COMP 598 Homework 4 – Bokeh Dashboard

30 pts
Assigned Sept 30, 2021
Due Oct 15, 2021 @ 11:59 PM (Bonus days due to reading week!)

In this homework, you are a data scientist working with the New York City data division. Your task is to develop a dashboard allowing city leaders to explore the discrepancy in service across zipcodes. You'll be using a derivate of the following dataset:

- https://data.cityofnewyork.us/Social-Services/311-Service-Requests-from-2010-to-Present/erm2-nwe9

NOTE: The original dataset is very large and therefore we have provided a subset of it.

For the purpose of this assignment:

1. Download `nyc_311.csv.tgz` from MyCourses.
2. Trim it down to only include the incidents that occurred in 2020 (for an added challenge, see if you can trim the dataset down using exactly one call to the grep command line tool).

For the remainder of this assignment, you should only work with the trimmed down dataset.

Task 1: Get Jupyter running on your EC2 (5 pts)

Setup Jupyter on your EC2 with password login support. For the EC2 instance, you can reuse the instance you created for previous assignments, or spin up an entirely new machine (recommended).

Create a Jupyter notebook in which you have loaded the data file you trimmed (mentioned above) and printed out the number of distinct zipcodes in the dataset. **You will submit the logs that Jupyter created** when it booted up and handled you logging in via a browser. Note that an automated grading script will be used to evaluate your logs, so they must contain the log text produced when Jupyter successfully boots up AND when you log into the server.

- To capture the log data, redirect the stdout and stderr from your Jupyter notebook command into the file `jupyter.log`

Task 2: Bokeh dashboard (25 pts)

The goal of your dashboard is to allow a city official to evaluate the difference in response time to complaints filed through the 311 service by zipcode. Response time is measured as the amount of time from incident creation to incident closed. Using the dataset from the previous exercise, build a bokeh dashboard which provides in a single column, the following:

- A drop down for selecting zipcode 1
- A drop down for selecting zipcode 2
- A line plot of monthly average incident create-to-closed time (in hours)
    - Don't include incidents that are not yet closed
    - The plot contains three curves:
        - For ALL 2020 data
        - For 2020 data in zipcode 1
        - For 2020 data in zipcode 2
        - A legend naming the three curves
        - Appropriate x and y axis labels

When either of the zipcode dropdowns are changed, the plot should update as appropriate.

Other details:

- Your dashboard should be running on **port 8080**. The **dashboard name (in the route) should be "nyc_dash"**.
- The bokeh dashboard should authenticate any user who logs in with URL params username = "`nyc`" and password = "`iheartnyc`" (**quotes not included**). Failed authentications just need to fail to allow the user in (i.e., they don't need to route the user to a login page that actually exists).
- On any change to either zipcode, your dashboard must update within 5 seconds.
- The IP address of your instance must the specified in a file named `ip_address.txt`
- `ip_address.txt` should only contain a **single line** with ip address to your server for example `54.175.131.58`
- You are welcome to submit a valid domain name, but it will break the unit tests. It will also break our grading scripts - do this at your own risk.
- A design tip: there are WAY too many incidents in 2020 for you to be able to load and process quickly (at least quickly enough for your dashboard to meet the 5 second rule). The way to solve this is to pre-process your data (in another script) so that your dashboard code is just loading the monthly response-time averages for each zipcode … not trying to compute the response-time averages when the dashboard updates.

How to solve this assignment - Hints

1. Please read all the instructions, especially the ones in the **GitHub README**.
2. Familiarize yourself with Bokeh by reading its documentation
3. Ensure your dashboard is accessible
4. You can trim down a subset of the 2020 dataset and do experiments on your local machine
5. Watch out for the EC2 instance type - a micro instance (free tier) might not be enough.

FAQ

1. For some rows the closed date is before the open date resulting in a negative response time. How to handle those rows? - *These rows should be removed. You can use some filter.*
2. If the start date is in Jan and close date in Feb, which month does it belong to? - *Feb. Since that is when the issue was resolved.*
3. Do rows with missing zip codes be included in the overall average? - *These should be removed. At the beginning when you trim the data, you should only choose the rows with zipcode.*
4. Do we include cases based on "open in 2020" or "closed in 2020"? - *Open in 2020.*
5. Should we calculate by day or by hour? - *In hours.*
6. What if there is no end date for an zipcode event? - *Remove it.*

Submission Instructions

Leave your instance up and running until Oct 22nd, 11:59PM EST.

PLEASE read the README.md on Github for HW4.

https://github.com/druths/comp598-2021/blob/main/hw4/README.md

NOTE: It is essential that you follow the submission guidelines as stated in the README file. **Failure to follow exact guidelines would result in losing points.**