M Mani Sumanth
ce21btech11052

Scraping Methodology

To gather product information from Amazon, I utilized the Beautiful Soup library, focusing on smartwatches and shoes.

Main Procedure

I established a main function in my script to oversee the scraping process. This function accepted parameters such as the Amazon page URL, the page response object, and a DataFrame for data storage.

Scraping Steps

Product Link Extraction:

- Initially, I parsed the HTML content of the Amazon page to extract the URLs of smartwatches and shoes.
- Individual Product Details: For each product link, I developed separate functions to scrape various details:
- Product Name: Extracted from the product page.
- Price: Obtained from the same product page.
- Description: Retrieved using Beautiful Soup.
- Reviews and Ratings: Scraped from user feedback.
- Image URL: Lastly, the product image URL was extracted.

Data Handling

After accumulating all relevant details for each product, I structured the information in a dictionary format. This dictionary was then transformed into a Pandas DataFrame for easy manipulation and analysis. The resultant dataset, organized in CSV format, is suitable for further analysis or integration into machine learning models like GPT-4.

Challenges

Encountered difficulties in handling dynamically loaded content and circumventing anti-scraping measures by Amazon. To tackle these issues, I relied on Beautiful Soup's flexibility in parsing HTML content and navigating intricate website structures. Additionally, I implemented strategies to manage dynamic content loading, ensuring accurate data capture.

Utilized Libraries

The primary libraries used in this project were:
Beautiful Soup: For parsing HTML content and navigating Amazon's DOM structure.
Requests: To fetch the HTML content of Amazon pages.
Pandas: Used for creating and manipulating DataFrames, facilitating efficient storage and handling of scraped data.

Ethical Considerations

To comply with Amazon's terms of service and legal requirements:
Implemented rate limiting mechanisms to prevent overloading Amazon servers.
Adhered to robots.txt directives, refraining from scraping disallowed pages explicitly.