Roll Num: i19-0622

Name: Muhammad Shahmeer

Course: MLOps

Section: B

Assignment : 2

The primary purpose of this script is to demonstrate the implementation of MLOps practices within Apache Airflow. Specifically, it showcases the following:

1. Data Extraction: The script extracts data from two news websites, Dawn and BBC, including links, titles, and descriptions of articles displayed on their homepages. This is achieved using web scraping techniques with the BeautifulSoup library.
2. Data Transformation: Once the data is extracted, it undergoes transformation by converting it into JSON format. This transformation is necessary to prepare the data for further analysis or storage.
3. Data Storage and Version Control: The transformed data is then saved to a JSON file and version-controlled using DVC. This allows for tracking changes to the data over time.

Working:

1. Extraction Functions: The fetch_data_from_dawn() and fetch_data_from_bbc() functions are responsible for extracting data from the Dawn and BBC websites, respectively. These functions use the requests library to fetch the HTML content of the websites and then parse it using BeautifulSoup to extract the desired information.
2. Transformation Function: The convert_to_json() function combines the data extracted from both websites into a single dictionary and converts it into JSON format using the json library. This function prepares the data for storage and versioning.
3. Loading and Versioning Functions: The save_json_data() function saves the transformed data to a JSON file named combined_data.json. The version_and_push_data() function performs version control using DVC by adding, committing, and pushing changes to the data repository.

4. DAG Configuration: The DAG is configured with default arguments such as owner, start date, retries, and retry delay. Task dependencies are defined to ensure the sequential execution of tasks.

Execution Flow:

1. The DAG starts by executing the extract_dawn_task and extract_bbc_task tasks to extract data from the Dawn and BBC websites, respectively.
2. Once data extraction is complete, the transform_task combines the extracted data from both sources and converts it into JSON format.
3. The load_task then saves the transformed data to a JSON file named combined_data.json.
4. Finally, the version_task version-controls the data using DVC and pushes the changes to the data repository.

Challenges:

1. The first challenge was the installation of the Airflow setup. It took a lot of time because of the errors but in the end it was successfully installed using docker. I have also shared the docker compose file in the github repo.
2. Second challenge was combining the extracted data from the dawn and bbc sites. At first I tried to do it using the '+' operator but it was not valid for dictionaries in python. So I created an empty dictionary and used the update() method to merge the dictionaries extracted_data_dawn and extracted_data_bbc into extracted_data. Then, converted extracted_data to JSON format and pushed it back to XCom.

These were the major challenges that I faced during this assignment. All other tasks were implemented easily.