Data Version Control with Apache Airflow

Project Overview

This project implements an automated data pipeline that extracts data from BBC's homepage, processes the data, saves it in JSON format, and uploads it to Google Drive. We use Data Version Control (DVC) to manage versions of our datasets and Apache Airflow to schedule and automate the data processing workflows.

Setup Instructions

Prerequisites

- Python 3.x
- Git
- Apache Airflow
- DVC
- Access to Google Drive API

Installation

1. Clone the repository:
   ```
   git clone https://github.com/revolutionarybukhari/a2-mlops
   cd your-repository
   ```

2. Install the required Python libraries:
   ```
   pip install -r requirements.txt
   ```

3. Initialize DVC:
   ```
   ```

```
dvc init
dvc remote add -d myremote gdrive://14wkKMqVSnIND0IVz8g06_pwu8IGyPale
```

4. Start Apache Airflow:

```
airflow webserver -p 8080
airflow scheduler
```

5. Navigate to `localhost:8080` in your web browser to access the Airflow UI.

Configuration

Ensure you have set up your Google Drive credentials and saved them in your project directory. Follow the Google Drive API documentation to obtain `credentials.json` and configure `pydrive`.

Data Processing Steps

Data Extraction

- Source: Data is extracted from BBC's homepage.
- Data: Titles, descriptions, and links of articles.

Data Transformation

- Cleaning: Text data is converted to lowercase, punctuation is removed, and text is stripped of extra spaces.
- Formatting: JSON formatting is applied to ensure data is structured and ready for analysis.

Data Versioning

- DVC: We use DVC to track changes to our datasets. Each new version of the dataset is pushed to Google Drive for storage.

Workflow Automation with Apache Airflow

- DAG Setup: A DAG is defined to manage the workflow, which includes tasks for data extraction, transformation, and version control.
- Scheduling: The DAG is scheduled to run daily, ensuring the latest data is processed and versioned.

Challenges and Solutions

- Challenge 1: Initial difficulties with DVC authentication for Google Drive.
  - Solution: Configured `pydrive` correctly and ensured `credentials.json` was placed in the project directory.

- Challenge 2: Handling rate limits from the BBC website during data extraction.
  - Solution: Implemented retry logic with exponential backoff to manage request failures.

- Challenge 3: Ensuring the Airflow scheduler runs the tasks as expected.
  - Solution: Adjusted DAG parameters and monitored logs to troubleshoot task execution issues.

Conclusion

This project demonstrates the effective use of modern data engineering tools to automate the process of data extraction, processing, and version control. The implementation of DVC and Apache Airflow provides a robust solution for managing data workflows in a scalable and repeatable manner.