Introduction to Data Analytics

AIGC5000
Lecture10-Web Scraping

Instructor: Parisa Pouladzadeh
Email: parisa.pouladzadeh@humber.ca

HUMBER

What is Web Scraping ?

- Web scraping is the process of using bots to extract content and data from a website.
- web scraping extracts underlying HTML code and, with it, data stored in a database.
- It is a technique to fetch data and information from websites.
- Everything you see on a webpage can be scraped.

Use of Web Scraping

Companies can use this data to fix the optimal pricing for their products so that they can obtain maximum revenue.

Preparing dataset for your ML model.

Web scraping news sites can provide detailed reports on the current news to a company.

Companies can also use Web scraping for email marketing.

Use of Web Scraping

1. Marketing: Web scraping is used by many companies to collect information about their products or services from various social media websites to get a general public sentiment. Also, they extract email ids from various websites and then send bulk promotional emails to the owners of these email ids.

Use of Web Scraping

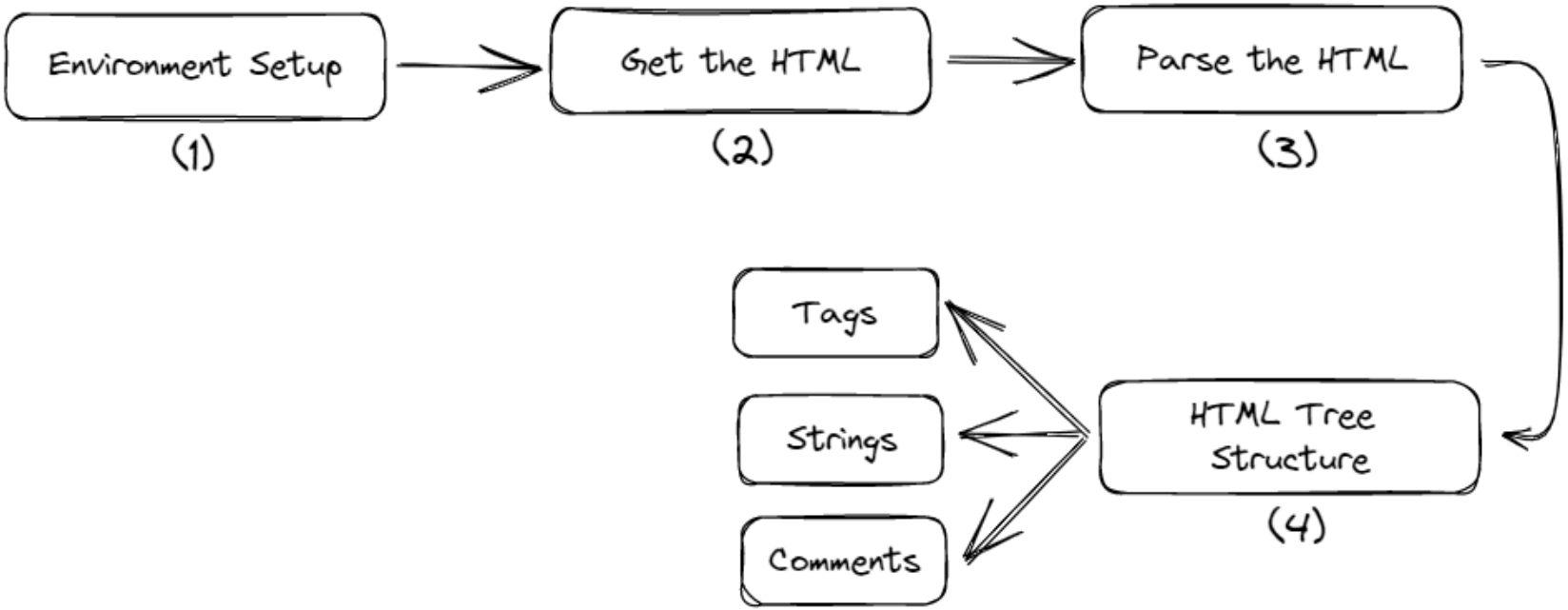
2. Content Creation: Web scraping can gather information from multiple sources like news articles, research reports, and blog posts. It helps the creator to create quality and trending content.

3. Price Comparison: Web scraping can be used to extract the prices of a particular product across multiple e-commerce websites to give a fair price comparison for the user. It also helps companies fix the optimal pricing of their products to compete with their competitors.

