Comparison of Off the Shelf Data Mining Methodologies in Educational Game Analytics

David J. Gagnon
University of Wisconsin-Madison
Madison, WI
david.gagnon@wisc.edu

Erik Harpstead
Carnegie Mellon University
Pittsburgh, PA
eharpste@cs.cmu.edu

Stefan Slater
University of Pennsylvania
Philadelphia, PA
slater.research@gmail.com

ABSTRACT

In this paper we compare the accuracy of nine common machine learning algorithms in predicting quitting and performance on knowledge assessment tests in the context of two middle school science learning games. The games being studied, the *Crystal Cave* and *Wave Combinator*, are both short duration (played for an average of 25 and 28 minutes respectfully), web-based games designed for use in classroom contexts. We used samples of 1,254 and 5,308 anonymous internet players respectively collected during Fall of 2018. We recorded raw clickstream data and used feature engineering methods to calculate simple descriptive features such as average timings between events and the number and types of player moves. We then used these features to model players quitting the game at each level, as well as content knowledge measured by subsequent assessment. We found that logistic regression produced the best models overall and model quality was influenced by specific game levels and assessment items. We conclude by discussing future work to improve predicting player quitting and player knowledge assessment.

Keywords

Feature engineering, digital games, videogames, modeling, prediction, quitting, assessment

1. INTRODUCTION

Digital games are increasingly being used to support learning in educational contexts across a wide variety of subjects, including social studies [4], mathematics [3], physics [9], and history [10]. Beyond content knowledge, games have also been used to support the development of cognitive and noncognitive skills, such as persistence and spatial reasoning [8]. As video games see increasing use in classroom contexts, the need to analyze the rich interaction data that they produce for meaningful behavioral and learning indicators from play becomes greater as well.

Educational data mining (EDM) is well-suited to the problem of analyzing digital games which feature rich interaction data, and methods common to EDM have been frequently deployed to better understand data produced by digital games. For instance, EDM techniques have been used to model quitting behavior among students playing an educational physics simulation game [2], problem-solving in a game-based programming task [5], and computational thinking skills in *Zoombinis* [7, 14].

In this paper, we use EDM techniques to predict quitting behavior and content knowledge within two middle school science games, *Crystal Cave* and *Wave Combinator* [1,11]. We sought to model these outcomes because of their relevance for the use of these games in educational contexts. The identification of quitting behavior affords game designers and educators the opportunity to intervene with scaffolds or feedback that can help keep students on-task and working productively [2]. Prediction and modeling of content understanding in students enables game designers and educators the opportunity to generate additional opportunities for a student to practice a given skill, or correct specific misconceptions that might exist about that content. While such techniques have been employed in intelligent tutoring systems via knowledge tracing and knowledge inference methods [14], the open-ended structure of many educational games makes these methods difficult to employ successfully.

1.1 Games Being Studied

The two games used in this study, *Crystal Cave* and *Wave Combinator*, are available online for free public use and are short duration experiences, played for an average of 25 and 28 minutes respectfully. They are primarily used in classroom contexts.

In *Wave Combinator*, players must manipulate the amplitude, frequency and offset of a wave in order to match the shape of a target wave (Figure 1). Once the player's wave is within a certain range of the target wave, they are allowed to continue to the next level. At key points of the game, a multiple-choice question appears on screen that assess the vocabulary used in the game (Figure 2). While these assessment items are presented as being asked by in-game characters, they are not situated within a broader narrative context, but were retrofitted into the game for the sake of this research. This study will be examining play data from the first 7 levels of the game and the 2 multiple choice items that follow.

