|  |  |
| --- | --- |
| **报告成绩** |  |
| **演示成绩** |  |
| **答辩成绩** |  |
| **总评** |  |

**中北大学**

**课 程 设 计 说 明 书**

数据可视化课程设计

|  |  |  |  |
| --- | --- | --- | --- |
| **学生姓名:** | **宋津京** | **学 号：** | **1907040446** |
| **学 院:** | **大数据学院** | | |
| **专 业:** | **数据科学与大数据技术** | | |
| **题 目:** | **VSAS综艺节目分析系统** | | |
|  |  | | |
| **指导教师：** | **耿艳兵、崔方舒** | | |

2022年 11 月24日

摘 要

弹幕和评论是年轻人对于视频作品交流的一种体现，在青年用户占比较多的bilibili视频网站更是将弹幕作为其核心进行推广，因此想要对于一款节目进行系统的分析得出改进意见，对于评论及弹幕的数据分析和情感分析是机器有必要的。

本次课程设计的VSAS（Variety Show Analysis System）综艺节目分析系统，旨在通过对bilibili视频网站中某一综艺节目（本项目中选择的节目为bilibili@黄金眼）的弹幕及评论数据进行网络爬虫爬取，进行相关的数据处理、数据存储和数据可视化，结合相关图表对多种元素进行有关的联系分析，系统的认识节目本质，发现节目规律并分析推理得到未来趋势。

本项目采用Django技术实现可视化的网站框架并利用Bootstrap技术进行前端页面渲染，网络爬虫技术采用基础的python-request库结合正则表达式爬取数据，数据清洗和处理方面选择pandas、json等python库，情感分析库使用snownlp，数据存储使用Django框架默认的sqlite3轻量级关系型数据库，可视化技术使用Apache Echarts。

项目涉及环境及版本：

Python – 3.7

Django – 3.2.16

Pandas – 1.3.5

Json – 4.17

snownlp – 0.12.3

SQLite – v3

Bootstrap – 3.4.1

Echarts – 5.4.0

目 录

[1 概述 - 1 -](#_Toc119961777)

[1.1 设计内容 - 1 -](#_Toc119961778)

[1.2 意义 - 1 -](#_Toc119961779)

[1.3 设计要求 - 2 -](#_Toc119961780)

[1.4 工作要求 - 2 -](#_Toc119961781)

[2 需求分析 - 3 -](#_Toc119961782)

[2.1 项目概要 - 3 -](#_Toc119961783)

[2.1.1 网站架构概览 - 3 -](#_Toc119961784)

[2.1.2 数据采集 - 3 -](#_Toc119961785)

[2.1.3 数据清洗和预处理 - 4 -](#_Toc119961786)

[2.1.4 数据分析与挖掘 - 5 -](#_Toc119961787)

[2.1.5 数据可视化 - 5 -](#_Toc119961788)

[2.2 功能化模块需求 - 6 -](#_Toc119961789)

[2.2.1 登录功能 - 6 -](#_Toc119961790)

[2.2.2 表的预览 - 7 -](#_Toc119961791)

[3 概要设计 - 8 -](#_Toc119961792)

[3.1 系统整体设计 - 8 -](#_Toc119961793)

[3.2 数据库设计 - 8 -](#_Toc119961794)

[3.2.1 用户表设计 - 8 -](#_Toc119961795)

[3.2.2 用户项目表 - 9 -](#_Toc119961796)

[3.2.3 用户数据表 - 9 -](#_Toc119961797)

[3.3 模块说明 - 9 -](#_Toc119961798)

[3.3.1 描述性统计 - 9 -](#_Toc119961799)

[3.3.2 情感分析 - 10 -](#_Toc119961800)

[3.3.3 其他 - 10 -](#_Toc119961801)

[3.4 界面设置 - 11 -](#_Toc119961802)

[3.4.1 登录界面 - 11 -](#_Toc119961803)

[3.4.2 工作台界面 - 11 -](#_Toc119961804)

[3.4.3 数据分析界面 - 12 -](#_Toc119961805)

[3.4.4 情感分析界面 - 12 -](#_Toc119961806)

[3.4.5 表预览界面 - 13 -](#_Toc119961807)

[4 详细设计 - 14 -](#_Toc119961808)

[4.1 数据采集 - 14 -](#_Toc119961809)

