DSCI 510 Project

Submission 1 (*due March 22)*

1. **Objectives:**
    - Find three (3) data sets on the web that are of interest to you.
        1. One must require "scraping" (i.e. not available via external API)
        2. One must be available via external public web API. (You should be able to access it without a ton of trouble)
        3. The third can be of any type: API, scraped, database, CSV, etc.

        Sites 1 and 2 must be such that they require automation to extract data. If you can just cut-and-paste the data, or just download it in one file, it's not appropriate.
    - Describe in 1 page what analysis or presentation can be expected from the combined data. The data from the different sources must be related and contribute to the joint analysis.

2. **Deliverables:** A PDF document (1-page, including figures and citations, 11pt font) describing the project and the datasets, including:
    - Links (URLs) to the data, and/or other instructions on how to get the data
    - A description of the analysis/questions student would like to answer based on their data.

    You will receive feedback on this submission.

Submission 2 (*due April 12)*:

1. **Objectives:**
    a. Scraping of the data, show code and sample data..
    b. Model data into SQL schema, Data Frame, CSV Files, ORM, etc., and create a diagram to show how the data joins across sources. The diagram should be an Entity-Relation diagram (no need to be fully formal) or just a set of tables showing how the data relates across the tables.

2. **Deliverables:** Submission should contain:
    a. A drawing/diagram of the data (all relations and attributes) at each source and how the data relate across sources (join fields). Essentially, an informal Entity-Relationship diagram.
    b. README.txt (plain text) file that explains what the script does.
    c. Static dataset files with the scraped data
    d. A Python script, **scraper.py,** with the code that scrapes one of your sources. The main script can call other python files or libraries you have developed. All the code must be submitted. The main script needs to be runnable from the command line with three different inputs:

1. **`scraper.py`**
   The script invocation without input arguments just prints (to standard output) the complete scraped dataset as rows of data.
2. **`scraper.py --scrape N`**
   This script invocation with the flag --scrape N prints (to standard output) the first N entries of the dataset.
3. **`scraper.py --save <path_to_dataset>`**
   This script invocation saves the complete scraped dataset into the file passed as input (<path_to_dataset> is just a placeholder for the path to the file). Sample invocations could be:
   ```
   scraper.py --save my_scraped_data.csv
   scraper.py --save dir1/dir2/football_stats.csv
   ```

Final submission (*due May 2)*

1. **Objectives:**
   a. Analyze/Draw conclusions from data.
   b. Explain how the whole combined data system work
   c. What facts the data tells you?
   d. Provide graphs, insights, analysis, etc.
   e. Describe maintainability/extensibility of project, e.g., changes to sources, further analysis, future work.

2. **Deliverables:** Submission should contain:
   a. Project description in pdf, at least 3 pages 11pt font, including:
      i. Motivation surrounding project topic
      ii. Brief description of data sources
      iii. Analysis performed
      iv. Conclusions drawn
   b. Files that contain the full datasets you used in your project. The files can be database files (e.g., sqllite3), or static dataset files like .csv, .json, .xlsx, etc.
   c. README file (.txt) explaining how code is structured, configuration, and how to run code from the command line.
   d. Python file/s (.py) that extracts the data from websites and from APIs, as well as analyzes your data. The Python code should be annotated with informative comments.

   Notes:
   - You can organize all scraping, web API code and analysis code in only one .py file which should be executed from the command line.
   - You could also split the code in multiple files and introduce import statements so that all the scraping and analysis occurs by running one script from the command line.
   - The main script can save files and pictures in the same directory. Make sure to explain what the expected output files are in the README.txt file.

You can use any publicly available source of data on the web.

Here are some sites that list available datasets/APIs if you need inspiration:

https://datasetsearch.research.google.com/

https://github.com/awesomedata/awesome-public-datasets

https://www.programmableweb.com/category/all/apis