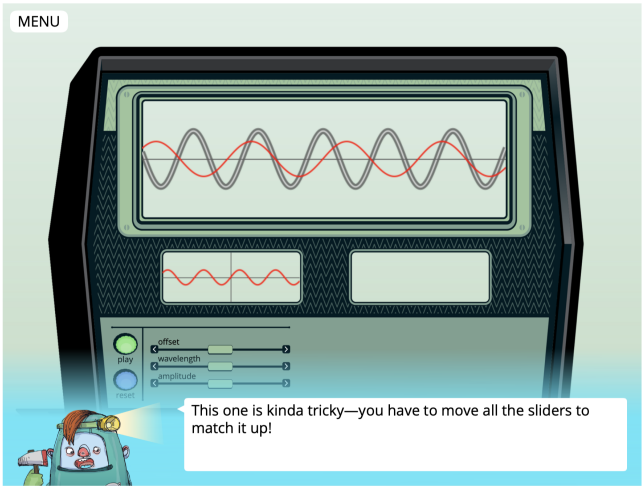

Figure 1. Initial levels of *Wave Combinator*.

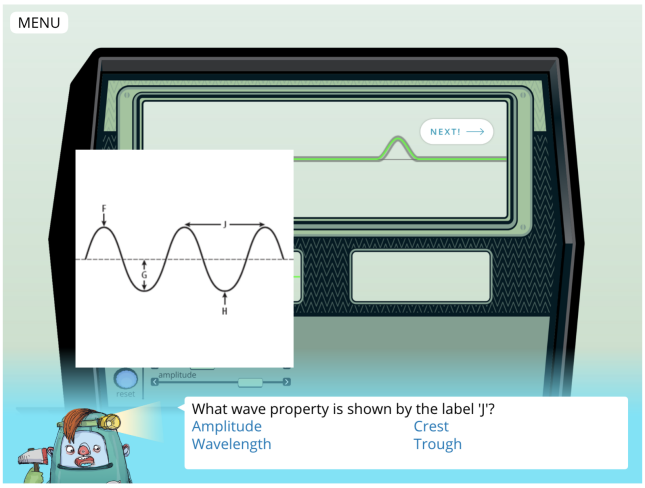

Figure 2. A multiple choice question embedded in *Wave Combinator*.

In *Crystal Cave*, players assemble differently shaped molecules to form crystals of varying stabilities (Figure 3). Stability is determined by the density of the resultant molecular pattern as well as the proper alignment of the positive and negative charges on portions of the molecules. For each level, different thresholds of stability for the players' molecular design will result in completing the level with 1 to 3 stars. Each level unlocks when the player has achieved a certain number of stars, leading to a semi-structured progression through the game that allows students to repeat challenges to find optimal molecular arrangements or to progress to new challenges. As with the *Wave Combinator*, multiple choice quiz items are presented by game characters, but without meaningful integration into the game context. These questions appear after completing specific levels (Figure 4). This study will be examining play data from the entire set of 7 levels of the game and the 3 multiple choice items that are intermixed.

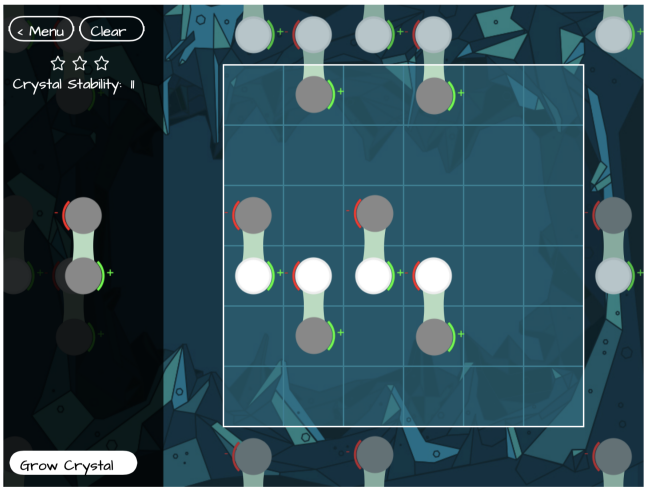

Figure 3. Assembling simple molecules to create stable crystals in Crystal Cave.

