depren, S. K., Aşkın, Ö. E., & Öz, E. (2017). Identifying the Classification Performances of Educational Data Mining Methods: A Case Study for TIMSS. *Educational Sciences: Theory & Practice*, *17*(5), 1605–1623. https://doi.org/10.12738/estp.2017.5.0634

Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters*, *27*(8), 861–874. https://doi.org/10.1016/j.patrec.2005.10.010

Gabriel, F., Signolet, J., & Westwell, M. (2018). A machine learning approach to investigating the effects of mathematics dispositions on mathematical literacy. *International Journal of Research & Method in Education*, *41*(3), 306–327. https://doi.org/10.1080/1743727X.2017.1301916

Geurts, P., Ernst, D., & Wehenkel, L. (2006). Extremely randomized trees. *Machine Learning*, *63*(1), 3–42. https://doi.org/10.1007/s10994-006-6226-1

Goldhammer, F., Hahnel, C., Kroehne, U., & Zehner, F. (2021). From byproduct to design factor: On validating the interpretation of process indicators based on log data. *Large-Scale Assessments in Education*, *9*(1), 20. https://doi.org/10.1186/s40536-021-00113-5

Goldhammer, F., Martens, T., & Lüdtke, O. (2017). Conditioning factors of test-taking engagement in PIAAC: An exploratory IRT modelling approach considering person and item characteristics. *Large-Scale Assessments in Education*, *5*(1), 18. https://doi.org/10.1186/s40536-017-0051-9

Goldhammer, F., Naumann, J., Stelter, A., Tóth, K., Rölke, H., & Klieme, E. (2014). The time on task effect in reading and problem solving is moderated by task difficulty and skill: Insights from a computer-based large-scale assessment. *Journal of Educational Psychology*, *106*(3), 608–626. https://doi.org/10.1037/a0034716

Greenwell, B. M., Boehmke, B. C., & McCarthy, A. J. (2018). A Simple and Effective Model-Based Variable Importance Measure. *ArXiv:1805.04755 [Cs, Stat]*. http://arxiv.org/abs/1805.04755

Greiff, S., Wüstenberg, S., & 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. https://doi.org/10.1016/j.compedu.2015.10.018

Hahnel, C., Goldhammer, F., Naumann, J., & Kröhne, U. (2016). Effects of linear reading, basic computer skills, evaluating online information, and navigation on reading digital text. *Computers in Human Behavior*, *55*, 486–500. https://doi.org/10.1016/j.chb.2015.09.042

Han, Z., He, Q., & von Davier, M. (2019). Predictive Feature Generation and Selection Using Process Data From PISA Interactive Problem-Solving Items: An Application of Random Forests. *Frontiers in Psychology*, *10*, 2461. https://doi.org/10.3389/fpsyg.2019.02461

Hanley, J. A., & McNeil, B. J. (1982). The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology*, *143*(1), 29–36. https://doi.org/10.1148/radiology.143.1.7063747

He, Q., Borgonovi, F., & Paccagnella, M. (2019). *Using process data to understand adults' problem-solving behaviour in the Programme for the International Assessment of Adult Competencies (PIAAC): Identifying generalised patterns across multiple tasks with sequence mining*. https://doi.org/10.1787/650918f2-en

He, Q., & von Davier, M. (2015). Identifying Feature Sequences from Process Data in Problem-Solving Items with N-Grams. In L. A. van der Ark, D. M. Bolt, W.-C.

Wang, J. A. Douglas, & S.-M. Chow (Eds.), *Quantitative Psychology Research* (pp. 173–190). Springer International Publishing.

Huang, Y., & Khan, S. M. (2021). Advances in AI and Machine Learning for Education Research. In A. A. von Davier, R. J. Mislevy, & J. Hao (Eds.), *Computational Psychometrics: New Methodologies for a New Generation of Digital Learning and Assessment: With Examples in R and Python* (pp. 195–208). Springer International Publishing. https://doi.org/10.1007/978-3-030-74394-9_11

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2021). Statistical Learning. In G. James, D. Witten, T. Hastie, & R. Tibshirani (Eds.), *An Introduction to Statistical Learning: With Applications in R* (pp. 15–57). Springer US. https://doi.org/10.1007/978-1-0716-1418-1_2

Kuhn, M. (2021). caret: Classification and Regression Training. R package version 6.0-90. https://CRAN.R-project.org/package=caret

Liu, H., Liu, Y., & Li, M. (2018). Analysis of Process Data of PISA 2012 Computer-Based Problem Solving: Application of the Modified Multilevel Mixture IRT Model. *Frontiers in Psychology*, *9*. https://www.frontiersin.org/article/10.3389/fpsyg.2018.01372

McHugh, M. L. (2012). Interrater reliability: The kappa statistic. *Biochemia Medica*, *22*(3), 276–282.

