目录

[一． 系统概述 3](#_Toc3116373)

[二．所用工具 3](#_Toc3116374)

[2.1运行环境 3](#_Toc3116375)

[2.2前置库 3](#_Toc3116376)

[三、数据采集 3](#_Toc3116377)

[3.1确定待采集数据 3](#_Toc3116378)

[3.2确定采集对象及采集思路 3](#_Toc3116379)

[3.3采集准备 4](#_Toc3116380)

[3.4开始采集数据 5](#_Toc3116381)

[分析网页 5](#_Toc3116382)

[思考 9](#_Toc3116383)

# 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
* requests
* random
* 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

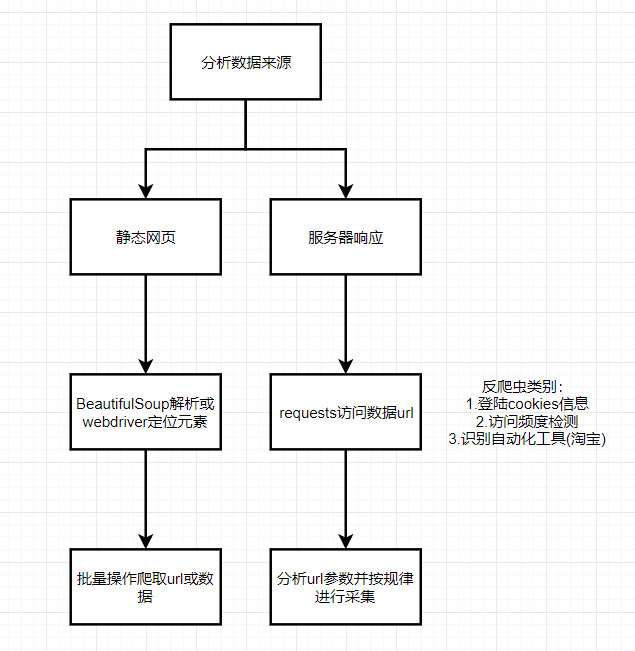


图3-1 数据采集思路

## 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

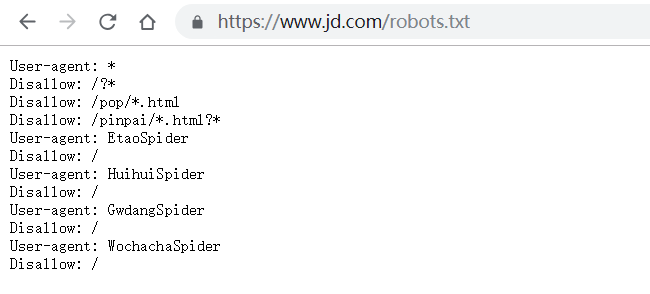


图3-2 京东网站robots协议

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。

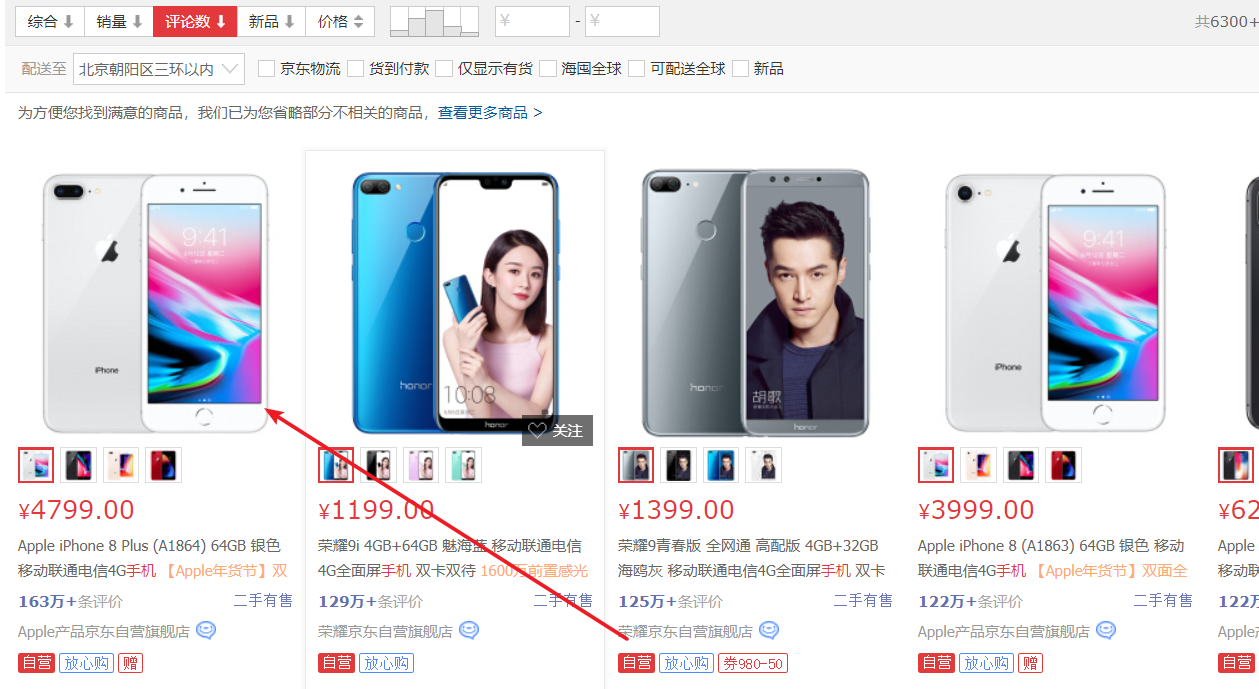


图3-3 京东手机评论降序

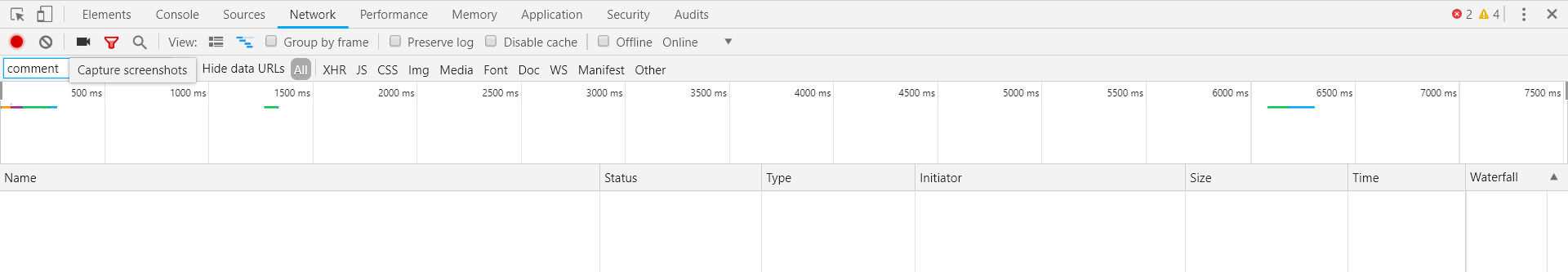
点击进入手机信息页面，在默认手机参数选择下，按F12打开调试界面，打开network面板并在过滤器中填入“comment”，如图3-4 Chrome开发人员工具所示