Figure 4. A multiple choice question embedded in *Crystal Cave.*

These two games were chosen because they represent different archetypes of educational games. *Wave Combinator* provides players with controls that manipulate the outputs of a simple simulation in real-time. Players have to construct meanings about the purpose of each control in order to find a solution. *Crystal Cave* is a more constructive task that delays feedback for several moves and requires players to apply simple chemistry rules to develop reasonable strategies. Our goal in looking at two different games was to explore the degree to which similar minimal feature engineering approaches would perform across differing game structures and design attributes.

2. METHODS

2.1 Process of Data Collection & Instrumentation

Data is collected from the games using both Google Analytics (GA) as well as a researcher developed event logging system. GA were used to quickly record and visualize overall game metrics such as number and location of player sessions, session length, and high-level progression through each game (i.e., completed level 5, then quit). GA were primarily used during development and to understand audiences but are not included in the current analyses beyond understanding the audience and usage patterns.

Multiple choice knowledge assessment measures were designed by the researchers for both games. Each item was aligned with the documented learning goals of the game. The instruments were designed to use a similar visual style as the rest of the game play (See Figures 2 and 4). Players completed the assessment measures after finishing gameplay; the assessment items were not embedded into the game itself.

Our analyses focused on two labels – quitting behavior, when a player quits a level before it is completed and leaves the game, and performance on the post-test assessment measures. Population and Sampling Process

Based on GA, 93% of the games' usage was from United States, based on IP addresses. Gameplay sessions were primarily recorded during school hours and on weekdays, leading researchers to believe that the games are used primarily in classroom contexts. Gameplay sessions were recorded anonymously making it impossible to tell if a session represented a new or returning player. While we acknowledge this limitation,

our analysis assumes that an individual session represents a unique player. During the data collection period of September 1 through December 31, 2018 the *Crystal Cave* was played 20,963 times with an average of 24.68 minutes/session. The *Wave Combinator* was played 23,353 times with an average of 28.78 minutes/session. Of these sessions, 1,254 of the *Crystal Cave* sessions and 5,308 *Wave Combinator* sessions are included in this study based on the availability of the logging system and data exclusion rules described below.

2.2 Data Logging

Within each game, a JavaScript based logging client captures and transmits clickstream events to a server for storage. Events are recorded for all discrete player actions such as starting a level, making a move, and completing a challenge. Each event is time-coded using the client browser's native time, an automatically generated session identifier, and details about the event that took place. The events are encoded as JSON and sent via an HTTP POST request. These requests are scheduled for delivery to the backend logging server using a first-in-first-out queue and are only dismissed after delivery is confirmed.

The backend server is comprised of a researcher built, open source, PHP-based web service. Client requests are parsed, appended with the server's system time, and inserted as individual records into a MySQL database. As each clickstream event is sent as a seperate network request and recorded as a individual row, the system is easily parallelized for large numbers of clients. For this study, a single quad core Apache / PHP server Virtual Machine and a single quad core MySQL Virtual Machine server were provisioned in a University data-center.

2.3 Feature Engineering and Distribution

We designed features that describe the actions players are able to take in each game. We intentionally explored basic features that could conceivably be extended to other educational games. We developed features that describe the counts of each (game specific) move type, the average number of each move type in each level of play, timings and attributes of each move, the scoring the game provided to the player in each level, and attributes of re-starting and replaying challenges by players. Features were calculated using data collected chronologically before the outcome being modeled. For example, features to model quitting in level 5 were calculated using play data derived from levels 1 through 4, and any available data from level 5 play, but not level 6 or greater. This was done to preserve the predictive nature of the research and to create a model that could conceivably be used to make predictions in real-time within a gameplay session. Play sessions with less than 10 total moves were excluded from the final dataset. The table below describes the features used for each game and attempts to align them across games when appropriate.

We defined "quitting" on a given level as a session where the session ends (log events halt abruptly) before the current level is completed. Using this definition, each session is labeled with either a "quit" or a "complete" for each level. The distribution for these events leans strongly toward completing, with an average of 70.3% and 81.4% of sessions completing each level in *Crystal Cave* and *Wave Combinator* respectfully. We use each level's quitting distributions as our baseline model.