Use of Web Scraping

4. Job Postings: Web Scraping can also be used to collect data on various job openings across multiple job portals so that this information can help many job seekers and recruiters.

Code Implementation



Workflow and Libraries

Libraries: Requests and BeautifulSoup

The Requests library :

- is used to make HTTP requests in Python. It is an easy-to-use library with a lot of features. We will be using it to get the content from the URL of the webpage that we wish to scrap
- Send Request and Load the webpage.
  (Requests, urllib, httplib)

BeautifulSoup:

- is a library that is used to parse HTML documents. We create a new BeautifulSoup object by passing the HTML content and the type of parser we want to use

Setting Up the Environment

1.Installing the Libraries

$ pip install requests
$ pip install bs4

```
from bs4 import BeautifulSoup
import requests
import pandas as pd
```

Setting Up the Environment

The requests module extracts the HTML content from a URL. It extracts all the data in a raw format as a string that needs further processing.

The bs4 is the Beautiful Soup module. It will parse the raw HTML content obtained from the `request` module in a well-structured format.

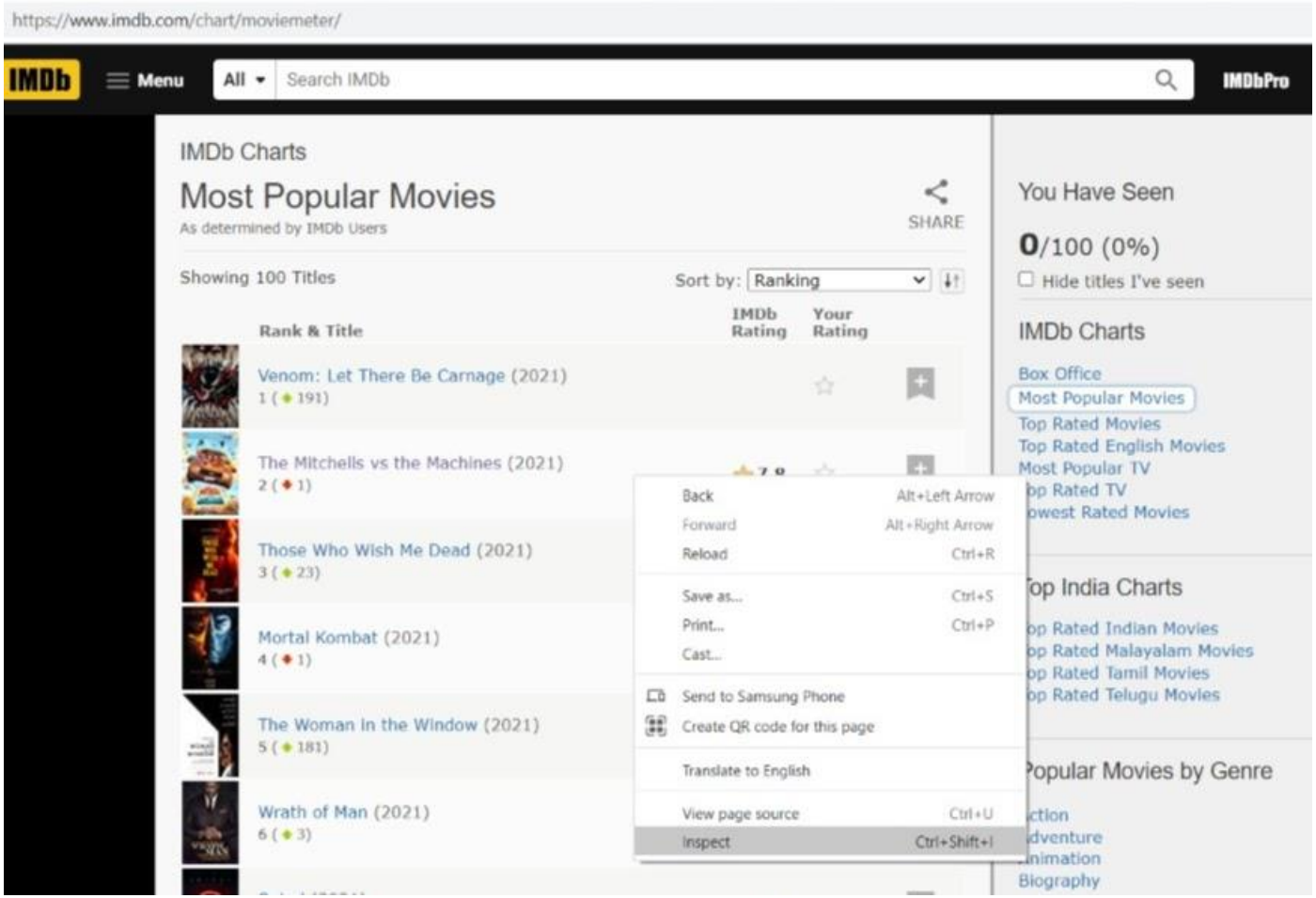
Implementation

2. Understanding the Webpage structure

The most important step in data scraping is to understand the structure of the webpage and the structure of the data we want to scrap

In order to see the HTML code for a website, do a “Right-Click” and click on “Inspect Source”

