Final Report

Predicting Mastery in Academic Assessments

Students taking advantage of practice exercises through software applications seek an evaluation of their preparedness for formal academic assessments. The goal of this project is to help test developers understand user progress by predicting whether a student answers the next test question correctly.

Data are from students preparing for three test groups - ACT, GMAT and SAT. Question fields in the dataset indicate outcome, group, track (test subject), sub-track, 'game' structure, and times for when the question was started and answered or deactivated. Relationships that could predict observations under the field labeled correct are explored and modeled throughout the project.

The dataset spans three years of students answering questions to prepare for the different tests. In order to predict a user's next test question, the data were split (by the test developers) in the following manner:

- If a user answered at least 6 questions, one of their answers was selected randomly from the 6th question to the last. This plus all later question attempts by this user were removed.
- Earlier answers were included in the **`training.csv`** set, which is used for this project.
- All answers from users answering fewer than 6 questions are also included in the **`training.csv`** set.

APPROACH

1. **Initial Data Exploration** - Import and inspect raw data. Isolate relevant variables, fill or calculate new variables, and organize the dataframe. Resolve missing, invalid, corrupted, duplicate values.

2. **Exploratory Data Analysis** - Create data visualizations to explain variables. Detect and possibly remove or mark outliers. Explore variable dependence and correlations. Consider a hypothesis to explore. Leverage statistical inference to test the hypothesis. More generally, begin to develop a preliminary likeness of the solution.

3. **Machine Learning** - Build, fit, and validate a method to model the data. Evaluate the performance of each model tested, including Logistic Regression, Random Forest, Linear Mixed Effects, Mixed Effects Random Forest, XGBoost.

INITIAL DATA EXPLORATION

Organize Columns: Redundant columns such as **`outcome, answer_id, question_set_id`** were removed. A new column was created, **`round_ended_at`**, filled with `answered_at` values if present. If no value was found in any `answered_at` field, then the value from `deactivated_at` was used to populate `round_ended_at`.

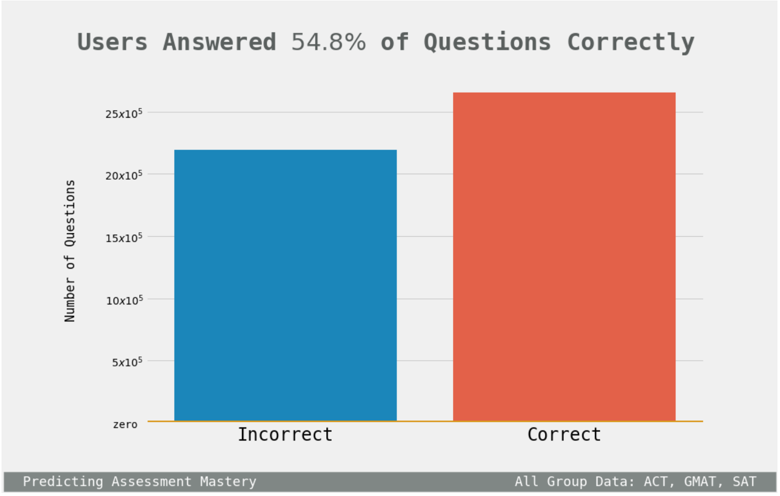
Evaluate & Resolve Missing Data: NULL values in the dataframe indicate that the value is missing, invalid or corrupted. These were expected for **`answered_at`** since this denotes that the question timed-out. Missing values for **`date_of_test`** were also no surprise since it was known that users may or may not enter an expected test date. NULL values for **`round_started_at`** and **`deactivated_at`** were less understood and, therefore, examined more closely. All observations with missing data for all timedate variables were eliminated. Another 21 rows were dropped for missing **`answered_at`** and **`deactivated_at`** values since round duration cannot be calculated in those rows. About 87% of users did not provide an expected test date, so the **`date_of_test`** column was dropped. NULL values also existed in the **`game_type`** column, as a value of 6 represents a data collection failure. Most, if not all, of the observations involve rounds with end times earlier than start times, so they were dropped.

Evaluate & Resolve Duplicate Data: Given the reasonable expectation that it is impossible to start and end a round at the same time, 4264 observations meeting such conditions were dropped. After weird time issues were resolved, duplicate observations for the combination of **`user_id`**, **`round_started_at`** and **`round_ended_at`** were dropped.

