AMAZON DATA SCRAPER

Github Link: https://github.com/VinitaChowkekar/Amazon-Data-Scrapper

The script is a web scraper designed to extract product details from Amazon India. It retrieves product information, saves it to a CSV file, and cleans the data to maintain a structured database. Below is a detailed explanation of each section:

1. Base URL and Headers

- BASE_URL: The search results page for a specific category on Amazon.

  Base_URL:
  https://www.amazon.in/s?bbn=21541572031&rh=n%3A976419031&dc&qid=1733993930&rnid=3576079031&ref=sr_nr_n_0

- HEADERS: A dictionary with headers to mimic a browser and avoid getting blocked by Amazon's anti-bot systems.

2. Function to Fetch Product Data

- **Purpose**: Scrape product details (title, price, original price, discount, rating, reviews, and brand) from a specific page.
- **Logic**:
  - The URL for each page is dynamically constructed using BASE_URL and the page number.
  - A GET request fetches the HTML content.
  - BeautifulSoup parses the HTML to find relevant data like product title, price, and reviews.
  - Data is processed and stored in a dictionary format.
- **Error Handling**:
  - Catches and prints exceptions during the processing of individual products to avoid breaking the loop.

3. Pagination and Scraping Multiple Pages

- **Pagination**: The script loops through the first 20 pages (adjustable by changing the range).
- **Break Condition**: Stops if 200 products are scraped.
- **Throttling**: A delay (time.sleep(2)) between page requests prevents overwhelming Amazon's server.

4. Saving to CSV

- The collected data is saved to a CSV file named amazon_products_database.csv.
- **Appending Mode**: If the file already exists, new data is added without overwriting. If the file doesn't exist, it creates a new one with a header.
- The header parameter ensures headers are added only when creating the file.

5. Cleaning the Database

- The clean_database function loads the CSV file, cleans missing and noisy data, and saves the cleaned data back to the file.
- **Cleaning Steps**:
  - Drops rows with missing values.
  - Ensures numeric columns (Price, Rating, etc.) are valid.
  - Filters for realistic ranges (e.g., ratings ≤ 5, positive prices).

How to Run the Scraper

1. Prerequisites

Install required Python libraries:
bash
Copy code

```
pip install requests beautifulsoup4 pandas
```

-

2. Running the Script

1. Copy the code into a Python file (e.g., amazon_scraper.py).

Run the script:
bash
Copy code

```
python amazon_scraper.py
```

2.
3. The script will:
   - Scrape data from the first 20 pages (or until 200 products are collected).
   - Save the data to amazon_products_database.csv.

3. Testing the Scraper

- **Verify Output**:

    - Check the amazon_products_database.csv file to confirm the data is saved correctly.
- **Inspect Logs**:
    - Look for any "Error while processing a product" messages to identify issues during scraping.

4. Testing Cleaning

- Run the clean_database function:
    - Open the CSV file and introduce some dummy missing values or unrealistic data.
    - Re-run the script to clean the database.
    - Check the cleaned file to verify corrections.

After running the scraper, here's how the results and their quality can be analyzed:

1. Data Coverage

- **Product Count**: The script is designed to collect up to 200 products. Verify if it successfully collected this number or stopped prematurely due to missing data or errors.
- **Fields Collected**: Check the amazon_products_database.csv file for columns like Title, Price, Original Price, Discount (%), Rating, Reviews, and Brand. All columns should be populated for most products, with some exceptions if data was missing on Amazon.

2. Data Quality

- **Title**: Review product titles for clarity and completeness. Ensure there are no placeholders like "Title Not Available" for a large portion of the data.
- **Price and Discount**: Validate price and discount calculations. A high percentage of missing or inaccurate prices may indicate scraping issues.
- **Ratings and Reviews**:
    - Check if ratings are in the 0–5 range.
    - Look for numerical inconsistencies in reviews (e.g., text instead of numbers).
- **Brand**: Ensure extracted brand names match the actual product brands.

3. Data Cleaning Effectiveness

- **Missing Values**:
    - Post-cleaning, the CSV file should not contain any rows with missing Price, Original Price, Rating, or Reviews.
    - If too many rows were removed during cleaning, it may indicate poor data quality or scraping errors.
- **Unrealistic Values**:
    - Products with prices ≤ 0 or ratings > 5 should have been filtered out.

    - Remaining rows should represent valid data.

4. Challenges Observed

- **Incomplete Data**: Amazon may not display all fields (e.g., Original Price or Discount) for certain products.
- **Anti-Bot Protection**: If the scraper fetched fewer products than expected or stopped prematurely, it might have been blocked or throttled.
- **Noise in Reviews or Ratings**: Non-standard formats in scraped data could require additional cleaning.

5. Final Output

After cleaning, the amazon_products_database.csv should have:

- A structured dataset with valid and relevant product details.
- All rows containing essential fields without missing or invalid values.
- Clean numeric fields (e.g., Price, Rating, and Discount (%)).

You can use this database in AI /ML algorithms taking appropriate atrributes as per your requirement.