目录

一． 系统概述

对评论所含内容进行简单的分析，初步确定所需要抓取的内容。首先，在京东商城手机信息界面的用户评价中含有昵称、会员等级、评价星级，评价内容、手机型号、购买时间等等。

其中通过分析会员等级与购买的关系，可以给为不同会员提供不同的产品做参考。分析评价内容可以大概可以推断出消费者对该产品的态度、以及哪些回复关注度比较高等等。分析购买时间可以了解到消费者集中的购买时间段。**这些分析对商品广告的精准投放以及为消费者提供更个性化的服务提供了重要参考**。

二．所用工具

2.1 运行环境

- Chrome 版本 72.0.3626.109（正式版本）（64 位）
- Python 3.5.2 :: Anaconda 4.2.0 (64-bit)

2.2 前置库

- json
- time
- numpy
- requests
- BeautifulSoup
- fake_useragent

三、数据采集

3.1 确定待采集数据

用户 ID、评论内容、会员级别、点赞数、回复数、评价星级、购买时间、手机型号

3.2 确定采集对象及采集思路

选择按评论数降序排列的手机型号，选择 Apple iPhone 8 Plus(A1864) 64GB 的评论数据进行采集

采集思路如图 3-1 所示

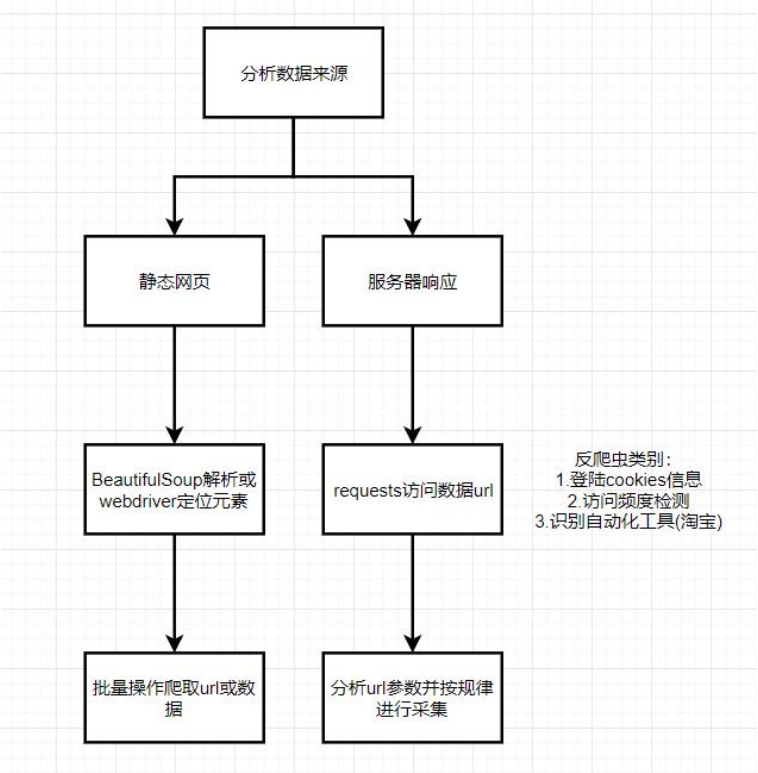