We defined "incorrect" answers as a session selecting any of the 3 options that were not correct for each assessment item. As with quitting, the distribution for the assessments leans toward correct answers being provided. An average of 65.6% of sessions selected correct answers for the three Crystal Cave questions while an average of 65.8% of sessions selected correct answers for the two Wave Combinator questions. As with quitting, we use each question's distribution as the baseline model.

Table 1. Gameplay features for Wave Combinator and Crystal Cave.

	Wave Combinator	Crystal Cave
Move Counts	Total Slider Moves Total Offset Moves Total Amplitude Moves Total Wavelength Moves Total Move Type Changes	Total Molecule Moves Total Molecule Rotates Total Stamp Rotates
Averages / Level	Av. Slider Moves Av. Offset Moves Av. Amplitude Moves Av. Wavelength Moves Av. Move Type Changes Av. % Offset Moves Av. % Amplitude Moves Av. % Wavelength Moves	Av. Molecule Moves Av. Molecule Rotates Av. Stamp Rotates
Timing / Move Attributes	Slider St. Dev. Slider Min / Max Av. Slider St. Dev. / Level Av. Slider Max / Min Total Time Av. Time / Level	Av. Molecule Move Time Total Time Av. Time / Level Total Time in "Museum"
Scoring		Av. Score / Level
Resets / Replays	Av. Valid / Invalid Transitions Total Valid / Invalid Transitions	Av. Resets / Level Av. Completes / Level

2.4 Modeling process

We modeled the data using several algorithms provided by RapidMiner, a multiplatform data science tool [6]. This tool was chosen for ease of use and free use in educational contexts across the vast majority of computational platforms (OSX, Windows, and Linux). Individual models were generated for quitting at levels 1, 3, 5 and 7 for each game. Individual models were also generated for each knowledge assessment, where the results of the assessment were represented as a binomial indicating a correct vs. incorrect response. Models that were used included RapidMiner's implementations of Naive Bayes, Generalized Linear Model, Logistic Regression, Fast Large Margin, Deep Learning[1],

[1] RapidMiner's Deep Learner component is based on a multi-layer feed-forward artificial neural network. For more details see:

Decision Tree, Random Forest, Gradient Boosted Trees and Support Vector Machine. RapidMiner's default hyperparameters were used for all models, including a preprocessing step to standardize all values to have zero mean and unit variance as well as the option to use a single thread to ensure reproducibility. Model specific hyperparameters are seen in Appendix A. A single 60/40 split process randomly divided the source data into a 60% training set and 40% validation set. Accuracy percentage of each model was determined, along with baseline accuracies for quitting and knowledge assessment. The same initial feature space was used to predict both quitting behavior and post-test performance.

3. RESULTS

Table 3. Performance for predicting instances of quitting at each level within *Crystal Cave*.

	Accuracy				
	LV1	LV3	LV5	LV7	Av.
Baseline	0.789	0.769	0.679	0.576	0.703
Naive Bayes	0.819	0.749	0.685	0.687	0.735
Generalized Linear Model	0.883	0.772	0.785	0.687	0.782
Logistic Regression	0.897	0.763	0.786	0.667	0.778
Fast Large Margin	0.863	0.771	0.666	0.460	0.690
Deep Learning	**0.908**	**0.737**	**0.786**	**0.707**	**0.784**
Decision Tree	0.863	0.767	0.666	0.649	0.736
Random Forest	0.861	0.762	0.676	0.660	0.740
Gradient Boosted Trees	0.850	0.762	0.777	0.686	0.769
Support Vector Machine	0.762	1.091			

Table 4. Performance for predicting instances of quitting within *Wave Combinator*.

