Assignment 1

Due: 3rd feb 23 03:59 pm

Case summary: You are working for a Geospatial startup and are tasked to build a data exploration tool that leverages publicly available data and makes it easier for data analysts to download data. The data sources are on the NOA website and you have chosen the NexRad and GOES satellite datasets for exploration.

Tasks:

- GEOS
    a. Explore and download selected datasets for the GOES satellite dataset
    b. Given a filename, construct the hyperlink of data location.
    c. Write Unit tests for all the use cases
    d. Test using the links from [3]
- NexRad
    a. Explore and download selected datasets for the NexRad dataset
    b. Given a filename, construct the hyperlink of data location.
    c. Write Unit tests for all the use cases
    d. Test using the links from [3]
- Use a python package of your choice and plot the NexRad locations from [4]

Grading:

1. Accomplishing all tasks for Geos 40%
2. Accomplishing all tasks for NexRad 40%
3. Plotting and visualization of the NexRad site locations

Part 1: Exploration of the GEOS dataset [1]

The data analyst has two use cases:

1. The analyst wants to be able to explore datasets using Streamlit and get the data sourced from the NOAA's GEOS dataset[1]. However, they need to track the dataset downloads and don't want the datasets to be downloaded directly from the GOES website. To build a prototype, the analysts have agreed to work on one channel[1].
   The workflow of the application would look something like this:
    - An analyst would first choose the station, year, day, hour from the streamlit interface. (Note, the metadata should be scraped from the GOES website and stored in SQLLite.. See technical details below)
    - Based on selection, the app will list all the files from the site. (Note, the listing should come realtime and shouldn't be from the SQL table).
    - Once the file is selected for download by the analyst, the file will be transferred from the GOES location to your AWS S3 bucket and a link to "your" S3 bucket would be made available to the user. You should also provide the link to the GOES site from where you did the transfer so users can compare the files.
2. THe analyst wants to be able to enter a file name and retrieve the file from the GOES datasource.
   For example:

OR_ABI-L1b-RadC-M6C01_G18_s20230020101172_e20230	Corresponds to	https://noaa-goes18.s3.amazonaws.com/ABI-L1b-RadC/2023/002/01/OR_ABI-L1b-RadC-

020103548_c20230020103594.nc		M6C01_G18_s20230020101172_e20230020103548_c20230020103594.nc

The analyst should be able to enter the file name and get back the hyperlink as shown in the example above.
Note: You should build a data format validator.
If the filename is invalid, you should throw an error back
If the filename is valid but no such file exists you should list that.

Common requirements

- You should log all the file downloads.
- Ensure you write unit tests to test the correct/incorrect use cases
- Use 5 greatexpectations for each use case (10 in total)
- Submit all code on Github
- Host the streamlit app on streamlit public

Part 2:
Repeat Part 1 for the NexRad data [2]

Part 3:

- Use your favorite plotting package to plot all the NexRad locations[4] on the US map.
- Integrate the plot on to the Streamlit app

Additional notes:

1. Required attestation and contribution declaration on the GitHub page:
   WE ATTEST THAT WE HAVEN'T USED ANY OTHER STUDENTS' WORK IN OUR ASSIGNMENT AND ABIDE BY THE POLICIES LISTED IN THE STUDENT HANDBOOK
   Contribution:
   - member1: 25%
   - member2: 25%
   - member3: 25%
   - member4: 25%
2. Make sure you do not push anything to your GitHub after submission date (Editing Readme.md is ok but no code pushing after deadline)
3. Create a Codelab document describing everything you did.In your GitHub you should have a readme.md files which would tell what all things are there in this GitHub repository.

References:

1. https://noaa-goes18.s3.amazonaws.com/index.html#ABI-L1b-RadC/
2. https://noaa-nexrad-level2.s3.amazonaws.com/index.html#2023/
3. https://docs.google.com/spreadsheets/d/1o1CLsm5OR0gH5GHbTsPWAEOGpdqqS49-P5e14ugK37Q/edit
4. https://en.wikipedia.org/wiki/NEXRAD

Technical notes (Ask the TA for clarifications)

- Collect metadata of the AWS bucket & Doppler Radar Sites

- Create a AWS s3 bucket which is has public read access
- Design a SQLite db with the metadata collected above
- Create and Deploy streamlit application
- Great Expectation report on the Class worksheet

1. Collect metadata of the AWS bucket

During this step you are expected to collect all the metadata for the bucket storage which would be related for user input search.
For the purpose of this assignment, only focus on ABI-L1b-RadC and get the meta data
https://noaa-goes18.s3.amazonaws.com/index.html#ABI-L1b-RadC/2022/ - has 154 entries for 2022 year
https://noaa-goes18.s3.amazonaws.com/index.html#ABI-L1b-RadC/2023/ - has 28 entries for 2023 year

SQLite db

A db should store the bucket metadata and Doppler Radar Sites with latitude and longitude coordinates for plotting a map.
Design schemas for both GOES-18 and NEXRAD with relevant tables. Additional, you can have a ER diagram explaining the database design you created.

References:
https://www.ncdc.noaa.gov/nexradinv/map.jsp

Streamlit UI

The streamlit application should primarily have two features for the users to download a file

- Enables to fetch file URL by filtering through the fields
- Enable to fetch file URL by the complete file name
- Map of Doppler Radar Sites

Two separate pages within one streamlit app for GOES-18 and NEXRAD.
https://blog.streamlit.io/introducing-multipage-apps/

Make use of rich user input elements rather than plain text input fields. Example Selectbox, Date input, Number input, etc.