图 3-1 数据采集思路

3.3 采集准备

（robots 协议说明：robots 是网站对爬虫的限定规则，它规定了那些爬虫可以爬，那些数据可以爬）

因此在采集之前，查看京东商城的 robots 协议，如图 3-1 所示

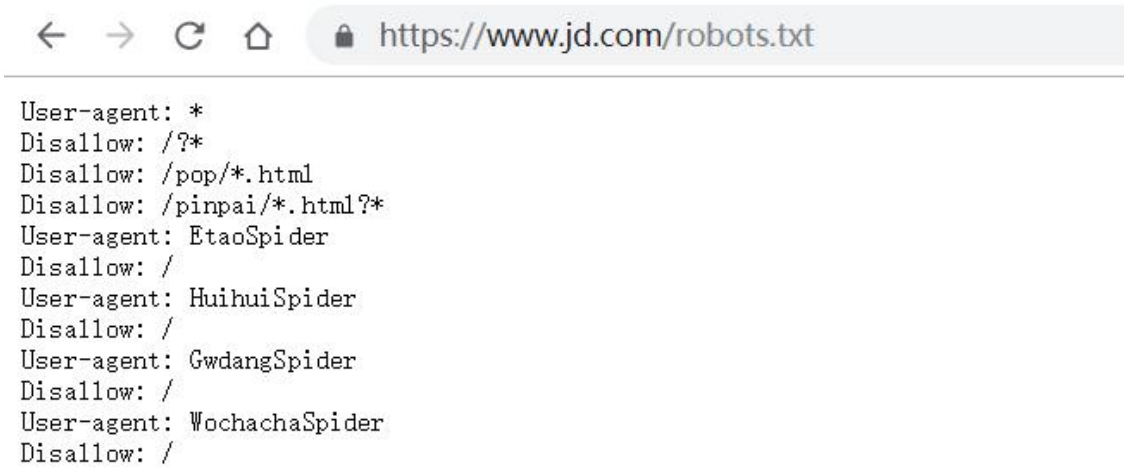

图 3-2 京东网站 robots 协议

参考 robots 协议规则：

`User-agent:` 指定对哪些爬虫生效
`Disallow:` 指定不允许访问的网址
`Allow:` 指定允许访问的网址

通过分析 robots 协议的内容，而下面即将采集的目录在根目录的 comment 的子目录下，不涉及到用户的个人隐私，因此可以继续进行采集。但是在采集过程中，添加 sleep 函数，既为了防止频繁访问 ip 被封，也防止高频度访问对网站带来的负荷。

3.4 开始采集数据

分析网页

首先选择一款评论数目多的手机，按照评论数降序排列，如图 3-3 京东手机评论降序所示。

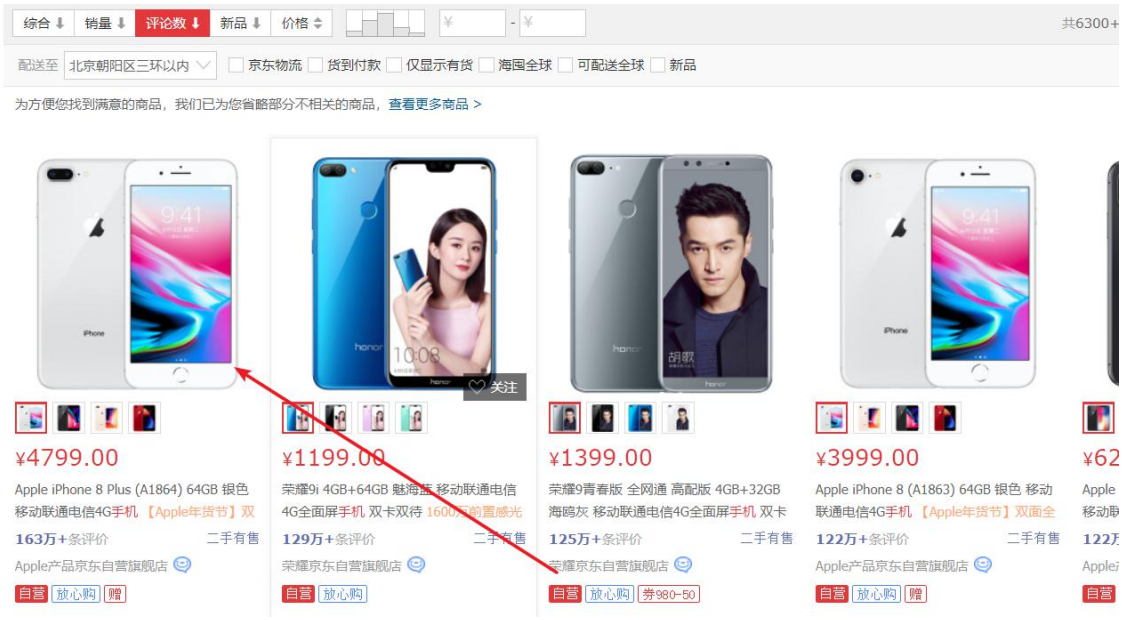

图 3-3 京东手机评论降序

点击进入手机信息页面，在默认手机参数选择下，按 F12 打开调试界面，打开 network 面板并在过滤器中填入“comment”，如图 3-4 Chrome 开发人员工具所示