Compute and Explore Time Data: The **`round_duration`** column was created and filled by subtracting `round_started_at` values from `round_ended_at` values. A view of time data statistics revealed that some rounds were days long and many were over within only a few seconds. Rather than delete the 167,564 observations outside a threshold of 5 seconds to 10 minutes, new columns were created to categorize fast and slow extremes. One variable indicating the number of **`prior_attempts`** of a particular question by the user, and one variable indicating when these become **`excessive_attempts`** (> 100) were also created.

	Total Observations	Total Variables	Number of Users	Number of Unique Questions
Pre Cleaning	4,851,475	17	179,106	6,045
Post Cleaning	4,846,730	22	179,033	6,045
ELIMINATED	4,745	(9 added) 4	73	0

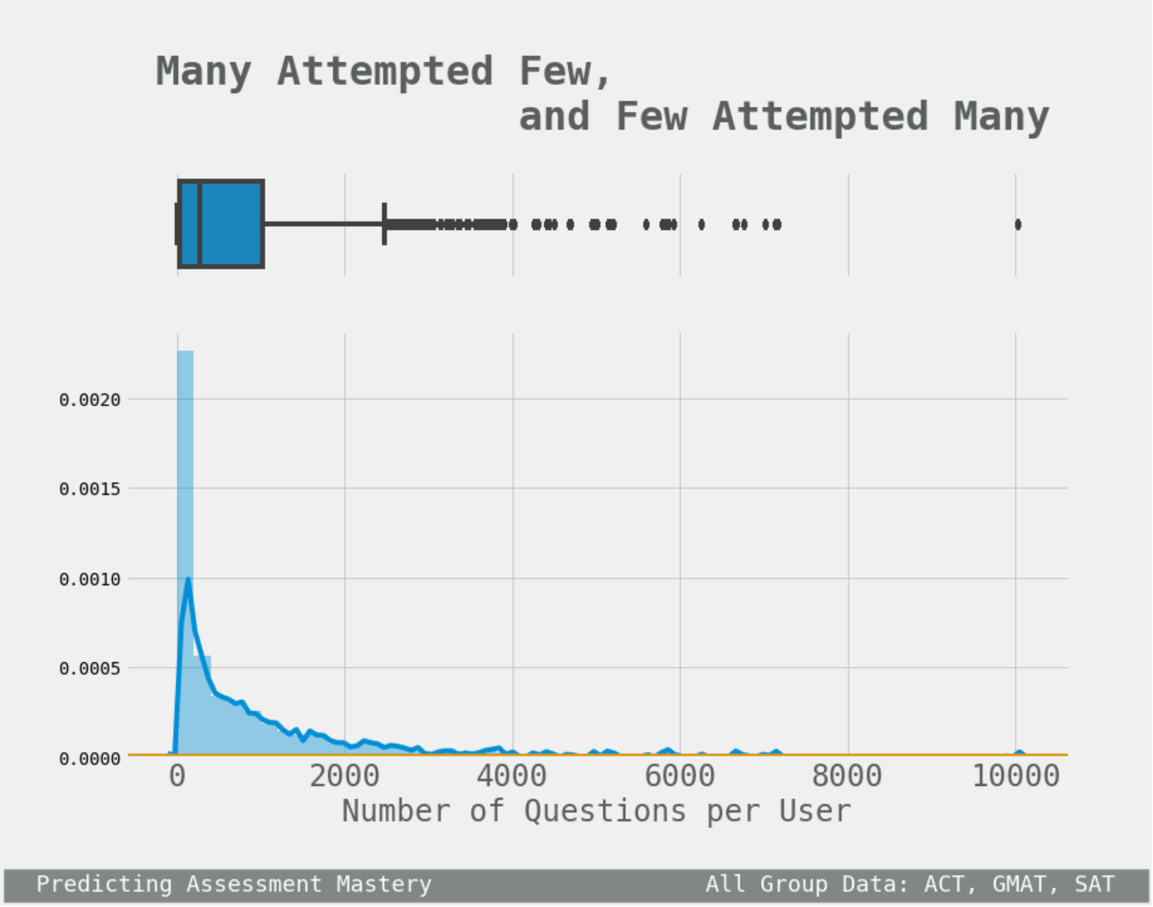
EXPLORATORY DATA ANALYSIS



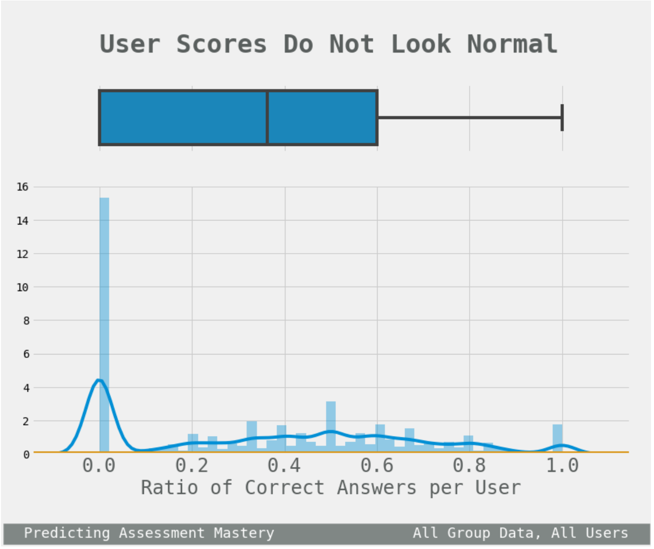
Dataset Statistics: For all records in the clean dataset the mean percentage of correct answers is 54.8% and the mean round duration is 1 minute, 8 seconds.

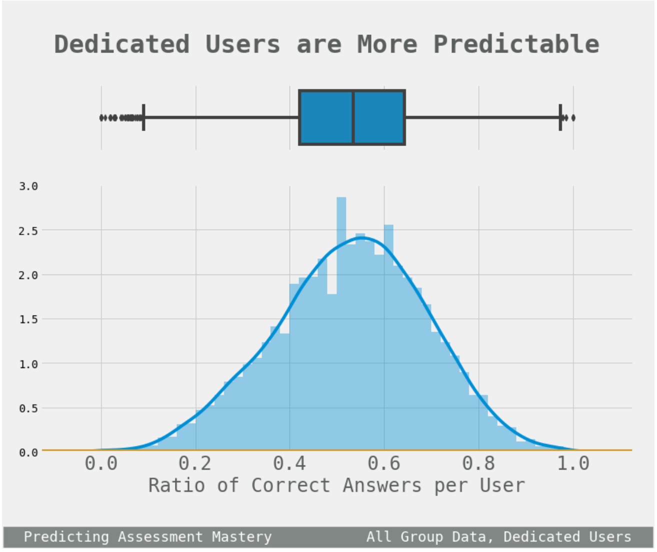
While more than half of all questions in the dataset were answered correctly, it's difficult to evaluate this alone from a performance point of view since so many users attempted so few questions…

User Data: Users attempting more than 4,000 questions are scarce, but at least one attempted at least 10,000 questions.