[4.1.1 爬取b站视频评论信息 - 14 -](#_Toc119961810)

[4.1.2 爬取b站视频弹幕信息 - 15 -](#_Toc119961811)

[4.2 数据处理 - 15 -](#_Toc119961812)

[4.3 数据分析 - 16 -](#_Toc119961813)

[4.3.1 一般数据分析 - 16 -](#_Toc119961814)

[4.3.2 情感分析 - 18 -](#_Toc119961815)

[4.4 数据汇总 - 18 -](#_Toc119961816)

[4.5 数据可视化 - 19 -](#_Toc119961817)

[5 可视化结果分析 - 20 -](#_Toc119961818)

[5.1 描述性统计 - 20 -](#_Toc119961819)

[5.1.1 评论月分布折线图 - 20 -](#_Toc119961820)

[5.1.2 评论星期分布直方图 - 21 -](#_Toc119961821)

[5.1.3 节目受众占比饼图 - 21 -](#_Toc119961822)

[5.1.4 24小时男女观看统计图 - 22 -](#_Toc119961823)

[5.1.5 等级-点赞关系图 - 22 -](#_Toc119961824)

[5.1.6 热评前十漏斗图 - 23 -](#_Toc119961825)

[5.1.7 热词云图 - 24 -](#_Toc119961826)

[5.1.8 关键词热力图 - 24 -](#_Toc119961827)

[5.2 情感分析 - 25 -](#_Toc119961828)

[5.2.1 弹幕评论情感统计直方图 - 25 -](#_Toc119961829)

[5.2.2 第五集情感变化图 - 26 -](#_Toc119961830)

[5.2.3 平均情感仪表盘 - 26 -](#_Toc119961831)

[5.2.4 情感词云图 - 27 -](#_Toc119961832)

[6 课程设计体会 - 28 -](#_Toc119961833)

[参 考 文 献 - 29 -](#_Toc119961834)

# 概述

## 设计内容

本次课程设计以pycharm为主要开发工具，jupter notebook为测试工具：

1. 准备数据：遵循网络爬虫协议使用网络爬虫对bilibili视频网站《bilibili@黄金眼》节目的弹幕和评论相关信息进行采集，并暂存入csv文件中。
2. 数据预处理：使用python库工具对爬取的数据清洗加工，对必要内容进行转换或添加。
3. 搭建系统：使用Django + Bootstrap搭建一个网站框架，准备接收后端处理好的数据信息。
4. 数据汇总：数据分析数据、情感分析数据汇总到一个JSON文件中并传递到前端。
5. 数据存储：将必要的信息持久的写入到轻量级关系型数据库SQLite中。
6. 数据可视化：利用相关数据和Echarts技术，选择合适的图表进行数据的可视化。
7. 代码复现：对需要的依赖包进行安装，根据详细设计完成相关阶段设计。
8. 总结并书写实验周数据可视化项目说明书。

## 意义

数据可视化的意义是帮助人更好的分析数据，信息的质量很大程度上依赖于其表达方式。对数字罗列所组成的数据中所包含的意义进行分析，使分析结果可视化。其实数据可视化的本质就是视觉对话。数据可视化将技术与艺术完美结合，借助图形化的手段，清晰有效地传达与沟通信息。一方面，数据赋予可视化以价值;另一方面，可视化增加数据的灵性，两者相辅相成，帮助企业从信息中提取知识、从知识中收获价值。

本次数据可视化课程设计的目的在于提升本人对于数据的敏感度，能独立自主的完成一系列的数据可视化系统设计，诸如数据采集、数据处理、数据存储、数据可视化等。本次课程我选题对《bilibili@黄金眼》节目的评论和弹幕进行数据分析和情感分析，并着重实现数据可视化部分。

通过为期两周后的数据可视化课程设计，进一步熟悉大数据系统开发的步骤与实现，掌握软件开发的整体思想，培养大数据开发的综合应用能力是本项目的关键所在。

## 设计要求

（1）深入学习和了解大数据系统体系及相关实现技术；

（2）理解爬虫程序的构建方式并能实际应用；

（3）理解关系型数据库系统的存储方式；

（4）设计并实现数据分析与挖掘模块；

（5）设计并实现合理的可视化结果；

（6）符合课题要求，实现相应功能；

（7）注意程序的实用性、可读性。

## 工作要求

在题目设计初期，需要完成基本的题目分析，并进行思路规划；在题目完成中期，需要具体设计相关算法，进一步优化，并编写程序实现，学习并且制作可视化界面；在题目完成后期，需要对代码进行进一步的整合和规范化，并完成课程设计说明书文档。