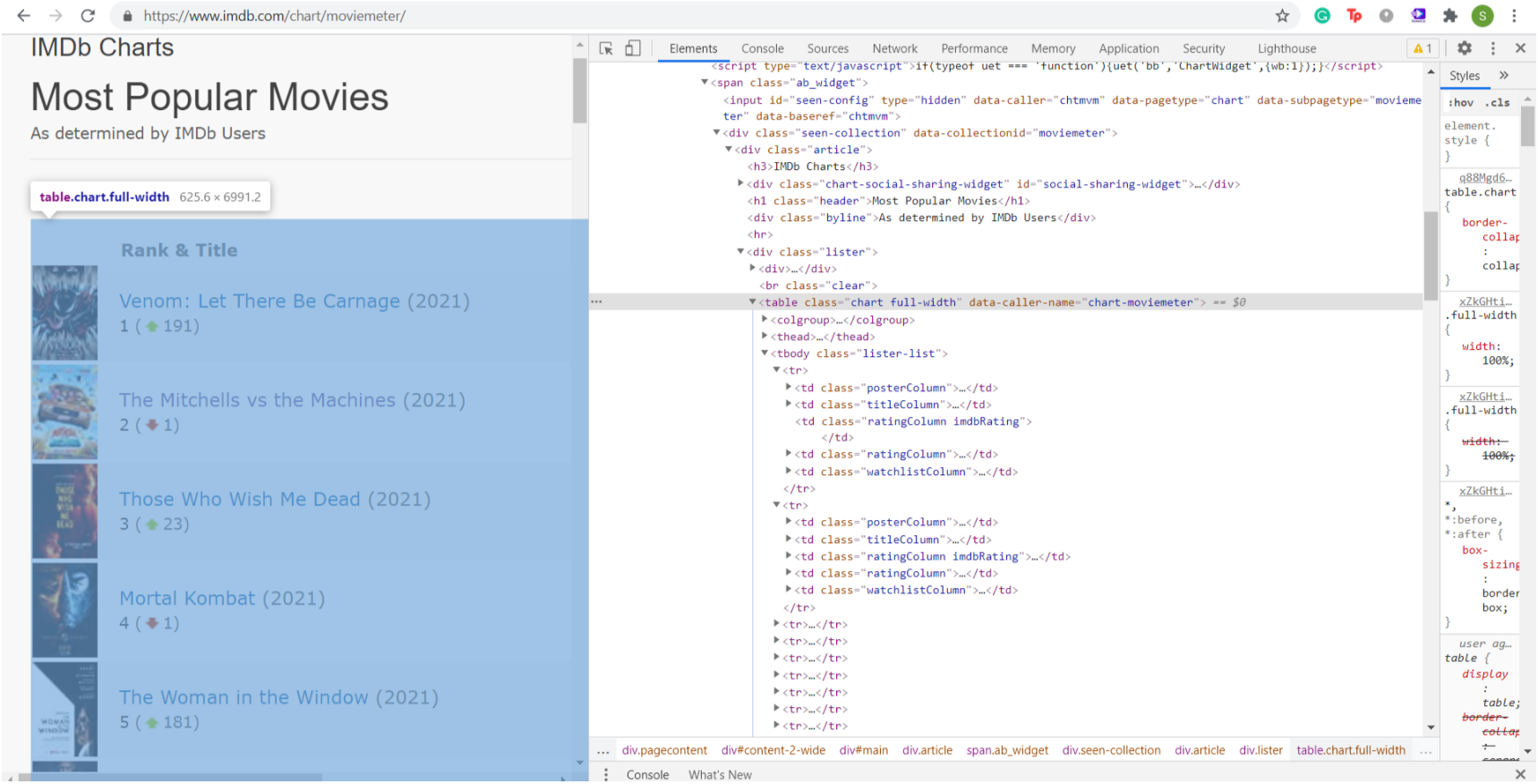
Example


https://www.imdb.com/chart/moviemeter/
IMDb
Menu
All
Search IMDb
IMDbPro
IMDb Charts
Most Popular Movies
As determined by IMDb Users
SHARE
Showing 100 Titles
Sort by: Ranking
Rank & Title
IMDb Rating
Your Rating
Venom: Let There Be Carnage (2021)
1 (191)
The Mitchells vs the Machines (2021)
2 (1)
Those Who Wish Me Dead (2021)
3 (23)
Mortal Kombat (2021)
4 (1)
The Woman in the Window (2021)
5 (181)
Wrath of Man (2021)
6 (3)
Back Alt+Left Arrow
Forward Alt+Right Arrow
Reload Ctrl+R
Save as... Ctrl+S
Print... Ctrl+P
Cast...
Send to Samsung Phone
Create QR code for this page
Translate to English
View page source Ctrl+U
Inspect Ctrl+Shift+I
You Have Seen
0/100 (0%)
Hide titles I've seen
IMDb Charts
Box Office
Most Popular Movies
Top Rated Movies
Top Rated English Movies
Most Popular TV
Top India Charts
Popular Movies by Genre
Biography

Example

- We are going to scrap the most popular movies from IMDb.
- This shows us the HTML code for the web page. Now we wish to scrap the following.
- 1.The rank of the movie
- 2.Name of the movie
- 3.IMDb rating of the movie
- Ongoing through the HTML code, we see that this information is stored in the form of a table using the <table> tag

HTML Example



HTML structure

3. Understanding HTML structure and tags

All the data on a webpage is stored inside different tags based on the content and presentation. The various important tags are :

1. tag: It is used to define a section of the webpage. For example, you can observe that the table which we want to scrap is present inside a tag with class =” lister” and this is further inside a tag with class = “article”

HTML structure

2. <a> tag: This is used to create a hyperlink to another webpage. It consists of the “href” attribute which specifies the URL of the webpage it links to. For example, when you click on a word and it directs you to another webpage on Wikipedia, it is because the word is present inside a tag along with the href attribute containing the URL of the web page you are directed to

3. tag: This is used to create unordered lists on the webpage. Each entry of the list is marked with a <li> tag

4. tag: This is used to create ordered lists on the webpage. Similar to the <ul> tag, each entry is marked with a <li> tag

HTML structure

