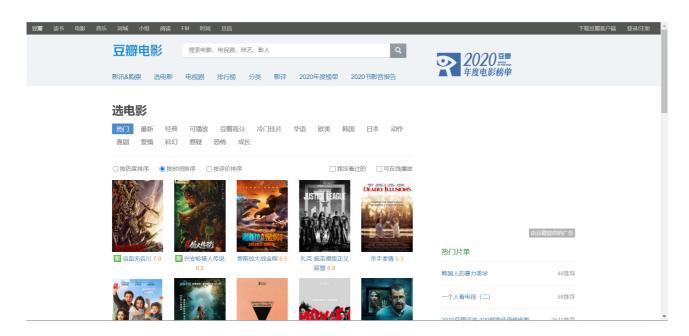
Selenium技术按标签爬取豆瓣电影

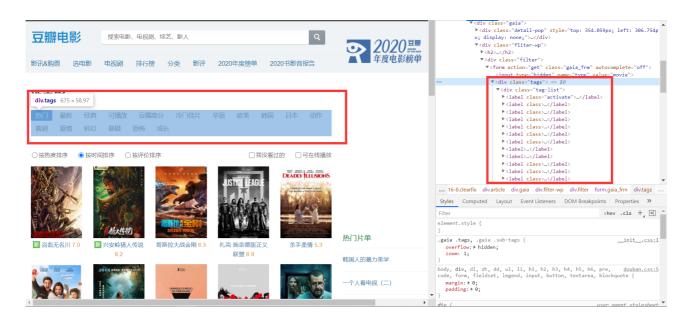
1.爬取准备

这次爬取的是豆瓣的"选电影"页面

"页面。



简单的分析可以发现,导航栏的所有标签在一个div中,而每个标签存放在一个label中。我们只需要模拟点击去访问这些label标签就可以实现下方电影内容的变化。

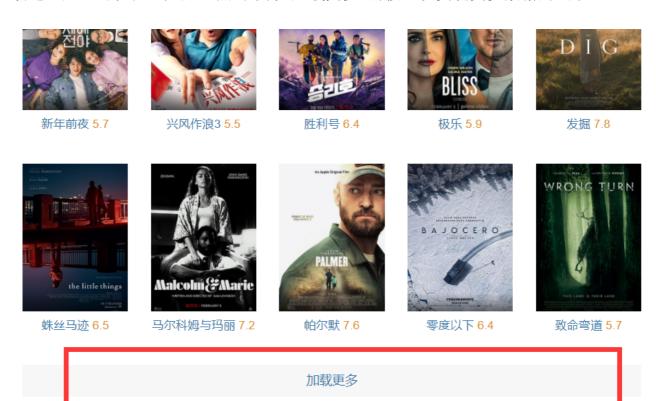


电影信息存放较为简单,一个a标签存放了其转跳的连接,而电影名和评分直接在下方的两个字符标签中。



思路整理:外层的循环遍历所有的tags,遍历的同时捕获页面下方的电影信息,从而达到爬取所有标签下电影的效果。最后将数据保存在csv文件中。

改进之处:可以在日后加入点击下方"加载更多"的按钮以实现更多数据的爬取。



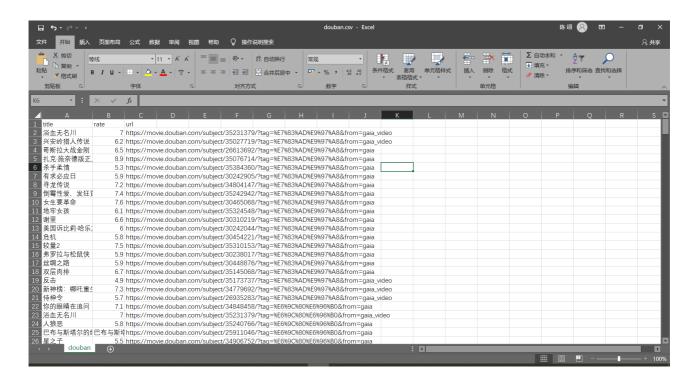
2.爬取代码

from selenium import webdriver
import selenium
from bs4 import BeautifulSoup
from time import sleep

```
import pandas as pd
def get_cur_page():
    # 爬取当前页面的所有内容
    buffer = []
    items_list = browser.find_elements_by_css_selector(".list a")
    for ele in items_list:
        tmp = ele.text.split(" ")
        print(tmp[0], tmp[-1], ele.get_attribute('href'))
        try:
            buffer.append([tmp[0], tmp[-1],
ele.get_attribute('href')])
        except exception as ex:
            print("此电影没有评分")
    return pd.DataFrame(buffer, columns=['title','rate','url'])
if __name__ == '__main__':
    df = pd.DataFrame(columns=['title','rate','url'])
   # 1.爬取tag-list
    browser =
selenium.webdriver.Edge(executable_path="S:/web_browser_driver/msed
gedriver.exe")
    target_url =
'https://movie.douban.com/explore#!type=movie&tag=%E7%83%AD%E9%97%A
8&sort=time&page_limit=20&page_start=0'
    browser.implicitly_wait(5)
    browser.get(target_url)
    page_list = browser.find_elements_by_css_selector(".tag-list
label")
    # 2. 爬取每个页面的内容
    for page in page_list:
        page.click()
        df = df.append(get_cur_page(), ignore_index=True)
        sleep(2)
        print(df)
    # 关闭selenium并保存
    path = "E:/douban.csv"
    browser.close()
```

```
df.to_csv(path, index=False, encoding="utf_8_sig")
print("保存成功", path)
```

3.爬取结果



演示gif (加载时间或许很长)