# 需求分析

## 项目概要

### ****网站架构概览****

网站功能的基础实现应当包含用户登录和用户注册，用户通过登录之后通过页面展示其能使用的功能，包括但不限于相关表的预览、数据可视化项目的展示。如图2.1.1 网站功能用例图所示：

图示

描述已自动生成

图 2.1.1 网站功能用例图

其中表的预览和项目的可视化为重点实现对象，对于Client端可模拟一个admin用户用于在实现过程中的测试，测试主要包含登录注册测试、主页功能测试、管理测试。

### 数据采集

对bilibili视频的弹幕的爬取，需要访问固定的格式网址：

[https://comment.bilibili.com/{cid}.xml](https://comment.bilibili.com/%7bcid%7d.xml)。

其中cid需要使用特殊的工具进行获取（这里使用bilitool.top网址提供的功能），弹幕的具体格式如下图 2.1.2所示。

可知只需要使用正则表达式，爬取指定的p标签就能获取弹幕的信息。

文本

低可信度描述已自动生成

图 2.1.2 弹幕格式

而评论的获取方式同理，需要一定的url格式：

[https://api.bilibili.com/x/v2/reply/main?jsonp=jsonp&next={i}&type=1&oid={oid}&mode=3&plat=1&\_=1647577851745](https://api.bilibili.com/x/v2/reply/main?jsonp=jsonp&next=%7bi%7d&type=1&oid=%7boid%7d&mode=3&plat=1&_=1647577851745)。

其中i表示页面的编号，而oid则表示一个视频的av号，同样可以使用上面的方法来获取，下图 2.1.3是评论的网页格式。

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

图 2.1.3 评论格式

很明显可以看出评论的信息是JSON格式，只要根据指定的路径就可以得到评论信息，额外的还能得到发布者的相关信息、时间戳、点赞回复数等信息。

### 数据清洗和预处理

数据的清洗和预处理直接使用pandas库，将数据转换为DataFrame格式，然后对数据进行去重去空、数据聚合等操作，最后持久化的存储到数据库中。

图示

描述已自动生成

图 2.1.4 数据处理流程

### 数据分析与挖掘

数据可视化主要旨在借助于图形化手段，清晰有效地传达与沟通信息。但是，这并不就意味着数据可视化就一定因为要实现其功能用途而令人感到枯燥乏味，或者是为了看上去绚丽多彩而显得极端复杂。为了有效地传达思想概念，美学形式与功能需要齐头并进。

数据分析是指用适当的统计分析方法对收集来的大量数据进行分析，将它们加以汇总和理解并消化，以求最大化地开发数据的功能，发挥数据的作用。而数据挖掘是指从大量的数据中通过算法搜索隐藏于其中信息的过程。

本次项目的数据分析与挖掘使用的技术仍然是python的pandas工具库，pandas工具库提供DataFrame方法包括但不限于新建、查询、删除、聚合算法、lambda算法、持久写入文件。

### 数据可视化

数据可视化部分使用echarts.js的可视化技术，echarts的技术十分强大，支持多种图形如：折线图、柱状图、饼图、散点图、热力图等等。

不仅如此echarts还提供多种扩展以适应不同的需求场景，如python版本的pyecharts、用于绘制词云图的wordcloud-echarts。

图形用户界面, 应用程序

描述已自动生成

图 2.1.5 Echarts官方示例

**节目数据分析实现：**

1. 评论-月份关系折线图
2. 评论-星期关系直方图
3. 观众性别占比饼图
4. 24小时男女观看情况并列直方图
5. 等级-点赞数量关系直方图
6. 热门评论漏斗图
7. 评论词云图
8. 等级-点赞箱线图
9. 《1818黄金眼》关键词热力图

