教学方法与技巧分享

Python爬虫实践：《流浪地球》豆瓣影评分析

张敏

目录

教学目标确认

1. 让学生掌握Chrome开发者工具的使用。
2. 让学生掌握使用Selenium库模拟浏览器操作的方法。
3. 让学生掌握HTML网页解析方法，如正则表达式、XPath、BeautifulSoup等。
4. 让学生学会对获取的数据进行简单的分析和数据可视化。
5. 学生能够动手实现爬取其他主流网站数据。
6. 让学生掌握爬虫工程师的主要工作技能，为相关求职工作做准备。

目录

案例任务点拆解

获取网页源代码	网页解析	循环爬取	分析评论关键信息	评论数量、分值分析	用户城市分析
Selenium库	XPath	翻页实现	预处理	数据统计	数据统计
HTTP请求	正则表达式	等待响应	统计词频	可视化	可视化
	Beautiful Soup		绘制词云		

案例任务点拆解

- 获取网页源代码
  - Selenium库
  - HTTP请求

```
from selenium import webdriver
from lxml import etree
import requests
import pandas as pd
import time
```

```
driver = webdriver.Chrome()    # 启动chrome浏览器
url = 'https://movie.douban.com/subject/26266893/comments?status=P'
driver.get(url)    # 获取网页源码数据
```

案例任务点拆解

网页解析

- XPath
- 正则表达式
- Beautiful Soup

```
def get_web_data(dom=None, cookies=None):
    '''
    获取每页评论数据
    '''
    names = dom.xpath('//div[@class="comment-item"]//span[@class="comment-info"]/a/text()')          # 用户名
    ratings = dom.xpath('//div[@class="comment-item"]//span[@class="comment-info"]/span[2]/@class')   # 评分
    times = dom.xpath('//div[@class="comment-item"]//span[@class="comment-info"]/span[@class="comment-time "]/@title')     # 评论发布时间
    message = dom.xpath('//div[@class="comment-item"]//div[@class="comment"]//span[@class="short"]/text()')    # 短评正文
    user_url = dom.xpath('//div[@class="comment-item"]//span[@class="comment-info"]/a/@href')           # 用户主页网址
    votes = dom.xpath('//div[@class="comment-item"]//div[@class="comment"]//span[@class="votes"]/text()')    # 赞同数量
```

```
    load_times.append(load_time)
    time.sleep(3)
ratings = ['' if 'rating' not in i else int(re.findall('\d{2}', i)[0]) for i in ratings]   # 评分数据整
load_times = ['' if i == [] else i[1].strip()[:-2] for i in load_times]   # 入会数据整理
cities = ['' if i == [] else i[0] for i in cities]   # 居住地数据整理
```

```
for i in user_url:
    web_data = requests.get(i, cookies=cookies)
    dom_url = etree.HTML(web_data.text, etree.HTMLParser(encoding='utf-8'))
    address = dom_url.xpath('//div[@class="basic-info"]//div[@class="user-info"]/a/text()')                    # 用户居住地
    load_time = dom_url.xpath('//div[@class="basic-info"]//div[@class="user-info"]/div[@class="pl"]/text()')   # 用户入会时间
    cities.append(address)
```

案例任务点拆解

- 循环爬取
- 翻页实现
- 等待响应

```
# 对所有页面进行数据爬取及解析操作，并进行数据保存
all_data = pd.DataFrame()
wait = WebDriverWait(driver, 20)
while True:

    wait.until(
        EC.element_to_be_clickable(    # 通过该项条件确认网页是否已经加载进来
            (By.CSS_SELECTOR, '#comments > div:nth-child(20) > div.comment > h3
        )
    )
```

```
if driver.find_element_by_css_selector('#paginator > a.next')==[]:    # 判定是否还有“后页”按钮
    break

confirm_bnt = wait.until(
    EC.element_to_be_clickable(
        (By.CSS_SELECTOR, '#paginator > a.next')
    )
)
confirm_bnt.click()    # 执行翻页操作
```

案例任务点拆解

- 分析评论关键信息
  - 预处理
  - 统计词频
  - 绘制词云

```
def my_word_cloud(data=None, stopWords=None, img=None):
    dataCut = data.apply(jieba.lcut)  # 分词
    dataAfter = dataCut.apply(lambda x: [i for i in x if i not in stopWords])  # 去除停用词
    wordFre = pd.Series(_flatten(list(dataAfter))).value_counts()  # 统计词频
    mask = plt.imread(img)
    wc.fit_words(wordFre)
    plt.imshow(wc)
    plt.axis('off')
```