Marvin N. Wright, Andreas Ziegler (2017). ranger: A Fast Implementation of Random Forests for High Dimensional Data in C++ and R. Journal of Statistical Software, 77(1), 1-17. doi:10.18637/jss.v077.i01

Naumann, J. (2015). A model of online reading engagement: Linking engagement, navigation, and performance in digital reading. *Computers in Human Behavior*, *53*, 263–277. https://doi.org/10.1016/j.chb.2015.06.051

OECD. (2010). *PISA Computer-Based Assessment of Student Skills in Science*. Organisation for Economic Co-operation and Development. https://www.oecd-ilibrary.org/education/pisa-computer-based-assessment-of-student-skills-in-science_9789264082038-en

OECD. (2013). *PISA 2012 Assessment and Analytical Framework: Mathematics, Reading, Science, Problem Solving and Financial Literacy*. OECD. https://doi.org/10.1787/9789264190511-en

Qiao, X., & Jiao, H. (2018). Data Mining Techniques in Analyzing Process Data: A Didactic. *FRONTIERS IN PSYCHOLOGY*, *9*. https://doi.org/10.3389/fpsyg.2018.02231

Reis Costa, D., & Leoncio Netto, W. (2019). LOGAN: Log File Analysis in International Large-Scale Assessments. R package version 1.0.0. https://CRAN.R-project.org/package=LOGAN

R Core Team (2021). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

Reis Costa, D., Bolsinova, M., Tijmstra, J., & Andersson, B. (2021). Improving the Precision of Ability Estimates Using Time-On-Task Variables: Insights From the PISA 2012 Computer-Based Assessment of Mathematics. *Frontiers in Psychology*, *12*. https://www.frontiersin.org/article/10.3389/fpsyg.2021.579128

Reis Costa, D., & Leoncio Netto, W. (2022). Process Data Analysis in ILSAs: An Ecological Framework and Literature Review. In T. Nilsen, A. Stancel-Piątak, & J.-E. Gustafsson (Eds.), *International Handbook of Comparative Large-Scale Studies in Education* (pp. 1–27). Springer International Publishing. https://doi.org/10.1007/978-3-030-38298-8_60-1

Reis Costa, D., & Qin, Q. (2022). LOGANTree: Tree-Based Models for the Analysis of Log Files from Computer-Based Assessments. R package version 0.1.0. https://cran.r-project.org/web/packages/LOGANTree/index.html

Romero, C., & Ventura, S. (2020). Educational data mining and learning analytics: An updated survey. *WIREs Data Mining and Knowledge Discovery*, *10*(3), e1355. https://doi.org/10.1002/widm.1355

Saarela, M., & Jauhiainen, S. (2021). Comparison of feature importance measures as explanations for classification models. *SN Applied Sciences*, *3*(2), 272. https://doi.org/10.1007/s42452-021-04148-9

Salles, F., Dos Santos, R., & Keskpaik, S. (2020). When Didactics Meet Data Science: Process Data Analysis in Large-Scale Mathematics Assessment in France. *Large-Scale Assessments in Education*, *8*.

