Forward School

Program Code: J620-002-4:2020

Program Name: FRONT-END SOFTWARE DEVELOPMENT

Title : Case Study - IMDB Web Scraping

Name: Chuay Xiang Ze

IC Number: 021224070255

Date : 6/7/2023

Introduction : Learning how to scrape data from IMDB website.

Conclusion : Managed to complete tasks relating to the topic.

Reference : [https://medium.com/better-programming/the-only-step-by-step-guide-youll-need-to-build-a-web-scraper-with-python-e79066bd895a (https://medium.com/better-programming/the-only-step-by-step-guide-youll-need-to-build-a-web-scraper-with-python-e79066bd895a)](https://medium.com/better-programming/the-only-step-by-step-guide-youll-need-to-build-a-web-scraper-with-python-e79066bd895a)

In [132]:

```
import requests
from requests import get
from selenium.webdriver.common.by import By
from bs4 import BeautifulSoup
import pandas as pd
import numpy as np
from selenium import webdriver
import time
```

1. Import Data by using webscrapping

Open the URL with headless webdriver and parse the page source into html with beautifulsoup

In [175]:

```
driver = webdriver.Chrome('C:\\Users\Xiang Ze\Downloads\chromedriver_win32\chromedriver.
url = 'https://www.imdb.com/search/title/?groups=top_1000&ref_=adv_prvt'

driver.get(url)

time.sleep(5)

data = []

soup = BeautifulSoup(driver.page_source, 'html.parser')

# for tr in soup.find_all('div', attrs={'class': 'lister-item-content'}):
#     for td in tr:
#         gg = td.find_all('p', attrs={'class':'text-muted'})
#         data.append(gg.text.rstrip().replace('\n', ''))

# data

# Send a GET request to the IMDb URL
url = "https://www.imdb.com/search/title/?groups=top_1000&ref_=adv_prvt"
response = requests.get(url)

# Parse the HTML content using BeautifulSoup
soup = BeautifulSoup(response.content, "html.parser")

# Find all the movie items
movie_items = soup.find_all("div", attrs = {'class' : "lister-item"})

data = []
count = 0
# Loop through each movie item and extract the required information
driver = webdriver.Chrome('C:\\Users\Xiang Ze\Downloads\chromedriver_win32\chromedriver.
url = 'https://www.imdb.com/search/title/?groups=top_1000&ref_=adv_prvt'

driver.get(url)

time.sleep(5)

data = []

soup = BeautifulSoup(driver.page_source, 'html.parser')

# Create an empty list to store the extracted data
data = []

count = 0

while count < 20:
    movie_items = soup.find_all("div", attrs={'class': 'lister-item'})
    for item in movie_items:
        title = item.h3.a.text.strip()
        year = item.h3.find("span", class_="lister-item-year").text
        imdb_score = item.strong.text.strip()
        runtime = item.find("span", class_="runtime").text.strip()
        genre = item.find("span", class_="genre").text.strip()
        certificate_span = item.find('span', attrs={'class':'certificate'})
        certificate = ''
        if certificate_span:
            certificate = certificate_span.text.rstrip()
```

```
#         rating = item.find("span", class_="certificate").text.strip() if len(item.find_
        metascore = item.find("span", class_="metascore").text.strip() if item.find("spa
        description = item.find_all("p")[1].text.strip()
        votes = item.find_all("span", attrs={"name": "nv"})[0]["data-value"]
        gross = item.find_all("span", attrs={"name": "nv"})[1]["data-value"] if len(item
        directors_and_stars = item.find_all("p")[2].text.strip().replace("\n", "")
        directors, stars = directors_and_stars.split("|")
        directors = directors.replace('Director:', '').replace('Directors:', '').lstrip(
        stars = stars.replace('Star:', '').replace('Stars:', '').lstrip()

        data.append([title, year, certificate, genre, runtime, imdb_score, metascore, di

        print("Title:", title)
        print("Year:", year)
        print("Rating", certificate)
        print("Category:", genre)
        print("Runtime:", runtime)
        print("IMDb Score:", imdb_score)
        print("Metascore:", metascore)
        print("Directors:", directors)
        print("Stars:", stars)
        print("Description:", description)
        print("Votes:", votes)
        print("Gross:", gross)

    next_button = driver.find_element(By.LINK_TEXT, 'Next »')
    next_button.click()
    count += 1
    time.sleep(1)
    soup = BeautifulSoup(driver.page_source, 'html.parser')
```

```
Title: Spider-Man: Across the Spider-Verse
Year: (2023)
Rating PG
Category: Animation, Action, Adventure
Runtime: 140 min
IMDb Score: 8.9
Metascore: 86
Directors: Joaquim Dos Santos, Kemp Powers, Justin K. Thompson
Stars: Shameik Moore, Hailee Steinfeld, Brian Tyree Henry, Luna Lauren
Velez
Description: Miles Morales catapults across the Multiverse, where he en
counters a team of Spider-People charged with protecting its very exist
ence. When the heroes clash on how to handle a new threat, Miles must r
edefine what it means to be a hero.
Votes: 171744
Gross: 12
Title: Titanic
Year: (1997)
Rating PG-13
```

