Final Project Overview

SI 507 Winter 2022

Milestones

There are 3 milestones that need to be turned in.

- Project Proposal, due March 22
- Data Checkpoint & Interactive Presentation Design, due April 12
- Final Project Demo and Repository Link Submission, due April 26 (keep in mind this is during finals week)

Project Overview

The goal of the final project is for you to showcase what you've learned in 507 regarding:

- Accessing data from the web
  - using either web APIs, including those that require authentication, or web scraping
  - Accessing data efficiently and responsibly using caching
- Use advanced data structures and operations to analyze and process data in "interesting" ways
  - you will need to make a tree or graph of the data
- Use a presentation tool or framework to present data to a user
  - Command Line interface is fine, but using flask or any other presentation framework will be looked upon favorably
- Support basic interactivity by allowing a user to choose among different data presentation options

Here are a couple of examples that would be reasonable final projects:

- A program that access restaurant data from a website, organizes that data into a tree, and then asks users questions (about price, cuisine, location etc) until it provides a set of recommendations that meet the user options
- A graph that is generated based on some set of relationships (such as retweets from a twitter data set, or citations within a corpus of articles), and a program

that can return information about how connected or not connected (what is the shortest path, what are common neighbors etc) two members of the network are.

- The best projects will integrate two related but distinct dataset and therefore be able to tell us something interesting that couldn't be found with just one

Project Components

There are several components that your project must contain. Each of these are detailed in this section.

Data Sources

- You must select data sources that, combined together, give you a "challenge score" of at least 8. Additionally, you must use *either* a Web API that requires authorization *or* a website where you crawl and scrape multiple pages as *one* of your data sources (these options are marked with below). Here's how the scoring works:

Data Source	**Example**	**Challenge Score*** **
Web API you've used before	iTunes, newsapi	2
Web API you haven't used before that requires no authorization	Wikipedia, Google Books	3
Web API you haven't used before that requires API key or HTTP Basic authorization ✣	Yelp Fusion, Open Movie Database	4
Web API you haven't used before that requires OAuth ✣	Open Table, Reddit, many more	6
Scraping a page/site you've worked with before**	nps.gov, si.umich.edu	1
Scraping a new single page**	So many!	4
Crawling [and scraping] multiple pages in a site you haven't used before ✣	So many!	8
CSV or JSON file you haven't used before with	Dataset from	2

> 1000 records	data.gov	
Multiple related CSV or JSON files with at least one file containing > 1000 records	Python Questions from Stack Overflow	4

**: If you choose “scraping a new single page” you can only use this option for *one* of your project sources (i.e., you can’t scrape 2 pages you haven’t scraped before and count it as 8 challenge points).

***: The challenge scores listed here are a guideline, but specific sources may be determined to be more or less challenging depending on the details of the source and how you’re planning to use it.
✣: You *must* use at least one of these options as one of your data sources.

From each source, also need to capture at least 100 records (for CSV/JSON sources you need to capture at least 1000), and each record must have at least 5 “fields” associated with it.

If you have a source you’d like to use that you don’t think fits neatly into one of these categories, consult with your GSI.

Data Access and Storage

For data from APIs or web pages you must cache the raw results (JSON or HTML) you fetch from the source. You will need to demonstrate your use of caching for the Data Checkpoint milestone.

You will also need to load some or all of the data that your application uses into a Database. Generally speaking, any data that is not dynamic (e.g., “recent headlines” from a web API) should be stored in and accessed from a database. Your database must have at least two tables, and there must be at least one relation (primary key - foreign key) between the two tables. Your data processing code (see below) must draw any non-dynamic data from the database.

You will need to be able to create and populate your database from code. You will also need to demonstrate this at the Data Checkpoint.

Data Processing

This is largely up to you, but you need to do whatever is necessary to support the data presentation(s) your program provides. This will require creation of at least one Tree or Graph.

Data Presentation

Use a tool or framework to present data to users on demand. The data should be presented in some way *other* than print( ) statements that output to the terminal. Your program must be able to produce at least 4 different graphs/displays/presentation. These can be different groupings of data, different graph types, or can differ in other ways (if you're not sure if they're "different" enough, check with your GSI).

We will look at two options for data presentation during the final month of class, so you may need to wait until then to get started. Don't worry there's plenty to do before then!

Here are potential approaches for the interactive presentation portion of your project.

1. Provide an interactive command line prompt for user to choose data/visualization options. Display selected graphs using plotly or described via text
2. Create a Flask App that uses HTML links/form elements to prompt for the user to choose data/visualization options. Display selected data using HTML tables (or other elements, as long as the output looks good).
3. If you're feeling ambitious, you can figure out how to use plotly with Flask. It doesn't look too hard, actually: https://stackoverflow.com/questions/36288134/plotly-offline-with-flask. YMMV.

If you wish to use a different data presentation approach, you should check with your GSI.

Upcoming Milestones, in Brief

- March 22: Proposal
  - ~1 page, describing your data sources and how you plan to use them (processing, interaction, and presentation).
  - You will receive feedback on this proposal by Friday, Dec 3.
- April 12: Data Checkpoint & Interactive Presentation Design

    - Data Checkpoint: Demonstrate that you are successfully collecting, caching, and storing in a database all relevant data from your sources.
    - Interactive Presentation Design: ~1 page, describing your plans for implementing interactive presentation capabilities, including user options supported and presentation types
    - You will only receive feedback on this checkpoint if something is wrong!
- April 26: Demo and final submission
    - Refer to the **Final Submission Instructions.**