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。

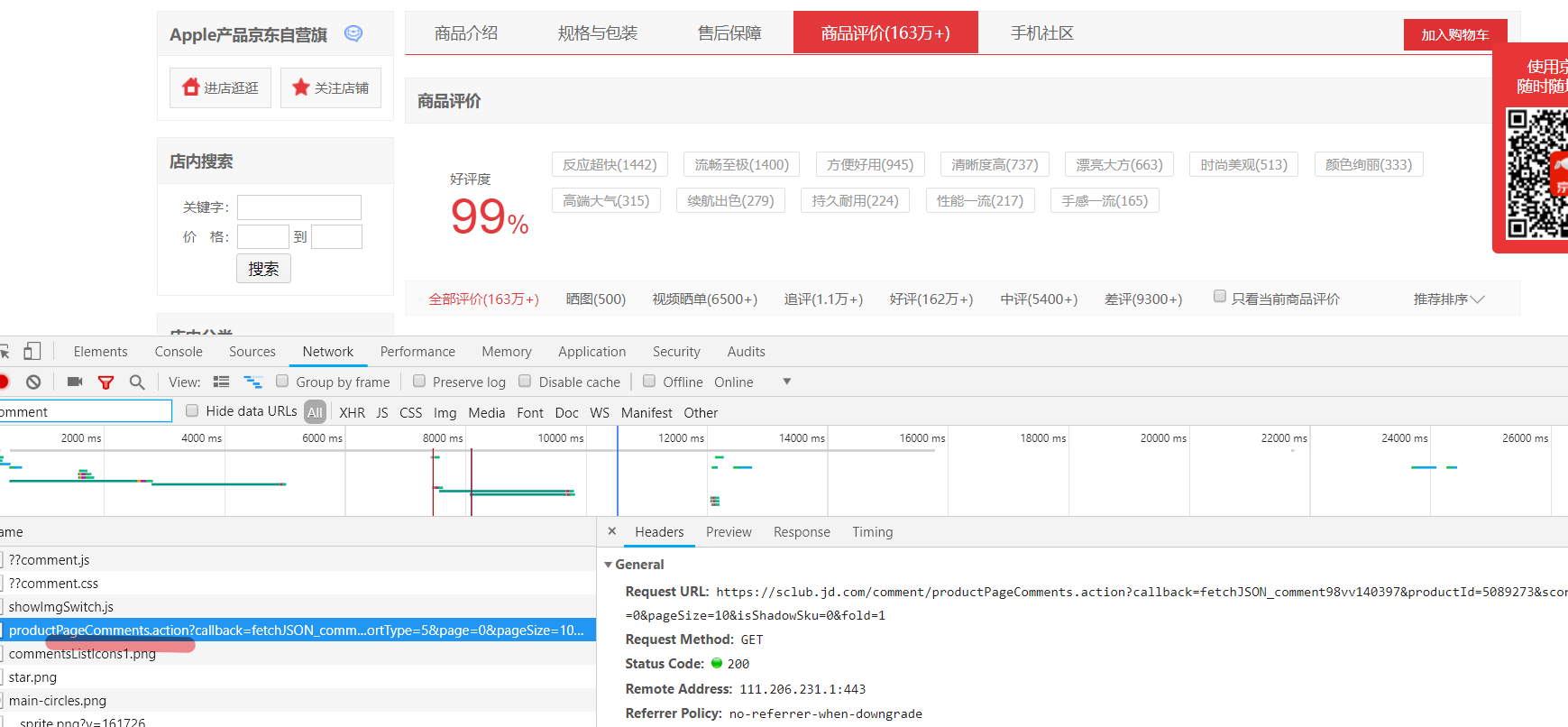


图3-5 评论数据捕获

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

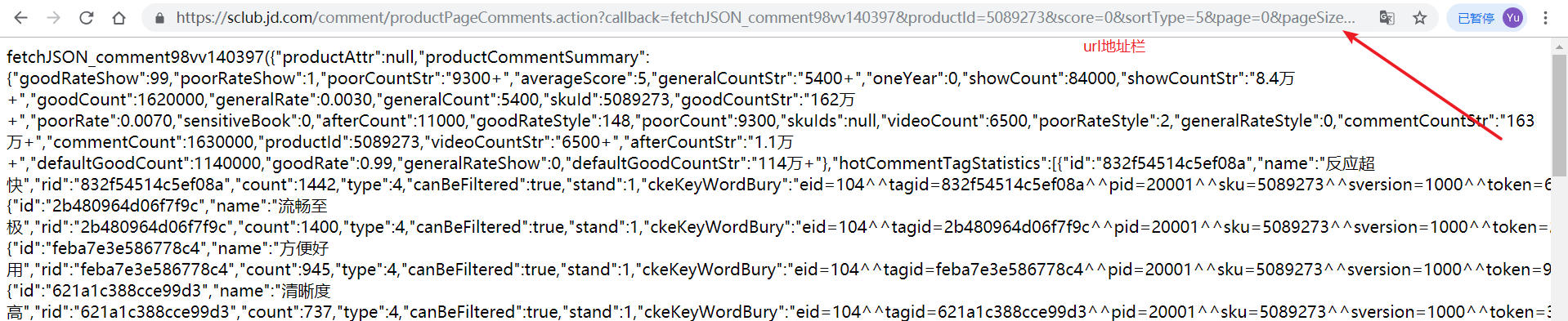


图3-6 json评论数据

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

