# 社会计算期末大作业

姓名：阮翔宇　　学号：2022302181257

## 一、实验简介

研究对象 ： 电影《铃芽之旅》　　评论网站：bilibili.com

代码环境 ： python 3.12.x

## 二、爬虫技术简介以及数据去重

采用了 request 库发送请求，通过 cookie 保持登陆状态。

通过分析评论网站的网络包可以看出来评论展示页是一个动态网页， 当用户向下滑动页面时，javascript 通过 Ajax 技术向服务器请求内容，更新网页数据。在浏览器开发者工具中查看网络请求，查看 xhr 类文件，每次下滑时，服务器都会发送一个 xhr 类响应，文件内容是 json 格式当。请求 url[1]通过加入 cursor 参数从服务器中获取后续参数。观察 url 发现每次请求时 cursor 变化没有规律，考虑是通过加密生成的参数。但观察返回的 json 内容后发现，下一次下滑的 cursor 参数由 josn 中的一个 next 项给出。于是可以通过上一响应返回的内容递归地向服务器请求数据。

将返回的 text 通过 json.loads()处理为 json 类，用关键字提取表中数据。提取了以下几项数据

- uname ： 评论用户名
- uid ： 评论用户账号的唯一 id
- ulevel ： 评论用户账号等级，用户账号使用时间越长，等级越高。共 6 级
- comment ： 用户评论内容
- title ： 用户评论内容的标题
- score ： 用户打分

将得到的数据整合存入 csv 文件。

在进行数据处理时，将 csv 文件读入为 pandas 库中的 dataframe 结构。利用类的 drop_duplicates 方法以 comment 列为标准对数据进行去重。为了提高数据的有效性，还可将评论字数于 5 的内容丢弃，然后将行号重新排序。

爬虫爬取了 b 站评论中的所有评论，共 5694 条短评，433 条长评。经过去重与丢弃低价值数据，共丢弃了 974 条短评，1 条长评。

## 三、数据分析与情感分析

情感分析采用 snownlp 库, 其中 SnowNLP 对象的 sentiment 方法可以进行情感分析。

---

[1]https://api.bilibili.com/pgc/review/short/list?media_id=28370944&ps=20&sort=0'

SnowNLP 的情感分析基于朴素贝叶斯分类器，这是一种基于贝叶斯定理的分类方法。朴素贝叶斯分类器假设特征之间是独立的，即一个特征或者单词出现的可能性并不依赖于其他特征或单词是否出现。

SnowNLP 的情感分析模块预先训练了一个朴素贝叶斯模型，该模型在大量的带有情感标签的文本数据上进行训练，学习到每个词语对情感的影响。在进行情感分析时，SnowNLP 会将输入的文本分词，然后利用训练好的模型计算每个词语的情感概率，最后将这些概率综合起来，得到整个文本的情感倾向。

SnowNLP 的情感分析结果是一个介于 0 到 1 之间的浮点数，数值越接近 1，表示文本的情感越积极，数值越接近 0，表示文本的情感越消极。

由于采用机器学习的方法进行情感分析，该方法在对一些短句子的预测上正确率偏低，如“好看！！！”会被识别为消极词，且 snownlp 中的模型并不是针对影评训练的，原始数据采用的是商品交易。于是利用 300 百万条豆瓣评论对模型进行了重新训练（模型参数在 data_newmodel 中）。但重新训练的模型对整体长句子不如初始模型，总体评价分数的偏差更大。两种模型的预测如下表

| | actual_score | predict_score |
|---|---|---|
| initial model | 8.223689320388349 | 7.477090705626724 |
| new model | 8.223689320388349 | 6.966182309587365 |

根据资料显示初始参数的正确率约为 80%，偏差一部分可能是模型本身造成的。

并且考虑到 bilibili 平台打分只能是 2 的倍数所以用户打分在十分制下存在正负一点的误差是正常现象。

bilibili 平台自己统计的打分均值是 9.8，而我通过爬取所有评论得到的均值数据是 8.2，所以 bilibili 平台统计分数是不真实的。

豆瓣平台的统计评分是 7.2 分与模型通过 bilibili 平台评论预测的结果更加接近。豆瓣对单部电影的评论访问条数限制为 500 条，所以无法通过爬取所有评论统计平台数据的真实性。但其结果与 bilibili 平台中的预测结果接近，可以假设豆瓣的数据是真实的，更能反应网络舆论。