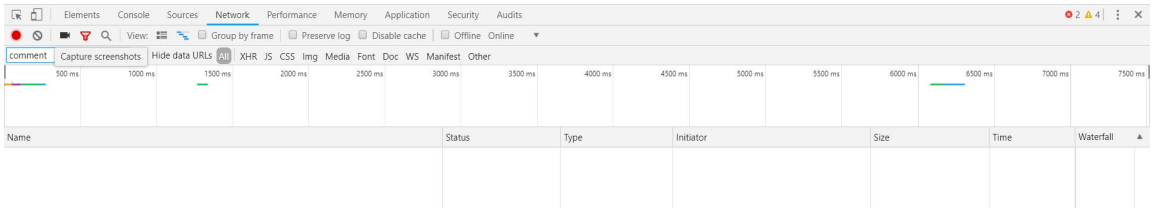

图 3-4 Chrome 开发人员工具

此时，点击商品评价，筛选到如图 3-5 评论数据捕获所示。

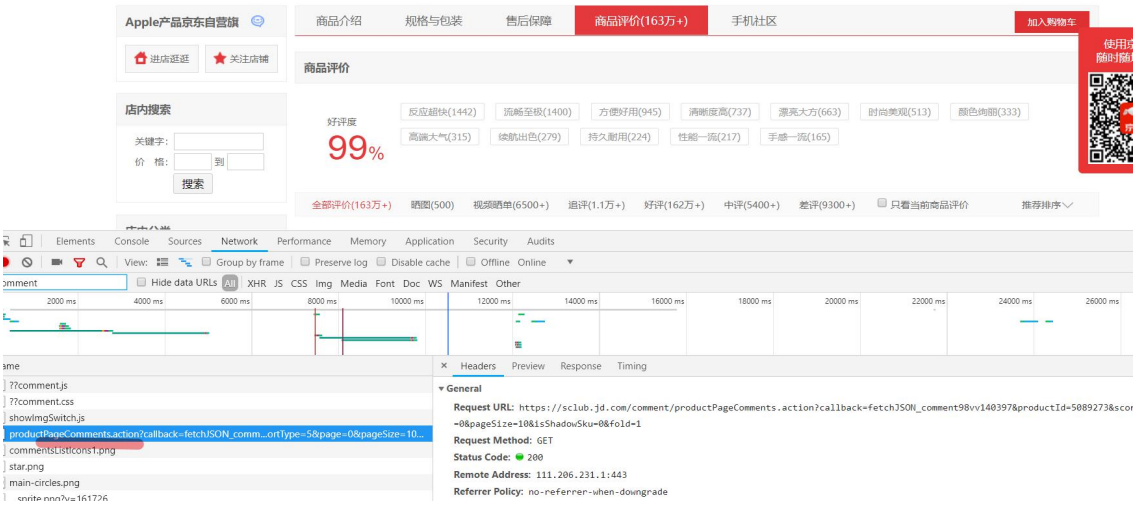

图 3-5 评论数据捕获

含有 pageComment 字段的即为服务器返回的页面评论数据，右键该文件->copy->Copy link address 复制 url 并在 url 地址栏进行访问。访问结果如图 3-6 所示

