# stat29000project06solutions

March 6, 2020

## 1 STAT29000 Project 6 Solutions

[4]:
```
import requests
import pandas as pd
from bs4 import BeautifulSoup as bsoup
```

### 1.1 Question 1

#### 1.1.1 1a

[5]:
```
%%html
<table class="chart full-width" data-caller-name="chart-top250movie"></table>
```

```
<IPython.core.display.HTML object>
```

#### 1.1.2 1b

[16]:
```
html = requests.get("https://www.imdb.com/chart/top/?ref_=nv_mv_250")
soup = bsoup(html.text)
print(soup.find_all(attrs={"data-caller-name":"chart-top250movie"})[0].
 ↪prettify()[:800])
```

```
<table class="chart full-width" data-caller-name="chart-top250movie">
 <colgroup>
  <col class="chartTableColumnPoster"/>
  <col class="chartTableColumnTitle"/>
  <col class="chartTableColumnIMDbRating"/>
  <col class="chartTableColumnYourRating"/>
  <col class="chartTableColumnWatchlistRibbon"/>
 </colgroup>
 <thead>
  <tr>
   <th>
   </th>
```

```
   <th>
    Rank &amp;amp; Title
   </th>
   <th>
    IMDb Rating
   </th>
   <th>
    Your Rating
   </th>
   <th>
   </th>
  </tr>
 </thead>
 <tbody class="lister-list">
  <tr>
   <td class="posterColumn">
    <span data-value="1" name="rk">
    </span>
    <span data-value="9.222316949890466" name="ir">
    </span>
    <span data-value="7.791552E11" name="us">
    </span>
    <span data-value="2192004" name="nv">
    </span>
    <span data-value="-1.777
```

### 1.1.3 1c

[7]:
```
movie_and_year_soup = soup.find_all(class_="titleColumn")
stars_soup = soup.find_all(class_="ratingColumn imdbRating")

stars = [star.strong.string for star in stars_soup]

movie = [movie.a.string for movie in movie_and_year_soup]
year = [year.span.string.replace('(', '').replace(')', '') for year in␣
 ↪movie_and_year_soup]

print(stars[:10])
print(movie[:10])
print(year[:10])
```

```
['9.2', '9.1', '9.0', '9.0', '8.9', '8.9', '8.9', '8.9', '8.8', '8.8']
['The Shawshank Redemption', 'The Godfather', 'The Godfather: Part II', 'The
Dark Knight', '12 Angry Men', "Schindler's List", 'The Lord of the Rings: The
Return of the King', 'Pulp Fiction', 'The Good, the Bad and the Ugly', 'The Lord
of the Rings: The Fellowship of the Ring']
['1994', '1972', '1974', '2008', '1957', '1993', '2003', '1994', '1966', '2001']
```

#### 1.1.4 1d

[8]:
```
def top250() -> pd.DataFrame:
    html = requests.get("https://www.imdb.com/chart/top/?ref_=nv_mv_250")
    soup = bsoup(html.text)

    movie_and_year_soup = soup.find_all(class_="titleColumn")
    stars_soup = soup.find_all(class_="ratingColumn imdbRating")

    stars = [float(star.strong.string) for star in stars_soup]
    movie = [movie.a.string for movie in movie_and_year_soup]
    year = [int(year.span.string.replace('(', '').replace(')', '')) for year in␣
 ↪movie_and_year_soup]

    return pd.DataFrame(data={"movie": movie, "year": year, "stars": stars})

top250().head()
```

[8]:

| | movie | year | stars |
|---|---|---|---|
| 0 | The Shawshank Redemption | 1994 | 9.2 |
| 1 | The Godfather | 1972 | 9.1 |
| 2 | The Godfather: Part II | 1974 | 9.0 |
| 3 | The Dark Knight | 2008 | 9.0 |
| 4 | 12 Angry Men | 1957 | 8.9 |

[60]:
```
t2 = top250()
t2.groupby(t2['year']>=2000).mean()
```

[60]:

| year | year | stars |
|---|---|---|
| False | 1972.355263 | 8.276316 |
| True | 2009.540816 | 8.228571 |

### 1.2 Question 2

#### 1.2.1 2a

[142]:
```
html = requests.get("https://www.imdb.com/title/tt0110357")
soup = bsoup(html.text)

metascore = soup.find_all(class_="metacriticScore score_favorable␣
 ↪titleReviewBarSubItem")
metascore[0].span.string
```

[142]: '88'

### 1.2.2 2b

[13]:
```
def get_metascore(id: str) -> int:

    html = requests.get(f"https://www.imdb.com/title/{id}")
    soup = bsoup(html.text)
    metascore = soup.find_all(class_="metacriticScore")

    if len(metascore) > 0:
        return int(metascore[0].span.string)
    else:
        return None

result = get_metascore('tt0110357')
print(result)
# 88
print(type(result))
# int
print(get_metascore('tt0095327'))
# None
```

```
88
<class 'int'>
None
```

### 1.2.3 2c

[12]:
```
def top250() -> pd.DataFrame:
    html = requests.get("https://www.imdb.com/chart/top/?ref_=nv_mv_250")
    soup = bsoup(html.text)

    movie_and_year_soup = soup.find_all(class_="titleColumn")
    stars_soup = soup.find_all(class_="ratingColumn imdbRating")
    id_soup = soup.find_all(class_="ratingColumn")


    ids = [i.find_all("div", attrs={"data-titleid": True}) for i in id_soup]
    ids = [i[0]['data-titleid'] for i in ids if len(i) > 0]
    metascore = [get_metascore(i) for i in ids]


    stars = [float(star.strong.string) for star in stars_soup]
    movie = [movie.a.string for movie in movie_and_year_soup]
    year = [int(year.span.string.replace('(', '').replace(')', '')) for year in␣
↪movie_and_year_soup]
```

```
    return pd.DataFrame(data={"movie": movie, "year": year, "stars": stars,␣
 ↪"metascore": metascore})


top250().head()
```

[12]:

| | movie | year | stars | metascore |
|---|---|---|---|---|
| 0 | The Shawshank Redemption | 1994 | 9.2 | 80.0 |
| 1 | The Godfather | 1972 | 9.1 | 100.0 |
| 2 | The Godfather: Part II | 1974 | 9.0 | 90.0 |
| 3 | The Dark Knight | 2008 | 9.0 | 84.0 |
| 4 | 12 Angry Men | 1957 | 8.9 | 96.0 |