Scraping and Saving Top 250 Rated Movies from IMDb Using Python and Pandas

Introduction

This Python script scrapes the top 250 rated movies from IMDb and saves the data to a CSV file using Pandas.

Libraries Used

The script uses the following libraries:

- pandas
- requests
- BeautifulSoup

Code Explanation

The script starts by defining the URL to scrape and making a request to that URL using the `requests` library. It then parses the HTML content using the `BeautifulSoup` library and extracts the movie data from the table using BeautifulSoup's `find` and `find_all` methods.

The movie data is stored in an empty list called `movie_data`, and the script loops through each movie and extracts its rank, name, year, and rating. This data is then appended as a sublist to the `movie_data` list.

The `movie_data` list is then used to create a Pandas DataFrame using the `pd.DataFrame` function. The column names are specified as `Rank`, `Movie Name`, `Year`, and `Rating`.

Finally, the script saves the DataFrame to a CSV file using the `to_csv` method, with the filename `top_movies.csv`. The `index` parameter is set to `False` to exclude the index column from the CSV file.

Usage

To use the script, simply run it in a Python environment. The output CSV file will be saved in the same directory as the script.

Conclusion

This script demonstrates how to scrape data from a website and save it to a CSV file using Python and Pandas. It can be easily modified to scrape data from other websites and to extract different types of data.

Code

importing required libraries

In [34]:

```python
import pandas as pd
import requests as r
from bs4 import BeautifulSoup as bs
```

Getting the permission from website

In [35]:

```python
data = 'https://www.imdb.com/chart/top/?ref_=nv_mv_250'
df = r.get(data)
```

Creating a BeautifulSoup Object to Parse HTML Content

In [36]:

```python
soup = bs(df.content, 'html.parser')
```

Extracting Movie Data from IMDb HTML Content Using BeautifulSoup

In [37]:

```python
movies = soup.find('tbody', class_="lister-list").find_all('tr')
```

Create an empty list to store the movie data

In [38]:

```python
movie_data = []
```

Loop through the movies and extract the data

```
In [39]: for movie in movies:
             name = movie.find('td', class_="titleColumn").a.text
             Rating = movie.find('td', class_="ratingColumn imdbRating").strong.text
             year = movie.find('td', class_="titleColumn").span.text.strip('()')
             Rank = movie.find('td', class_="titleColumn").get_text(strip=True).split('.')[0]
```

Appending the data to empty list movie_data

```
In [40]: movie_data.append([Rank, name, year, Rating])
```

Create a Pandas DataFrame from the movie data

```
In [41]: df = pd.DataFrame(movie_data, columns=['Rank', 'Movie Name', 'Year', 'Rating'])
```

Save the DataFrame to a CSV file

```
In [42]: df.to_csv('top_movies.csv', index=False)
```

```
In [43]: print('Data saved to top_movies.csv file.')
```

```
Data saved to top_movies.csv file.
```

```
In [ ]:
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js