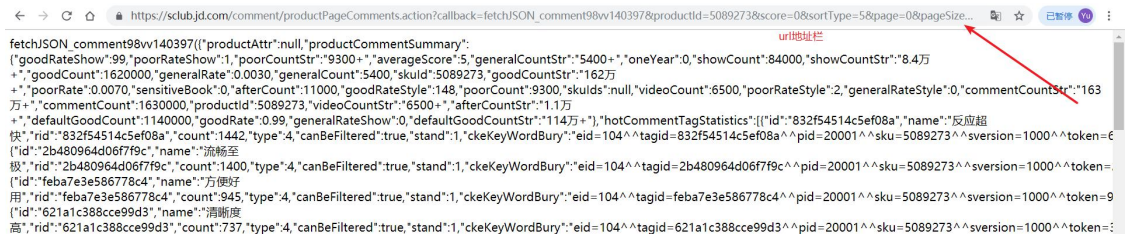

图 3-6 json 评论数据

可以很容易的看到，服务器返回给页面的数据是 JSON 格式的数据。

可以先使用 JSON 在线编辑器进行 json 解析来验证这部分数据，在解析时发生错误，这是由于页面的数据的头部和尾部有一些其他的字符使得页面内容不完全是 json 数据，去掉第一个’(‘以及其之前的字符，同时去掉最后一个’)’以及其之后的字符即可。整理之后的结果如图 3-7 所示

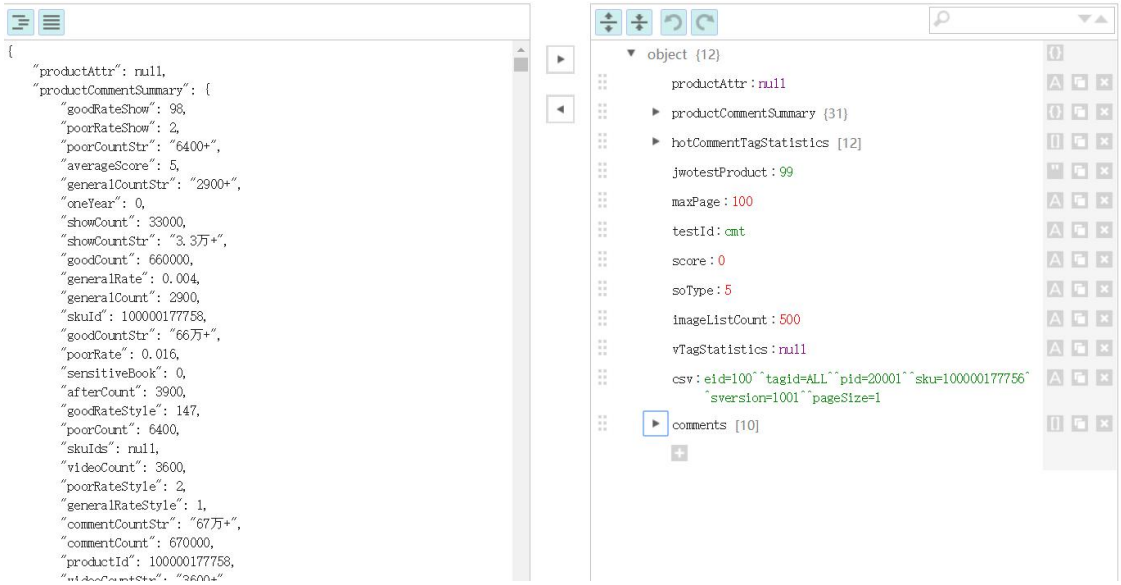

图 3-7 格式化 json 评论数据

从图中可以清楚的看到，评论共有 100 页，每页的 comment 有 10 条。单独点开其中一条评论数据如下图

图 3-8 页面评论数据

可以清楚的看到我们所需要的数据。

那么新的问题来了，京东界面所写评论有 163 万+条，那其他的数据都去哪了？查看一下第 100 页后面，看有没有发现

图 3-9 查看评论区隐藏评价

从图 3-9 中可以看出，还有 114 万+用户给了默认评价，为了分析更准确，加上这部分数据(其实点开也就 100 页，其他的可能服务器就没留着)。按照同样的方式，获取这部分评论的通用 url

