PLURALSIGHT | DATA

Data Scientist Take-Home Exercise

Problem Statement

One of Pluralsight's key product offerings is adaptive skill assessments. We offer these for roughly 50 different skills, ranging from developer tools to design applications (like Photoshop) and engineering tools (like AutoCAD), to basic business skills. Each assessment session usually takes less than 5 minutes, and users are shown a skill score, along with targeted course recommendations based on their results. Users can track their progress as they learn, and Pluralsight can use the results to quantify the effectiveness of our content. We have shared a small, anonymized subset of our assessments data with you to observe how you explore and understand its structure, relationships, and distributional properties.

Brief Scoring Algorithm Primer

The assessment algorithm is a modified Glicko* scoring rule that quantifies question difficulty and user skill in the same statistical space, so they are directly comparable. Questions and users are characterized by 2 parameters, referred to as ranking (the mean of the distribution) and RD (essentially the standard deviation).

The algorithm starts questions and users with an average ranking and a high degree of uncertainty, and then homes in on their true rankings as users answer questions. Each time a user answers a particular question correctly, the user's ranking improves, and that question's ranking decreases (and vice-versa) by an amount corresponding to the uncertainty of both the user and the question at that point in time. Once a user's uncertainty is low enough, the session is concluded and a score is assigned.

*It may help to read the Wikipedia entry on Glicko, but you shouldn't need intimate knowledge of it to answer the questions.

Data Background Information

We are providing you with a SQLite database consisting of several tables. They contain data for a selection of user and question scores, and question meta-data for four of our skill assessments.

These tables include

Table Name	Description
user_assessment_sessions	A collection of summary statistics detailing several initiated, but not necessarily completed assessment sessions
user_interactions	A record of all user-oriented interactions over the course of the sessions in user_assessment_sessions
question_interactions	A record of all question-oriented interactions for all sessions in user_assessment_sessions
question_details	The content ID and topic for each question that appears in question_interactions

Instructions

Please submit responses to the following questions, along with your supporting code (in SQL, R and/or Python), and any additional materials relevant to your analysis.

Please clearly state any assumptions you apply to your analyses or observations. Rest assured that we recognize there are nuances in how our algorithm and business work that you would learn on the job, and that you are not expected to know for this exercise.

It is our intention that you will not have to spend more than 6-8 hours on this exercise.

Questions

We will not “grade” your answers to these questions, rather we will use your work to guide a review during a potential next round of interviews. Be prepared to explain your questions, thought process, assumptions, and decisions.

Data Exploration Questions

1. Describe and visualize how the distributions of user and question rankings compare and relate between assessments.
2. How does it appear the algorithm determines when a user’s assessment session is complete?
3. Which of the assessments has the highest and lowest dropout rates, respectively?
4. Is there significant variance in question difficulty by topic within a given assessment?
5. How many times must a question be answered before it reaches its certainty floor? Does that number appear to be constant or does it vary depending on question or assessment?

More Involved/Open-ended Questions

1. Identify a metric that could be used to identify questions that are performing poorly, and consequently might need to be reviewed, changed, or removed.
2. Suppose an update to Python causes a question’s answer to change, but our question authors don’t notice, and the now-outdated question remains in the test. How might that scenario reveal itself in the data?
3. Given your response to number 2 in the Data Exploration Questions above, what is a method we could use to determine ideal points to stop a user’s assessment session (i.e. identify the right balance between certainty and burden on the user)?
4. How could we calculate the overall difficulty level of a particular topic? How might we then calculate a topic-level score for a single user?