用 Python 来了解一下《安家》

原创 野客 Python技术 1周前

如果要选一部近期最火的电视剧，一定非《安家》莫属，你可能没有具体看过，但如果你看微博的话一定听过这个名字，这部电视剧多次登上微博热搜榜，好像还有几次冲上了热搜榜首，该剧主要讲述的是关于房产中介卖房的故事，电视剧原名也是叫卖房子的人。

使用 Python 分析这部电视剧，主要包括两个步骤：获取数据和分析数据，数据来源我们选取《安家》的豆瓣评论区数据。

获取数据

豆瓣中《安家》的地址是：`https://movie.douban.com/subject/30482003/`，我们打开看一下，如下图所示：

从图中我们可以直观的看出截止目前有 9 万多人进行了打分，从评分上来看，打三星和四星的人数居多，总体评分 6.2 属于及格分，算是中规中矩吧。

我们把页面向下拉到评论区位置，如下图所示：

我们可以看到目前有 3 万多条评论数据，豆瓣对查看评论数据的限制是：未登录时最多可以查看 200 条，登录用户最多可以查看 500 条，也就是说我们最多可以抓取 500 条评论相关的信息，我们要抓取的数据项包括：用户昵称、星级、评论时间、评论内容，将这些信息抓取后我们再将其存到 csv 文件，代码实现如下：

```
import requests, time, random, pandas as pd
from lxml import etree

def spider():
    url = 'https://accounts.douban.com/j/mobile/login/basic'
    headers = {"User-Agent": 'Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.0; Trident/4.0)'}
    # 安家评论网址，为了动态翻页，start 后加了格式化数字，短评页面有 20 条数据，每页增加 20 条
    url_comment = 'https://movie.douban.com/subject/30482003/comments?start=%d&limit=20&sort=new_sco
    data = {
        'ck': '',
        'name': '自己的用户',
        'password': '自己的密码',
        'remember': 'false',
        'ticket': ''
    }
    session = requests.session()
    session.post(url=url, headers=headers, data=data)
    # 初始化 4 个 list 分别存用户名、评星、时间、评论文字
    users = []
    stars = []
    times = []
    content = []
    # 抓取 500 条，每页 20 条，这也是豆瓣给的上限
    for i in range(0, 500, 20):
```

```
    for i in range(0, 500, 20):
        # 获取 HTML
        data = session.get(url_comment % i, headers=headers)
        # 状态码 200 表是成功
        print('第', i, '页', '状态码：',data.status_code)
        # 暂停 0-1 秒时间，防止 IP 被封
        time.sleep(random.random())
        # 解析 HTML
        selector = etree.HTML(data.text)
        # 用 xpath 获取单页所有评论
        comments = selector.xpath('//div[@class="comment"]')
        # 遍历所有评论，获取详细信息
        for comment in comments:
            # 获取用户名
            user = comment.xpath('.//h3/span[2]/a/text()')[0]
            # 获取评星
            star = comment.xpath('.//h3/span[2]/span[2]/@class')[0][7:8]
            # 获取时间
            date_time = comment.xpath('.//h3/span[2]/span[3]/@title')
            # 有的时间为空，需要判断下
            if len(date_time) != 0:
                date_time = date_time[0]
            else:
                date_time = None
            # 获取评论文字
            comment_text = comment.xpath('.//p/span/text()')[0].strip()
            # 添加所有信息到列表
            users.append(user)
            stars.append(star)
            times.append(date_time)
            content.append(comment_text)
    # 用字典包装
    comment_dic = {'user': users, 'star': stars, 'time': times, 'comments': content}
    # 转换成 DataFrame 格式
    comment_df = pd.DataFrame(comment_dic)
    # 保存数据
    comment_df.to_csv('data.csv')
    # 将评论单独再保存下来
    comment_df['comments'].to_csv('comment.csv', index=False)
```

分析数据

现在数据取到了，我们使用 Python 来对这些数据进行分析一下。

评论数量

首先，我们来统计一下这 500 条数据每天的评论数量，然后利用折线图进行数据展示，代码实现如下：

```
import pandas as pd, matplotlib.pyplot as plt
```

```
csv_data = pd.read_csv('data.csv')
df = pd.DataFrame(csv_data)
df_gp = df.groupby(['time']).size()
values = df_gp.values.tolist()
index = df_gp.index.tolist()
# 设置画布大小
plt.figure(figsize=(10, 6))
# 数据
plt.plot(index, values, label='评论数')
# 设置数字标签
for a, b in zip(index, values):
    plt.text(a, b, b, ha='center', va='bottom', fontsize=13, color='black')
plt.title('评论数随时间变化折线图')
plt.xticks(rotation=330)
plt.tick_params(labelsize=10)
plt.ylim(0, 200)
plt.legend(loc='upper right')
plt.show()
```