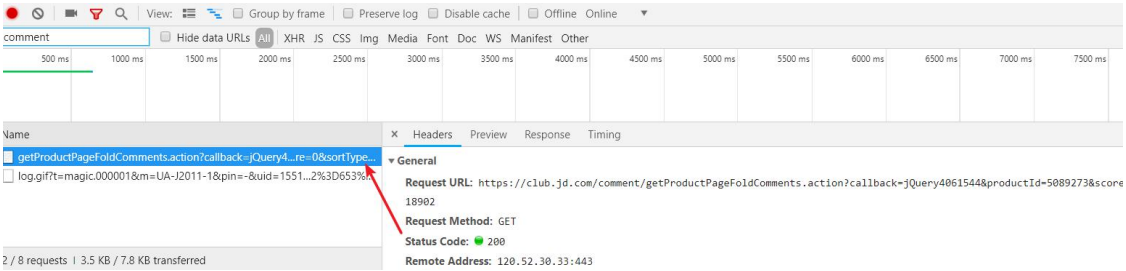

图 3-10 获取隐藏评论 url

尽管如此，也才仅有 1500 条左右的数据，不过在评论的菜单中还有追评、好评、中评、差评等，把这些也算进来，对比比较差异，见图 3-11url 参数分析

图 3-11 url 参数分析（1）

发现不同评价的 score 不同，综合大约有 4000~5000 条数据

继续统计手机不同参数所含的评论。在这里我将手机颜色从“金色”改为了“深空灰色”，按照同样的方式查看 url。并与之前获取的进行对比。如图 3-12

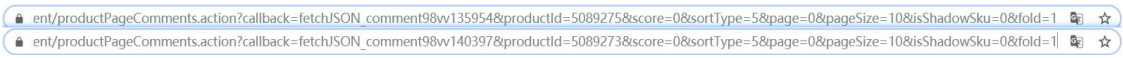

图 3-12 url 参数分析（2）

对比之后发现，不同颜色的产品 ID(productid)发生了变化。但其实评论区域还是各种颜色都有，所以这也是手机评论数据，只不过为了美观，在每次更改手机参数选择时进行了刷新(动态生成)。

根据这些 url 参数，就可以尽可能多的爬取该款手机的评论数据，具体代码请移步 JDComment_Spider(**https://github.com/YuleZhang/JDComment_Spider**)，里面的 SpiderScript 是一个完整的京东评论爬虫脚本，并且采用了随机浏览器和延时访问来防止爬虫被封，为了获取完整的数据，加入了 **try...except** 防止程序中断崩溃。

思考

经过上面的分析，可以看到数据量非常有限，远远没有达到 163 万条。经分析，有以下两种可能：

1. 出现了**数据造假**，这个数字可能是刷出来的（机器或者水军）
2. 真的有这么多的评论，但这时候系统可能只显示其中比较新的评论，而对比较旧的评论进行了存档。

四．数据清洗与分析

4.1 清洗过程简述

在进行数据清洗之前要做的一项工作是先观察数据，看看数据中的哪些部分是合理的，哪些是不合理的，来确定待清洗的部分。使用 python 语句读取的数据如图 4-1

图 4-1 读取评论数据

从图中可以清晰的看出，有大量的用户并未填写评价，这部分数据没有任何用处，并且会影响到我们最后统计的，结果，因此这部分数据需要清空。其次每一列中还存在一些空值，这些空值会严重影响我们的分析判断，并且很容易出现语法错误，因此这部分空值还要进行列格式的统一。

