豆瓣Top250爬虫与数据分析

Top网址

> https://movie.douban.com/top250

项目涉及内容

- 获取豆瓣Top250电影的基本信息，并导入到表格
- 根据的到的基本信息表格用matplotlib，绘制图表，分析其各字段的相关性
- 获取所有电影的内容简介，并导入到txt文件中

Top250电影信息

涉及文件

- **Top250details.py**
- **top250.xlsx**

要点/步骤

- 爬虫的时候会出现禁止访问的情况，首先需要定义headers 并在requests中传参

  > requests.get(url=MainPage, headers=headers)

  是在没办法是可以通过重启路由器 或重启手机换自己的网络ip

- Top250网址有10页，需要对网址进行翻页操作

  主要通过对网址的结构观察得到start=后的数字不同，每一页都是累加25 对该数字累加实现跳转

  > MainPage = "https://movie.douban.com/top250?start=" + str(x) + '&filter='

- 每部电影的信息采集需要匹配个字段 需要存在各变量中，通过find查找

```
Soup = BeautifulSoup(strMainpage, 'lxml')
# 在获取解析好BeautifulSoup类Soup中寻找class='grid_view'的'ol'标签 25条记录的
details都写在该标签下
ol_tag = Soup.find('ol', class_='grid_view')
# 在找到的'ol'标签中寻找所有的li标签 共会得到25条
details = ol_tag.find_all('li')
# li标签迭代 即每条记录中寻找各类信息
for detail in details:
    # 电影排名 存放在em标签中 获取其text属性
    movie_rank = detail.find('em').text
    # 电影标题 存放在class='title' 的span标签 获取其text属性
    movie_title = '《' + detail.find('span', class_='title').text + '》'
    # 电影别名 与title同理 存放在class='other'的span标签中
    movie_other = '《' + detail.find('span',
class_='other').text.split('\xa0')[-1].strip() + '》'
    # 电影评论数 运用正则表达式 寻找 xxx人评价的字段 将他变成字符串 用切片的方法取倒数第
三个字以前的内容
    movie_CommentNum = detail.find(text=re.compile(r'\d+人评价')).string[:-3]
    # 电影评分 寻找class='rating_num'的span标签 获取其text属性
```

```
        movie_score = detail.find('span', class_='rating_num').text
        # 查看页面结构 会发现上映时间等信息都存在p标签中 所以先获取p标签的所有内容
        movie_P = detail.find('p').text
        # 用split方法根据回车分割成两个字符串
        movie_P1 = movie_P.split('\n')[1]
        movie_P2 = movie_P.split('\n')[2]
        # 电影导演 P1中用split 根据'\xa0'分割 取第一个元素 strip方法去除空格 再获取其前第4个字符以后的字符即导演
        movie_director = movie_P1.split('\xa0')[0].strip()[4:]
        # 电影上映时间 P2中用split 根据'\xa0'分割 取第一个元素 strip方法去除空格 再获取其前4个字符即上映时间
        movie_year = movie_P2.split('\xa0')[0].strip()[:4]
        # 制片国家 P2中用split 根据'\xa0'分割 取第二个元素
        movie_country = movie_P2.split('\xa0')[2]
        # 电影类型 P2中用split 根据'\xa0'分割 取最后一个元素
        movie_type = movie_P2.split('\xa0')[-1].strip()
        # 将必要的属性加入到details的列表中
        movie_details.append([movie_rank, movie_title,movie_other,
movie_CommentNum, movie_score, movie_year,
movie_director, movie_country,movie_type])
```

```