5.<table> tag: This is used to create a table on the webpage. For example, we can see on the IMDb page, the information we require is present in a tabular form.

6.<tr> tag: This is used to specify the different rows under the table. For example, the different movies on the IMDb page are present in different rows and hence, in different <tr> tags

7.<td> tag: This is used to specify the content for each column for a given row in a table. For example, the content for the different columns (“Rank and Title”, “IMDb Rating”, “My Rating”) for each row is specified under the <td> tag

Implementation

4. Creating a BeautifulSoup object

```
url="https://www.imdb.com/chart/moviemeter/"
res=requests.get(url)
soup=BeautifulSoup(res.text,"html.parser")
print(soup)
```

Example

This creates an object named “soup” which has the HTML code for the URL given and can be used to select certain sections of the data

```
url="https://www.imdb.com/chart/moviemeter/"
res=requests.get(url)
soup=BeautifulSoup(res.text,"html.parser")
print(soup)
```

```
</thead>
<tbody class="lister-list">
<tr>
<td class="posterColumn">
<span data-value="1" name="rk"></span>
<span data-value="0.0" name="ir"></span>
<span data-value="1.631664E12" name="us"></span>
<span data-value="0" name="nv"></span>
<span data-value="-11.0" name="ur"></span>
<a href="/title/tt7097896/"> <img alt="Venom: Let There Be Carnage" height="67" src="https://m.media-amazon.com/images/M/MV5BYzljNzQ1MzMtODI5NS00MDRlLTgzYmQtNjE1NDk4MTIxODI0XkEyXkFq
</a> </td>
<td class="titleColumn">
<a href="/title/tt7097896/" title="Andy Serkis (dir.), Tom Hardy, Michelle Williams">Venom: Let There Be Carnage</a>
<span class="secondaryInfo">(2021)</span>
<div class="velocity">1
<span class="secondaryInfo">(
<span class="global-sprite titlemeter up"></span>
191)</span>
</div>
</td>
<td class="ratingColumn imdbRating">
</td>
<td class="ratingColumn">
<div class="seen-widget seen-widget-tt7097896 pending" data-titleid="tt7097896">
<div class="boundary">
<div class="popover">
<span class="delete"> </span><ol><li>1<li>2<li>3<li>4<li>5<li>6<li>7<li>8<li>9<li>10</li></li></li></li></li></li></li></li></li></ol> </div>
</div>
<div class="inline">
<div class="pending"></div>
<div class="unseeable">NOT YET RELEASED</div>
<div class="unseen"> </div>
```

