**WEB SCRAPING IN PYTHON**

**Given task:** Scrape products from the URL of bags in Amazon with details about their URL, name, price, ratings and reviews and store all data into a csv file.

**My work flow:**

Web scraping refers to the process of automatically extracting data from websites. It involves writing code or using tools to access and parse the HTML or structured data of a web page, extracting the desired information, and storing it in a structured format for further analysis or use.

The process typically involves the following steps:

* Sending HTTP requests: Using programming libraries or tools, you send an HTTP request to a specific URL of a website you want to scrape.
* Retrieving the HTML: Upon receiving the request, the website's server responds by sending back an HTML document containing the webpage's content.
* Parsing the HTML: The HTML document is parsed to extract the relevant data using techniques like CSS selectors or XPath.
* Data extraction: You identify the specific elements or patterns in the HTML that contain the data you want to scrape and extract that information.
* Data storage: The extracted data is typically stored in a structured format such as CSV, JSON, or a database for further analysis or use.

Web scraping can be used for various purposes, such as market research, price monitoring, content aggregation, sentiment analysis, lead generation, and much more.

In my project, I used three Python libraries: requests library for making HTTP requests like get() that help retrieve data from the website, bs4 or BeautifulSoap library which is used to parse HTML documents and convert them into a structured tree-like representation and also to access the text and other data inside HTML elements or CSS selectors and lastly pandas to access the data inside the CSV file and perform manipulations on the data.

So, to start my project, I went to the given URL which was the search result of bags in Amazon.com. I opened a few of the products and inspected the source code using inspect element. Since I had to retrieve details like name, price, ratings etc. I looked into the code to see the class that was given to these items on the different pages and noted the classes that were of the same name. For eg. a-offset was the class given to the span tag that had the price of the bag as text, so I chose a-offset to be the class I would use for searching and retrieving the price.

After selecting the classes, I started coding for the first question. The code is available in the repository as Q1. Since I coded on Colab, I didn’t need to install any library as they are already available. First, I defined the template URL and the number of pages needed to be hit. Then I used requests.get(URL) method and checked if I was able to hit my target page and it was successful as it returned a 200 status code.

The next step was to use BS4 to get the links of all products in the URL. After that in a for loop I retrieved the text of all the product details using the class names that I had found initially and stored them in their respective arrays and printed them out.

In Q2, a few new items had to be retrieved and the data had to be put into a csv file. I defined the headings of the columns of the csv file and wrote it onto the csv file using the csv library of Python. Then, I performed the same operations as before to get the required data but for more number of pages. I wrote all the acquired data onto the csv file and printed a message saying that the csv file was created and data was written onto it. The code took a while to finish running and at the end, I got the csv file that was created in the files section in Colab. I opened it and saw that the data had been successfully retrieved and was written onto the CSV file.

As I have done a few projects on machine learning and data analysis, I decided to use the data in the CSV file and analyse it. I decided to sort the products based on the ratings field. Since the data in the rating field was like “4.3 out of 5”, I had to first trim it down to only the value needed, so I split the string and replaced the values in the rating fields with the float values that I obtained. Then I sorted it in descending order and using pandas, I was able to display the sorted data.

**Limitations of the program:**

* I was not able to access the ASIN values as I do not know how to access elements inside a nested HTML tag because the ASIN value did not contain any class of it’s own.
* Despite accessing the id feature-bullets, I was not able to get the product description and the manufacturer for some products.
* A lot of values under the “Number of Reviews” section is “More Like This”, which I assume is a placeholder of some sort as the webpage is dynamically loaded and might not contain the exact number at all times of request.