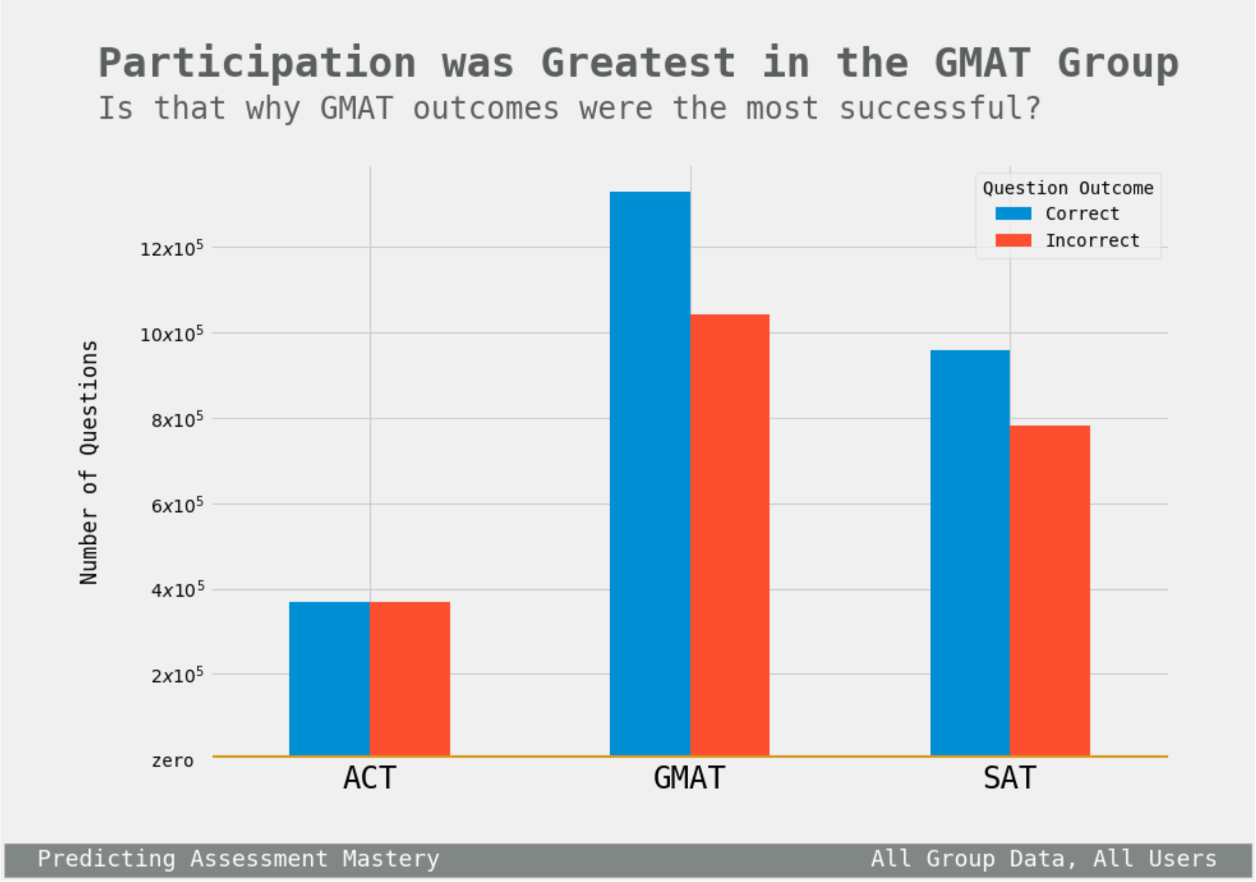
This certainly helps explain the distribution of the ratio of correct answers by user. When filtered for dedicated users, defined as having attempted at least 30 questions, the distribution is normal.





Question Data: The distribution of unique questions showed that a large number are underrepresented in the dataset. This is explained by the distribution of test groups (next section). One user attempted one particular question 1268 times. This user was eventually dropped. There were at least 12 other questions associated with excessive attempts but the closest count amongst them was for 274 prior attempts.

Group Name Data: A cross-tabulation of **`group_name, track_name,`** and **`subtrack_name`** determined that there is no overlap of tracks or subtracks across groups. The groups seem to be independent, so the dataframe was split to explore each group individually. Helped by the graph below, GMAT was ultimately deemed best for modeling.



ACT Data: About 50% of questions in the ACT group were answered correctly, and the average round duration was about 53 seconds.

- Average score for ACT users: 0.347707
- Number of ACT users with no correct answers: 12198
- Average score for dedicated ACT users: 0.507080
- Number of dedicated ACT users with no correct answers: 8

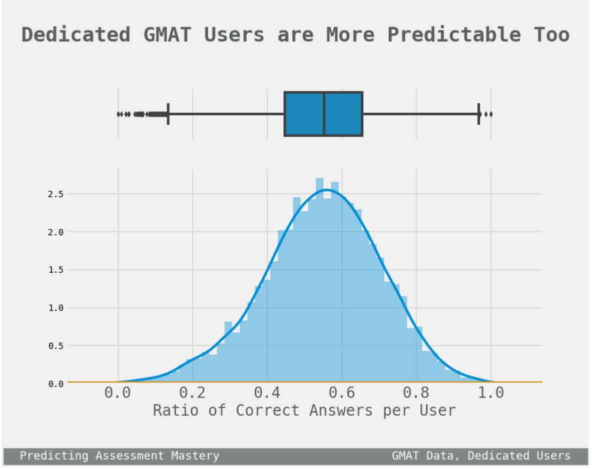
GMAT Data: About 56% of questions in the GMAT group were answered correctly, and the average round duration was about 1 minute 18 seconds.