4.2 可视化与分析

这部分主要使用 python 结合 pyecharts 库来实现整体的功能，将上面清洗过的数据接着进行分析。

消费时段分析

按照最初的想法进一步分析下我们现有的数据，首先是购买一天中对不同时间段购买情况的分析，如图 4-2 所示

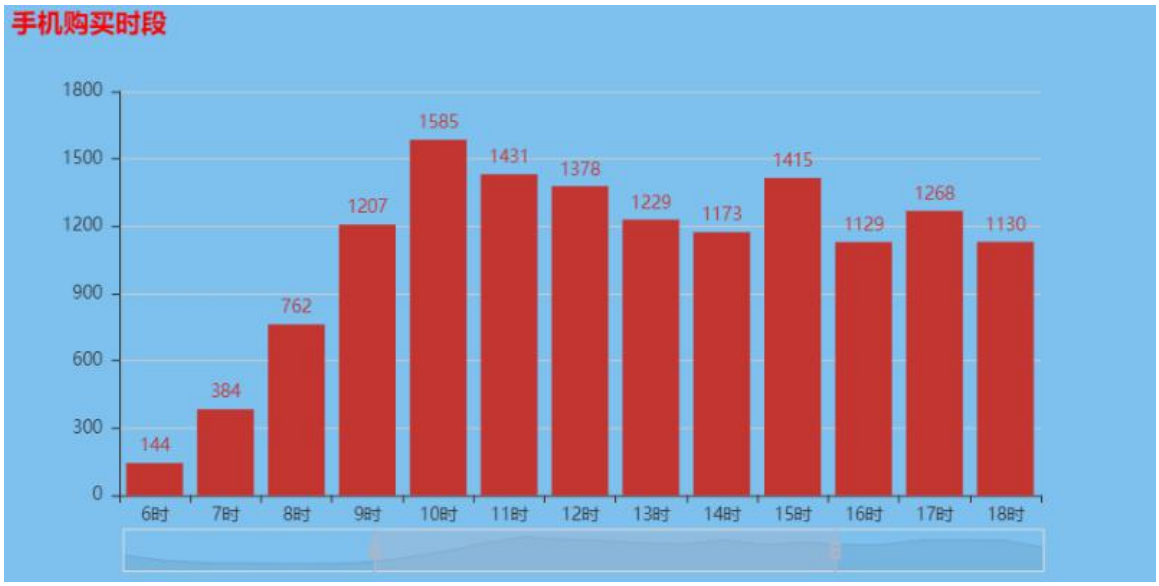

图 4-2 日消费时段分析

分析一天中不同时段消费者的手机购买情况，可以分析得出，大部分消费者在午时（10-12 时）和晚上（20-22 时）出现了消费高峰期，在此时段顾客购买商品的概率更大，他们浏览商品的机会更多。

月消费与会员等级分析

从规格化的数据中看出，会员等级是有限个，但都是字符串形式，只有对它们进行统一编码，才能统计类别个数。猜测会员等级和销售额存在一定的关系，并且通过分析月份和小时能更精确的为消费者提供服务，因此用折线图来表示销量与月份、会员级别的关系是非常有必要的。

绘制出折线图如图 4-3 所示

4-3 月消费与会员等级分析折线图

通过分析上图可知，不论是哪种会员，他们在三月份前后以及十一月份前后购买该款手机的数量最多，三月份换手机的原因推测是新年到来，更新换面的想法。十一月份换手机的原因推测与双十一有关，这时有很大的优惠，也是人们多样化选择手机的时段。

分析不同会员的购买情况可知，他们购买的频次降序排列为 PLUS 会员>金牌会员>银牌会员=钻石会员>PLUS 会员（试用）>企业会员。 分析原因，大致是因为 PLUS 会员大部分都属于高消费群体，他们购买这些价值昂贵的手机概率更大。因此，大致可以推断，这款手机的主要消费对象是京东商城的 PLUS 会员（31.00%）、金牌会员（21.87%）、银牌会员（18.52%）。

评论内容分析

将评论内容的数据整合到一起，采用 jieba 库分词，并使用 wordcloud 生成词云，结果如图 4-4 所示

图 4-4 评论内容词图

通过上面此图基本可以看出消费者对该款手机的整体评价，但其中可能还有一些刷单的情况，这些量无法控制，没法排除，因此还是要理性的看待一款手机综合性能。

五．设计过程中存在的问题和解决过程

问题 1 描述：

评论数据量超出 jupyter 可处理的范围

```
IOPub data rate exceeded.
The notebook server will temporarily stop sending output
to the client in order to avoid crashing it.
To change this limit, set the config variable
`--NotebookApp.iopub_data_rate_limit`.
```