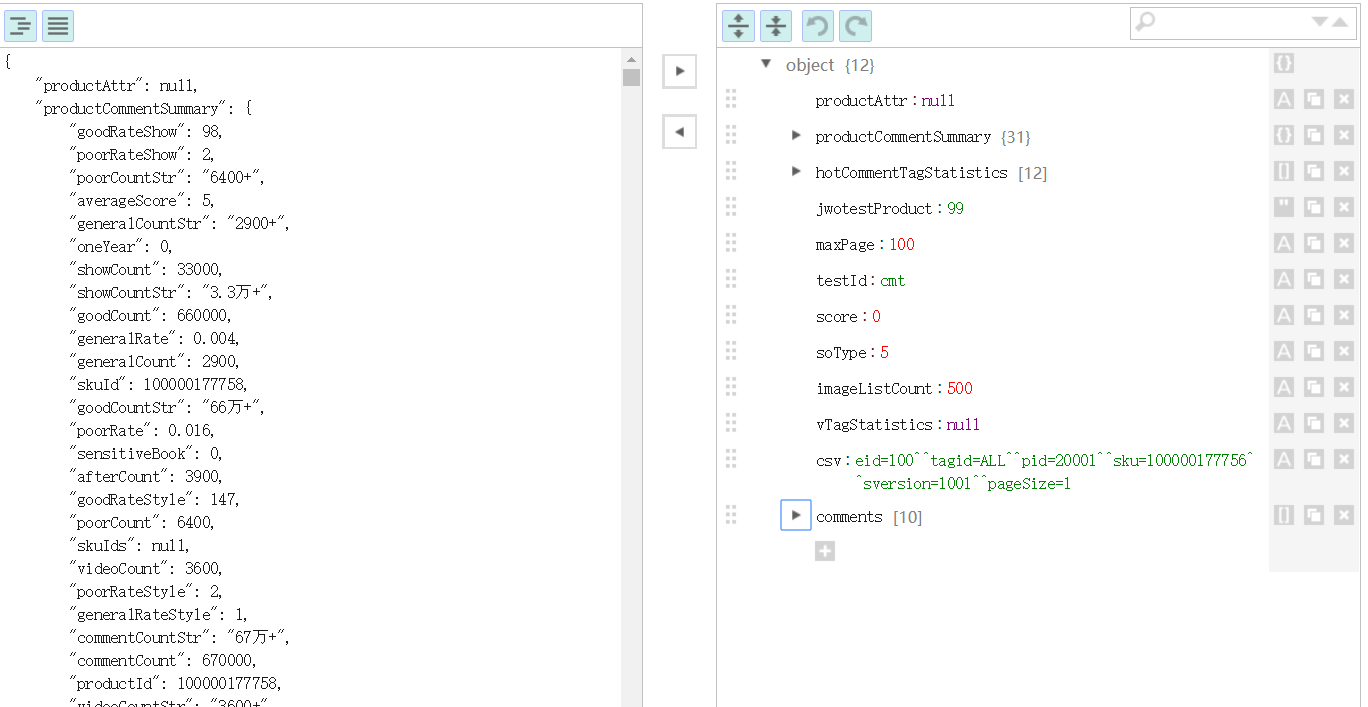


图3-7 格式化json评论数据

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

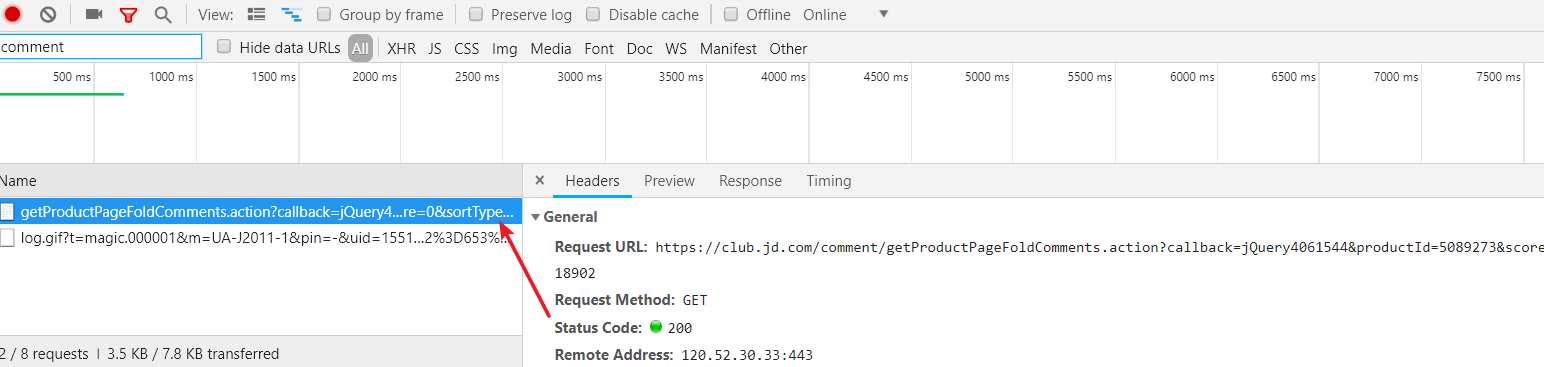


图3-10 获取隐藏评论url

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

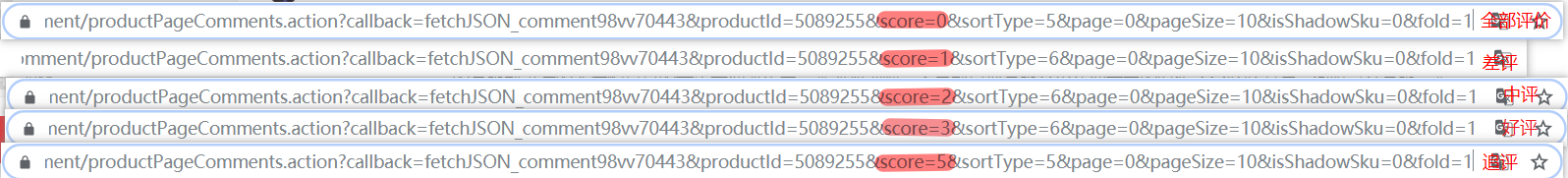


图3-11 url参数分析（1）

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

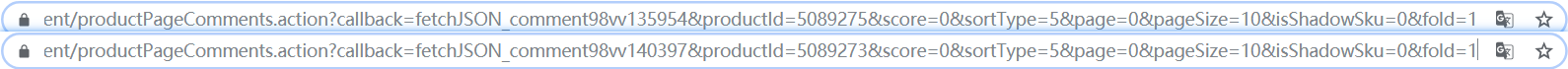


图3-12 url参数分析（2）

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,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

在JDComment\_Processing.ipynb文件中