'9', '《盗梦空间》', '《潜行凶间(港)  /  全面启动(台)》', '1564248', '9.3', '2010', '克里斯托弗·诺兰 Christo
'10', '《忠犬八公的故事》', '《秋田犬八千(港)  /  忠犬小八(台)》', '1071447', '9.4', '2009', '莱塞·霍尔斯道
'11', '《海上钢琴师》', '《声光伴我飞(港)  /  一九零零的传奇》', '1281294', '9.3', '1998', '朱塞佩·托纳多雷
'12', '《星际穿越》', '《星际启示录(港)  /  星际效应(台)》', '1236808', '9.3', '2014', '克里斯托弗·诺兰 Chri
'13', '《楚门的世界》', '《真人Show(港)  /  真人戏》', '1158976', '9.3', '1998', '彼得·威尔 Peter Weir', '美
'14', '《三傻大闹宝莱坞》', '《三个傻瓜(台)  /  作死不离3兄弟(港)》', '1428479', '9.2', '2009', '拉库马·希拉
'15', '《机器人总动员》', '《太空奇兵·威E(港)  /  瓦力(台)》', '1009524', '9.3', '2008', '安德鲁·斯坦顿 Andr
'16', '《放牛班的春天》', '《歌声伴我心(港)  /  唱诗班男孩》', '993368', '9.3', '2004', '克里斯托夫·巴拉蒂 C
'17', '《大话西游之大圣娶亲》', '《西游记完结篇仙履奇缘  /  齐天大圣西游记》', '1138276', '9.2', '1995', '刘
'18', '《熔炉》', '《无声呐喊(港)  /  漩涡》', '698308', '9.3', '2011', '黄东赫 Dong-hyuk Hwang', '韩国', '
'19', '《疯狂动物城》', '《优兽大都会(港)  /  动物方城市(台)》', '1368864', '9.2', '2016', '拜伦·霍华德 Byro
'20', '《无间道》', '《Infernal Affairs  /  Mou gaan dou》', '927208', '9.2', '2002', '刘伟强 / 麦兆辉', '中
'21', '《龙猫》', '《邻居托托罗  /  邻家的豆豆龙》', '954797', '9.2', '1988', '宫崎骏 Hayao Miyazaki', '日本
'22', '《教父》', "《Mario Puzo's The Godfather》", '697727', '9.3', '1972', '弗朗西斯·福特·科波拉 Francis F
'23', '《当幸福来敲门》', '《寻找快乐的故事(港)  /  追求快乐》', '1150764', '9.1', '2006', '加布里尔·穆奇诺
'24', '《怦然心动》', '《萌动青春  /  青春萌动》', '1336232', '9.1', '2010', '罗伯·莱纳 Rob Reiner', '美国',
'25', '《触不可及》', '《闪亮人生(港)  /  逆转人生(台)》', '744051', '9.2', '2011', '奥利维·那卡什 Olivier N
'26', '《控方证人》', '《雄才伟略  /  情妇》', '309359', '9.6', '1957', '比利·怀尔德 Billy Wilder', '美国',
'27', '《蝙蝠侠：黑暗骑士》', '《蝙蝠侠前传2：黑暗骑士  /  黑暗骑士(台)》', '774762', '9.2', '2008', '克里斯
'28', '《活着》', '《人生  /  Lifetimes》', '601151', '9.3', '1994', '张艺谋 Yimou Zhang', '中国大陆 中国香
'29', '《末代皇帝》', '《末代皇帝溥仪(港)》', '558855', '9.3', '1987', '贝纳尔多·贝托鲁奇 Bernardo Bertoluc
'30', '《寻梦环游记》', '《玩转极乐园(港)  /  可可夜总会(台)》', '1153206', '9.1', '2017', '李·昂克里奇 Lee

'31', '《乱世佳人》', '《飘》', '512981', '9.3', '1939', '维克多·弗莱明 Victor Fleming / 乔治·库克 George Cu
'32', '《何以为家》', '《迦百农  /  星仔打官司(港)》', '717385', '9.1', '2018', '娜丁·拉巴基 Nadine Labaki',
'33', '《指环王3：王者无敌》', '《魔戒三部曲：王者再临(台 / 港)》', '566778', '9.2', '2003', '彼得·杰克逊 Pe
'34', '《摔跤吧！爸爸》', '《我和我的冠军女儿(台)  /  打死不离3父女(港)》', '1175853', '9.0', '2016', '涅提·
'35', '《飞屋环游记》', '《冲天救兵(港)  /  天外奇迹(台)》', '1000660', '9.0', '2009', '彼特·道格特 Pete Doc
```