Append data found into list according to the category

In [102]:

```
```

Check if the data is webscrapped successfully

In [105]:

```
```

```
Title: The Departed
Year: (2006)
Rating <span>1</span>
Category: Crime, Drama, Thriller
Runtime: 151 min
IMDb Score: 8.5
Metascore: 85
Directors: Director:
Martin Scorsese

Stars:
    Stars:
Leonardo DiCaprio,
Matt Damon,
Jack Nicholson,
Mark Wahlberg
Description: An undercover cop and a mole in the police attempt to identif
y each other while infiltrating an Irish gang in South Boston.
Votes: 1363095
Gross: 132,384,315
```

2. Building a DataFrame With pandas

Put the data into data frame with Pandas

In [176]:

```
import pandas as pd
# pd.set_option('display.max_rows', None)

df = pd.DataFrame(data, columns=['Title', 'Year', 'Rating', 'Genre', 'Runtime', 'IMDb Sc
df = df.set_index('Title')
df
```

Out[176]:

Title	Year	Rating	Genre	Runtime	IMDb Score	Metascore	Directors	Stars
Spider-Man: Across the Spider-Verse	(2023)	PG	Animation, Action, Adventure	140 min	8.9	86	Joaquim Dos Santos, Kemp Powers, Justin K. Tho...	Shameik Moore, Hailee Steinfeld, Brian Tyree H...
Titanic	(1997)	PG-13	Drama, Romance	194 min	7.9	75	James Cameron	Leonardo DiCaprio, Kate Winslet, Billy Zane, K...
Avatar: The Way of Water	(2022)	PG-13	Action, Adventure, Fantasy	192 min	7.6	67	James Cameron	Sam Worthington, Zoe Saldana, Sigourney W...

3. Data Cleaning

Data cleaning - remove the '()' from year

In [177]:

```
df['Year'] = df['Year'].str.strip('()')
```

Data cleaning - remove the min from the timemin value

In [178]:

```
df['Runtime'] = df['Runtime'].str.replace(" min", "")
```

Data cleaning - remove the $ and M from the data value

In [114]:

```
#already done initially when scraping
```

Data cleaning - clear the ',' from the votes value

In [ ]:

```
#already done initially when scraping
```

4. Display Cleaned and Converted Code in Pandas

In [179]:

```
df
```

Out[179]:

Title	Year	Rating	Genre	Runtime	IMDb Score	Metascore	Directors	Stars
Spider-Man: Across the Spider-Verse	2023	PG	Animation, Action, Adventure	140	8.9	86	Joaquim Dos Santos, Kemp Powers, Justin K. Tho...	Shameik Moore, Hailee Steinfeld, Brian Tyree H...
Titanic	1997	PG-13	Drama, Romance	194	7.9	75	James Cameron	Leonardo DiCaprio, Kate Winslet, Billy Zane, K...
Avatar: The Way of Water	2022	PG-13	Action, Adventure, Fantasy	192	7.6	67	James Cameron	Sam Worthington, Zoe Saldana, Sigourney

5. Saving Your Data to a CSV

In [180]:

```
df.to_csv('imdb_data.csv', index=False)
```

6. Conclusion

What have you leanrt from this practice?

In [ ]:

```

```