**情感分析实现：**

1. 平均情感统计直方图
2. 单集弹幕情感变化时间图
3. 总体情感仪表盘
4. 情感得分词云图

## 功能化模块需求

### 登录功能

网站功能要求每个用户之间的数据是相互隔离的，这样的话数据之间才不会造成数据污染，对于管理者数据管理也更容易。

登录功能的实现需要结合数据库进行验证，当数据库验证成功则对用户展示专属个人界面，如果失败则进行失败提示驳回请求，或跳转至注册界面。

图示

描述已自动生成

图 2.2.1 登录功能流程

对于注册界面的设计和登录界面异曲同工，我们需要采集足够可靠的数据开确定用户的唯一性，并将用户信息持久化写入关系型数据库，必要的是将密码进行加密编码存储。

### 表的预览

要求对于用户的每一张表都能够进行预览，这要求所有的表需要一个特定的id来保证访问表的唯一性和正确性。

将表的数据转换为一个二维列表传递到前端，前端则使用table标签进行渲染展示。

图示

描述已自动生成

图 2.2.2 表预览功能

通过表的预览，我们可以清楚的得知表的结构，从而能够更加准确的对可视化项目设定目标和方向。

# 概要设计

## 系统整体设计

系统整体的概要设计如图 3.1.1 所示：

图示

描述已自动生成

图 3.1.1 系统总体概要设计

## 数据库设计

数据库的设计主要是用户表、用户项目表以及用户数据表三个部分。这三个部分的主要功能如下：

用户表 – 用于用户登录验证和用户注册信息写入；

用户项目表 – 用于区分可视化项目所属用户，并根据项目ID展示可视化项目；

用户数据表 – 用于区分可视化数据表所属用户，并根据表的ID预览用户所拥有的表。

### 用户表设计

用户的主要字段是用户的id以及用户的用户名和密码，这三者是最为重要的部分，此外可能还会需要搜集用户的性别、年龄、邮箱等重要信息，但为了方便起见可以尽量少的设定需要字段。

表格

描述已自动生成

图 3.2.1 用户表样式

### 用户项目表

用户项目表如果需要确定一个唯一一个项目，只需要确认项目的id和和其所属的用户id就能做到。

门上写着字

描述已自动生成

图 3.2.2 用户项目表样式

### 用户数据表

用户数据表和用户项目表差别不大，需要注意的是数据是存储在单个或多个表中的，因此表的path属性可以有多个，对于表来说这是可重复的选项，而对于设计者来说这也很好的防止了一次读入大量数据的场景：

