Data Analytics Web App Summary (STEM Center Internship)

SHORTEST VERSION: Faculty advised project for Foothill's tutor-center as to visualize/predict tutor wait-times -- I am writing the automated data processing back-end in Python.

As far as the project is concerned, it will be hosted on a virtual server with new tutor request data added daily to a MySQL database. The back end is being written in Python with Pandas, and the front end in JavaScript with D3 (data visualization library). From the STEM Center website, a graph for the given subject will be displayed over a given **time range** (*quarter/academic year/...*). The **x-axis** will be the **time** (*in days/weeks/...*), while the **y-axis** will be the **average wait-time** (*for any given day/week*).

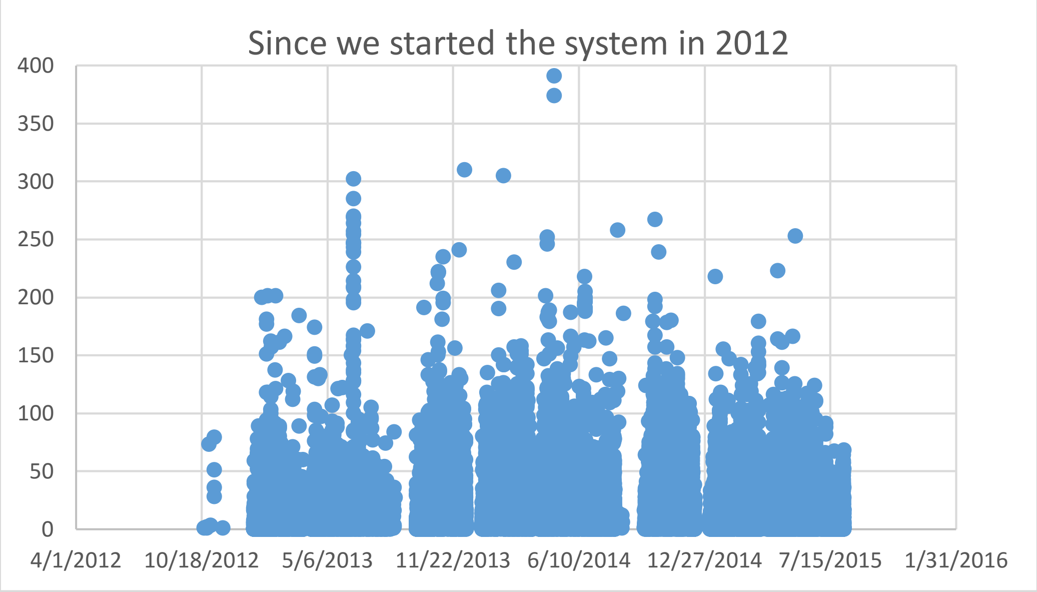
For example, let's say a user was interested in how long students would [on average] be expected to wait for tutoring help in Introductory Java within the fall 2013 quarter. Given the constraints, the data would be filtered, and the corresponding graph would then plot the average wait-time for each day in the fall 2013 quarter.

This would be especially useful for those in management -- as areas close to 0 average wait-times indicate a surplus of tutors available -- whereas wait-times in excess of 30 minutes indicate a sheer lack of tutors for the given subject. In the future, we wish to further optimize the proper scheduling/hiring of tutors by adding a predictive element to the application as well.

