Weather Dataset Creation: A Comprehensive Guide

Rik Halder

October 25, 2024

Overview

This abstract of the document is to explain the creaion of a weather dataset for different stations accross India. The dataset contains **341,641 rows** of weather data from **39 stations** spread across India , covering various hourly weather parameters for the all the months of the year **2023**. This project demonstrates an automated pipeline for downloading, extracting, processing, and combining weather data from multiple sources.

Tools and Technologies Used

- **Python**: Used for data manipulation and processing of data using `pandas`.
- **Meteostat API**: It is the Source of hourly weather data for different weather stations.
- **PeaZip**: PeaZip is used for extracting compressed files (.gz files).
- **Wget**: Wget is used for downloading files from the internet.
- **7z**: It is a command-line utility for decompressing `.csv.gz` files.
- **JSON**: JSON is used to store and parse metadata of weather stations.

Dataset Structure

Each weather entry contains the following fields:

- **Station Name & Number**: Identification details of the weather station.
- **Date and Hour**: Timestamp of the recorded weather parameters.
- **Temperature (Temp)**: Temperature recorded in °C.
- **Dew Point**: Dew point temperature in °C.
- **Humidity**: Percentage of humidity.
- **Precipitation**: Rainfall measured in mm.
- **Snow Depth**: Depth of snow recorded (if any).
- **Wind Direction**: Direction of wind in degrees.
- **Wind Speed**: Speed of wind in km/h.
- **Peak Wind Gust**: Maximum recorded wind gust.
- **Air Pressure**: Pressure in hPa.
- **Sunshine Total**: Total sunshine in minutes per hour.
- **Weather Condition Code**: Coded representation of the weather condition.

Visualizing Weather Stations on India Map

The map below shows the geographic locations of the 39 weather stations across India used in this dataset. Each red point corresponds to a station's position, with labels indicating the station names. This visualization offers insight into the distribution of weather stations, ensuring broad geographical coverage.