## 四、正向和反向关键词提取。

首先尝试了利用 snownlp 自带的关键词提取，但效果不佳，得到的多为副词，停止词。且 snownlp 没有词性分析功能。后采用 jieba 库进行分词处理，利用 jieba 库中的 analyse 模块进行分析。

jieba 库的关键词提取基于 TF-IDF 算法。TF-IDF 是一种统计方法，用以评估一个词语对于一个文件集或一个语料库中的其中一份文件的重要程度。

TF-IDF 算法的主要思想是：如果某个词或短语在一篇文章中出现的频率 TF（Term Frequency，词频）高，并且在其他文章中很少出现，则认为此词或者短语具有很好的类别区分能力，适合用来分类。

TF-IDF 实际上是两个部分：TF 和 IDF。

TF (Term Frequency) 表示词条在文本中出现的频率。这个数字是对词数(term count)的归一化，防止它偏向长的文件。（同一个词在长文件里可能会比短文件有更高的词数，即便它在两个文件中占据的比例是一样的）

IDF（Inverse Document Frequency）表示词条的通用性。计算公式是以词条的文档频率为底的对数。

jieba 库的 extract_tags 方法就是基于 TF-IDF 算法来提取关键词的。

jiaba 库但 extract_tags 可以通过 allowpos 参数来选取获得关键词但词性

在提取关键词时通过 extract_tags 获取了两个名词，三个形容词。在数据处理上，将不同 sentiment 为界限，划分正向句子和反向句子，组合成一个 string，提取了两个 string 中的关键词。sentiment 越大，特征提取选择的评论越激进。

| sentiment rate | pos keywords | neg keywords |
|---|---|---|
| 0.9 | 剧情 女主 不错 优秀 很棒 | 剧情 巨人 特效 生硬 无聊 |
| 0.9 | 剧情 女主 不错 优秀 很棒 | 剧情 电影院 特效 生硬 无聊 |
| 0.7 | 剧情 电影 不错 很棒 优秀 | 剧情 电影院 特效 无聊 成功 |
| 0.5 | 剧情 电影院 特效 无聊 成功 | 剧情 电影院 无聊 不错 清楚 |

第一个数据中出现了“巨人”这一与剧情无关的要素，经检测是一条恶意评论将巨人重复了 21 次。手动去除后计算第二个结果。

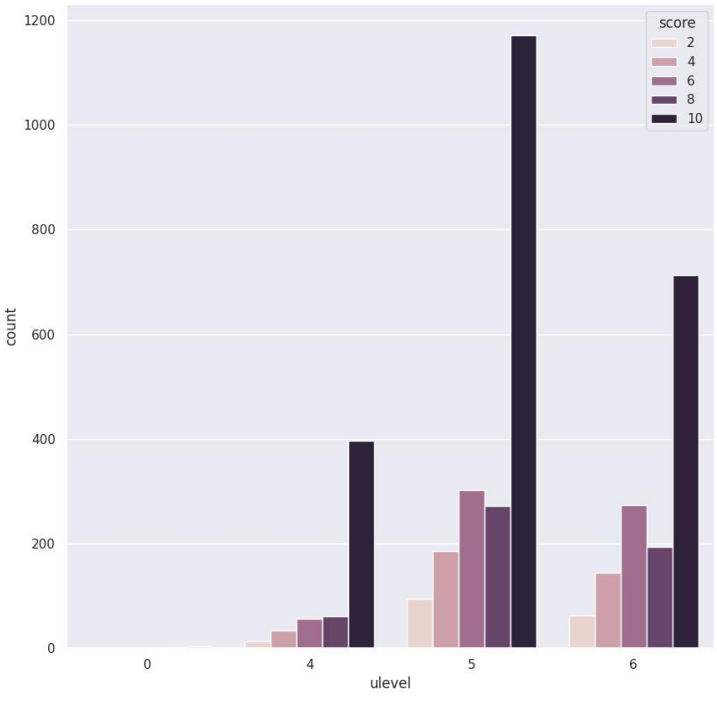
可以看到关注的正向特征和反向特征是相一致的，都是剧情和电影院，区别在于评价形容词。

作为一个整体评分达到 8 分以上的电影，提取的负面特征句子不多，负面特征词和实际可能会存在偏差。

## 五、评论用户特征，水军分析

通过分析用户等级与打分的分布，查看是否存在批量的水军僵尸账号或其他异常。

以 ulevel 为横坐标，score 为权重，数量为高度，做图如下。



可以看到评论用户在各个等级下分布均匀，且均为高等级用户，不存在明显的水军行为。

uid_counts = df['uid'].value_counts()通过这个方法计算每个 uid 出现的次数，统计无重复评论的情况。