MLOps Assignment 02

Name: Aans Rehman Khan Roll Number: 20I-0865

Workflow Overview

This report provides a summary of the workflow implemented for Assignment II, focusing on the automation of data extraction, transformation, and storage using Apache Airflow, along with version control using DVC and integration with Google Drive. The workflow consists of the following steps:

1. **Data Extraction**:
    - Utilized Python scripts to extract data from dawn.com and BBC.com, capturing links, titles, and descriptions from the landing pages.
    - Implemented error handling to manage exceptions during the extraction process.
    - Output is two csv files named as bbc_scrapper.csv and dawn_scrapper.csv.
2. **Data Transformation**:
    - Pre-processed the extracted text data using Python scripts, applying cleaning and formatting techniques to prepare the text for analysis.
    - Ensured consistency and accuracy in the transformation process through careful inspection and validation.
    - Output of cleaning is two files named as: bbc_scrapper_cleaned.csv and dawn_scrapper_cleaned.csv.
    - I have merged both file (named as merged_data.csv) and moved to the directory named as data. (data/merged_data.csv).
3. **Data Storage and Version Control**:
    - Configured Google Drive as the storage destination for the processed data.
    - Implemented Data Version Control (DVC) to track versions of the data and manage dependencies.
    - Integrated DVC with GitHub to version metadata against each **dvc push**.
4. **Apache Airflow DAG Development**:
    - Installed the Apache Airflow on the WSL Ubuntu.

- Running Apache Airflow on that.
- Developed an Airflow DAG to automate the data pipeline, comprising tasks for extraction, transformation, and storage.
- Configured task dependencies and error handling mechanisms within the DAG to ensure smooth execution.

Challenges Encountered

During the implementation of the workflow, several challenges were encountered:

- **Data Scrapping from bbc.com:** bbc.com code seems to be too dynamic so that there was no clear indication that we gat the description of every news, but I tried my best and get as many descriptions as I can.
- **Installation of Apache Airflow:** I have tried to download the Apache Airflow on Windows but there were too many issues. I have tired to install airflow through docker and through the method mentioned on the site, but i was facing some problems related to database. Finally, I decided to move to Ubuntu and installed WSL Ubuntu there I setup my environment and install the airflow, and airflow was working right.
- **Configuring DVC with Google Drive**: Integrating DVC with Google Drive required careful configuration to ensure that data versions were accurately recorded and synchronized. Some troubleshooting was necessary to resolve authentication and authorization issues with Google Drive APIs.
- **Ensuring Consistency Across Environments**: As I was developing my application on Windows, and I was running the airflow on Ubuntu there were some issues related to setting up GitHub and DVC. So, there is a little bit of dependency issue that it is not configuring the DVC and GitHub.

GitHub Repository Maintenance

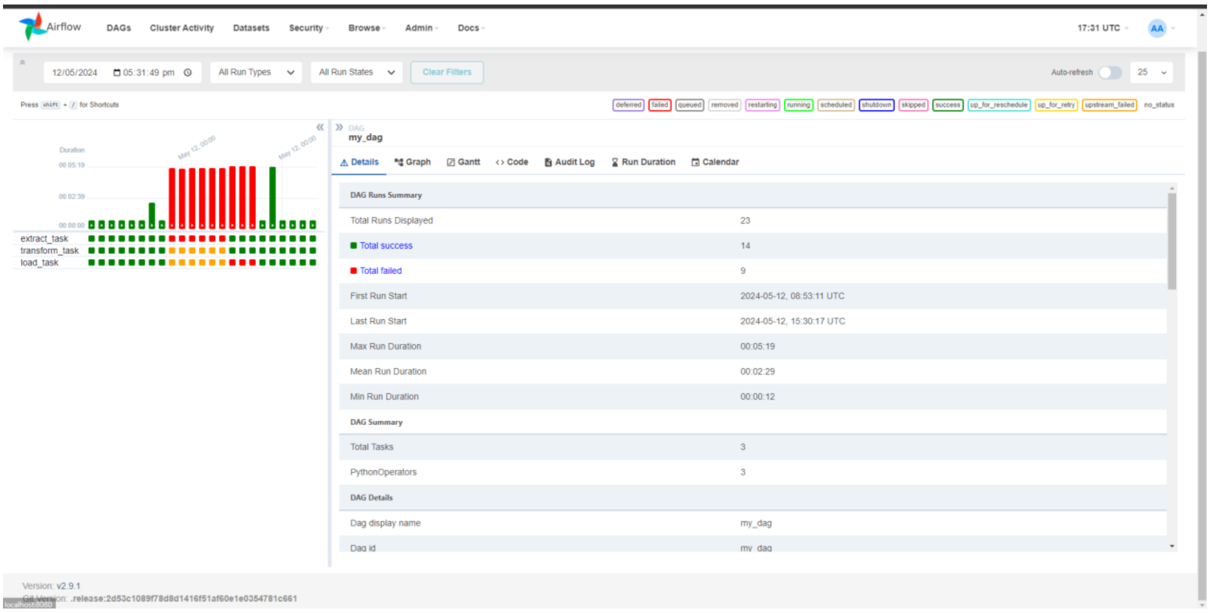
The GitHub repository for this assignment (https://github.com/AansRehman/MLOps_Assignment2.git) was structured to provide a clear overview of the project and its components. The repository contains the following files and directories:

- **README.md**: Detailed documentation of the assignment, including objectives, tasks, and workflow overview.
- **Data Extraction**: Includes two files named as **bbc_scrapper.py** and **dawn_scrapper.py.** These files contain the code to scrap the data from the sites (BBC.com and dawn.com).

- **Data Transformation**: Includes three files named as **cleaning_dawn_dataset.py** and **cleaning_bbc_dataset.py** and **combining_files.py**. These files contain the code to clean the data scrapped from the sites and merge both files two make a single dataset.
- **Airflow Dag**: Includes a file named as **my_dag.py**. In this file, I have written the code to specify the workflow of the entire application. It includes extracting data from sites, cleaning and merging of data, setting up the DVC, uploading the data to DVC and version controlling of the data.
- **DVC**: Directory containing DVC configuration files and metadata for version control.
- **Report**: File having a detailed documentation of the entire application.
- **requirements.txt**: File specifying the Python dependencies required for the project.

GitHub Repository Link:

https://github.com/AansRehman/MLOps_Assignment2.git



DAG Runs Summary	
Total Runs Displayed	23
🟩 Total success	14
🟥 Total failed	9
First Run Start	2024-05-12, 08:53:11 UTC
Last Run Start	2024-05-12, 15:30:17 UTC
Max Run Duration	00:05:19
Mean Run Duration	00:02:29
Min Run Duration	00:00:12
DAG Summary	
Total Tasks	3
PythonOperators	3
DAG Details	
Dag display name	my_dag
Dag id	my_dag

Version: v2.9.1
Git Version: .release:2d53c1089f78d8d1416f51af60e1e0354781c661

Conclusion

Through diligent planning, implementation, and testing, the workflow for Assignment II was successfully completed, achieving the objectives of automating data processing tasks and implementing version control. The integration of Apache Airflow, DVC, and Google Drive provides a robust framework for MLOps, enabling efficient and scalable data management and analysis workflows. Ongoing maintenance and monitoring of the GitHub repository will ensure that the project remains accessible and well-documented for future reference.