Sinharay, S. (2016). An NCME Instructional Module on Data Mining Methods for Classification and Regression. *Educational Measurement: Issues and Practice*, *35*(3), 38–54. https://doi.org/10.1111/emip.12115

Vista, A., Care, E., & Awwal, N. (2017). Visualising and examining sequential actions as behavioural paths that can be interpreted as markers of complex behaviours. *Computers in Human Behavior*, *76*, 656–671. https://doi.org/10.1016/j.chb.2017.01.027

von Davier, A. A., Mislevy, R. J., & Hao, J. (Eds.). (2021). *Computational Psychometrics: New Methodologies for a New Generation of Digital Learning and Assessment: With Examples in R and Python*. Springer International Publishing. https://doi.org/10.1007/978-3-030-74394-9

Xu, H., Fang, G., Chen, Y., Liu, J., & Ying, Z. (2018). Latent Class Analysis of Recurrent Events in Problem-Solving Items. *Applied Psychological Measurement*, *42*(6), 478–498. https://doi.org/10.1177/0146621617748325

Zachary A. Deane-Mayer and Jared E. Knowles (2019). caretEnsemble: Ensembles of Caret Models. R package version 2.0.1. https://CRAN.R-project.org/package=caretEnsemble

Appendix I

GDPR documents & Ethical approval

Anonymous data were used in the current study, thus the NSD (Norsk Senter for Forskningsdata) notification form is presented below instead of assessed GDPR documents.

Which personal data will be processed?

What are personal data?

Personal data consist of any data relating to an identified or identifiable person. Collected data that can be linked directly or indirectly to individual persons are considered personal data. Select if you are processing personal data, including if there exists a link between the collected data and personally identifiable information (e.g. name, identity number, contact details etc).

What is processing?

Processing means any operation that is performed on personal data, such as collection, recording, registering, organisation, structuring, storage, adaptation, alteration, retrieving, transferring, distributing, publishing, deleting or destroying.

General categories of personal data

Name (also with signature/written consent)

☐Yes ☑ No

National ID number or other personal identification number

☐Yes ☑ No

Date of birth

☐Yes ☑ No

Address or telephone number

☐Yes ☑ No

Email address, IP address or other online identifier

☐Yes ☑ No

Photographs or video recordings of people

☐Yes ☑ No

Sound recordings of people

☐Yes ☑ No

GPS data or other geolocation data (electronic communications)

☐Yes ☑ No

Background data that can identify a person

☐Yes ☑ No

Other data that can identify a person

☐Yes ☑ No

Special categories of personal data

Racial or ethnic origin

☐Yes ☑ No

Political opinions

☐Yes ☑ No

Religious beliefs

☐Yes ☑ No

Philosophical beliefs

☐Yes ☑ No

Trade Union Membership

☐Yes ☑ No

Health data

☐Yes ☑ No

Genetic data

☐Yes ☑ No

Biometric data

☐Yes ☑ No

Sex life or sexual orientation

☐Yes ☑ No

Criminal convictions and offences

☐Yes ☑ No

Project

Master's Thesis: Application of Tree-based Data Mining Techniques to Examine the Log File Data from a 2012 PISA Computer-based Mathematic Item

If you will only be processing anonymous data you should not notify your project

Anonymous data are data where individual persons are not/no longer identifiable; not directly, indirectly or via email/IP address or scrambling key.

Appendix II

Data Management & Analysis Code

For reproducibility of the findings, the data sets and R scripts for data management and analyses used in this study are available on Open Science Framework (OSF) platform: https://osf.io/u8mfp/?view_only=76ae51b4fd834d20897814f1335ae39e

Appendix III

Supplemental Material

Table 8

Descriptive Statistics for Generated Features (n=64)

