Data Project 2

50 points
Due 12/11/2023

The goal of the second data project, building upon the first project, is to further demonstrate (1) an understanding of and (2) competence creating and implementing basic data science systems such as pipelines, scripts, data transformations, containers, APIs, databases and cloud services.

Submit your projects in your Git Repo

1. Data ingestion & analysis – individual project
2. Discord bot – individual or pairs

Data ingestion and analysis:

Deliverable: Write and deploy a process that executes exactly once every minute, retrieving data from a remote API (provided for you) and write all retrieved values to a database for 60 minutes. Using code- based data analysis techniques against the database, try to (i) describe any patterns or changes in the data over time; and (ii) explain the logic of these changes.
The remote data API can be found here:
https://4feaquhyai.execute-api.us-east-1.amazonaws.com/api/pi

Benchmarks:
1. Your solution must execute precisely once per minute at the same time each minute. (You should not use a sleep 60 command to simply "wait" between executions or other methods that will drift over time.) Therefore, your solution must be designed carefully.
2. Your solution must run for exactly one hour, starting at 00 minutes and finishing at 59 minutes. The API is available 24/7 for such testing.
3. Your solution will retrieve all data fields from an API and write them to the database of your choice and design – relational or NoSQL.
4. Submit all code in a standalone GitHub repository in your account.
5. Your analysis should look at the relationship between all data fields and their changes over time. In a brief statement, describe any changes or patterns you observe, and propose an explanation for them. Include this in your GitHub repository.

o Successful deployment – 10 points.
o Functionality that meets all benchmarks – 12 points.
o Documentation – Describe your process, code, deployment strategy – 3 points.

Discord Bot

Your bot should recognize a "help" message and reply with user instructions.
2. Your bot should provide at least three commands or functions.
3. Your bot should reply promptly with an intelligent response or an informative error message.

4. Your bot should integrate with at least one external data source that you can document and describe. This could be a database system or API.
5. Submit all code in a standalone GitHub repository in your account.

Grading:

- Successful build of the bot and API solution on PythonAnywhere or anything of your choice – 10 points
- Functionality that meets all benchmarks – 10 points
- Creativity / Innovation / Quality – 2 points
- Documentation – Describes how to use the bot and the elements that make it operational –
- 3 points