门上写着字

描述已自动生成

图 3.2.3 用户数据表样式

## 模块说明

项目的实现分为三个模块：数据采集、数据处理和存储、可视化展示。

### 描述性统计

1. 评论-月份关系折线图

该图以月份为划分依据，展示了不同月份的评论数量变化。

1. 评论-星期关系直方图

该图以星期数为划分依据，展示了不同星期的评论数量变化。

1. 观众性别占比饼图

该图描述综艺节目《1818黄金眼》的受众分布情况。

1. 24小时男女观看情况并列直方图

该图以性别和评论时间为划分依据，展示了不同性别的人不同时间段评论数量变化。

1. 等级-点赞数量关系直方图

该图展示不同等级的评论被点赞的情况。

1. 热门评论漏斗图

该图显示热评前十的评论详情，并以漏斗的形式展示。

1. 评论词云图

对爬取到的评论进行分词并可视化。

1. 等级-点赞箱线图

该图展示等级对于点赞数的影响。

1. 《1818黄金眼》关键词热力图

该图显示节目中出现频繁的关键词之间的联系。

### 情感分析

1. 平均情感统计直方图

该图展示了对每一条评论和弹幕的情感打分，分数越靠近1，表示这条评论的情感越积极；分数越靠近0，表示这条评论的情感越消极。

1. 单集弹幕情感变化时间图

该图显示在一集节目中观众情感的起伏变化，得出较为热门的时间断。

1. 总体情感仪表盘

该图表示全体观众对于该节目的情感均值。

1. 情感得分词云图

筛选关键词，对于关键词情感较强的评论、弹幕进行词云展示。

### 其他

其中数据采集模块使用request库和re库对《bilibili@黄金眼》进行弹幕和评论的爬取。

数据处理采用pandas库的DataFrame数据结构，对表结构的数据进行数据的去重、去空、以及统计处理。

数据存储采用Django自带的轻量级关系型数据库进行存储，这样做的好处是直连方便快捷。

## 界面设置

### 登录界面

图形用户界面, 网站

描述已自动生成

图 3.4.1 登录界面

### 工作台界面

图形用户界面

描述已自动生成

图 3.4.2 工作台界面

### 数据分析界面

图形用户界面, 应用程序

描述已自动生成

图 3.4.3数据分析界面

### 情感分析界面

图形用户界面

低可信度描述已自动生成

图 3.4.4 情感分析界面

### 表预览界面

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

图 3.4.5 表预览界面

# 详细设计

## 数据采集

### 爬取b站视频评论信息

**编写思路：**

采用面向对象的思想，将爬虫程序进行类封装，类名为JsonProcess。类的属性包括掩饰头header、 存储爬取信息的dataList列表属性、Json\_data字符串属性用于中转变换的Json数据；使用spider 函数发起请求开始爬虫程序，spider需要传入一个字符串类型URL参数作为请求目标。

整体的爬虫程序包含进一个名为getReplies的方法中，其需要一个JsonProcess实例对象、指定开始页数i、指定视频的oid。

**JsonProcess类的设计：**

class JsonProcess:

def \_\_init\_\_(self):

self.dataList = []

self.Json\_data = ''

# 请求头

self.headers = {

'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/99.0.4844.51 Safari/537.36 Edg/99.0.1150.39'

}

# 发送爬取请求

def spider(self, URL):

url = URL

response = requests.get(url, headers=self.headers, verify=False)

response.encoding = 'utf-8'

self.Json\_data = response.json()['data']['replies']

**getReplies函数设计：**

def getReplies(jp, i, oid):

# 不知道具体有多少页的评论，所以使用死循环一直爬

while true:

url = f'https://api.bilibili.com/x/v2/reply/main?jsonp=jsonp&next={i}&type=1&oid={oid}&mode=3&plat=1&\_=1647577851745'

jp.spider(url)

# 如果当前页为空（爬到头了），跳出循环，程序结束。

if jp.Json\_data is None:

break

for node in jp.Json\_data:

print('===================')

# 对感兴趣的字段提取（以评论为例）

content = node['content']['message'] # 评论

# 将字段放入列表

data = [content]

jp.dataList.append(data)

# 每爬完一页，页数加1

i += 1

time.sleep(2)

### 爬取b站视频弹幕信息

**编写思路：**

将视频的cid放入一个名为cids的列表，方便迭代爬取不同视频的弹幕信息，将弹幕爬取之后持久的写入csv文件中，这里可以借用pandas.DataFrame.to\_csv()函数直接写入文件。

**代码设计：**

if \_\_name\_\_ == '\_\_main\_\_':

fileName = "1818黄金眼弹幕\_1-14.csv"

barrage\_list = []

for i in range(len(cids)):

url = "https://comment.bilibili.com/" + cids[i] + ".xml"

page = requests.get(url=url, headers=headers)

page.encoding = page.apparent\_encoding

tmpList = re.findall('<d p=".\*?">(.\*?)</d>', page.text)

print(tmpList, '\n')

barrage\_list.append(tmpList[:]) # 添加进二维列表

time.sleep(2) # 太快了不好

if not os.path.exists(barrageDir):

os.mkdir(barrageDir)

df = pd.DataFrame()

for i in range(len(barrage\_list)):

df = pd.concat([df, pd.DataFrame(barrage\_list[i])], axis=1)

df.to\_csv(fileName, index=False)

## 数据处理

**编写思路：**

简的来说就是将数据文件读取为DataFrame然后将空值和重复值去掉，对一些特殊值进行进一步修改添加必要的列表。

**读取数据：**

# 弹幕

brg\_df = pd.read\_csv('../barrages/1818黄金眼弹幕\_1-14.csv')

# 5集的评论

cmt\_df1 = pd.read\_csv('../comments/1818-1.csv')

cmt\_df2 = pd.read\_csv('../comments/1818-2.csv')

cmt\_df3 = pd.read\_csv('../comments/1818-3.csv')

cmt\_df4 = pd.read\_csv('../comments/1818-4.csv')

cmt\_df5 = pd.read\_csv('../comments/1818-5.csv')

# 总的评论

cmt\_df = pd.concat([cmt\_df1,cmt\_df2,cmt\_df3,cmt\_df4,cmt\_df5],axis=0)