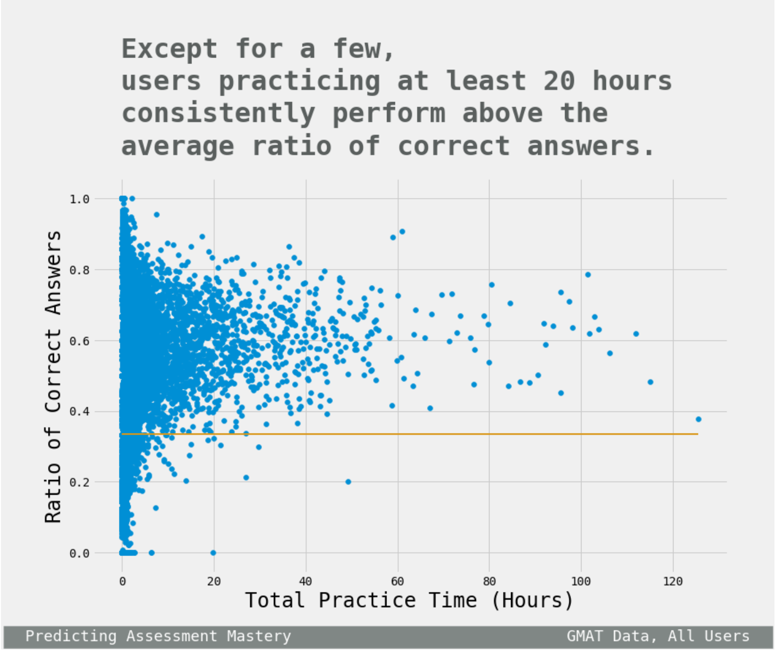
- Average score for GMAT users: 0.333503
- Number of GMAT users with no correct answers: 28007
- Average score for dedicated GMAT users: 0.547034
- Number of dedicated GMAT users with no correct answers: 2

SAT Data: About 55% of questions in the SAT group were answered correctly, and the average round duration was about 1 minute.

- Average score for SAT users: 0.381569
- Number of SAT users with no correct answers: 19281
- Average score for dedicated SAT users: 0.524551
- Number of dedicated SAT users with no correct answers: 3

For each test group, outcomes are better across all subjects for dedicated users. This effect is most dramatic for the GMAT group. Of all scenarios, average scores are best amongst the dedicated GMAT users.





Hypothesis Testing: To ensure the independence of test groups, statistical analysis comparing group means was performed.

Null Hypothesis: All test groups are from the same distribution.

- For this, a one-way ANOVA test was performed on the average number of correct answers for each group.
- `alpha=0.05 > p-value=0.00`, and `F-value=4070.79 > F-critical=2.9957`
- Reject the null hypothesis - **At least one of the three test groups is from a different distribution.**

Null Hypothesis: The GMAT and SAT groups are from the same distribution.

- For this, a T-test for two independent samples was performed on the average number of correct answers for each group.
- `alpha=0.05 > p-value=0.00`, and `t-statistic=1.61`
- Reject the null hypothesis - **The GMAT and SAT groups are from different distributions.**

Null Hypothesis: GMAT users and dedicated GMAT users are from the same distribution

- For this, a T-test for two related samples was performed on the average number of correct answers for each group.
- `alpha=0.05 > p-value=0.00`, and `t-statistic=-40.1945`
- Reject the null hypothesis - **GMAT users and dedicated GMAT users are from different distributions.**

Summary: A small number of users practiced in more than one test group; however, EDA and statistical analysis indicate the test groups are independent. For each test group, outcomes are better across all subjects for dedicated users. In every scenario, user score distributions were more interpretable (normal) amongst the dedicated users.

MACHINE LEARNING ANALYSIS

Build, fit, and validate a method to model the GMAT data. Make predictions for each model and evaluate the performance. Return the strongest predictors for the best model.