	Accuracy				
	LV1	LV3	LV5	LV7	Av.
Baseline	0.899	0.819	0.625	0.911	0.814
Naive Bayes	0.956	0.819	0.629	0.914	0.830
Generalized Linear Model	1.000	0.818	0.624	0.914	0.839
Logistic Regression	**0.999**	**0.819**	**0.630**	**0.914**	**0.841**
Fast Large Margin	1.000	0.819	0.619	0.914	0.838
Deep Learning	0.999	0.657	0.425	0.717	0.700
Decision Tree	0.997	0.819	0.629	0.914	0.840
Random Forest	0.974	0.819	0.630	0.914	0.834
Gradient Boosted Trees	n/a	0.699	0.582	0.888	n/a
Support Vector Machine	0.999	0.818	0.621	0.908	0.836

Baseline calculations were quite high for predicting quitting at each of the different levels across both games. For example, 91.1% of players who start level 7 in *Wave Combinator* also complete it. By predicting that all players will complete level 7, a model will have a 0.911 accuracy, leaving very little room for improvement. Across all levels, a baseline model that always predicts completing the level will have an average accuracy of 0.703 for *Crystal Cave* and 0.814 for *Wave Combinator*.

For predictions of quitting at each level in *Crystal Cave*, Deep Neural Networks performed best on average. The most accurate prediction was 0.908 for level 1, followed by 0.786 for level 5, 0.737 for level 3 and 0.707 for level 7. The largest improvement over the baseline was for level 7, with the model performing 22.7% more accurately. This was followed by level 5 with a 15.8% improvement over baseline, and level 1 with a 15.1% improvement over baseline. The model performed slightly worse than baseline (0.958) predicting quitting for level 3. On average, the Deep Neural Networks predicted quitting with an accuracy of 0.784. All models performed better than the baseline model except for Fast Large Margin.

For predictions of quitting at each level in *Wave Combinator*, Logistic Regression was the most accurate, but offered little improvement over baseline for most levels. The most accurate prediction was 0.999 for quitting in level 1 followed by 0.914 for level 7, 0.819 for level 3 and 0.630 for level 5. The largest improvement over baseline was seen in level 1 with the model performing 11.2% more accurately. This advantage quickly dissolves with only a 0.8% improvement in level 5, a 0.3% improvement for level 7 and a 0.1% improvement for level 3. On average, Logistic Regression predicted quitting with an accuracy of 0.841. Deep Learning and Gradient Boosted Tree algorithms failed to perform better than the baseline model for this prediction.

Table 5. Performance for predicting incorrect answers for each assessment question in *Crystal Cave*.

	Accuracy			
	Q0	Q1	Q2	Av.
Baseline	0.588	0.724	0.574	0.656
Naive Bayes	0.594	0.732	0.575	0.634
Generalized Linear Model	0.594	0.732	0.575	0.634
Logistic Regression	**0.603**	**0.745**	**0.600**	**0.649**
Fast Large Margin	0.585	0.732	0.625	0.647
Deep Learning	0.500	0.591	0.450	0.514
Decision Tree	0.581	0.732	0.600	0.638
Random Forest	0.585	0.732	0.500	0.606
Gradient Boosted Trees	0.543	0.706	0.525	0.591
Support Vector Machine	0.568	0.691	0.650	0.637

https://docs.rapidminer.com/latest/studio/operators/modeling/predictive/neural_nets/deep_learning.html

Table 6. Performance for predicting incorrect answers for each assessment question in *Wave Combinator*.

	Q0	Q1	Av.
Baseline	0.540	0.776	0.658
Naive Bayes	0.446	0.718	0.582
Generalized Linear Model	0.588	0.771	0.680
Logistic Regression	**0.590**	**0.776**	**0.683**
Fast Large Margin	0.453	0.774	0.614
Deep Learning	0.489	0.585	0.537
Decision Tree	0.549	0.774	0.661
Random Forest	0.546	0.774	0.660
Gradient Boosted Trees	0.578	0.728	0.653
Support Vector Machine	0.550	0.776	0.663

Baseline predictions for the assessment items were lower than for quitting, but still much higher than a fair coin toss. Averaging across the 3 items in the *Crystal Cave* and the 2 items in the *Wave Combinator*, a baseline model that always predicts a correct answer will have an accuracy of 0.656 and 0.658, respectfully.

