Social Media Data Science Pipelines Project 3: Live Dashboard-esque Tool

November 29, 2021

1 Introduction

Now that you have a live data collection system running as well as a set of analyses you have performed on the data, it's time to provide some interactivity. In this project, you will thus be developing a simple dashboard that allows a user to vary some parameter of some of the analyses you performed and be shown a result.

2 Project Description

Simply put, you will provide some sort of dashboard (I suggest making it Web based) that will allow for interactive querying and results from *two* analyses from Project 2. The interface is not expected to be super complicated. For example, the demo I will show in class allows the user to choose dates by putting them in the URL, and such an interface (if properly documented) is acceptable.

We are also open to other tools. E.g., a command line tool that allows us to perform some analysis (with different parameters) or a Web API, etc.

3 Project Deliverables

There are three deliverables for this project.

1. Project proposal.
    - FINAL DATE: December 9th
2. Project implementation.
    - FINAL DATE: December 17th

3. Project report.
    - FINAL DATE: December 17th

3.1 Project Proposal

The proposal for this project is much simpler than for the other projects.

You must simply:

1. Specify which two (or more) analyses from your Project 2 you intend to include in your dashboard.
2. Specify which tools you intend to use (e.g., plotting libraries, Web app frameworks, etc.)

3.2 Project Implementation

You will be required to submit all the the code you created. You are also free to use essentially any library or tool that you want (if you have any concerns, reach out as soon as possible). Please note that this code *must* be executable by us, or, at minimum, a live implementation must be reachable by us to grade your work.

For those of you that use Python, you might want to check out Flask (`https://github.com/pallets/flask`) which is approximately equivalent to Sinatra.

You can also check out Django (`https://www.djangoproject.com/`) which is a more full featured Web application framework in Python, or Ruby on Rails (`https://rubyonrails.org/`) which is a more full featured Web application framework in Ruby.

Again, we are not expecting polish at the level of something like SMAT, but your dashboard must not be static: we *must* be able to vary at least one parameter of each of the analyses that your dashboard provides. That said, a slick interface is always appreciated :)

3.3 Project Report

The report for this project should, for the most part be documentation of what you ended up doing in the end. It should also include *clearly and concise* instructions on how to use your dashboard. For example, if it is not actively running, you need to provide us with instructions on how to launch it (including your VM IP address, any authentication necessary, etc.). In all cases, you need to clearly explain the different features (ideally including screenshots) as well as how to use those features. If there are any known bugs, limitations, edge cases, etc. then you also need to let us know about it.

Basically, the report needs to give us enough information to know what your dashboard is supposed to do (ideally you include screenshots in case something goes horribly wrong), how to access it, and how to use it.

4 Grading

- Proposal is worth 25 points.
- Implementation is worth 50 points.
- Final report is worth 25 points.