**Environmental Data Management with DVC**

**Table of Contents**

1. Overview
2. Features
3. Technologies Used
4. Getting Started
   * Prerequisites
   * Installation
   * Configuration
5. Data Collection
   * Air Quality Data
   * Weather Data
6. DVC Integration
   * Initializing DVC
   * Remote Storage Configuration
   * Data Versioning
7. Automation
   * PowerShell Script
   * Scheduling with Task Scheduler
8. Manual Usage
9. Acknowledgments

**1. Overview**

The **Environmental Data Management with DVC** project provides an efficient way to collect, version, and manage real-time environmental data streams using **Data Version Control (DVC)**. The system integrates live environmental data streams from reputable APIs, ensuring that air quality and weather metrics are consistently recorded, tracked, and available for analysis.

By leveraging **DVC**, it allows seamless data management, reproducibility, and integration with remote cloud storage services like Google Drive. The automation pipeline ensures regular data fetching without manual intervention, making it suitable for real-time monitoring and predictive analysis tasks.

**2. Features**

* **Real-Time Data Collection**: Automatically fetches current, historical, and forecasted air quality and weather data.
* **Data Version Control**: Tracks all changes to datasets with DVC for reproducibility.
* **Remote Storage**: Integrates Google Drive to store large datasets.
* **Automation**: PowerShell scripts automate the process of data fetching and updating.
* **User-Friendly Configuration**: Clear steps for setting up environment variables and dependencies.
* **Task Scheduling**: Scheduled scripts ensure regular updates using Windows Task Scheduler.

**3. Technologies Used**

|  |  |
| --- | --- |
| Technology | Purpose |
| Python | Core programming language for scripts. |
| DVC (Data Version Control) | Version control for managing data. |
| Git | Version control for codebase. |
| Google Drive | Remote storage integration for DVC. |
| PowerShell | Automation script execution. |
| Windows Task Scheduler | Automates recurring tasks. |
| Dotenv | Manages environment variables securely. |
| OpenWeatherMap API | Fetches air quality data. |
| Visual Crossing API | Fetches weather data. |

**4. Getting Started**

**4.1 Prerequisites**

Ensure the following tools and credentials are ready:

* **Python** (version 3.8 or above)
* **Git** for version control
* **DVC** installed on your system
* **Google Drive Account** for remote storage
* API Keys:
  + [OpenWeatherMap API Key](https://openweathermap.org/api/air-pollution)
  + [Visual Crossing Weather API Key](https://www.visualcrossing.com/weather-api)

**4.2 Installation**

Follow these steps to set up the project:

1. **Create a Virtual Environment**:

python -m venv .venv

1. **Activate the Virtual Environment**:
   * On Windows:

.\.venv\Scripts\Activate.ps1

* + On Unix/Linux:

source .venv/bin/activate

1. **Install Dependencies**:

pip install -r requirements.txt

**4.3 Configuration**

**1. Environment Variables** Create a .env file in the root directory and add the following:

OPENWEATHER\_API\_KEY=your\_openweather\_api\_key

VISUAL\_CROSSING\_API\_KEY=your\_visual\_crossing\_api\_key

LATITUDE=your\_latitude

LONGITUDE=your\_longitude

**2. Configure DVC Remote Storage**: Place your Google Drive dvc-key.json authentication file in the root directory and configure DVC:

dvc remote add -d gdrive\_remote gdrive://<folder\_id>

dvc remote modify gdrive\_remote gdrive\_use\_service\_account true

dvc remote modify gdrive\_remote gdrive\_service\_account\_json\_file\_path "path/to/dvc-key.json"

**5. Data Collection**

**5.1 Air Quality Data**

* **Script**: scripts/air\_collector.py
* **Description**: Fetches air quality data (current, forecasted, and historical) from the OpenWeatherMap API.

**5.2 Weather Data**

* **Script**: scripts/weather\_collector.py
* **Description**: Fetches weather data (current, forecasted, and historical) from the Visual Crossing Weather API.

**6. DVC Integration**

**6.1 Initializing DVC**

To initialize DVC in the repository:

dvc init

**6.2 Remote Storage Configuration**

Follow these steps:

1. Add a Google Drive remote:

dvc remote add -d gdrive\_remote gdrive://<folder\_id>

1. Modify settings for Google Drive service account:

dvc remote modify gdrive\_remote gdrive\_use\_service\_account true

dvc remote modify gdrive\_remote gdrive\_service\_account\_json\_file\_path "path/to/dvc-key.json"

dvc remote default gdrive\_remote

**6.3 Data Versioning**

Add DVC tracking to your scripts and outputs. For example:

**dvc.yaml**\*\* Configuration\*\*:

stages:

air\_quality:

cmd: python scripts/air\_collector.py

outs:

- data/air\_quality/current/air\_quality\_current.json

- data/air\_quality/forecast/air\_quality\_forecast.json

- data/air\_quality/historical/air\_quality\_historical.json

weather:

cmd: python scripts/weather\_collector.py

outs:

- data/weather/current/weather\_current.json

- data/weather/forecast/weather\_forecast.json

- data/weather/historical/weather\_historical.json

Commit all changes:

git add .

git commit -m "Add DVC data pipelines"

**7. Automation**

**7.1 PowerShell Script**

* **Script**: collect\_and\_push.ps1
* **Purpose**: Automates running the DVC pipeline, pushing updates to remote storage, and syncing changes with Git.

**Script Outline**:

# Run DVC Pipeline

python scripts/air\_collector.py

python scripts/weather\_collector.py

dvc repro

dvc push

git add .

git commit -m "Automated data update"

git push

**7.2 Task Scheduler**

To schedule automated updates:

1. Open Windows **Task Scheduler**.
2. Create a new task:
   * **Trigger**: Set the desired schedule (e.g., daily).
   * **Action**: Run the PowerShell script:

powershell.exe -ExecutionPolicy Bypass -File "path\to\collect\_and\_push.ps1" ```

**8. Manual Usage**

**Run Data Collection:**

python scripts/air\_collector.py

python scripts/weather\_collector.py

**Execute DVC Pipeline:**

dvc repro

**Push Data to Remote Storage:**

dvc push

git push

**Acknowledgments**

* **DVC Documentation**: <https://dvc.org/doc>
* **OpenWeatherMap API**: <https://openweathermap.org>
* **Visual Crossing Weather API**: <https://www.visualcrossing.com>