在读取数据之后需要将评论的DataFrame的ctime时间戳这一列进行加工，诸如转换为日期、月份、星期、时间等列元素。

# 时间戳转日期

cmt\_df['time'] =

cmt\_df['ctime'].map(lambda x:time.strftime('%Y-%m-%d %H:%M:%S',time.localtime(x)))

# 获取天数

cmt\_df['day'] = cmt\_df['time'].map(lambda x:x.split(' ')[0])

#获取月份数

cmt\_df['month'] = cmt\_df['time'].map(lambda x:x.split(' ')[0][:7])

# 得到星期数

cmt\_df['week'] = cmt\_df['ctime'].map(lambda x:time.localtime(x).tm\_wday)

# 得到24小时

cmt\_df['hour'] = cmt\_df['ctime'].map(lambda x:time.localtime(x).tm\_hour)

# 重新设置索引

cmt\_df.index = [i for i in range(len(cmt\_df))]

## 数据分析

### 一般数据分析

**时间分组编写思路：**

利用dataframe时间time这一列，分离出来的月份、星期、日期分别对以上三组分组进行统计得到需要的数据。

**月评论数量数据：**

y\_month = cmt\_df.groupby('month').count()['time']

x\_month = list(y\_month.reset\_index()['month'])

**周评论数量数据：**

x\_week = ['星期一','星期二','星期三','星期四','星期五','星期六','星期七']

y\_week = list(cmt\_df.groupby('week').count()['time'])

**日评论数量数据：**

y\_day = cmt\_df.groupby('day')['rcount'].sum()

x\_day = list(y\_day.reset\_index()['day'])

**性别分组编写思路：**

利用dataframe的性别sex这一列，对性别进行分组统计，值得注意的是bilibili用户的性别分为3种类，分别是女、男、保密。

**性别分组数据：**

x\_sex = ['保密','女','男']

y\_sex = cmt\_df.groupby('sex').count()['time']

y\_sex = list(i\*9 for i in y\_sex)

data\_sex = [

{"value":y\_sex[0],'name':x\_sex[0]},

{"value":y\_sex[1],'name':x\_sex[1]},

{"value":y\_sex[2],'name':x\_sex[2]},

]

**观看时间段编写思路：**

利用dataframe的时间hour这一列，可以得出节目观众看视频的喜好时间段，考虑到性别差异，可以在这个基础之上再进行一次男女分组。

**时间段分组编写：**

# 男

y\_male\_hour = cmt\_df[cmt\_df['sex'] == '男'].groupby('hour').count()['time']

y\_male\_hour = list(i\*9 for i in y\_male\_hour)

# 女

y\_female\_hour = cmt\_df[cmt\_df['sex'] == '女'].groupby('hour').count()['time']

y\_female\_hour = list(i\*9 for i in y\_female\_hour)

**等级分组编写思路：**

利用dataframe的等级level这一列，可以根据不同等级对该节目的喜爱程度或者说活跃程度，需要注意的是bilibili的等级分为7个阶段为0～6级别，这个级别中0等级的用户一般是注销的用户，非必要可以不统计。

**等级分组编写：**

x\_lv = [0,1,2,3,4,5,6]

y\_lv = cmt\_df.groupby('level').count()['time']

y\_like\_lv = cmt\_df.groupby('level')['like'].mean()

**热评提取编写思路：**

利用dataframe的点赞like这一列，来判断哪些评论的点赞率较高，对其评论content进行提取。

**热评提取编写：**

tmp\_df = cmt\_df.sort\_values(by='like',ascending=False).head(10)

x\_comment\_10 = list(tmp\_df['content'])

y\_comment\_10 = list(tmp\_df['like'])

data\_comment\_10 = list()

for i in range(10):

data\_comment\_10.append({'value':y\_comment\_10[i],'name':x\_comment\_10[i]})

### 情感分析

SnowNLP是一个python写的类库，可以方便的处理中文文本内容，是受到了TextBlob的启发而写的，由于现在大部分的自然语言处理库基本都是针对英文的，于是写了一个方便处理中文的类库，并且和TextBlob不同的是，这里没有用NLTK，所有的算法都是作者自己实现的，并且自带了一些训练好的字典。

