Final project

December 27, 2022

[6]:
```
import requests
from bs4 import BeautifulSoup
import pandas as pd
import numpy as np
```

[7]:
```
#This HTTP headers let the client and the server pass additional information␣
 ↪with an HTTP request or response.
headers = {'User-Agent': 'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.
 ↪36 (KHTML, like Gecko) Chrome/56.0.2924.76 Safari/537.36'}
```

[8]:
```
url = 'https://www.themoviedb.org/movie'  # website url
```

[9]:
```
#This block of code is for pagination
url_lst = []  #Creating empty url_list
page_url = 'https://www.themoviedb.org/movie?page='  #Pagination url
for page in range(1,51):  #will loop all 50 pages
    url_lst.append(page_url + str(page))  #concatenating to get all pages url


#This block of code will scrape all the cards information
all_movie_list = []  #creating empty list
for url_ in url_lst:  #looping all url
    r = requests.get(url_,headers = headers).text  # Send a request to the␣
 ↪website
    soup = BeautifulSoup(r,'lxml')  # Parse the HTML content
    card1 = soup.find_all('div',class_='card style_1')  # Find all the block of␣
 ↪cards on the page

    for item in card1:  #This loop will get you all details
            movie = item.find('a')['title']  #Title name
            rating = item.
 ↪find('div',class_="user_score_chart")['data-percent'] #Will get you ratings
            movie_lst = []
            movie_url = item.find('a')['href']  #Will get you movie_url
            common_url = 'https://www.themoviedb.org'  #common_url
            inner_url =  common_url + movie_url  #adding common_url and␣
 ↪movie_url
```

```
                r2 = requests.get(inner_url,headers = headers).text  #send a␣
↪request to inner card details
                soup2 = BeautifulSoup(r2,'lxml')  #Parse the HTML content
                inner_card = soup2.find('div',class_="header_poster_wrapper") ␣
↪#inner_card div
                span_genre = inner_card.find_all('span',class_="genres") ␣
↪#inner card genre

                #THIS BLOCK IS FOR GENRE
                genre = []  #empty list for genre
                for span in span_genre:  #looping all genre
                    links = span.find_all('a')  #extracting data and storing␣
↪into variable
                    for link in links:  #nested loop to find genre
                            genre.append(link.text)  #will append to empty list
                genre = str(genre)[1:-1].replace("'",'')  #removing spaces from␣
↪list

                #THIS BLOCK IS FOR RELEASE_DATE
                release_date = soup2.find('span',class_="release").text.strip().
↪rstrip().split(" ")[0]  #removing spaces from release date


                #THIS BLOCK IS FOR RUNTIME
                inner_runtime = inner_card.find('span',class_='runtime')␣
↪#Storing data to a variable
                runtime_ = []  #empty list for runtime
                if inner_runtime != None:  #filtering a none type
                    inner_runtime = inner_runtime.text.strip().rstrip() ␣
↪#removing spaces from runtimes
                    runtime_.append(inner_runtime)  #appending in runtime empty␣
↪list
                    runtime_ = str(runtime_).replace("['",'').replace("']",'')␣
↪#converting empty list to str and removing brackets
                else:
                    continue

                #THIS BLOCK IS FOR DIRECTOR
                list_people = soup2.find_all('li',class_="profile") #Extracting␣
↪and stroing into variable
                dir_list =[]  #empty list
                for item in list_people:  ##will loop to get name of the␣
↪director
                        if 'Director' in item.find('p',class_="character").text:
↪ #will filter and convert into text
```

```
                              dir_list.append(item.find('a').text) #appending␣
↪into empty list
                dir_list = str(dir_list)[1:-1].replace("'",'') # removing␣
↪spaces

                #THIS BLOCK WILL STORE ALL THE DATA TO DICTIONARY
                all_movie_info = {
                     'Name' :movie,
                     'Rating':rating,
                     'Genre':genre,
                     'Release date':release_date,
                     'Runtime':runtime_,
                     'Director':dir_list,
                     'Url':inner_url
                 }
                all_movie_list.append(all_movie_info)  #appending into empty␣
↪list
```

[10]:
```
df = pd.DataFrame(all_movie_list)
```

[36]:
```
# df2 = df.mask(df == '')
# df = df.replace('r^\s*$',np.nan,regex = True,inplace=['Genre','Director'])
# df = df.mask(df.applymap(str).eq(''))
# mask = df.eq('')
# df = df.mask(mask,np.nan)
# print(df)
```

[11]:
```
df
```

[11]:
```
                                                  Name Rating  \
0                               Avatar: The Way of Water     80
1                                                  Troll   67.0
2                                             Black Adam   72.0
3     The Chronicles of Narnia: The Lion, the Witch …   71.0
4                          Guillermo del Toro's Pinocchio   84.0
..                                                   …     …
972                                 Fistful of Vengeance   56.0
973                                         Cold Pursuit   57.0
974                                          Titanic 666   62.0
975                                        Scary Movie 5   48.0
976                             Wrong Turn 5: Bloodlines   54.0

                                       Genre Release date Runtime  \
0    Science Fiction, Adventure, Action   12/16/2022  3h 12m
1            Fantasy, Action, Adventure   12/01/2022  1h 44m
2      Action, Fantasy, Science Fiction   10/21/2022   2h 5m
3            Adventure, Family, Fantasy   12/09/2005  2h 23m
4             Animation, Fantasy, Drama   11/17/2022  1h 57m
..                                  …           …      …
972                     Action, Fantasy   02/17/2022  1h 37m
973             Action, Crime, Thriller   02/08/2019  1h 59m
974                    Thriller, Horror   04/15/2022  1h 31m
975                              Comedy   04/12/2013  1h 26m
976                    Horror, Thriller   10/23/2012  1h 31m

                                Director  \
0                          James Cameron
1                            Roar Uthaug
2                     Jaume Collet-Serra
3                         Andrew Adamson
4    Guillermo del Toro, Mark Gustafson
..                                    …
972                           Roel Reiné
973                    Hans Petter Moland
974                             Nick Lyon
975                         Malcolm D. Lee
976                      "Declan OBrien"

                                         Url
0     https://www.themoviedb.org/movie/76600
1    https://www.themoviedb.org/movie/736526
2    https://www.themoviedb.org/movie/436270
3       https://www.themoviedb.org/movie/411
4    https://www.themoviedb.org/movie/555604
..                                         …
972  https://www.themoviedb.org/movie/890656
973  https://www.themoviedb.org/movie/438650
974  https://www.themoviedb.org/movie/945657
975    https://www.themoviedb.org/movie/4258
976  https://www.themoviedb.org/movie/125509

[977 rows x 7 columns]
```

[38]:
```
df.to_excel('final_project1.xlsx')
```

[ ]:
```
```

[ ]:
```
```