

# 使用说明

## 数据集部分

数据集在data目录下，里面包含了测试集数据[tmp\\_test\\_data.csv](#)，大约有5000条评论数据，共11款手机，便于演示。同时也包含了完整的数据集[JDComment\\_data](#)，大约2，包含进60款手机的评论数据。[test\\_result.csv](#)是采用完整数据集计算出的各个手机的评论得分，可以用做功能演示。

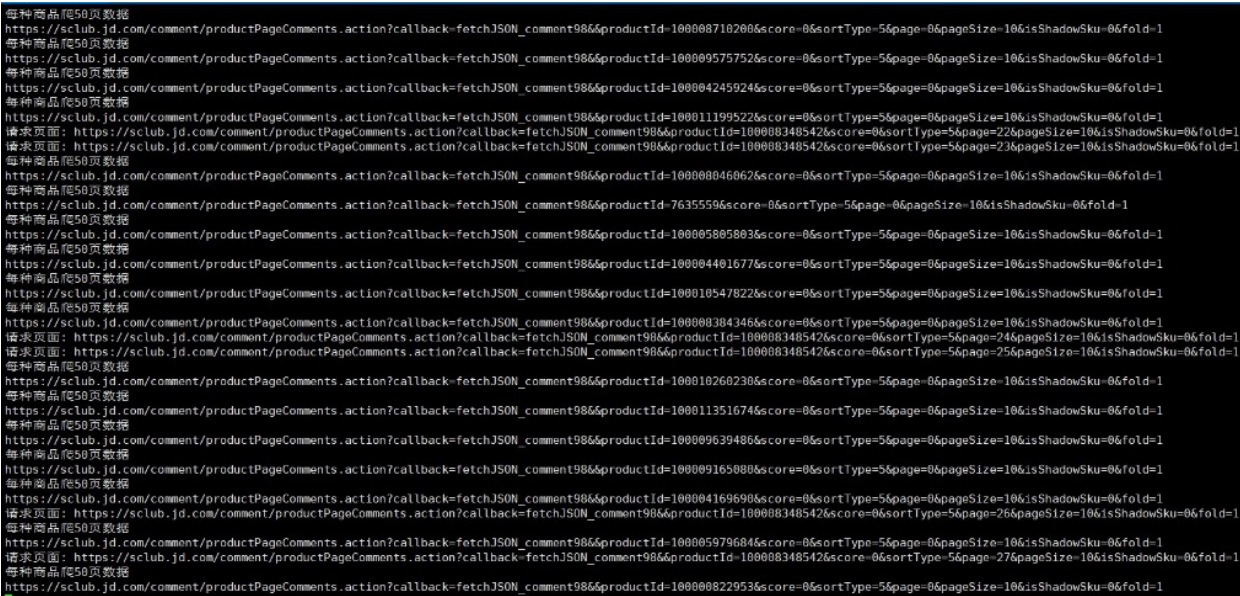
## 评论数据采集

除了已经采集好的数据集，也可以通过脚本SpiderScript重新爬取

在安装好Python，以及配置好pip或conda环境之后，在[当前位置打开cmd控制台](#)(windows)或者在终端输入(Linux)以下语句执行脚本

```
python SpiderScript.py
```