看一下效果图：

从图中我们可以看出 2 月 21、22 这两天评论数最多，其中 2 月 21 号为开播日，评论数较多很正常， 2 月 22 号评论数多于开播日，我们大致可以推测是开播后网络等渠道进一步扩散的因素，之后随着时间的推移热度有所下降，评论数量呈下降至相对平稳的趋势。

角色分析

我们接着统计评论区中几个主要角色被提及的次数，然后再利用柱状图进行数据展示，代码实现如下所示：

```
import pandas as pd, jieba, matplotlib.pyplot as plt

csv_data = pd.read_csv('data.csv')
roles = {'姑姑':0, '房似锦':0, '王子':0, '闪闪':0, '老油条':0, '楼山关':0, '鱼化龙':0}
names = list(roles.keys())
for name in names:
    jieba.add_word(name)
for row in csv_data['comments']:
    row = str(row)
    for name in names:
        count = row.count(name)
        roles[name] += count
plt.figure(figsize=(8, 5))
# 数据
plt.bar(list(roles.keys()), list(roles.values()), width=0.5, label='提及次数', color=['g', 'r', 'dodg
# 设置数字标签
for a, b in zip(list(roles.keys()), list(roles.values())):
    plt.text(a, b, b, ha='center', va='bottom', fontsize=13, color='black')
plt.title('角色被提及次数柱状图')
plt.xticks(rotation=270)
```

```
plt.xticks(rotation=270)
plt.tick_params(labelsize=10)
plt.ylim(0, 30)
plt.legend(loc='upper right')
plt.show()
```

看一下效果图：

我们从角色被提及的次数可以大致推测出角色的受欢迎程度。

星级变化

我们接着根据获取数据来看一下这几天星级变化的大致趋势，一天中如果有多条星评我们取其平均值即可，代码实现如下所示：

```
import pandas as pd, numpy as np, matplotlib.pyplot as plt

csv_data = pd.read_csv('data.csv')
df_time = csv_data.groupby(['time']).size()
df_star = csv_data.groupby(['star']).size()
index = df_time.index.tolist()
value = [0] * len(index)
# 生成字典
dic = dict(zip(index, value))
for k, v in dic.items():
    stars = csv_data.loc[csv_data['time'] == str(k), 'star']
    # 平均值
    avg = np.mean(list(map(int, stars.values.tolist())))
    dic[k] = round(avg ,2)
# 设置画布大小
plt.figure(figsize=(9, 6))
# 数据
plt.plot(list(dic.keys()), list(dic.values()), label='星级')
plt.title('星级随时间变化折线图')
plt.xticks(rotation=330)
plt.tick_params(labelsize=10)
plt.ylim(0, 5)
plt.legend(loc='upper right')
plt.show()
```

看一下效果图：

从现有数据来看，《安家》的星级整体维持在 2 星左右，我们可以发现尽管该剧比较热，但观众对该剧的满意并不是很高。

词云展示

最后，我们对所有评论进行词云效果展示，这样可以让我们更加直观的看出评论区哪些词汇出现的频率较高，实现代码

如下所示：

```
from wordcloud import WordCloud
import numpy as np, jieba
from PIL import Image

def jieba_():
    # 打开评论数据文件
    content = open('comment.csv', 'rb').read()
    # jieba 分词
    word_list = jieba.cut(content)
    words = []
    # 过滤掉的词
    remove_words = ['以及', '不会', '一些', '那个', '只有',
                    '不过', '东西', '这个', '所有', '这么',
                    '但是', '全片', '一点', '一部', '一个',
                    '什么', '虽然', '一切', '样子', '一样',
                    '只能', '不是', '一种', '这个', '为了']
    for word in word_list:
        if word not in remove_words:
            words.append(word)
    global word_cloud
    # 用逗号隔开词语
    word_cloud = ', '.join(words)

def cloud():
    # 打开词云背景图
    cloud_mask = np.array(Image.open('bg.jpg'))
    # 定义词云的一些属性
    wc = WordCloud(
        # 背景图分割颜色为白色
        background_color='white',
        # 背景图样
        mask=cloud_mask,
        # 显示最大词数
        max_words=100,
        # 显示中文
        font_path='./fonts/simhei.ttf',
        # 最大尺寸
        max_font_size=80
    )
    global word_cloud
    # 词云函数
    x = wc.generate(word_cloud)
    # 生成词云图片
    image = x.to_image()
    # 展示词云图片
    image.show()
    # 保存词云图片
    wc.to_file('anjia.png')
```

```
49
50 jieba_()
51 cloud()
```

看一下效果图：

总结

本文通过爬取豆瓣中《安家》的评论区数据并对其进行可视化，我们可以大致了解观众对《安家》这部电视大致的评价情况，当然因为我们所获取的样本数量有限，可能或多或少还会与用户实际的评价情况有一点偏差。

【代码获取方式】

识别文末二维码，回复：666

PS： 公号内回复「Python」即可进入 Python 新手学习交流群，一起 **100天计划！**

-END-

Python 技术

关于 Python 都在这里

文章已于2020-03-25修改