DS2002 Data Project 1

50 Points

The Goal of this project is to demonstrate (1) an understanding of and (2) competence of implementing and using basic data science systems rooted in SQL and other data sources like flat files (CSV), Open Data and other relational and data sources as well as APIs and data transformation. For this project you will use GitHub to store and manage your code.

You may find a partner to do this or do it alone. If you do it in a group, please submit ONE Github link to both submissions and let me know in the Readme.md

This will be due on Oct 20th at 11:59 PM. Submit it to Git, copy the invite /link to me and Dylan.

ETL data processor

1. Deliverable: Author a segment of an ETL pipeline that will ingest or process raw data. You must also submit a URL to a GitHub repository for your solution. In python you'll need to know how to open files/call an API, iterate files, pattern match and output files.
2. Benchmarks:
   - i. Your data processor should be able to ingest a pre-defined data source and perform these operations:
     1. **Fetch / download / retrieve** a remote data file by URL, or ingest a local file mounted. Suggestions for remote data sources are listed at the end of this document.
     2. **Convert** the general format and data structure of the data source (from JSON to CSV, from CSV to JSON, from JSON into a SQL database table, etc. I want the option to convert any source to any target. So, if I get a CSV as an input, I want the user to choose an output)
     3. **Modify** the number of columns from the source to the destination, reducing or adding columns. If you add data cols you can put any other useful information in that column you wish.
     4. **Store** The converted (new) file should be written to disk (local file) or written to a SQL database.
     5. In your code, Generate a brief summary of the data file ingestion including:
        1. Number of records
        2. Number of columns
     6. In your code Generate a brief summary of the post processing including:
        1. Number of records
        2. Number of columns
   - ii. The processor should produce informative errors should it be unable to complete an operation.
   - iii. You must do this from CSV and JSON which can come from a file dump or an API Call (so this will have two data sources…) These two sources do NOT need to be merged into one.

iv. Submit a 1 pager on the your experience reflecting on the challenges, what was easier than you thought and what was harder. How would a utility like this be useful for other data projects you may encounter?

3. Grading:
   i. □ Successful build/Execution of the solution with no errors in syntax…etc. It can be a python notebook or python code. – 10 points
   ii. □ Functionality that meets all benchmarks – 30 points
   iii. □ Creativity / Innovation / Quality – 5 points
   iv. □ Documentation -- Describes how to use the data processor and the elements that make it operational. This can be done with comments in the code. – 5 points

Publicly-available datasets:

- https://www.kaggle.com/datasets
- https://data.world/
- https://www.data.gov/
- https://opendata.charlottesville.org/

You can Choose/find data from anywhere you like…these are just suggestions.

Publicly-available APIs:

- https://docs.github.com/en/rest
- HUGE LIST: https://github.com/public-apis/public-apis