For predicting incorrect answers on the 3 assessment items in the *Crystal Cave*, Logistic Regression was the most accurate. The model best predicts the outcome of question 1 with an accuracy of 0.745, followed by question 0 with an accuracy of 0.603 and question 2 with an accuracy of 0.600. Compared to the baseline, the greatest improvement was 4.4% on question 2. The model demonstrated a 2.8% improvement on question 1 and 2.5% improvement on question 0. On average, the model predicted incorrect answers with an accuracy of 0.649.

For predicting incorrect answers on the 2 assessment items in the *Wave Combinator*, Logistic Regression was the most accurate. The model has an accuracy of 0.776 for question 1 followed by an accuracy of 0.590 for question 0. This translates to an improvement of 9.2% for question 0 and identical accuracy to baseline for question 1. On average, the model predicted incorrect answers with an accuracy of 0.683.

4. DISCUSSION

This paper compares the accuracy of 9 common modeling algorithms for predicting quitting and knowledge assessment in two different learning games using the simplest possible feature engineering. We found that, on the whole, these models were able to successfully predict quitting behavior and correct answers in our two games and their associated post-tests. This is a promising finding for continuing to deploy educational data mining methods in order to capture and identify learning and behaviors of interest within digital games.

Accurate prediction of quitting behaviors and post-test performance has a number of practical applications within educational settings. For instance, players who are identified as being at-risk for quitting a level may be given targeted behavioral or affective scaffolds to keep them on-task and working productively. Players who have a low predicted score for a post-test assessment can be given additional practice opportunities on-demand, based on the specific misconceptions or difficulties they are having.

That said, there is room for improvement in the performance of these models. More complex, move sequence features may lead to more meaningful descriptors of the player's thinking. While the features that were used in this paper were certainly grounded in the interactions afforded to the player, they were only computed in terms of simple counts and averages. One possible next step would be to use sequential pattern mining to first identify common sequences of moves that correlate with outcomes of interest [12]. The presence of these patterns could then be used as an engineered feature to train the models.

The extreme accuracy (0.999) of level 1 quitting predictions for the wave combinator invites speculation for the usefulness of building models based only on very recent events. The approach used here was to use all player actions leading up to the quitting or assessment event. This may have the unintended consequence of diluting player moves that may immediately lead to a success in a specific level, with moves from much earlier in the gameplay that are now irrelevant to the challenge at hand. A next step would be to modify the feature generating scripts to experiment with different time windows for modeling.

Another limitation of this work is that accuracy may not be the best measure of the effectiveness of the predictions. In future work, the performance of the models should be reported by providing precision, recall and F1 scores. This issue is compounded by the fact that baseline predictions, based only on the percentages of players that complete a level or correctly answer a quiz item, are quite high, leaving very little room for improvement. The authors are unable to conclude that the models are deriving their accuracy from the strength of the features and not simply the unbalanced distribution of the phenomena.

Finally, the validity of the answers provided for the multiple choice assessment items could be studied. These items are not standardized measures, but reasonable assessments designed by the researchers. Further evaluating their validity and reliability may highlight insights as to why they are harder to predict. Additionally, by modifying the system to record the time spent answering each assessment would help identify obvious issues such as spending less than 1 second before answering, not nearly enough time to read and decide on a correct answer.

5. SUMMARY

In summary, logistic regressions performed better than all competing algorithms for quitting in *Wave Combinator* and content knowledge tests in both games. Deep Learning models performed best in predicting quitting in the *Crystal Cave* game. Level quits can be predicted with an average accuracy of 0.784 for *Crystal Cave* and 0.841 for *Wave Combinator*, an improvement of 12.4% and 3.1% over baseline, respectfully. Correct answers across the embedded knowledge assessment items can be predicted with an average accuracy of 0.649 for *Crystal Cave* and 0.683 for the *Wave Combinator*. The models provided a 3.3% and 4.6% improvement over baseline for these games.

These results show that educational data mining techniques can provide some predictive value to different kinds of educational games even with relatively minimal feature engineering. We hope that other researchers can be encouraged to apply similar methods to their own games given our results.