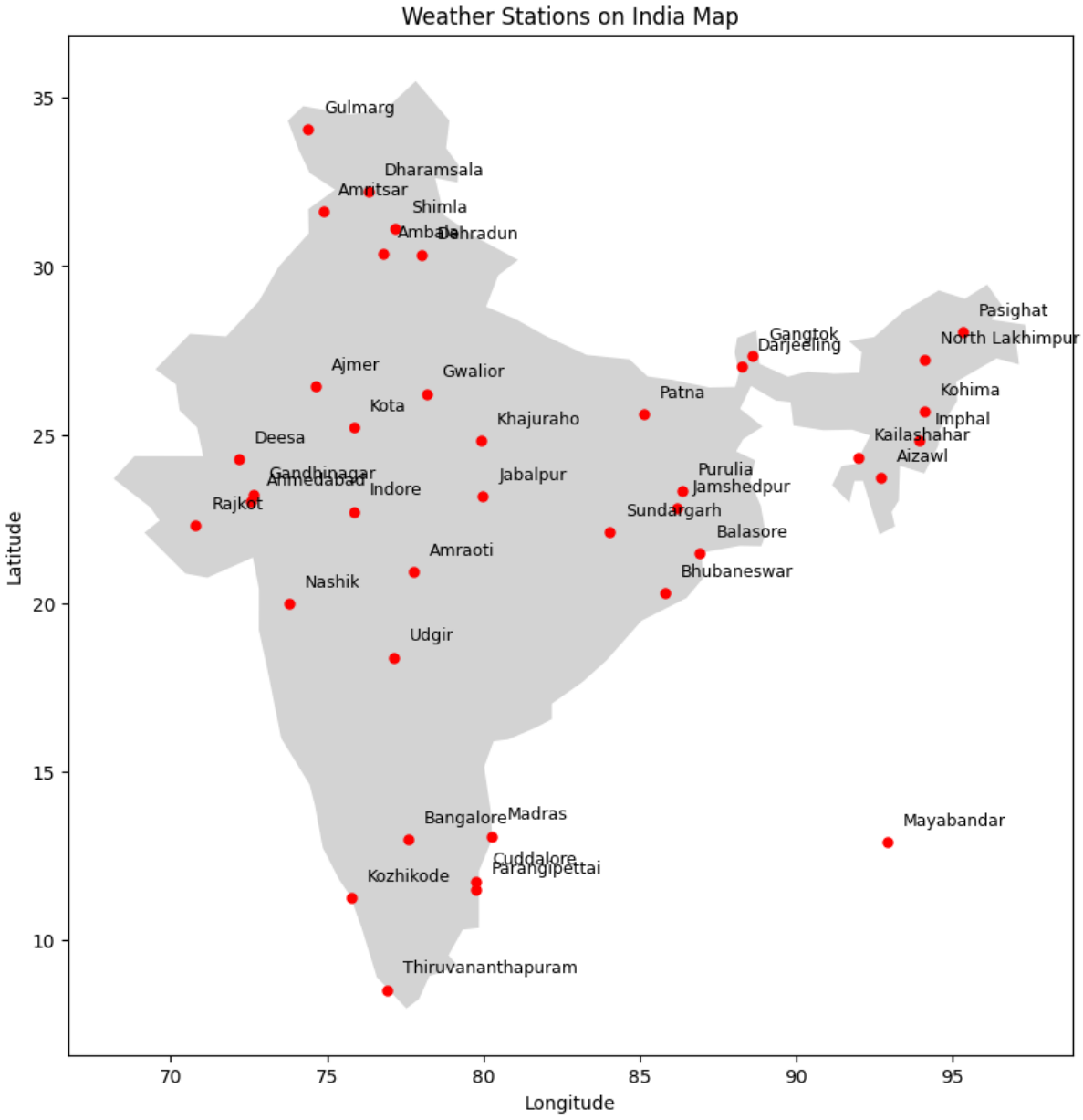

Figure 1: Weather Stations across India.

Step-by-Step Process

1. JSON Configuration File for Weather Stations

The first step was to define the weather stations using a **JSON file** containing metadata for each station. Each entry in the JSON includes:

- **WMO Identifier**: A unique code assigned to the station.
- **Station Name**: Name of the location where the station is situated.

2. Code for Downloading and Extracting Files

The script automates the process of downloading and decompressing hourly weather data from the Meteostat API. Below is the code snippet used:

```
def download_and_decompress(station_wmo, station_name):
    file_name_gz = f"2023-{station_wmo}-{station_name.lower()}.csv.gz"
    download_path = os.path.join(download_dir, file_name_gz)
    download_url = f"https://bulk.meteostat.net/v2/hourly/2023/{station_wmo}.
        csv.gz"

    try:
        print(f"Downloading {file_name_gz}...")
        subprocess.run(['wget', '-O', download_path, download_url], check=True)
    except subprocess.CalledProcessError:
        print(f"Failed to download {file_name_gz}")
        return

    try:
        print(f"Decompressing {file_name_gz}...")
        subprocess.run([seven_zip_path, 'e', download_path, f"-o{extract_dir}"
            ], check=True)
        os.remove(download_path)
    except subprocess.CalledProcessError:
        print(f"Failed to decompress {file_name_gz}")
```

Listing 1: Download and Extraction Script

3. Combining Individual CSV Files into a Single Dataset

Once all station data was extracted, the next step was to combine individual CSV files into a single dataset using the following code:

```
csv_dir = r'path_of_dir'
dfs = []
for csv_file in os.listdir(csv_dir):
    if csv_file.endswith('.csv'):
        file_path = os.path.join(csv_dir, csv_file)
        base_name = os.path.splitext(csv_file)[0]
        _, wmo_code, station_name = base_name.split('-')

        df = pd.read_csv(file_path)
        df['wmo_code'] = wmo_code
        df['station'] = station_name

        dfs.append(df)

combined_df = pd.concat(dfs, ignore_index=True)
combined_df.to_csv(output_file_path, index=False)
print(f"Combined CSV file saved at {output_file_path}")
```

Listing 2: Combining CSV Files into a Single Dataset

4. Dataset Statistics and Summary

- **Total Records**: 341,641 rows
- **Number of Stations**: 39
- **Parameters Recorded**: 14 (Temperature, Dew Point, Humidity, etc.)

Conclusion

This project effectively illustrates how to use data from several stations around India to create a large-scale meteorological dataset. To expedite data collection, extraction, and consolidation, the automated pipeline makes use of Python, the Meteostat API, and additional tools. Future weather analysis, forecasting, and research can be built upon this information.This project effectively illustrates how to use data from several stations around India to create a large-scale meteorological dataset. To expedite data collection, extraction, and consolidation, the automated pipeline makes use of Python, the Meteostat API, and additional tools. Future weather analysis, forecasting, and research can be built upon this information.