正常执行界面如下图所示，采集完成之后保存到路径data/JDComment\_data中



```
每种商品爬50页数据
https://club.jd.com/comment/productPageComments.action?callback=fetchJSON_comment986&productID=100008710200&score=0&sortType=56page=06pageSize=10&isShadowSku=0&fold=1
每种商品爬50页数据
https://club.jd.com/comment/productPageComments.action?callback=fetchJSON_comment986&productID=100009575752&score=0&sortType=56page=06pageSize=10&isShadowSku=0&fold=1
每种商品爬50页数据
https://club.jd.com/comment/productPageComments.action?callback=fetchJSON_comment986&productID=100004245924&score=0&sortType=56page=06pageSize=10&isShadowSku=0&fold=1
每种商品爬50页数据
https://club.jd.com/comment/productPageComments.action?callback=fetchJSON_comment986&productID=100011199522&score=0&sortType=56page=06pageSize=10&isShadowSku=0&fold=1
请求页面: https://club.jd.com/comment/productPageComments.action?callback=fetchJSON_comment986&productID=100008348542&score=0&sortType=56page=226pageSize=10&isShadowSku=0&fold=1
请求页面: https://club.jd.com/comment/productPageComments.action?callback=fetchJSON_comment986&productID=100008348542&score=0&sortType=56page=236pageSize=10&isShadowSku=0&fold=1
每种商品爬50页数据
https://club.jd.com/comment/productPageComments.action?callback=fetchJSON_comment986&productID=100008046062&score=0&sortType=56page=06pageSize=10&isShadowSku=0&fold=1
每种商品爬50页数据
https://club.jd.com/comment/productPageComments.action?callback=fetchJSON_comment986&productID=76355596&score=0&sortType=56page=06pageSize=10&isShadowSku=0&fold=1
每种商品爬50页数据
https://club.jd.com/comment/productPageComments.action?callback=fetchJSON_comment986&productID=100005805803&score=0&sortType=56page=06pageSize=10&isShadowSku=0&fold=1
每种商品爬50页数据
https://club.jd.com/comment/productPageComments.action?callback=fetchJSON_comment986&productID=100004401677&score=0&sortType=56page=06pageSize=10&isShadowSku=0&fold=1
每种商品爬50页数据
https://club.jd.com/comment/productPageComments.action?callback=fetchJSON_comment986&productID=100010547822&score=0&sortType=56page=06pageSize=10&isShadowSku=0&fold=1
每种商品爬50页数据
https://club.jd.com/comment/productPageComments.action?callback=fetchJSON_comment986&productID=100008384346&score=0&sortType=56page=06pageSize=10&isShadowSku=0&fold=1
请求页面: https://club.jd.com/comment/productPageComments.action?callback=fetchJSON_comment986&productID=100008348542&score=0&sortType=56page=246pageSize=10&isShadowSku=0&fold=1
请求页面: https://club.jd.com/comment/productPageComments.action?callback=fetchJSON_comment986&productID=100008348542&score=0&sortType=56page=256pageSize=10&isShadowSku=0&fold=1
每种商品爬50页数据
https://club.jd.com/comment/productPageComments.action?callback=fetchJSON_comment986&productID=100010260230&score=0&sortType=56page=06pageSize=10&isShadowSku=0&fold=1
每种商品爬50页数据
https://club.jd.com/comment/productPageComments.action?callback=fetchJSON_comment986&productID=100011351074&score=0&sortType=56page=06pageSize=10&isShadowSku=0&fold=1
每种商品爬50页数据
https://club.jd.com/comment/productPageComments.action?callback=fetchJSON_comment986&productID=100009639486&score=0&sortType=56page=06pageSize=10&isShadowSku=0&fold=1
每种商品爬50页数据
https://club.jd.com/comment/productPageComments.action?callback=fetchJSON_comment986&productID=100009165080&score=0&sortType=56page=06pageSize=10&isShadowSku=0&fold=1
每种商品爬50页数据
https://club.jd.com/comment/productPageComments.action?callback=fetchJSON_comment986&productID=100004169690&score=0&sortType=56page=06pageSize=10&isShadowSku=0&fold=1
请求页面: https://club.jd.com/comment/productPageComments.action?callback=fetchJSON_comment986&productID=100008348542&score=0&sortType=56page=266pageSize=10&isShadowSku=0&fold=1
每种商品爬50页数据
https://club.jd.com/comment/productPageComments.action?callback=fetchJSON_comment986&productID=100005979684&score=0&sortType=56page=06pageSize=10&isShadowSku=0&fold=1
请求页面: https://club.jd.com/comment/productPageComments.action?callback=fetchJSON_comment986&productID=100008348542&score=0&sortType=56page=276pageSize=10&isShadowSku=0&fold=1
每种商品爬50页数据
https://club.jd.com/comment/productPageComments.action?callback=fetchJSON_comment986&productID=10000822953&score=0&sortType=56page=06pageSize=10&isShadowSku=0&fold=1
爬取完成，数据已保存到data/JDComment_data中
```

**注意！**：保存的JDComment\_data文件格式为 `utf-8-sig`，需要打开后重新保存一下，转换成 `utf-8` 就可以正常操作了

## 评论数据情感分析并计算得分

tmp.py和comment\_analysis\_process都是计算情感分析的python源码，区别只是在于文件格式不同而已，可以根据不同的环境采用不同的脚本。

tmp.py运行方式为在控制台输入 `python tmp.py`

同时也可以使用 `python tmp.py -h` 查看并修改默认参数，示例如下(指定密码为1)

```
C:\Users\Administrator\Project-py\NLP_OutSourcing_Project>python tmp.py -h
Usage: tmp.py [options]

Options:
  -h, --help            show this help message and exit
  -u USER, --jobs=USER  mysql用户名, 默认为root
  -p PASSWORD, --pass=PASSWORD
                        mysql密码, 默认为空
  -d DATABASE, --Database=DATABASE
                        数据库名称
  -t TABLE, --table=TABLE
                        表名
  -o OUTPUT, --output=OUTPUT
                        保存路径, 默认为data//result.csv

C:\Users\Administrator\Project-py\NLP_OutSourcing_Project>python tmp.py -p '1'
Index(['用户ID', '评论内容', '会员级别', '点赞数', '回复数', '得分', '价格', '购买时间', '手机型号', '销量'], dtype='object')
预处理结束
input_comment目录已经清空
Apple_iPhone_11写入成功
荣耀V30_PRO_李现同款写入成功
```