6. ACKNOWLEDGMENTS

The authors gratefully acknowledge partial support of this research by NSF through the University of Wisconsin Materials

Research Science and Engineering Center (DMR-1720415) and the Wisconsin Department of Public Instruction.

7. REFERENCES

[1] Crystal Cave [Computer Software]. (2017). Madison: Field Day.

[2] Karumbaiah, S., Baker, R.S., Shute, V. (2018) Predicting Quitting in Students Playing a Learning Game. *Proceedings of the 11th International Conference on Educational Data Mining*, 21-31.

[3] Kiili, K., Devlin, K., Perttula, A., Tuomi, P., & Lindstedt, A. (2015). Using video games to combine learning and assessment in mathematics education. *International Journal of Serious Games*, *2*(4), 37-55.

[4] Maguth, B., List, S., & Wunderle, M. (2015). Teaching Social Studies with Video Games. *The Social Studies*, *106*(1), 32-36.

[5] Malkiewich, L., Baker, R.S., Shute, V., Kai, S., Paquette, L. (2016) Classifying behavior to elucidate elegant problem solving in an educational game. *Proceedings of the 9th International Conference on Educational Data Mining*, 448-453.

[6] Mierswa, I., & Klinkenberg, R. (2019). RapidMiner Studio (9.2) [Data science, machine learning, predictive analytics]. Retrieved from https://rapidminer.com/

[7] Rowe, E., Asbell-Clarke, J., Baker, R., Gasca, S., Bardar, E., & Scruggs, R. (2018). Labeling Implicit Computational Thinking in Pizza Pass Gameplay. In *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems*.

[8] Shute, V. J., Ventura, M., & Ke, F. (2015). The power of play: The effects of Portal 2 and Lumosity on cognitive and noncognitive skills. *Computers & Education*, *80*, 58-67.

[9] Shute, V. J., Ventura, M., & Kim, Y. J. (2013). Assessment and learning of qualitative physics in newton's playground. *The Journal of Educational Research*, *106*(6), 423-430.

[10] Watson, W. R., Mong, C. J., & Harris, C. A. (2011). A case study of the in-class use of a video game for teaching high school history. *Computers & Education*, *56*(2), 466-474.

[11] Wave Combinator [Computer Software]. (2017). Madison: Field Day.

[12] Wallner, G. (2015). Sequential Analysis of Player Behavior. In *CHI PLAY '15 Proceedings of the 2015 Annual Symposium on Computer-Human Interaction in Play*, 349–358. https://doi.org/10.1145/2793107.2793112

[13] Pavlik Jr., P.I., Cen, H., & Koedinger, K.R. (2009). Performance Factors Analysis – A New Alternative to Knowledge Tracing. In V. Dimitrova & R. Mizoguchi (Eds.), Proceedings of the 14th International Conference on Artificial Intelligence in Education. Brighton, England.

[14] Zoombinis [Computer Software]. (2015). TERC.

8. APPENDIX A: Hyperparameters used for each model

Model	Hyperparameters
Naive Bayes	n/a
Generalized Linear Model	Family = binomial Solver = L_BFGS
Logistic Regression	Solver = L_BFGS
Fast Large Margin	Strategy = 1 against all
Deep Learning	Activation = rectifier Hidden layer sizes = 50,50 Epochs = 10.0
Decision Tree	Criterion = gain_ratio Maximal depth = 2 Apply Pruning Confidence = 0.1 Minimal Gain = 0.05 Minimal Leaf Size = 2
Random Forest	Trees = 20 Criterion = gain_ratio Max Depth = 7 Apply Pruning Confidence = 0.25 Minimal gain = 0.05 Minimal Leaf Size = 2 Guess subset rratio Voting Strategy = confidence vote
Gradient Boosted Trees	Trees = 60 Max Depth = 2 Min Rows = 10 Min Spilt Improvement = 0 Bins = 20 Learning Rate = 0.1 Sample Rate = 1.0
Support Vector Machine	Type = C-SVG Kernel = rbf Gamma = 1.0E-4 C = 100.0 Epsilon = 0.001