Feature	Type	Metric	Descriptive Statistics					
			Min	1st.Qu	Median	3st.Qu	Max	SD
duration.pos.S	Amount of time	Original (in min)	0.00	0.24	0.54	0.90	3.20	0.49
duration.pos.W	Amount of time	Original (in min)	0.00	0.00	0.00	0.45	2.90	0.44
duration.pos.O	Amount of time	Original (in min)	-0.01	0.00	0.01	0.25	3.39	0.33
durationStart.pos.S	Amount of time	Original (in min)	0.00	0.44	0.71	1.03	4.90	0.47
durationStart.pos.W	Amount of time	Original (in min)	0.00	0.00	0.00	0.50	4.93	0.41
durationStart.pos.O	Amount of time	Original (in min)	0.00	0.00	0.05	0.50	3.23	0.43
AUS.TOT	Amount of time	Original (in min)	0.06	0.75	1.14	1.52	5.28	0.61
n_actions	Number of actions	Original	2.00	4.00	6.00	14.00	67.00	8.98
wgt.freq.W	Frequency of action	Weighted	0.00	0.00	0.00	2.45	4.24	1.39
wgt.freq.S	Frequency of action	Weighted	0.00	0.01	0.01	0.01	0.03	0.00
wgt.freq.O	Frequency of action	Weighted	0.00	0.00	0.68	1.15	3.16	0.76
wgt.freq.BW	Frequency of action	Weighted	0.00	0.00	0.00	0.00	1.68	0.65
wgt.freq.BS	Frequency of action	Weighted	0.00	0.00	0.59	0.59	0.59	0.29
wgt.freq.BO	Frequency of action	Weighted	0.00	0.00	0.00	1.37	1.37	0.60
wgt.freq.BE	Frequency of action	Weighted	0.00	0.00	0.00	0.00	5.54	0.35
wgt.freq.WW	Frequency of action	Weighted	0.00	0.00	0.00	2.30	4.11	1.28
wgt.freq.WS	Frequency of action	Weighted	0.00	0.00	0.00	1.03	2.16	0.67
wgt.freq.WO	Frequency of action	Weighted	0.00	0.00	0.00	0.00	4.89	0.97
wgt.freq.WE	Frequency of action	Weighted	0.00	0.00	0.00	0.00	3.88	0.55
wgt.freq.SW	Frequency of action	Weighted	0.00	0.00	0.00	1.34	2.27	0.61
wgt.freq.SS	Frequency of action	Weighted	0.00	0.00	0.51	0.51	2.05	0.30
wgt.freq.SO	Frequency of action	Weighted	0.00	0.00	0.00	0.00	3.24	0.69

Feature	Type	Metric	Descriptive Statistics					
			Min	1st.Qu	Median	3st.Qu	Max	SD
wgt.freq.SE	Frequency of action	Weighted	0.00	0.08	0.08	0.08	0.08	0.02
wgt.freq.OW	Frequency of action	Weighted	0.00	0.00	0.00	0.00	4.96	1.01
wgt.freq.OS	Frequency of action	Weighted	0.00	0.00	0.00	1.17	2.46	0.63
wgt.freq.OO	Frequency of action	Weighted	0.00	0.00	0.00	0.00	6.69	1.37
wgt.freq.OE	Frequency of action	Weighted	0.00	0.00	0.00	0.00	2.87	0.66
wgt.freq.WWW	Frequency of action	Weighted	0.00	0.00	0.00	2.17	4.79	1.38
wgt.freq.WWS	Frequency of action	Weighted	0.00	0.00	0.00	1.07	2.25	0.67
wgt.freq.WWO	Frequency of action	Weighted	0.00	0.00	0.00	0.00	4.30	0.91
wgt.freq.WSW	Frequency of action	Weighted	0.00	0.00	0.00	0.00	3.38	0.71
wgt.freq.WSO	Frequency of action	Weighted	0.00	0.00	0.00	0.00	5.22	0.67
wgt.freq.WSS	Frequency of action	Weighted	0.00	0.00	0.00	0.00	2.66	0.68
wgt.freq.WOS	Frequency of action	Weighted	0.00	0.00	0.00	0.00	4.69	0.68
wgt.freq.WOW	Frequency of action	Weighted	0.00	0.00	0.00	0.00	5.76	0.99
wgt.freq.WOO	Frequency of action	Weighted	0.00	0.00	0.00	0.00	5.99	0.82
wgt.freq.SSS	Frequency of action	Weighted	0.00	0.00	0.00	0.00	10.96	1.00
wgt.freq.SSW	Frequency of action	Weighted	0.00	0.00	0.00	0.00	4.65	0.45
wgt.freq.SSO	Frequency of action	Weighted	0.00	0.00	0.00	0.00	4.85	0.68
wgt.freq.SWS	Frequency of action	Weighted	0.00	0.00	0.00	0.00	7.49	0.18
wgt.freq.SWO	Frequency of action	Weighted	0.00	0.00	0.00	0.00	3.70	0.57
wgt.freq.SWW	Frequency of action	Weighted	0.00	0.00	0.00	0.00	2.43	0.62
wgt.freq.SOS	Frequency of action	Weighted	0.00	0.00	0.00	0.00	4.27	0.69
wgt.freq.SOW	Frequency of action	Weighted	0.00	0.00	0.00	0.00	4.98	0.68
wgt.freq.SOO	Frequency of action	Weighted	0.00	0.00	0.00	0.00	5.20	0.66
wgt.freq.OOO	Frequency of action	Weighted	0.00	0.00	0.00	0.00	8.91	1.58
wgt.freq.OOW	Frequency of action	Weighted	0.00	0.00	0.00	0.00	5.11	0.84
wgt.freq.OOS	Frequency of action	Weighted	0.00	0.00	0.00	0.00	4.31	0.76