图 5-1 python 数据溢出

解决办法：

重新生成 jupyter 配置文件，将里面的 iopub_data_rate_limit 限制值改一下就能处理，不过处理过程会比较慢。

问题 2 描述

通过采集之后发现京东的评论数据没有达到要求，于是到淘宝上看同款产品的评价进行搜集。如图 5-2 所示

5-2 淘宝评论区采集

能找到包含 json 格式的评论数据，但是使用 python 进行访问时，却没有跳转到应该有的数据界面，而且跳到了其他界面，如下图

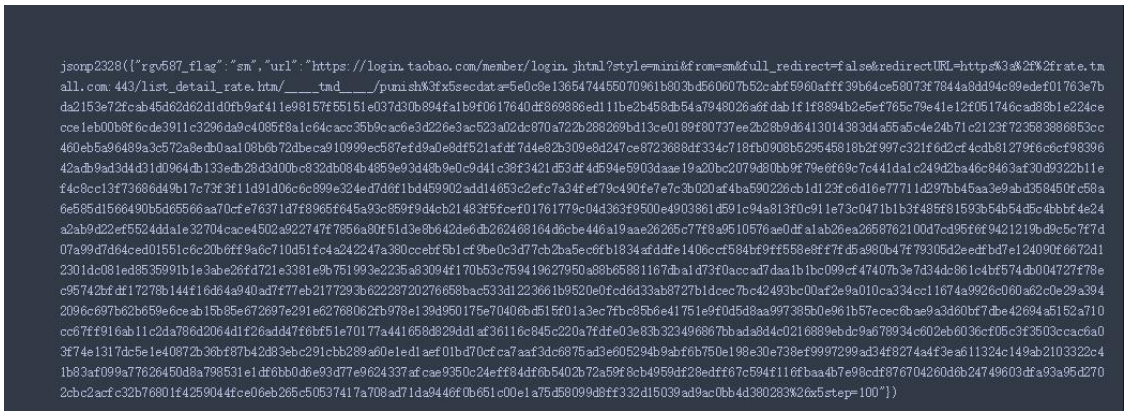

图 5-3 请求网页错误代码

这意味着，没有登陆授权，无法查看评论信息(上面能看到评论信息，也是因为事先登陆过了)，无法获取评论数据。

解决办法：

法 1：python+selenium+webdriver 的探索及问题：

通过一系列的碰壁探索，发现 selenium 无法绕过淘宝登陆界面的验证，淘宝网应该是有识别自动化脚本登陆的反爬机制，根本无法获取登陆状态，无法登陆。

法 2：通过伪造请求头来获取淘宝数据：

登陆淘宝后，在淘宝主界面刷新，来获取登陆的 cookies 信息，如图 3-14 所示

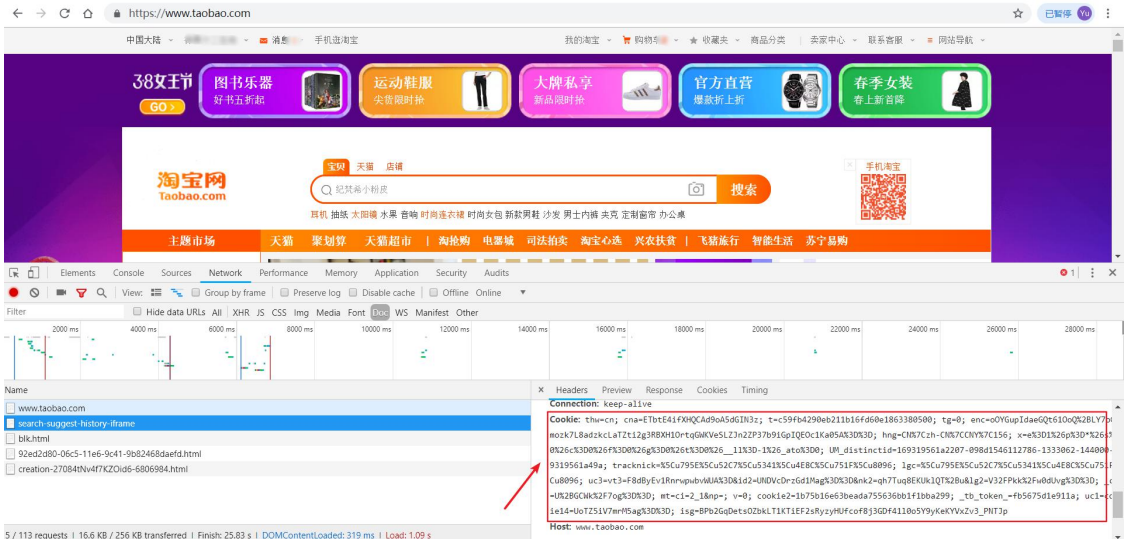

图 5-4 获取登陆 cookies 信息

随后用这部分 cookies 来构造请求头，如图 3-15 所示

```
url='https://rate.tmall.com/list_detail_rate.htm?itemId=558760911386&spuId=877095771&sellerId=2616970884&order=3&

headers={
    'Accept':'text/html, application/xhtml+xml, image/jxr, */*',
    'cookie':"thw=cn; cna=ETbtE4ifXHQCAd9oA5dGIN3z; t=c59fb4290eb211b16fd60e1863380500; tg=0; enc=o0YGupIdaeGQt61
    'user-agent': ua.random,
    'Accept - Encoding':'gzip, deflate',
    'Accept-Language':'zh-Hans-CN, zh-Hans; q=0.5',
    'Connection':'Keep-Alive'
}
requests.get(url=url,headers=headers)
```

图 5-5 构造请求头源码

这样，就能访问淘宝上需要登陆才能获取的页面信息，不过当你尝试就会发现，即使这样，在短时间连续访问多次，也很容易被检测到，从而限制页面的获取数量。我使用该爬虫在不同的时间段爬了六七次，每次 requests 到第 17 次就会出错了，有待完善。源码见 https://github.com/YuleZhang/JDComment_Spider/

