NeurIPS(Neural Information Processing Systems) Conference

Web Scraping

Mayur Mankar

20169

Data Science & Engineering

**Abstract:**

**What is Web Scrapping?**

Web scraping extracts underlying HTML code and, with it, data stored in a database. The scraper

can then replicate entire website content elsewhere. Web crawling is the process of automatically

browsing the World Wide Web. It is a method of web indexing, which is used by search engines to

create a searchable index of the millions of documents on the World Wide Web. Web crawlers are also

used by companies to monitor the web for changes to their web pages, such as price changes or new

products.

Web scraping is used in a variety of digital businesses that rely on data harvesting. Legitimate use

cases include:

Search engine bots crawling a site, analyzing its content and then ranking it.

Price comparison sites deploying bots to auto-fetch prices and product descriptions for allied

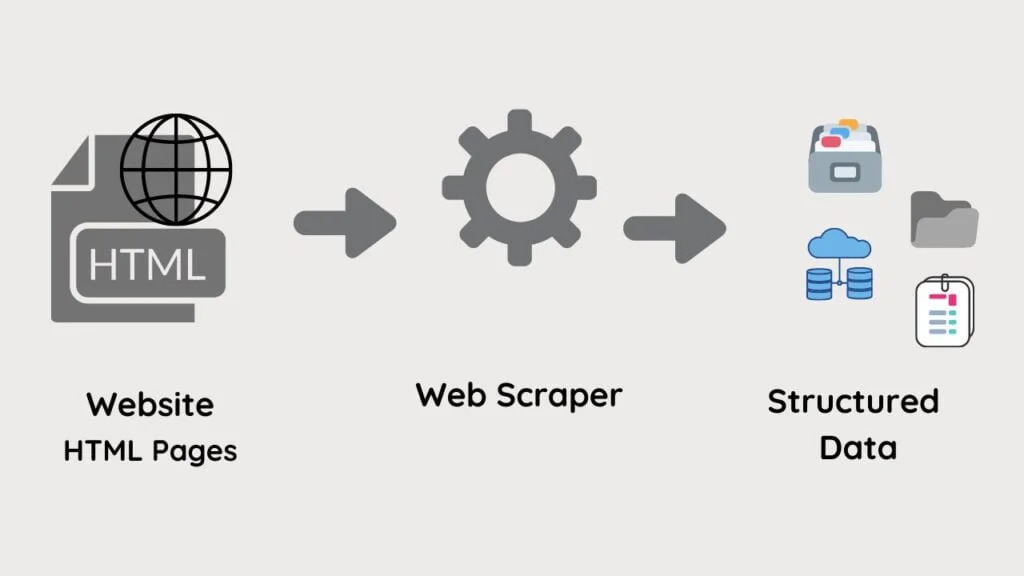
seller websites.Market research companies using scrapers to pull data from forums and social media (e.g., for

sentiment analysis).

Web scraping is also used for illegal purposes, including the undercutting of prices and the theft

of copyrighted content. An online entity targeted by a scraper can suffer severe financial losses,

especially if its a business strongly relying on competitive pricing models or deals in content

distribution.

**Fig 1.**

**Web Scraper**

Here we are scrapping the website <https://proceedings.neurips.cc/> to gather data about about all the previous proceedings of NeurIPS Conference to know the trends in current research in domain of natural information processing.

**Procedure:**

Step1: Setting up the Environment-

a) To use the empower of python to scrap websites, we don’t have to reinvent the wheel.

b)We can use existing libraries to get the job done!

c)We will install the following libraries using pip:

import requests

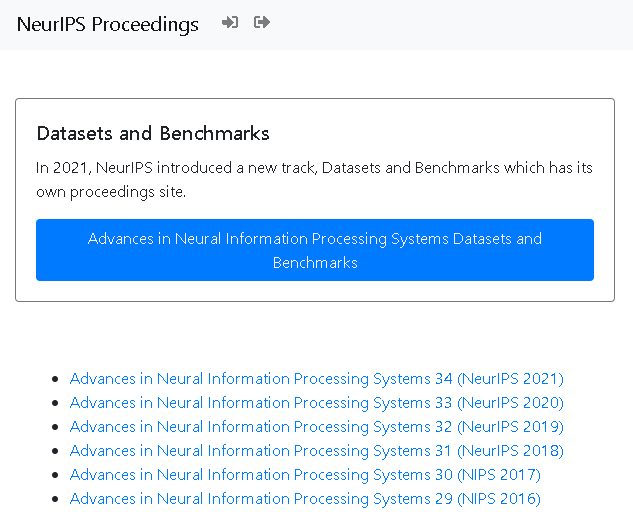
from bs4 import BeautifulSoup

import bs4

Step 2 : Get the HTML(fetching the HTML content)

response = requests.get("https://proceedings.neurips.cc/")

soup = bs4.BeautifulSoup(response.content, "lxml")



**Fig 2.**

**NeurIPS Proceedings Web Page**

Step 3 : I have stored paper links in csv file named “list\_of\_html.csv”.

list\_of\_html = []

for link in out\_url:

    response = requests.get(link)

    soup = BeautifulSoup(response.content, 'html5lib')

    links = soup.find\_all('a')

    for link in links:

      some\_text = link.get('href')

      if some\_text:

        if "html" in some\_text:

          list\_of\_html.append(link.get('href'))

