Managing and analyzing student learning data: A python-based solution for edX

Vita Lampietti
MIT OpenLearning
Cambridge, MA
vlampietti@gmail.com

Anindya Roy
MIT OpenLearning
Cambridge, MA
anindyar@mit.edu

Sheryl Barnes
MIT OpenLearning
Cambridge, MA
sherylb@mit.edu

ABSTRACT
Online learning platforms, such as edX, generate usage statistics data that can be valuable to educators. However, handling this raw data can prove challenging and time consuming for instructors and course designers. The raw data for the MIT courses running on the edX platform (MITx courses) are pre-processed and stored in a Google BigQuery database. We designed a tool based on Python and additional open-source Python packages such as Jupyter Notebook, to enable instructors to analyze their student data easily and securely. We expect that instructors would be encouraged to adopt more evidence-based teaching practices based on their interaction with the data.

Author Keywords
Online education; data analysis; science of learning; edX; python; learning analytics.

ACM Classification Keywords
K.3.1; D.2.3; E.1; H.5.2; J.1.

INTRODUCTION
Online learning platforms generate a large amount of usage data pertaining to each individual enrolled in the course. Instructors usually interact with the data via dashboards, which show the high-level features of the data (grades, usage of course components such as videos, course completion etc.), while the raw datasets usually lie behind layers of technology, often unfamiliar to the instructor. We believe that easy access to student learning data by the instructors would allow them to adopt more evidence-based teaching practices. All MITx courses have their usage data pre-processed from the edX platform [1] into a Google BigQuery [2] cloud database. Our tool would allow the existing community of instructors (those who use the edX to BigQuery dataflow) to analyze their students' learning data, and share their custom analyses and code snippets with other instructors, thus amplifying the pedagogical advantage. For this project, we used representative data from a large-enrollment freshman MITx course.

L@S 2018, June 26–28, 2018, London, United Kingdom

ACM ISBN 978-1-4503-5886-6/18/06…$15.00
https://doi.org/10.1145/3231644.3231706

Application Description and Underlying Technology
Our tool is a collection of Jupyter Notebooks [3] (to be described later) complete with a sample analysis of student learning data, and a dashboard (complete with the source code) to view individual student performance in comparison to the rest of the class. The dashboard can be run from within the Jupyter Notebook application, or as a stand-alone application. To use this application, the instructors will have to install the requisite Python libraries including Jupyter Notebook on their own personal computer, download the sample Notebooks and the dashboard application we provide, and run the codes. When supplied proper credentials (by the instructors), the Notebook pulls the course data from the Google BigQuery database. Alternatively, the Notebook can import data from the downloaded data files available locally. The dashboard and many examples on the Notebook files are intentionally kept simple to ease the end-users into exploring the datasets, and to make the customization and the coding

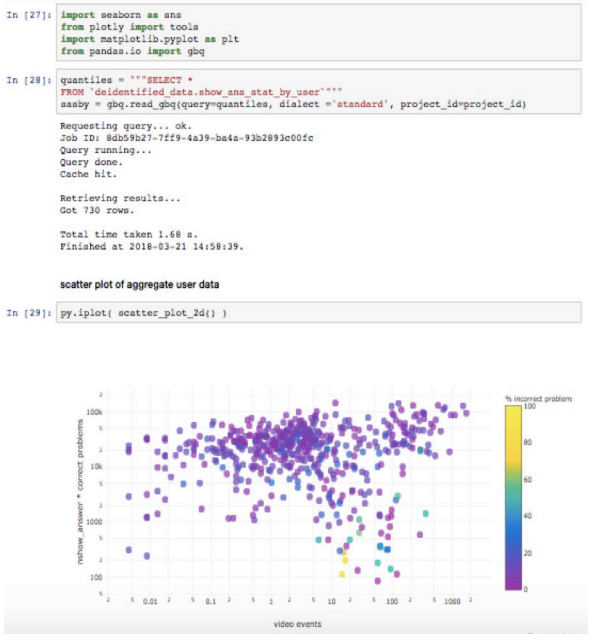

Figure 1: A typical layout of a Jupyter Notebook file. The gray panels contain user input (code, comments), separated by the output panels, containing code results, plots etc.

process transparent.

Jupyter Notebook, formerly known as iPython, is functionally in-between a static document such as PDF, and a collection of computer scripts/packages: it is composed of blocks that can be divvied out into code snippets (along with their output) and text blocks (see Figure 1). One could simply read through the Notebook, or execute the embedded codes contained in it. Jupyter Notebook supports several programming languages, but we have exclusively used the Python programming language.

In addition to the fundamental framework provided by Jupyter Notebook, we made use of several Python-based modules: Pandas to process data [4], Plotly [5] to visualize, and the Dash framework [6] to make the dashboard. Plotly and Dash use react.js [7], a JavaScript library. Dash is built on top of Flask [8], a Python-based web development framework. Our application requires working knowledge of Python, but no knowledge of JavaScript or web development is necessary.

Example Application on the Dashboard

Our example dashboard presents aspects of aggregate student data, and then presents details of individual student performance. This allows instructor to cross-filter students who are struggling in the course or those performing better than the rest. Different instructors may prefer to investigate aspects other than those presented in this example, and becoming familiar with this toolset would help them design what they need. In our sample dashboard (see Figure 2), the aggregate data of all of the course's students can be found in the scatter plot to the left, while the individual data are distributed in the two plots to the right as well as the centered time-series graph below. Each point on the scatter plot represents an individual student and the three other plots represent specific data on whichever student is selected. Selection of an individual student can happen through the clicking of a point on the scatter plot, or by choosing their username from the dropdown menu in the upper right-hand corner.

Figure 2: The dashboard showing cross-filtered data. The scatter plot contains one point per student in the course, approx. 750 students. The other graphs on the page pertain to one individual user, who can be identified either by clicking on the graph or by finding their user ID in the dropdown menu.

DEMONSTRATION

We anticipate the users to interact with the dashboard and Jupyter Notebook on their own devices as described earlier. The Notebooks themselves are interactive and allow for a good amount of manipulation and experimentation.

During the conference, we anticipate having this as an option while also providing a monitor or laptop to allow for easier access to the dashboard application and its functions.

Merit of the Application

Currently, we have a prototype for data analysis for an individual MITx course. Next steps involve deployment of the application and gathering of feedback from actual instructors to improve the Notebooks and the dashboard. We are optimistic about the versatility of our program, and anticipate it being easy to use with little to moderate programming knowledge. Once we have collected feedback and made improvements to our original Notebook, we will share this tool with the broader edX community. Ultimately, the value of a tool like this is twofold. First, it allows for the advancement of evidence-based teaching practices through data-supported experimentation and analysis by the instructors. Second, a common framework of analysis and sharing medium would facilitate the broader edX educator community to collaborate and share their work. Specifically, our example dashboard is useful for determining those students that are either performing under or over the mean, which can have a positive impact on catching students that might need extra support.

Online learning platforms collect a massive quantity of learning analytics. For the learning data to be useful, educators need easy access to the data, and multiple ways to share their use of students' learning analytics with the community. Our application provides a common framework for educators on edX using the edX to BigQuery dataflow.

REFERENCES

1. edX, https://www.edx.org/
2. Google BigQuery, https://bigquery.cloud.google.com/
3. Jupyter Notebook, http://jupyter.org/
4. Pandas, https://pandas.pydata.org/
5. Plotly, https://plot.ly/
6. Dash, https://plot.ly/products/dash/
7. Javascript library: react.js, https://reactjs.org
8. Flask, http://flask.pocoo.org/