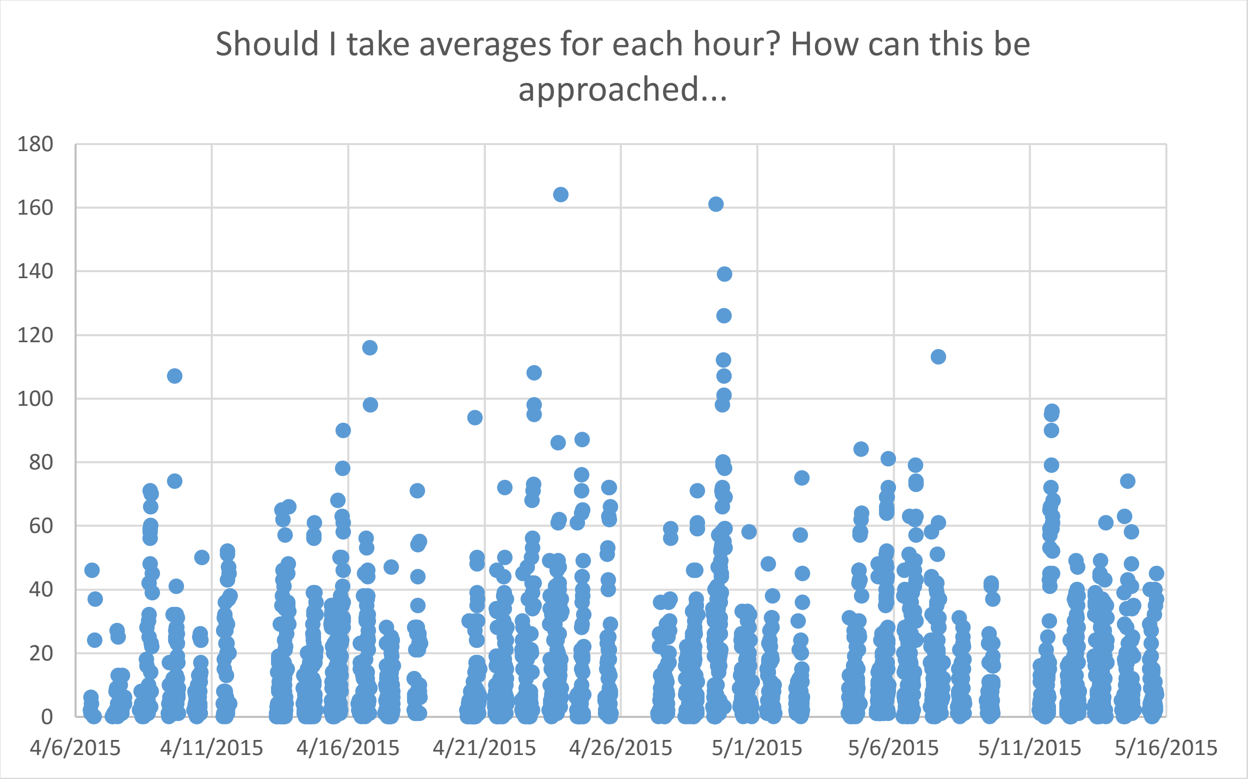
The above is the general summary I have compiled for any one is interest: I have some things I'd like to add specifically…

- We are beginning to get to the point where we need to figure out the best ways to visualize the data, so that we can narrow down how exactly we're going to continue data exploration. The line graph is our first idea, but we need to test things how to figure further to see how exactly we should tweak it…especially in regard to what gives us the most insight.

- Here's a chart I made at the very beginning of the project, with the time on the x axis and wait-times on the y-axis. As you can see, we can't really visualize anything. Even looking at individual days, the data is far too cluttered with lots of outliers floating around for us to do anything.



Data Analytics Web App Summary (STEM Center Internship)



Regardless of the fact that our concern (unlike Izzy's) is centered on the data processing, statistical, and (eventually, like in a quart or so at least) predictive elements – it's important to have better visualization to move on with the project (and so we can display on the website). So I bring this up to show how messy the data is…

That brings me to the next, most important, point. We need to figure out how we will deal with outliers. Most the times well over an hour are what I expect to be freak cases, but it's hard to say for sure, so we need to do lots of statistical analysis on it before Izzy and Bita can do any [useful] visualization – and before we can do any data mining/machine learning with it.

As far as the current status goes…

As of right now, the back-end data cleaning is working (I'll eventually need to refactor it but that's something else). To elaborate since the names are consistent and extreme abnormalities have been removed (incomplete test data from 2012 for example). Also, the table of the 46,000 students is setup and integrated into a Python Pandas DataFrame, with an equally well structured MySQL database – the core functionality of the backend is taken care of. The user-request layer (think of handling web transactions – the highest layer in the overall back-end). Speed is not much of an issue in the foreseen future, so using Pandas should not be an issue, and it'll be relatively straight forward to speed up future code if need be (like NumPy arrays which I can use directly in Pandas with the same efficiency).

Here's the breakdown of the most intermediate objectives for us:

- Do statistical analysis, figure out the range of wait-times that are most definitely extremes (say, is it safe to remove fields containing wait-times in excess of two standard deviations, etc.).
- Once the above is done, I can implement them in Python

  Once that's mostly covered…

- After that, mostly data exploration (through both queries, statistics, online research, and visualization) to figure out what machine learning algorithms would be appropriate, and how we can best approach [reliable] predictive features of the app

  Eventually…

- We will need to figure out how to deal with processing requests – whether we should store things in a database or calculate on the spot – but that's something that won't come up for a few months, since the approach will largely depend on exactly what processes will be necessary, and other constraints. Those are variables we're not certain with at the moment.

There's plenty more, but those are some of the big ones. Perhaps another step is to prioritize the steps and features and the order that would be best to execute.

Last thing I want to mention is in regards to further information: I can familiarize you with the general architecture of the back-end and other specifics when we meet tomorrow, but this should suffice for now.