For example, NEXRAD page could have the following

Input Components:

1. Year
2. Month
3. Day
4. NEXRAD Station Code
    a. State Selector
    b. List of all station within the selected state

These inputs can frame up the path: /<Year>/<Month>/<Day>/<NEXRAD Station/>/

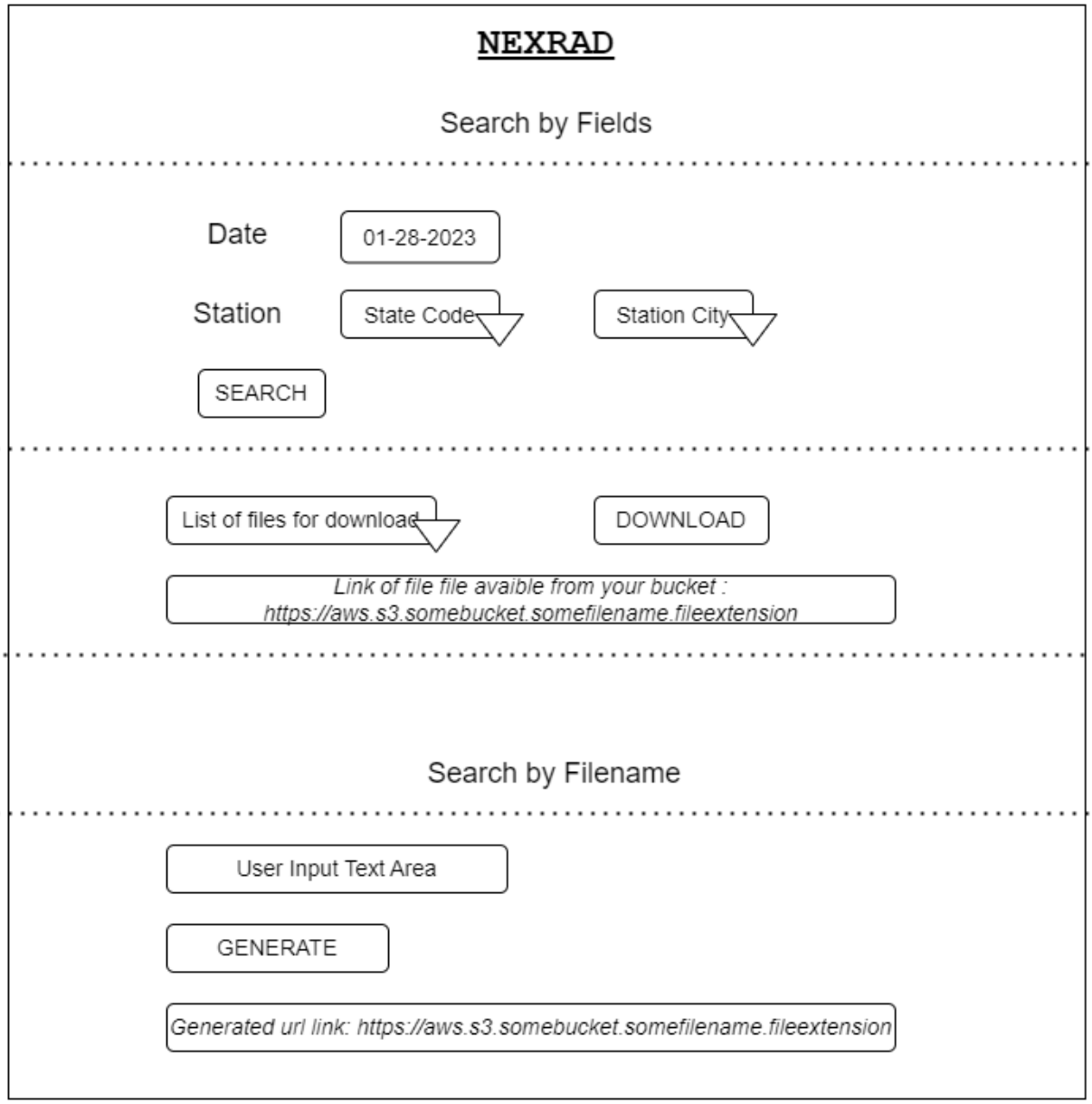
Workflow:
Search by Fields:

1. Users can provide inputs on the above four parameters
2. Based on the inputs, a search operation to be performed on nexrad aws buckets to retrieve list of files available
3. Users should have an option to select any one file
4. The selected file should be copied from nexrad bucket to you bucket (created in step 2). Note: Avoid duplicate file transfer if file already exist in you bucket.
5. Retrieve the URL of the file from your bucket and not the link of the nexrad bucket and displayed

Search by Filename:

1. User have the ability to input the filename
2. The app should display a link of the file available on nexrad bucket

Below is a prototype UI, Please DONOT consider this as a template that you have to follow. Come up with your own design.


NEXRAD
Search by Fields
Date
01-28-2023
Station
State Code
Station City
SEARCH
List of files for download
DOWNLOAD
Link of file file avaible from your bucket :
https://aws.s3.somebucket.somefilename.fileextension
Search by Filename
User Input Text Area
GENERATE
Generated url link: https://aws.s3.somebucket.somefilename.fileextension