Implementation

5. Extracting the data using BeautifulSoup

```
▼<table class="chart full-width" data-caller-name="chart-moviemeter"> == $0
  ▶<colgroup>…</colgroup>
  ▶<thead>…</thead>
  ▼<tbody class="lister-list">
    ▼<tr>
      ▶<td class="posterColumn">…</td>
      ▶<td class="titleColumn">…</td>
       <td class="ratingColumn imdbRating">
          </td>
      ▶<td class="ratingColumn">…</td>
      ▶<td class="watchlistColumn">…</td>
     </tr>
    ▼<tr>
      ▶<td class="posterColumn">…</td>
      ▶<td class="titleColumn">…</td>
      ▶<td class="ratingColumn imdbRating">…</td>
      ▶<td class="ratingColumn">…</td>
      ▶<td class="watchlistColumn">…</td>
     </tr>
    ▶<tr>…</tr>
    ▶<tr>…</tr>
    ▶<tr>…</tr>
    ▶<tr>…</tr>
    ▶<tr>…</tr>
    ▶<tr>…</tr>
    ▶<tr>…</tr>
    ▶<tr>…</tr>
    ▶<tr>…</tr>
    ▶<tr>…</tr>
    ▶<tr>…</tr>
    ▶<tr>…</tr>
    ▶<tr>…</tr>
```

The information regarding the movies is stored inside a <table> tag with the class attribute as “chart full-width”

So we will use the “find_all” function to find all such tables in the soup object

```
for i in soup.find_all("table",class_="chart full-width"):
  print(i)
```

Within the table, the data for each movie is stored in a <tr> tag so we will loop through these movies and store their data in a dataframe

The <td> tag containing the name and rank of the title has the class = “titleColumn”. The name of the movie is stored in a tag since it connects it to the webpage of that particular movie while the rank is stored inside a tag with the class attribute as “velocity”

Also, the <td> tag containing the IMDb rating of the movie has the class attribute as “ratingColumn IMDb rating”

```
▼<tr>
  ▶<td class="posterColumn">…</td>
  ▼<td class="titleColumn">
     <a href="/title/tt7097896/?pf_rd_m=A2FGELUUNOQJNL&pf_rd_p=ea4e08e1-c8a3-47b5-ac3a…QCPHR5C&p·
     Carnage</a>
     <span class="secondaryInfo">(2021)</span>
    ▼<div class="velocity">
       "1
       "
    ▶<span class="secondaryInfo">…</span>
     </div>
   </td>
   <td class="ratingColumn imdbRating">
       </td>
  ▶<td class="ratingColumn">…</td>
  ▶<td class="watchlistColumn">…</td>
```

We can extract the text within a tag using the following way:

```
for i in soup.find_all("table",class_="chart full-width"):
  for j in i.find_all("tr"):
    for k in j.find_all("td",class_="titleColumn"):
      for l in k.find_all("a"):
        print("Name of the movie is ",l.text)
```

```
Name of the movie is  Venom: Let There Be Carnage
Name of the movie is  The Mitchells vs the Machines
Name of the movie is  Those Who Wish Me Dead
Name of the movie is  Mortal Kombat
Name of the movie is  The Woman in the Window
Name of the movie is  Wrath of Man
Name of the movie is  Spiral
Name of the movie is  Oxygen
Name of the movie is  Tom Clancy's Without Remorse
Name of the movie is  Tenet
Name of the movie is  Things Heard & Seen
Name of the movie is  Army of the Dead
Name of the movie is  Nomadland
Name of the movie is  Nobody
Name of the movie is  The Green Knight
Name of the movie is  Radhe
Name of the movie is  A Quiet Place Part II
Name of the movie is  Promising Young Woman
```