Department Of Aerospace Engineering
Indian Institute Of Technology Madras

CS5830: Big Data Laboratory
Prof. Sudarsun

Lab Assignment 3:
Git + Version Control

Sanket Pramod Bhure (*AE20B108*)

March 30, 2024

Contents

1 Introduction

This assignment aims to set up a data pipeline to acquire public domain climatological data from the National Centers for Environmental Information (**NCEI**) website. The pipeline is divided into two main tasks: **DataFetch Pipeline** and **Analytics Pipeline**. It leverages **Apache Airflow** for task fetching and **Apache Beam** for data processing.
It further, utilizes **Git** for version control and tracks the changes made to the code, data, and configurations. Additionally, Data Version Control (**DVC**) is used to manage and version control the generated data. The repository structure consists of the following directories and files:

- `dags/`: Contains Apache Airflow DAGs for task scheduling.
- `data/`: Directory for storing generated data.
- `plots/`: Directory for storing generated visualizations.
- `requirements.txt`: List of Python dependencies.
- `docker-compose.yml`: Docker Compose file for containerized deployment.
- `dockerfile`: Dockerfile for building Docker images.

The assignment is available in the repository **Sanky18/CS5830-Big-Data-Laboratory-Assignment-3** on GitHub.

2 Building the repository and the data pipeline

1. **Clone the Repository:**
   Clone the repository from the remote repository to our local machine.
2. **Navigate to the Project Directory:**
   Change our current directory to the cloned repository.
3. **Initialize Git:**
   Initialize Git in the repository directory.
4. **Create a .gitignore File:**
   Create a `.gitignore` file to specify files and directories that should be ignored by Git. You can create this file manually or use tools like gitignore.io.
5. **Create a .dvc Directory:**
   Initialize Data Version Control (DVC) in the repository directory.
6. **Install Dependencies:**
   Install the required Python dependencies specified in the `requirements.txt` file.
7. **Start Apache Airflow:**
   If not already installed, set up Apache Airflow according to its documentation and start the Airflow webserver and scheduler.
8. **Start Docker Compose:**
   If using Docker Compose for containerized deployment, start the services defined in `docker-compose.yml`.

9. **Run DAGs:**
   Once Airflow is running, enable and trigger the DAGs from the Airflow UI. The DAGs should start running according to their schedules.

10. **DataFetch Pipeline (Task 1) Steps:**

    (a) **Fetch Data:**
    Fetch the HTML page containing location-wise datasets for a specific year from the NCEI website.

    (b) **Select Files:**
    Randomly select CSV files from the fetched HTML page.

    (c) **Fetch Files:**
    Download the selected CSV files.

    (d) **Zip Files:**
    Compress the downloaded CSV files into a ZIP archive.

    (e) **Move Zip File:**
    Move the ZIP archive to a specified location.

    (f) **DVC Add:**
    Once all the data is archived in the specified location inside the cloned repository, track it using DVC with the command `dvc add`.

11. **Analytics Pipeline (Task 2) Steps:**

    (a) **Wait for Archive:**
    Wait for the ZIP archive to be available at the specified location.

    (b) **Unzip Archive:**
    Unzip the contents of the archive into individual CSV files.

    (c) **Extract and Filter Data:**
    Extract required fields from CSV files and filter them based on string-only columns containing hourly values.

    (d) **Compute Monthly Averages:**
    Compute monthly averages of the required fields containing hourly values.

    (e) **Combine Data:**
    Combine data from different locations for at least one month files into a single DataFrame.

    (f) **Generate Geomaps:**
    Generate geospatial visualizations using the average values of combined data. Additionally, take the magnitude of negative average values into account, where larger average values result in bigger dots on the geomaps, while smaller magnitudes lead to negligible dot sizes.

12. Execute the above two DAG pipelines for the years 2023 and 2024, filtering those columns which have at least one month hourly data for required fields.

3 Results

The results of the Analytics Pipeline includes, Geospatial visualizations (geomaps) for different parameters. We have shown some sample geomaps for two different years 2023 and 2024, for month july and jan respectively.