因此我们可以利用这个库对数据直接进行情感分析，情感分析结果的区间为[0,1]，越接近0的结果越消极，越接近1越积极。

**评论弹幕情感得分：**

# 评论

cmt\_list = list(cmt\_df['content'])

cmt\_score\_df = pd.DataFrame([SnowNLP(str(x)).sentiments for x in cmt\_list],columns=['score'])

# 弹幕

brg\_score\_df = pd.DataFrame([SnowNLP(str(x)).sentiments for x in brg\_list],columns=['score'])

但是这样的到的分数集过于庞大，我将情感以0.125为一个区间，划分为了8个区间，按照这8个区间进行分组统计。

cmt\_score\_list = []

for i in range(8):

cmt\_score\_list.append(list(cmt\_score\_df[(cmt\_score\_df['score']>i\*0.125)&(cmt\_score\_df['score']<=(i+1)\*0.125)]['score']))

cmt\_score\_count\_list = []

for i in range(8):

cmt\_score\_count\_list.append(len(cmt\_score\_list[i]))

## 数据汇总

将之前的到的所有数据进行汇总成一个字典，然后转存为JSON文件持久存储，等项目启动时加载一次，然后前端就能够随意的调用这些数据，而不需要在调用数据的时候才开始加载而导致阻塞。

dict\_data ={

# 字典内容如上，这里省略

}

# 写入文件

json\_data = json.dumps(dict\_data,ensure\_ascii=False)

with open('../json\_data.json','w') as f:

f.write(json\_data)

## 数据可视化

如果要使用echarts进行数据可视化首先需要了解他的结构：

import \* as echarts from 'echarts';

var chartDom = document.getElementById('main'); // 关联div的id

var myChart = echarts.init(chartDom);

var option;

option = {

// 绘制图的参数

};

option && myChart.setOption(option);

显而易见我们首先需要一个id为main的div块，然后在javascript代码中向上面代码中规范书写，我们绘制的是什么样的图像取决于我们向option中传入的数据。

例如series中的type类型为line则说明我们绘制的是折线图、如果是bar则说明我们绘制的条形图，再进一步的就是我们需要向xAxis和yAxis中传递一些数据来让其显示我所希望看到的图像。

关于这些更详细的部分我觉得官方的API文档会帮助到你：

<https://echarts.apache.org/zh/api.html#echarts>

# 可视化结果分析

## 描述性统计

### 评论月分布折线图

图形用户界面, 文本, 应用程序

中度可信度描述已自动生成

图 5.1.1 评论月分布折线图

结果分析：

可以看出观看数量的统计高峰期在2019-01月～2019-05月之间，这段时间刚好是节目的播出时间，节目刚播出的时候热度很高，但之后骤减，说明节目的重复观看率或新观看人数并不多，预计评论数在之后也不会在有起伏。

### 评论星期分布直方图

图表, 条形图

描述已自动生成

图 5.1.2 评论星期分布直方图

结果分析：

这是根据评论发布的星期时间而统计出的直方图，不难看出在星期六和星期日的观看评论人数要远远多于星期一至星期五的人数，说明人们通常会选择在周末的闲暇时间观看这类节目，同时也意味着在周末播出的节目可能带来的收益和热度会更高。

### 节目受众占比饼图

图表, 饼图

描述已自动生成

图 5.1.3 节目受众占比饼图

结果分析：

节目的受众在不考虑保密人群的性别，男性的占比要远多于女性，也许男性人群更爱看着这种充满综艺性质的新闻节目。另一方面保密占比最多，或许也说明b站人群对于性别的保密观念较重。

### 24小时男女观看统计图

图片包含 图形用户界面

描述已自动生成

图 5.1.4 24小时男女观看统计图

结果分析：

这是根据评论发布的24小时时间段和评论用户性别而统计出的直方图，可以看出无论性别，大家都喜欢在夜间（19:00～0:00）观看该节目，这样充分的说明的该节目的有趣性以至于下饭或是让人熬夜的地步。