六、心得体会

在这次数据库课程设计中，我又重新熟悉了一下 python 来做数据爬取、分析挖掘及可视化的过程，对 requests、pyecharts 等库也有了更多的熟悉，中间还遇到了一些从没见到过的问题，通过一次次的修改，最终呈现出了一个较为完整的版本，当然这次爬取的数据还有一部分没有用上，其实再深层剖析的话，还是有很多可分析空间。不过，自己对数据的理解还很浅，还需要多看看别人的文章进行学习。坦白说，这次课程设计对自己的提高不大，有些新技术还没来的及尝试，课设就结束了，在以后的时间里，我还会多去了解一些新的技术和方法来更有效率的完成学习和工作！

这次课设让我们对数据库编程理解有了更深的了解，我们的课程设计要求是通过网络爬虫，采集京东某一型号手机的评论，根据评论的各类信息，来分析出购买手机顾客的类型和手机的方方面面的使用体验，这次我们使用的爬虫代码和主程序全部采用 python 编写，让我们对新的汇报语言有了更深的了解，懂得了如何使用这种语言去分析数据。

我们还从网上找到了很多可以利用的插件，学会合理利用资源，减少工作量提高工作效率，可以帮我们更精确的表达分析结果。当然更重要的是，通过这次课程设计，我们小组之间的分工合作非常重要，几个人之间相互配合，有效缩短了作业时间，大大提高了工作效率，还使我们学到了从大数据中分析出商品优劣的能力，让我们以后对数据也有了更敏感的思想。美中不足的是我们对于 python 了解比较浅显，在初步学习花费了较多的时间，导致我们后期进度比较紧张，但还是配合组长完美的完成了任务，这次课设经过组长悉心指导使我们受益匪浅。

七、参考文章

1. 搞定 python 多线程和多进程
(https://www.cnblogs.com/whatisfantasy/p/6440585.html)

2. JSON 在线编辑器

(http://www.bejson.com/oldbejson/jsoneditoronline/)

3. Python 爬虫，抓取淘宝商品评论内容

(https://www.cnblogs.com/qun542110741/p/9221040.html)

4. Python 入门网络爬虫之精华版

(https://github.com/lining0806/PythonSpiderNotes)

5. Selenium Python 文档：四、元素定位

(https://www.cnblogs.com/taceywong/p/6602736.html)

6. 为何大量网站不能抓取?爬虫突破封禁的 6 种常见方法

(https://www.cnblogs.com/junrong624/p/5533655.html)

7. 处理 Jupyter Notebook 报错：IOPub data rate exceeded

(https://blog.csdn.net/LaoChengZier/article/details/80705298)