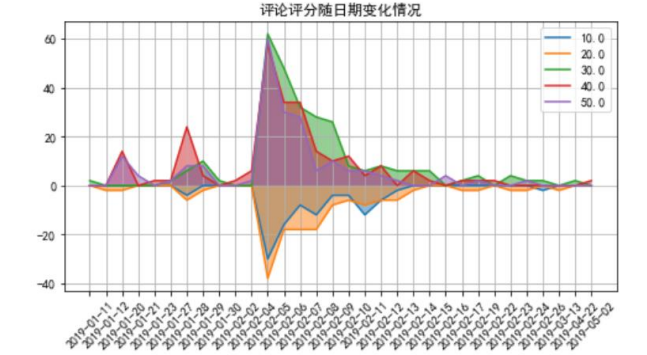
```
: my_word_cloud(data=data['短评正文'][index_positive], stopWor
```

案例任务点拆解

- 评论数量、分值分析
  - 数据统计
  - 可视化

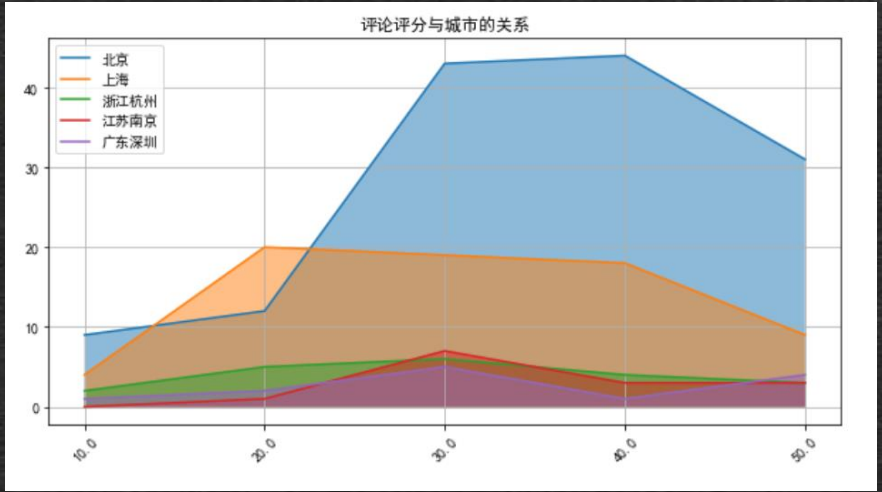
```
plt.plot(range(len(num)), num)
plt.xticks(range(len(num)), num.index)
plt.title('评论数量随时刻的变化情况')
plt.grid()
```


```
plt.legend(tmp.columns)
plt.xticks(range(n), tmp.index, rotation=45)
plt.show()
```



案例任务点拆解

- 用户城市分析
  - 数据统计
  - 可视化



目录

技能梳理与串联

基础技能	Python基础
	Selenium库
	Google开发者工具
	lxml库
	Pandas库
进阶技能	XPath
	显式等待
	Selenium翻页
	反反爬
	绘制词云
	可视化

- 实战技能：综合运用
- 前置课程：
  1. Python编程基础
  2. Python数据分析与应用
  3. Python数据可视化
  4. Python网络爬虫实战

重难点

技能梳理与串联

技能梳理

基础技能
- Python基础
- Selenium库
- Google开发者工具
- lxml库
- Pandas库

进阶技能
- XPath
- 显式等待
- Selenium翻页
- 反反爬
- 绘制词云
- 可视化

```
scores = dom.xpath('//*[@class="comment-info"]/span[2]/@class')
scores =['' if 'rating' not in i else int(re.findall('[0-9]{2}',
times = dom.xpath('//*[@class="comment-time "]/@title')  # 短评发
content = dom.xpath('//*[@class="short"]/text()')  # 评论正文
votes = dom.xpath('//*[@class="votes"]/text()')  # '//*[@class="
```

```
wait.until(
    EC.element_to_be_clickable(
        (By.CSS_SELECTOR, '#paginator > a:nth-child(1)')
    )
)
```

