Lab 3 Report

Name: Abdul Haque

Roll No: 21I-1769

BS-DS-M

Submitted to: Mr. Saad Munir

Challenges And Ip Rotation

- **LOAD MORE/JavaScript Button Challenge:** The first challenge was to find a static website which doesn't have any generic or dynamic html which means a load more button and a button you have to click in order to go to the next page this was the hardest challenge and most of the news channels are having dynamic html where the button doesn't have any a tag (The link to the next page) but rather has some JavaScript in the background scripts which changes the page.
- **No Page Number Changer in News Link:** After the first challenge the next one was that the pages which didn't have any load more button or JavaScript attached, the next problem was that there were no page's number getting increased when we press the > button to go to next page and have to find the page number increaser as well.
- **Restrictions of Websites:** After going through the above challenges another challenge was that whenever I found a website which had page numbers written in link that website was mostly restricted and whenever getting the html of that page or link it would only show some attributes and tags and not most which means it was all generically written html which I only got formats of the pages and not the main html text or content I needed.

- **Getting Date from Ap News:** I was using Ap News website to scrape data on Lionel Messi but the problem was that the publication Date was in the form of **Updated [hour]:[minute] [AMPM] [time zone], [month Full] [day], [year]** this is only giving the format of what the date will be so we don't specifically get the date or text inside the element I'm looking for so the solution to this was:

Get All dates: all_dates:

soup.find_all('bsp-timestamp')

Get dates one by one in loop:

timestamp_element = all_dates[count_dates]

(count_dates are the index used)

Rest of the logic:

if timestamp_element:

Get the timestamp data we have in bsp-timestamp:

timestamp_data = timestamp_element.get('data-timestamp')

The data we get is in milliseconds so divide 1000 to get in seconds:

timestamp = int(timestamp_data) / 1000

Using the datetime library I get the publication date I want:

date = datetime.fromtimestamp(timestamp).strftime('%B %d, %Y')

The date will come from this **Updated [hour]:[minute] [AMPM] [time zone], [month Full] [day], [year]** to **February 18, 2024**

- **The solution to Challenge 2:** The solution is to get a website which has P or Page in the end which we can change to get pages: **https://apnews.com/search?q=lionel+messi&f2=00000188-f942-d221-a78c-f9570e360000&s=0&p=20**

  Like in this link we have p in the end which we can change p=n, n is the number of pages we will be looking at.

- **IP Rotation and Finding Free IP's:** After all this the most difficult challenge was to find free ip's as before I wrote a code which would get free ip's, scrape them from freeproxylist.net but that didn't work as it would take a lot of time to fetch these and then keep trying out ip's and work them with my news links that didn't work so I looked up on google and searched for a free proxy website that was **webshare.io** which gave free 10 ip's which they themselves check every 2 minutes if the ip's working or not and you get a username and password to your ip's.

For example:

Username: tctsdtrb

Password: ncht9e985h7x

Ip's:

38.154.227.167:5868:tctsdtrb:ncht9e985h7x

185.199.229.156:7492:tctsdtrb:ncht9e985h7x

185.199.228.220:7300:tctsdtrb:ncht9e985h7x

185.199.231.45:8382:tctsdtrb:ncht9e985h7x

188.74.210.207:6286:tctsdtrb:ncht9e985h7x

188.74.183.10:8279:tctsdtrb:ncht9e985h7x

188.74.210.21:6100:tctsdtrb:ncht9e985h7x

45.155.68.129:8133:tctsdtrb:ncht9e985h7x

154.95.36.199:6893:tctsdtrb:ncht9e985h7x

45.94.47.66:8110:tctsdtrb:ncht9e985h7x

Now for Ip rotation I used **cycle** function from library **itertools** Now by using cycle function I can cycle through the ip's I have the 10 ip's for that I first created an ip's pool

Get ip's Function:

```
def get_ips():
    ip_list = []
    with open('myproxylist.txt', 'r') as file:
```

```
        for line in file:
            proxy = line.strip()
            ip = proxy.split(':')[0]
            ip_list.append(ip)
    return ip_list
```

Create Pool of ip's:

```
ip_list = get_ips()
proxy_pool = cycle(ip_list)
proxy = next(proxy_pool)
```

Using next we can cycle through each ip in our ip's pool and rotate the ip's after that I first check if the ip works or not and if it does it should scrape the website otherwise recursively call the function again so that it runs again and uses another ip.

Conclusion: The challenges I faced were difficult to solve but after this I am sure I can scrape most of the static websites using beautiful soup only. The websites had many information inside and getting that information while going through the inside links was difficult but was solved in the end and links publication date everything needed was scraped and put into three csv's which I combined later and made a combined dataset.