Dataset Creation Report

Objective

The objective was to create a dataset of Instagram user IDs associated with popular food-related hashtags. The focus was on food bloggers, allowing for the analysis of engagement metrics and audience demographics.

1. Setup of Web Scraping Environment

- Tool Used: Selenium, a browser automation tool, was set up to dynamically extract data from Instagram.

2. Logging into Instagram

- Process:
    - Automated login was achieved by navigating to the Instagram login page, entering credentials, and performing the login action.
    - A delay was added to ensure the page fully loaded before proceeding to the next step.

3. Scraping User IDs from Hashtags

Hashtag Selection

- Relevant hashtags were selected, including:
    - #food
    - #foodblogger
    - #foodphotography

Steps:

1. Navigating to Hashtag Pages: The script navigated to the Instagram page of each hashtag.
2. Loading Posts: A scrolling mechanism was implemented to load more posts to capture a larger sample size.
3. Extracting User IDs:
    - User IDs were extracted from the post URLs.
    - A set was used to store these IDs to ensure uniqueness and avoid duplicates.

4. Saving User IDs to a CSV File

- The collected user IDs were saved to a CSV file.

- Format: The file contained a header row followed by each unique user ID.
- Purpose: The CSV format made it easier to manipulate and analyze the data in future processes.

Data Extraction Overview

This part involved extracting relevant user data from Instagram profiles based on usernames sourced from the previously generated CSV file.

Methodology

1. Setup and Initialization

- WebDriver Configuration: Selenium WebDriver was configured to interact with Instagram and perform automated tasks.
- User Login: Logged into an Instagram account to access user profiles.

2. Reading Usernames from CSV

- The script read usernames from the CSV file and populated them into a list for processing.

3. Profile Information Scraping

For each user, the script extracted:

- Number of Posts
- Followers Count
- Following Count
- User Bio: Cleaned to remove emojis and non-ASCII characters for better readability.

4. Post URLs Extraction

- The script scrolled through the user's profile to load and collect post URLs, ensuring no duplicates.

5. Detailed Post Analysis

For each post, the following information was collected:

- Likes: Number of likes the post received.
- Hashtags: A list of hashtags used in the post.
- Location: Any tagged geographical location.
- Post Date: Captured in ISO 8601 format for consistency.

- Comments: Extracted to gain insights into user engagement.

6. Data Compilation and JSON Export

- All collected data was structured in a dictionary format.
- The final dataset was serialized into a JSON file for easy sharing and further analysis.

Methodology for JSON to CSV Conversion

1. JSON Data Loading

- The process began with loading the JSON data from the file instagram_profiles_full_data.json using Python's json library.

2. Defining CSV Structure

- Column headers were defined based on the data fields, including:
    - username: Instagram handle.
    - posts_count: Total number of posts.
    - followers_count: Number of followers.
    - following_count: Accounts the user follows.
    - bio: User bio.
    - post_url: URL of the post.
    - likes: Likes on the post.
    - hashtags: Hashtags used.
    - location: Geographical location tagged.
    - post_date: Date of post creation.
    - comments: Comments on the post.
    - comments_count: Total number of comments.

3. Writing Data to CSV

- CSV File Initialization: A new CSV file (updateddata.csv) was created with UTF-8 encoding.
- Header Row: The defined column headers were written first.
- Data Entry:
    - For each user profile, details like username, posts count, followers count, following count, and bio were extracted.

    - For each post, information such as likes, hashtags, location, post date, comments, and comments count was gathered.
    - Comments were processed to indicate user interactions when relevant.
    - Each completed record was written as a row in the CSV file.

4. Completion

- After the data writing process was completed, a confirmation message indicated the successful conversion of JSON to CSV format.