Feature	Type	Metric	Descriptive Statistics					
			Min	1st.Qu	Median	3st.Qu	Max	SD
wgt.freq.OSW	Frequency of action	Weighted	0.00	0.00	0.00	0.00	5.48	0.64
wgt.freq.OSS	Frequency of action	Weighted	0.00	0.00	0.00	0.00	3.46	0.69
wgt.freq.OSO	Frequency of action	Weighted	0.00	0.00	0.00	0.00	5.99	0.71
wgt.freq.OWS	Frequency of action	Weighted	0.00	0.00	0.00	0.00	5.80	0.64
wgt.freq.OWO	Frequency of action	Weighted	0.00	0.00	0.00	0.00	7.88	0.85
wgt.freq.OWW	Frequency of action	Weighted	0.00	0.00	0.00	0.00	4.28	0.94
wgt.freq.BSW	Frequency of action	Weighted	0.00	0.00	0.00	0.00	2.55	0.68
wgt.freq.BSO	Frequency of action	Weighted	0.00	0.00	0.00	0.00	2.76	0.67
wgt.freq.BSS	Frequency of action	Weighted	0.00	0.00	0.00	0.90	0.90	0.44
wgt.freq.BWW	Frequency of action	Weighted	0.00	0.00	0.00	0.00	1.73	0.66
wgt.freq.BWS	Frequency of action	Weighted	0.00	0.00	0.00	0.00	0.00	0.00
wgt.freq.BWO	Frequency of action	Weighted	0.00	0.00	0.00	0.00	4.71	0.44
wgt.freq.BOS	Frequency of action	Weighted	0.00	0.00	0.00	0.00	2.61	0.68
wgt.freq.BOW	Frequency of action	Weighted	0.00	0.00	0.00	0.00	3.44	0.61
wgt.freq.BOO	Frequency of action	Weighted	0.00	0.00	0.00	0.00	1.90	0.68
wgt.freq.SWE	Frequency of action	Weighted	0.00	0.00	0.00	0.00	7.49	0.18
wgt.freq.SOE	Frequency of action	Weighted	0.00	0.00	0.00	0.00	3.02	0.65
wgt.freq.SSE	Frequency of action	Weighted	0.00	0.00	0.62	0.62	0.62	0.31
wgt.freq.WWE	Frequency of action	Weighted	0.00	0.00	0.00	0.00	3.93	0.55
wgt.freq.WSE	Frequency of action	Weighted	0.00	0.00	0.00	0.00	1.40	0.60
wgt.freq.WOE	Frequency of action	Weighted	0.00	0.00	0.00	0.00	6.10	0.29
wgt.freq.OSE	Frequency of action	Weighted	0.00	0.00	0.00	0.00	2.05	0.69
wgt.freq.OWE	Frequency of action	Weighted	0.00	0.00	0.00	0.00	7.49	0.18
wgt.freq.OOE	Frequency of action	Weighted	0.00	0.00	0.00	0.00	5.29	0.37
wgt.freq.BSE	Frequency of action	Weighted	0.00	0.00	0.00	0.00	5.09	0.40
wgt.freq.BWE	Frequency of action	Weighted	0.00	0.00	0.00	0.00	0.00	0.00

