**Muhammad Kazim**

**20I-2310**

**SE**

**MLOPS – Assignment#2**

***Documentation***

**Data Preprocessing Steps:**

1. **Extracting Articles:**
   * Two functions **scrap\_bbc\_articles()** and **scrap\_dawn\_articles()** are defined to extract articles from BBC and Dawn websites, respectively.
   * Each function parses the HTML content of the respective website, finds article tags, and extracts relevant information like title, link, and description.
2. **Cleaning Data:**
   * The **clean\_data()** function removes any articles with empty descriptions.
3. **Saving to CSV:**
   * The **save\_to\_csv()** function saves the cleaned data to a CSV file named **articles.csv**.
4. **Pushing to DVC and Git:**
   * Two functions **dvcpush()** and **gitpush()** are defined to push the CSV file to DVC and Git repositories, respectively.
   * DVC is initialized, and the file is added and pushed to the specified remote storage.
   * Git is initialized, and the file is added, committed, and pushed to the specified remote repository.

**DVC Setup:**

* DVC is initialized using **dvc init**, and a remote storage named 'drive' is added using **dvc remote add**. I used google drive as remote storage.
* The CSV file is added to DVC using **dvc add** and pushed to the remote using **dvc push**.

**Setup Instructions:**

1. Install Apache Airflow and required Python libraries.
2. Clone the repository to your local machine.
3. Configure DVC and Git remotes for version control.
4. Update Airflow configuration to point to the DAG script.
5. Start the Airflow scheduler and web server.
6. Access the Airflow UI to monitor and manage the DAG execution.

**Usage:**

* Ensure proper setup and configuration before executing the Airflow DAG.
* Monitor DAG execution and logs in the Airflow UI.
* Use DVC and Git commands for versioning and collaboration.