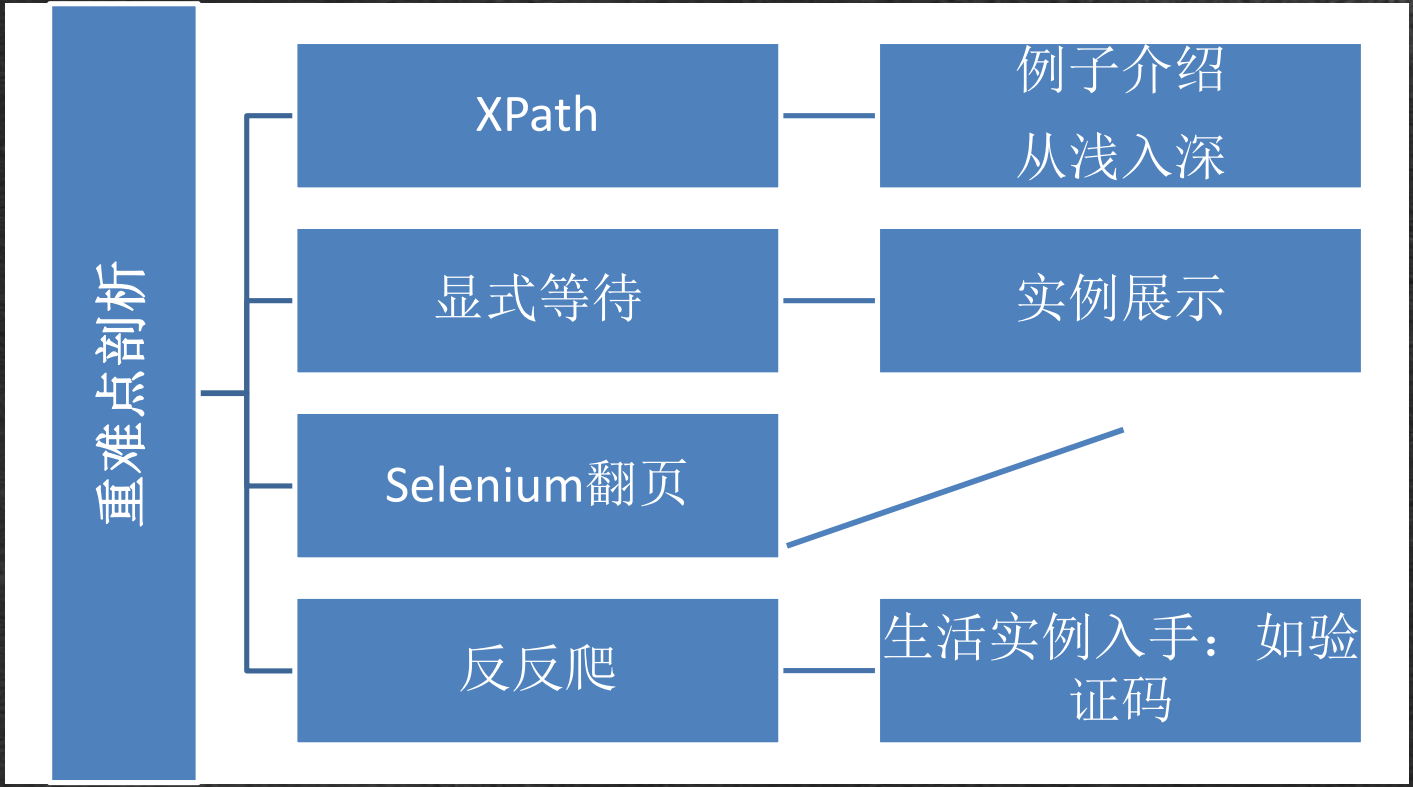
```
confirm_btn = wait.until(
    EC.element_to_be_clickable(
        (By.CSS_SELECTOR, '#paginator > a.next')
    )
)

confirm_btn.click()
```

```
import requests
for i in user_page:
    rq = requests.get(i, cookie
    dom3 = etree.HTML(rq.text,
    citys.append(dom3.xpath('//
    user_info.append(dom3.xpath
    time.sleep(1)
```

```
= 'bid=8B0N87v9JBU; dbcl2="1833
}
okie_str.split(';'):
i.split('=', 1)
[k] = v
er_page:
quests.get(i, cookies=cookies)
```