Feature	Type	Metric	Descriptive Statistics					
			Min	1st.Qu	Median	3st.Qu	Max	SD
wgt.freq.BOE	Frequency of action	Weighted	0.00	0.00	0.00	0.00	7.49	0.18
wgt.freq.gm	Frequency of action	Weighted	0.00	0.00	0.00	1.03	4.01	0.84
wgt.freq.bm	Frequency of action	Weighted	0.00	0.00	0.00	1.10	3.51	0.79
wgt.freq.gl	Frequency of action	Weighted	0.00	0.00	0.00	2.12	4.72	1.22
wgt.freq.bl	Frequency of action	Weighted	0.00	0.00	0.00	0.00	5.08	1.26
wgt.freq.gh	Frequency of action	Weighted	0.00	0.10	0.10	0.12	0.21	0.04
wgt.freq.bh	Frequency of action	Weighted	0.00	0.00	0.00	1.12	3.46	0.81
wgt.freq.gmbm	Frequency of action	Weighted	0.00	0.00	0.00	0.00	5.50	0.77
wgt.freq.bmgm	Frequency of action	Weighted	0.00	0.00	0.00	0.00	5.56	0.68
wgt.freq.gmghbmbh	Frequency of action	Weighted	0.00	0.00	0.00	0.00	4.45	0.71
wgt.freq.gmghbhbm	Frequency of action	Weighted	0.00	0.00	0.00	0.00	4.98	0.68
wgt.freq.gmbmghbh	Frequency of action	Weighted	0.00	0.00	0.00	0.00	6.10	0.29
wgt.freq.gmbmbhgh	Frequency of action	Weighted	0.00	0.00	0.00	0.00	4.92	0.42
wgt.freq.gmbhghbm	Frequency of action	Weighted	0.00	0.00	0.00	0.00	0.00	0.00
wgt.freq.gmbhbmgh	Frequency of action	Weighted	0.00	0.00	0.00	0.00	7.49	0.18
wgt.freq.ghgmbmbh	Frequency of action	Weighted	0.00	0.00	0.00	0.00	4.49	0.47
wgt.freq.ghgmbhbm	Frequency of action	Weighted	0.00	0.00	0.00	0.00	8.33	0.45
wgt.freq.ghbmgmbh	Frequency of action	Weighted	0.00	0.00	0.00	0.00	0.00	0.00
wgt.freq.ghbmbhgm	Frequency of action	Weighted	0.00	0.00	0.00	0.00	6.10	0.29
wgt.freq.ghbhbmgm	Frequency of action	Weighted	0.00	0.00	0.00	0.00	9.38	0.39
wgt.freq.ghbhgmbm	Frequency of action	Weighted	0.00	0.00	0.00	0.00	6.79	0.23
wgt.freq.bmgmghbh	Frequency of action	Weighted	0.00	0.00	0.00	0.00	5.54	0.35
wgt.freq.bmgmbhgh	Frequency of action	Weighted	0.00	0.00	0.00	0.00	0.00	0.00
wgt.freq.bmghgmbh	Frequency of action	Weighted	0.00	0.00	0.00	0.00	0.00	0.00
wgt.freq.bmghbhgm	Frequency of action	Weighted	0.00	0.00	0.00	0.00	7.49	0.18
wgt.freq.bmbhgmgh	Frequency of action	Weighted	0.00	0.00	0.00	0.00	4.78	0.44

Feature	Type	Metric	Descriptive Statistics					
			Min	1st.Qu	Median	3st.Qu	Max	SD
wgt.freq.bmbhghgm	Frequency of action	Weighted	0.00	0.00	0.00	0.00	4.19	0.51
wgt.freq.bhgmghbm	Frequency of action	Weighted	0.00	0.00	0.00	0.00	6.79	0.23
wgt.freq.bhgmbmgh	Frequency of action	Weighted	0.00	0.00	0.00	0.00	0.00	0.00
wgt.freq.bhbmgmgh	Frequency of action	Weighted	0.00	0.00	0.00	0.00	6.10	0.29
wgt.freq.bhbmghgm	Frequency of action	Weighted	0.00	0.00	0.00	0.00	6.39	0.26
wgt.freq.bhghgmbm	Frequency of action	Weighted	0.00	0.00	0.00	0.00	6.39	0.26
wgt.freq.bhghbmgm	Frequency of action	Weighted	0.00	0.00	0.00	0.00	0.00	0.00
wgt.Strategy1	Frequency of action	Weighted	0.00	0.00	0.00	0.00	4.60	0.77
wgt.Strategy2	Frequency of action	Weighted	0.00	0.00	0.00	0.00	5.60	0.86

Note. the feature “WSsequence” is not included given it contains non-numerical data.