# store the links in a csv file

df = pd.DataFrame(list\_of\_html,columns = ['PAPER LINK'])

df.to\_csv('list\_of\_html.csv', index = True, index\_label = 'ID')

Step 4 : I have stored paper names and author names in csv files named “list\_of\_papers.csv” and “list\_of\_authors.csv” respectively.

list\_of\_papers = []

list\_of\_authors = []

r1 = re.compile('html">(.\*?)</a>')

r2 = re.compile('<i>(.\*?)</i>')

for i in paper\_list:

  m1 = r1.search(i)

  m2 = r2.search(i)

  if m1:

    list\_of\_papers.append(m1.group(1))

  if m2:

    list\_of\_authors.append(m2.group(1))

df2 = pd.DataFrame (list\_of\_papers, columns = ['PAPER NAME'])

df2.to\_csv('list\_of\_papers.csv', index = True, index\_label = 'ID')

df3 = pd.DataFrame (list\_of\_authors, columns = ['AUTHORS NAME'])

df3.to\_csv('list\_of\_authors.csv', index = True, index\_label = 'ID')

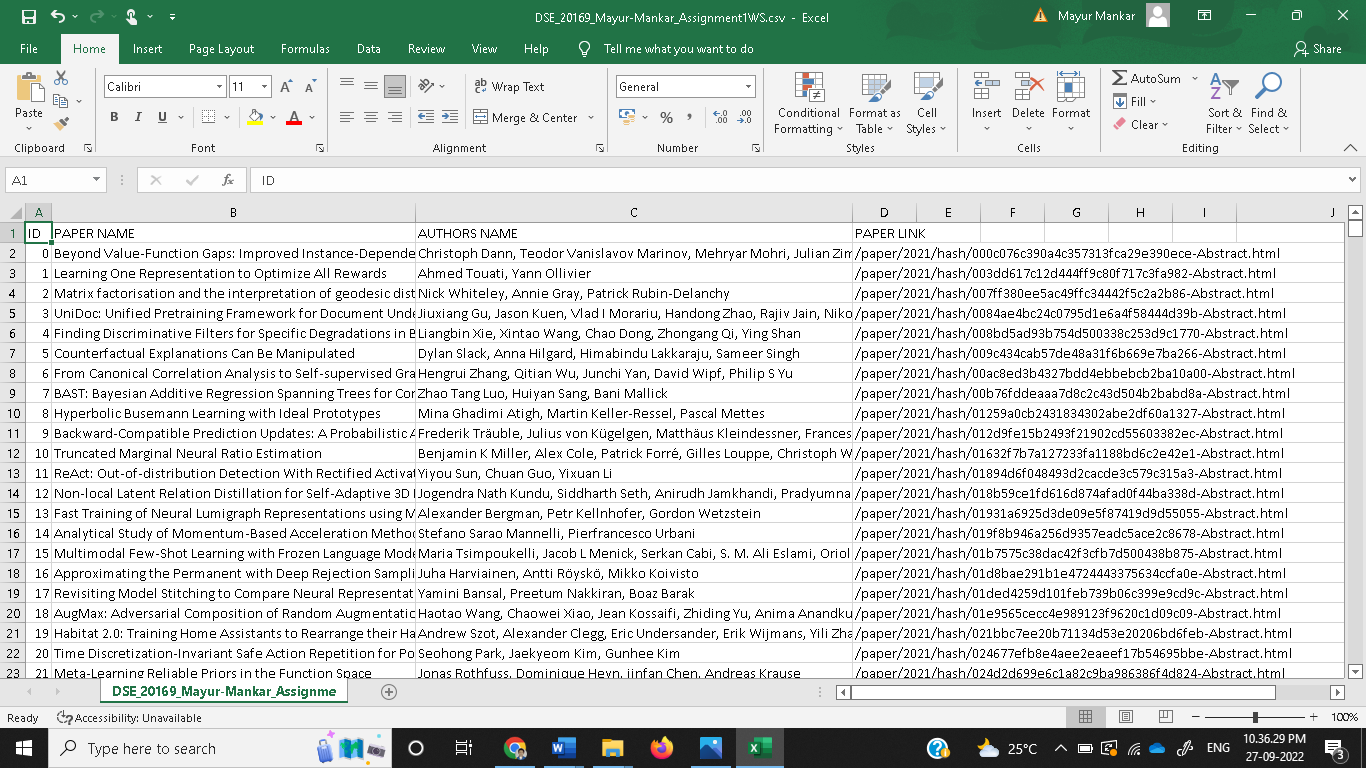
data2\_import = pd.read\_csv('list\_of\_papers.csv')

data3\_import = pd.read\_csv('list\_of\_authors.csv')

data\_merge1 = pd.merge(data2\_import,data3\_import,on = "ID",how = "outer")

data\_merge1.to\_csv('list\_of\_paper\_and\_author\_name.csv', index = False)

Step 5 : Finally, I have stored the paper name, author name and paper link in a csv file named “DSE\_20169\_Mayur-Mankar\_Assignment1WS.csv”.

**Fig 4.**

**Final data**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*