数据整理

解决完以上问题后将得到的列表通过构造pandas.DataFrame方法输出excel文件

```
# 用得到的details用pandas构造DataFrame对象 columns表示每一列的字段
movie_frame = pd.DataFrame(
        Movie_details,
        columns=['排名', '标题', '又名', '评论数', '评分', '上映时间', '导演', '制片国家', '电影类型'])
# 将整理好的DataFrame对象导出成excel表格
movie_frame.to_excel('Douban_DataAnalyze/top250.xlsx')
```

根据获取数据绘制表格

涉及文件

- **Top250chart.ipynb**
- **Top250chart.py**
- **Top250chart.html**

步骤

- 建立Jupyer Notebook 文件 扩展名为.ipynb，该文件将代码分块运行，像终端一样
- 导入整理好的表格

```
import pandas as pd
movie_frame = pd.read_excel('top250.xlsx')
# 删除'Unnamed: 0'这一列
movie_frame = movie_frame.drop(columns='Unnamed: 0')
```

- 导入matplotlib库 编写图表显示设置

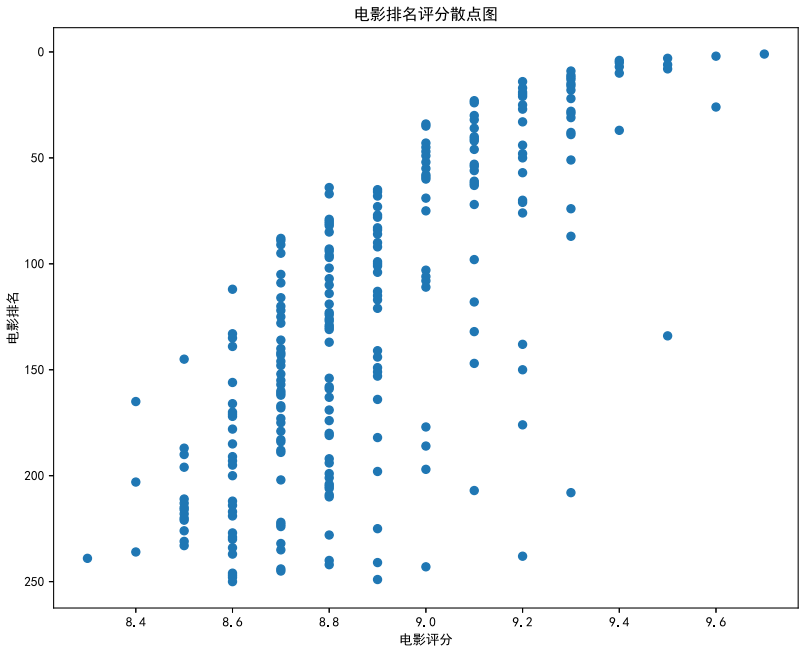
```
import matplotlib.pyplot as plt
import numpy as np

%matplotlib inline
plt.rcParams["font.sans-serif"]="SimHei" #解决中文乱码问题
plt.rcParams['axes.unicode_minus']=False #解决负号无法正常显示的问题
%config InlineBackend.figure_format='svg'#将图表设置成矢量图格式，使图表更清楚
```

- 根据不同的需求建立不同的表格 并导出svg

> plt.savefig('img/Movie-ScoreCount.svg',format='svg')

- 最后将该文件输出为html文件和py文件



获取每部电影的简介

涉及文件

- **Top250summary.py**
- **电影.txt**

要点/步骤

- 编写get_links()函数，在Top250主页中获取所有电影的链接

  网址链接构造，实现翻页，抓取每一页中所有电影条目的href属性，并将这些链接存放在列表对象中

```
# 用BeautifulSoup方法将得到的网页转换成可解析的类型
Soup = BeautifulSoup(strMainpage, 'lxml')
# 1-25循环在该页的25条电影记录中寻找链接
for i in range(1, 25):
    # 网页源代码得到网页链接a标签的css选择器
    Data = Soup.select(
        '#content > div > div.article > ol > li:nth-child(' + str(i) +
        ') > div > div.info > div.hd > a')
    # 将得到的标签内容中的href属性值添加到links列表 由于Data得到的是一个元组，所以需要用
for循环迭代
    for item in Data:
        links.append(item.get('href'))
        print('导入链接成功: %s' % item.get('href'))
```

- 根据获取的links后，用get方法获取每一页的内容，然后根据选择器获取每部电影的内容简介

```
strurl = requests.get(url=link, headers=headers, data=payload).text
# 接下来的操作与之前的一样 ： 获得text 再从text中清洗内容传入到film字典中
Soup = BeautifulSoup(strurl, 'lxml')
# 获取标题和简介的标签
Title_data = Soup.select('#content > h1 > span:nth-child(1)')
Summary_data = Soup.select('#link-report > span:nth-child(1)')
```

- 获取的Summary_data中存在很多多余的格式，需要对其split切割 与replace替换，并存到film字典中的'Summary'key中

```
for x in Summary_data:
    # 将空格变成'' 并按'\n'分割形成列表
    film['Summary'] = x.get_text().replace(' ', '').split('\n')
```

- 将处理好的Summary写入文件中，需要注意格式与文案清晰度 （''是不能写入文件的）

```
f.write(film['title'])
f.write('\n-----------------\n')
for x in film['Summary']:
    # 如果是''就不写入文件
    if x != '':
        f.write(x)
        f.write('\n')
f.write('\n------------------------------------------------------------\n')
```

```
导入链接成功：https://movie.douban.com/subject/1397546/
导入链接成功：https://movie.douban.com/subject/1300117/
导入链接成功：https://movie.douban.com/subject/5908478/
导入链接成功：https://movie.douban.com/subject/5964718/
导入链接成功：https://movie.douban.com/subject/1291824/
导入链接成功：https://movie.douban.com/subject/1304141/
导入链接成功：https://movie.douban.com/subject/27615441/
导入链接成功：https://movie.douban.com/subject/1292233/
导入链接成功：https://movie.douban.com/subject/27191492/
链接获取成功
导入简介:肖申克的救赎 The Shawshank Redemption
导入简介:霸王别姬
导入简介:阿甘正传 Forrest Gump
导入简介:这个杀手不太冷 Léon
导入简介:泰坦尼克号 Titanic
导入简介:美丽人生 La vita è bella
导入简介:千与千寻 千と千尋の神隠し
导入简介:辛德勒的名单 Schindler's List
导入简介:盗梦空间 Inception
导入简介:忠犬八公的故事 Hachi: A Dog's Tale
```