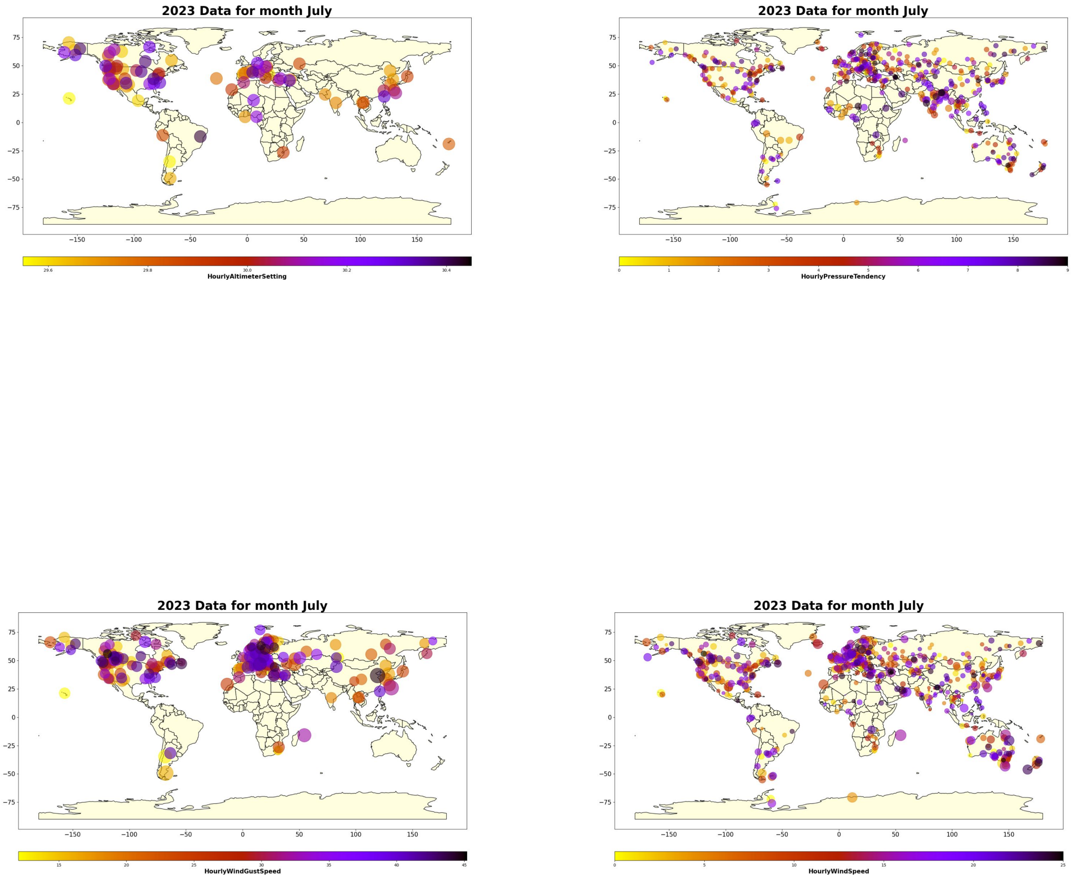

Figure 1: The series of geomaps depict climatological data for the month of July 2023, focusing on Hourly Altimeter Setting, Hourly Pressure Tendency, Hourly Wind Gust Speed, and Hourly Wind Speed. Each map provides a spatial representation of the respective parameter's distribution across the specified time frame.

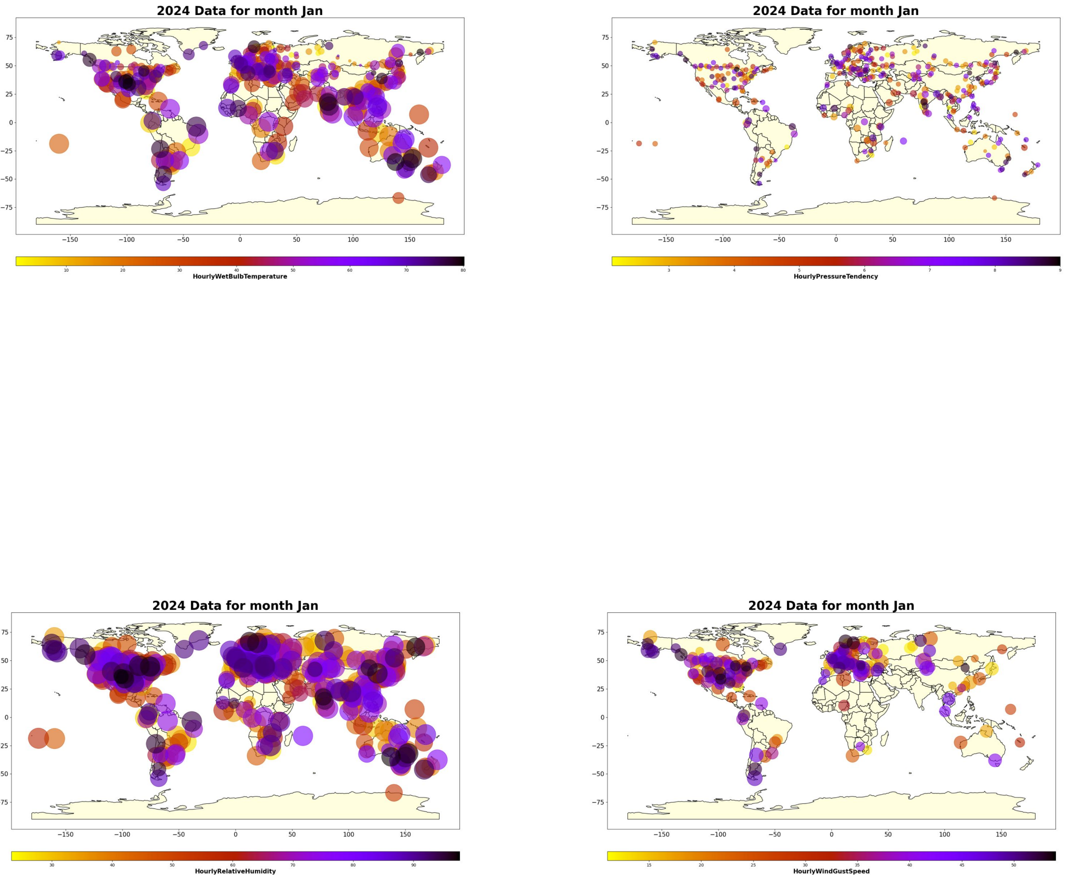

Figure 2: The series of geomaps depict climatological data for the month of Jan 2024, focusing on Hourly Wet Bulb Temperature, Hourly Pressure Tendency, Hourly Relative Humidity, and Hourly Wind Speed. Each map provides a spatial representation of the respective parameter's distribution across the specified time frame.