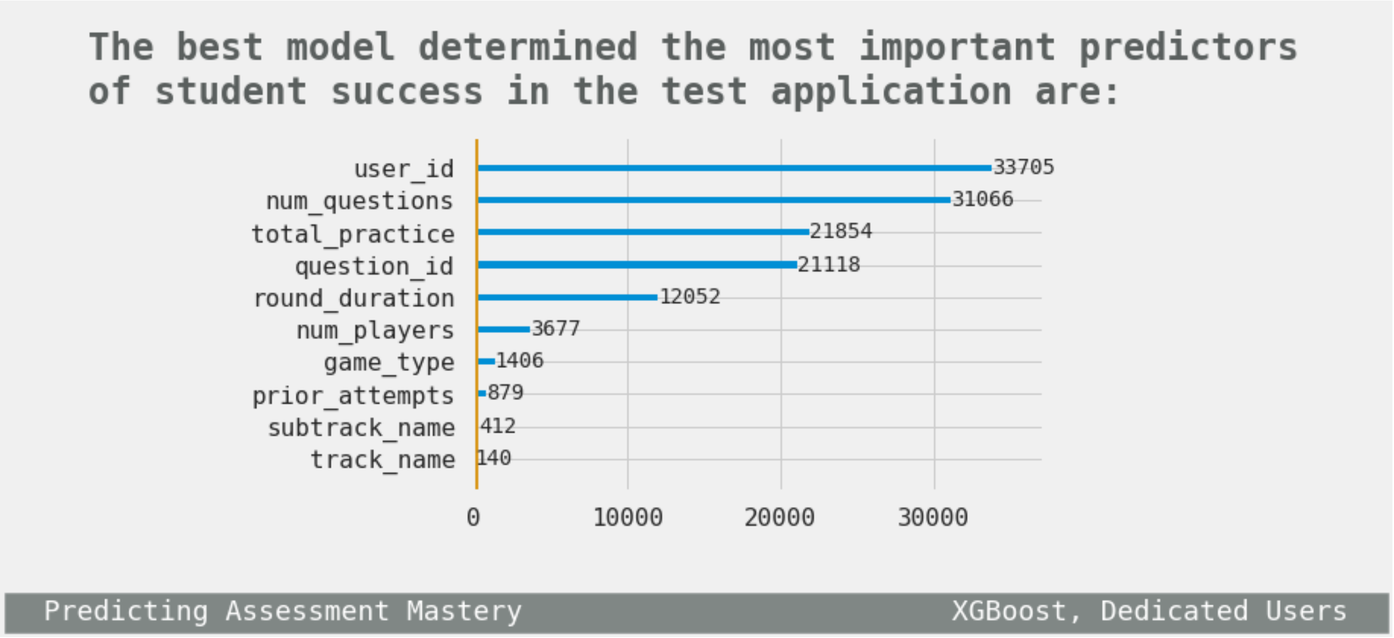
- **Logistic Regression** - A classification method for finding the probability of a given outcome, but assumes that observations are independent. Grouped by user, observations in this project are not independent. But this provides an easy and fast baseline for repeat simulations.
- **Random Forest** - This creates an ensemble of decision trees which are averaged to improve prediction accuracy, and it scores how important each feature was for classification.
- **XGBoost** - This implements gradient boosted decision trees, and scales very well to many samples. It works by iteratively learning a set of weak models on subsets of the data, weighing each weak prediction according to each weak model's performance, and combining the weighted predictions to obtain a single weighted prediction.
- **Linear Mixed Effects** - These models are useful for analyzing data that are dependent, hierarchical, or longitudinal. The tracks and subtracks suggest hierarchy and there is certainly a longitudinal nature of observations by user.
- **Mixed Effects Random Forest** - A method developed by a group of statisticians to account for mixed effects while using Random Forest as the estimator.

To compare models, two performance measures were computed for each: 1.) Accuracy of predictions on test data. Higher values are better. 2.) Capped Binomial Deviance, accounting for probability of responses. Lower values are better.

All models looked for patterns in these variables: **`user_id, question_id, track_name, subtrack_name, game_type, num_players, round_duration, prior_attempts, num_questions, total_practice`**. Removing the least important amongst these did not improve performance.

Model Name	Accuracy	CBD
xgboost	0.6983	0.2454
random_forest	0.6687	0.2554
mixed_effects_randforest	0.6683	0.2583
logistic_regression	0.6164	0.2889
linear_mixed_effects	0.5692	0.3017

Since XGBoost was the best model by both measures, it was then used to fit data filtered for dedicated users. Both performance measures improved. Accuracy: **0.7109** CBD: **0.2411**



The best model determined the most important predictors of student success in the test application are:

Predicting Assessment Mastery XGBoost, Dedicated Users

CONCLUSIONS

Exploratory data analysis and machine learning back up the intuition that the more a user practices the more successful they are in answering questions. More importantly, both show that dedicated users provide enough data for a more reliable analysis of their performance. Features related to subject matter and 'game' structure were less helpful as predictors.

Recommendation to Test Developers: To better understand user progress and improve test applications, developers are encouraged to incentivize users to attempt at least 30 questions.