Table 9

Descriptive Statistics for Total Time and Outcome

Statistics	Total Time	CM038Q03T=0	CM038Q03T=1
Total N	1785	555	1230
Min	0.06	0.06	0.06
1st.Qu	0.75	0.66	0.79
Median	1.14	1.06	1.18
Mean	1.15	1.10	1.18
SD	0.61	0.63	0.60
3st.Qu	1.52	1.44	1.54
Max	5.28	5.28	5.03

Table 10

Descriptive Statistics for WSsequence and Outcome

WSsequence / CM038Q03T	0	1	Total
1<=WS<3	57	292	349
Incomplete	12	0	12
S_only	374	660	1034
Sequence_Start_from_S	53	170	223
WS_only	57	105	162
WS>=3	2	3	5
Total	555	1230	1785

Table 11

Descriptive Statistics for Strategy1 and Outcome

wgt.Strategy1 / CM038Q03T	0	1	Total
0	516	1098	1614
2.345510138	30	104	134
3.971293877	6	19	25
4.922316398	2	7	9
5.597077616	1	2	3
Total	555	1230	1785

Table 12

Descriptive Statistics for Strategy2 and Outcome

wgt.Strategy2 / CM038Q03T	0	1	Total
0	499	980	1479
1.763588592	47	220	267
2.986015053	8	22	30
3.701088692	0	5	5
4.208441513	0	1	1
4.601974935	1	2	3
Total	555	1230	1785

Table 13

Confusion Matrices for Three Tree-based Methods from Caret Outputs

Decision tree		
Prediction\Reference	correct	incorrect
correct	360	33
incorrect	9	133
Random forest		
Prediction\Reference	correct	incorrect
correct	349	25
incorrect	20	141
Gradient boosting machine		
Prediction\Reference	correct	incorrect
correct	353	32
incorrect	16	134

Table 14

Chi-square Score Table for the N-gram Features (n=56)

Feature	OverallChisq	Rank.OverallChisq
wgt.freq.SS	42.46	1
wgt.freq.BSS	38.07	2
wgt.freq.OSO	28.56	3
wgt.freq.SSO	27.81	4
wgt.freq.SOS	27.70	5
wgt.freq.WSS	26.99	6
wgt.freq.SSE	26.85	7
wgt.freq.BS	21.62	8
wgt.freq.WSW	21.23	9
wgt.freq.SO	19.42	10
wgt.freq.gl	12.52	11
wgt.freq.BOS	11.81	12
wgt.freq.WWW	11.28	13
wgt.Strategy2	10.98	14
wgt.freq.OS	10.61	15
wgt.freq.SW	10.33	16
wgt.freq.SWW	10.13	17
wgt.freq.OE	9.88	18
wgt.freq.WSE	9.51	19
wgt.freq.gmghbhbm	8.87	20
wgt.freq.OSE	8.61	21
wgt.freq.bl	8.03	22
wgt.freq.WWS	7.21	23
wgt.freq.OSS	6.98	24
wgt.freq.WW	5.93	25
wgt.freq.bh	5.12	26
wgt.freq.OOS	4.64	27
wgt.freq.W	4.60	28
wgt.freq.bm	4.57	29
wgt.freq.WS	4.35	30
wgt.freq.SOE	4.10	31
wgt.freq.gm	4.03	32
wgt.freq.gmghbmbh	3.75	33
wgt.freq.BO	2.82	34
wgt.freq.O	2.58	35
wgt.freq.SE	1.64	36
wgt.freq.OWW	1.16	37
wgt.freq.WOO	1.12	38
wgt.freq.BSO	1.07	39

Feature	OverallChisq	Rank.OverallChisq
wgt.freq.WOS	1.00	40
wgt.freq.gmbm	0.72	41
wgt.freq.BOO	0.67	42
wgt.freq.gh	0.63	43
wgt.freq.WWO	0.61	44
wgt.freq.OW	0.57	45
wgt.freq.BSW	0.54	46
wgt.freq.OOW	0.52	47
wgt.freq.WO	0.52	48
wgt.freq.OO	0.18	49
wgt.freq.WOW	0.15	50
wgt.freq.S	0.09	51
wgt.freq.BW	0.09	52
wgt.freq.SOW	0.05	53
wgt.Strategy1	0.04	54
wgt.freq.BWW	0.03	55
wgt.freq.OOO	0.02	56

Note. Only n-gram features' Chi-square scores were computed.