comment\_analysis\_process可以用jupyter notebook或其他ipython IDE打开执行，内容同tmp.py一样

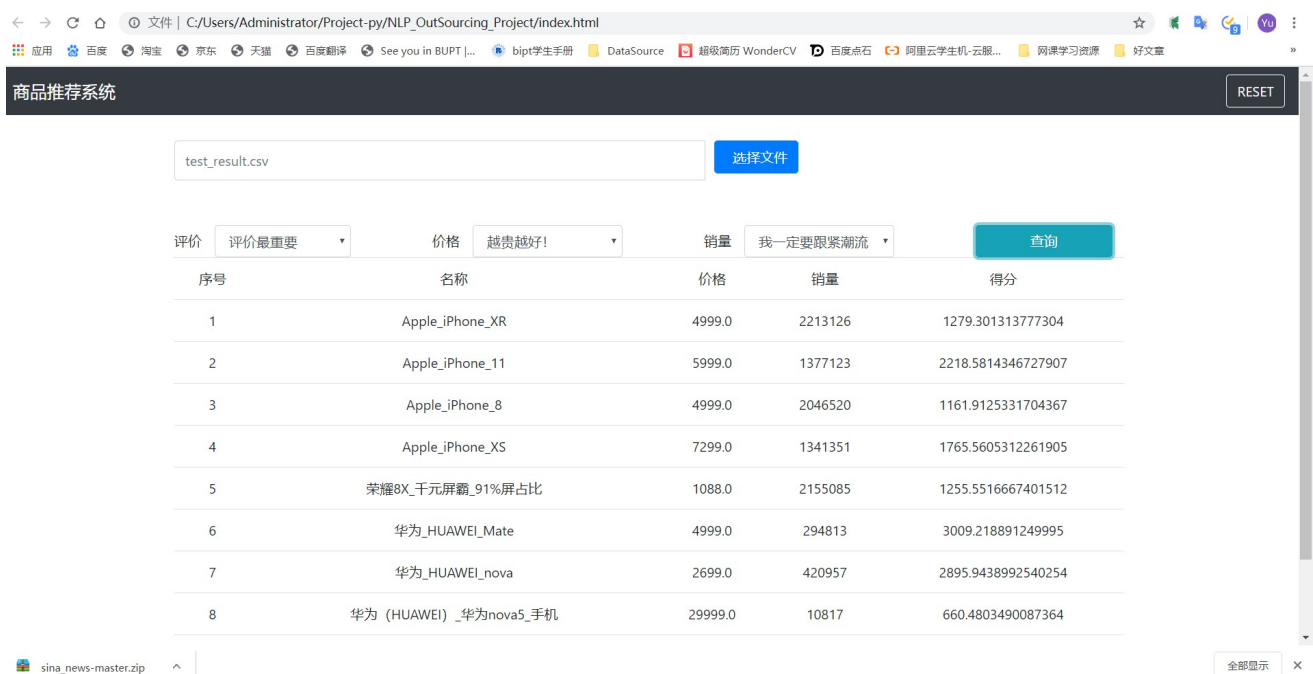
执行之后会将相同手机的评论整合到同一个TXT文件中，以手机名称命名，在input目录下。同时也能自动读取这些文件对其处理求评论得分，包含评论得分的完整手机信息默认存储在[data/result.csv](#)目录下。

## 关于数据库存储

在配置好本地机器Mysql环境之后，在控制台进行测试，详见[mysql配置环境变量 \(win 10\)](#)。tmp.py脚本执行后，会自动保存到数据库jd\_comment中（存储result.csv表）

## 页面测试

双击打开目录下的index.html，上传data目录下的result数据集进行测试即可，只能在本地演示。演示效果如下图



## 版本控制

为了便于管理和维护，我已将项目提交到[https://github.com/YuleZhang/NLP\\_Analysis\\_IDcomment](https://github.com/YuleZhang/NLP_Analysis_IDcomment)，可以自行下载查阅。

## 部分参考

[python实现多线程爬虫](#)

[MySQL的python连接](#)

[用python实现文本情感分析](#)

[optparse模块](#)