In [ ]:
```
import pandas as pd
import requests
import json
from bs4 import BeautifulSoup
import time
from urllib.parse import quote
```

In [ ]:
```
def get_scores(url):
        try:
                # debug
                print(url)

                # get the pages content
                page = requests.get(url)
                soup = BeautifulSoup(page.content, "html.parser")

                # get the elements we want, could be one of two possibilities
                job_elements = soup.find_all("score-board-deprecated")
                job_elements2 = soup.find_all("script", {"id": "media-scorecard-

                # rotten tomatoes new code
                if job_elements2:
                        data = json.loads(job_elements2[0].text)
                        if "score" in data["audienceScore"]:
                                audience = data["audienceScore"]["score"]
                        else:
                                audience = ""

                        if "score" in data["criticsScore"]:
                                critic = data["criticsScore"]["score"]
                        else:
                                critic = ""

                # rotten tomatoes legacy code
                else:
                        critic = job_elements[0]["tomatometerscore"]
                        audience = job_elements[0]["audiencescore"]

                return critic, audience
        except:
                return "", ""
```

In [ ]:
```
# load in the data
df = pd.read_csv('netflix_titles.csv')
titles = df[["type", "title"]]

# URL = "https://www.imdb.com/find/?q=%s"
URL = "https://www.rottentomatoes.com/search?search=%s"

for type, title in titles.values[:408]:

        # replace spaces with %20
        url_title = quote(title)
        url = URL % url_title
        print(f"\n{title}")

        # get the pages content
        page = requests.get(url)
        soup = BeautifulSoup(page.content, "html.parser")
        job_elements = soup.find_all("search-page-result")

        found = False

        for element in job_elements:

                # look for movie or tv series
                if type == "Movie":
                        if element["type"] == "movie":
                                list = element.find_all("a", {"slot" : "title"})

                else:
                        if element["type"] == "tvSeries":
                                list = element.find_all("a", {"slot" : "title"})

        # if there are elements, then get the first one ( rotten tomatoes best m
        if list:
                name = str.strip(list[0].text)
                movie_url = str.strip(list[0]['href'])

                # get the scores
                critic, audience = get_scores(movie_url)
                print(f"{name} - Critic: {critic}, Audience: {audience}")

                # update the dataframe
                df.loc[df["title"] == title, "audience_score"] = audience
                df.loc[df["title"] == title, "critic_score"] = critic

                # save the data
                df.to_csv("netflix_titles_scraped.csv", index=False)
        else:
                print("not found")

        # throttle the connections
        time.sleep(0.5)
```

```
Dick Johnson Is Dead
https://www.rottentomatoes.com/m/dick_johnson_is_dead
Dick Johnson Is Dead - Critic: 99, Audience: 78

Blood & Water
https://www.rottentomatoes.com/tv/blood_and_water
Blood & Water - Critic: , Audience: 84

Ganglands
https://www.rottentomatoes.com/tv/ganglands
Ganglands - Critic: , Audience: 

Jailbirds New Orleans
https://www.rottentomatoes.com/tv/jailbirds_new_orleans
Jailbirds New Orleans - Critic: , Audience: 100

Kota Factory
https://www.rottentomatoes.com/tv/kota_factory
Kota Factory - Critic: , Audience: 80

Midnight Mass
https://www.rottentomatoes.com/tv/midnight_mass
Midnight Mass - Critic: 87, Audience: 79

My Little Pony: A New Generation
https://www.rottentomatoes.com/m/my_little_pony_a_new_generation
My Little Pony: A New Generation - Critic: 92, Audience: 83

Sankofa
https://www.rottentomatoes.com/m/sankofa
Sankofa - Critic: 94, Audience: 89

The Great British Baking Show
https://www.rottentomatoes.com/tv/the_great_british_baking_show_holidays
```