目录

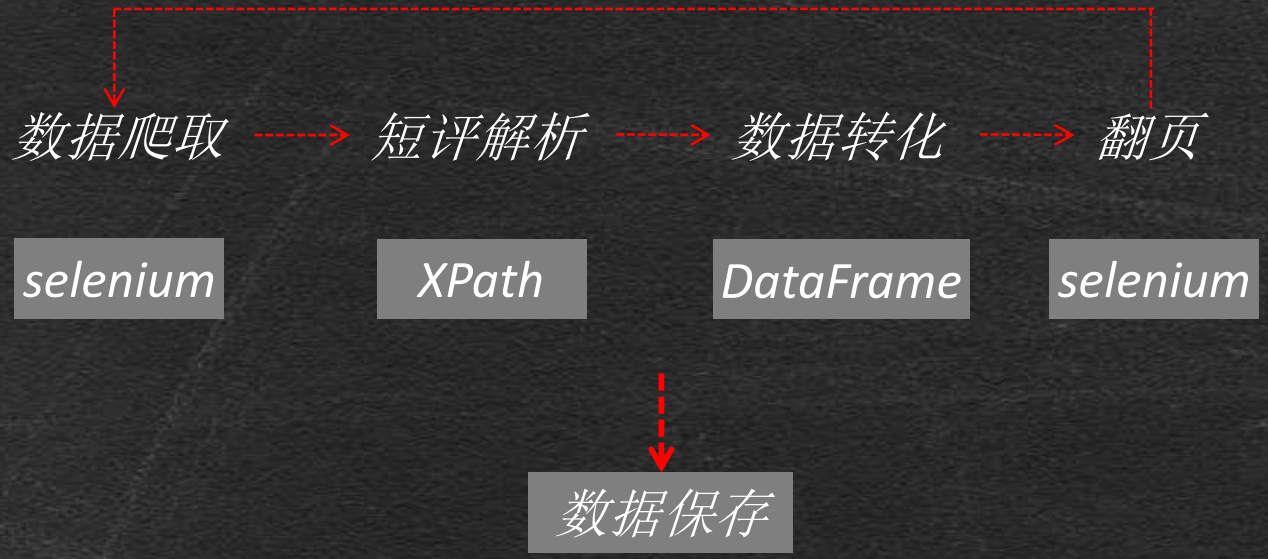

重难点剖析
XPath
例子介绍
从浅入深
显式等待
实例展示
Selenium翻页
反反爬
生活实例入手：如验证码

重难点解析

加入时间	发表时间	居住城市	用户名	短评正文	评分	赞同数量
2005-07-18加入	2019/2/4 15:56	北京	影志	电影比预期要更恢弘磅礴，晨昏线	40	69706
2008-08-30加入	2019/1/29 20:10	北京	陆支羽	1.终于，轮到我们仰望星空。2.后	50	60852
2018-10-07加入	2019/2/5 0:24	北京	沙雕电影	一个悲伤的故事：太阳都要毁灭，	40	37260
2006-04-13加入	2019/1/28 21:58	北京	艾晨	三星鼓励一下吧，四个字：太儿戏	30	21341
2011-06-28加入	2019/2/5 10:13	湖南长沙	妖孽	野心远远大于能力的作品。大刘小	10	14517
2008-03-12加入	2019/1/20 19:00	北京	乌鸦火堂	华语真正意义上的第一部科幻大片	50	14006
2012-07-29加入	2019/2/5 10:24	江苏南京	说给自己听	求求编剧，人类都快失去地球了，	10	13183
2008-03-04加入	2019/1/29 2:11	北京	张小北	从各个方面来说都是一部好看的类	50	12174

目录

流程梳理清楚



激发学生兴趣

1. 热门电影的数据为例，激发学生兴趣 豆瓣电影
2. 互动性强，学生能够动手获取知名网站（熟悉的网页）信息
3. 学习到的内容立马可以通过程序实现，并有相应结果返回

```
In [4]: # 用户名
        names = dom.xpath('//*[@class="comment-info"]/a/text()')
        names

Out[4]: ['陆支羽',
         '艾晨',
         '影志',
         '乌鸦火堂',
         'frozenmoon',
         '沙雕电影',
```

Thank you!