### 等级-点赞关系图

图表, 雷达图, 旭日形

描述已自动生成

图 5.1.5 等级点赞关系图

结果分析：

这张图表示等级和点赞之间的关系，不难看出越高等级的用户得到点赞数量越多，这表明用户更喜欢给高等级的用户点赞或是高等级用户更能评论出大家中意的评论。

### 热评前十漏斗图

图表, 漏斗图

描述已自动生成

图 5.1.6 热评前十漏斗图

结果分析：

根据点赞的数量得到前十的热评，可以看出其中大部分的评论都是对该节目的好评，该节目的受喜爱程度之高可见一斑。

### 热词云图

文本

描述已自动生成

图 5.1.7 热词云图

结果分析：

从热词中步伐看见“1818”、“黄金眼”、“哈哈哈”、“笑死我了”、“沙雕”等字眼，说明该节目是一款偏喜剧的综艺性质新闻节目。

### 关键词热力图

图表

描述已自动生成

图 5.1.8 关键词热力图

结果分析：

这是根据评论和弹幕出现频率较高的一些词语进行它们之间关联度的展示，除了和自己的关联，我们不难看出其中关联较深的几组词语，“店家”和“战术回避”、“记者”和“灵魂拷问”，“遇事不决”和“量子力学”。从这些我们可以推断出1818黄金眼是一个为消费者发声的新闻媒体，并受广大观众的喜爱。

## 情感分析

### 弹幕评论情感统计直方图

图表, 条形图

描述已自动生成

图 5.2.1 弹幕评论情感统计直方图

结果分析：

将情感的[0,1]区间分割为8个部分，按这8个部分对情感进行统计。可以看出无论是弹幕还是评论，在0.875~1的情感占大多数，说明观众对于节目颇有好感。随后就是在0.5周围的情感居多，这些都是模糊的表达，具有客观陈述的可能，并不奇怪。

### 第五集情感变化图

图表, 瀑布图

描述已自动生成

图 5.2.2 第五集情感变化趋势

结果分析：

将整个节目的时长均分为30个小份，其中在4-6分段和16-20分段，观众的情感是较为积极的，说明这段时间有较为滑稽的部分，综合整体的部分积极占比较多，这反应了这档节目整体氛围也是积极的。

### 平均情感仪表盘

手机屏幕的截图

描述已自动生成

图 5.2.3 平均情感仪表板

结果分析：

同上面的分析，平均情感积极程度高达72.26，这再一次印证了节目的整体氛围是积极快乐的。

### 情感词云图

图片包含 日程表

描述已自动生成

图 5.2.4 情感词云图

结果分析：

选择情感值较高的弹幕进行显示，可以发现其中“哈哈哈”的词汇暂居了很多，印证了观众对于该节目的喜爱，以及该节目的定位。

# 课程设计体会

本次可视化课程设计虽然有足足两周时间，但是由于是一个人独自完成一个项目，显得还是颇有挑战性，通过这一次课程设计也是十足的锻炼了我独立自主处理问题的能力。

在好几次课程的经历之后，我开始觉得制定一个计划去完成项目往往会事半功倍。在实际编写代码之前其实还有很多的事情需要确认，诸如技术的涉及和运用、数据库的选型、系统架构的模板等等，这些都是在编码时会遇见的问题，如果在那之前确定一个答案，这会让之后的路轻松不少。

这次的课程设计让我最头疼的就是页面的重复利用，对于一个并不熟练前端技术的人来说要实现这种有点难度的功能是比较头疼的，好在这并没让我放弃，在好几次的检索和阅读技术博客之后我终于将这个问题解决。原来只需要将公共的部分单独成一个html页面，需要使用就利用Django的include关键字引入即可。虽然之后再看答案会觉得这并不是难事，但是一点点熟练技术的过程让我受益良多。

俗话说，冰冻三尺，非一日之寒，我相信技术的累积也同样如此。再困难的技术只要之际肯花费功夫，就一定会有回报，哪怕这个回报并不大，那也是值得去做的。

# 参 考 文 献

[1]王冉阳.基于Django和Python的Web开发[J].电脑编程技巧与维护,2009(02):56-58.

[2]李迎辉.Python开发Rails框架——Django框架介绍[J].程序员,2006(11):107-110.

[3]杨志庆.基于Django的Blog系统的开发与实现[J].机电一体化,2013,19(09):69-72.

[3]舒后,熊一帆,葛雪娇.基于Bootstrap框架的响应式网页设计与实现[J].北京印刷学院学报,2016,24(02):47-52.DOI:10.19461/j.cnki.1004-8626.2016.02.014.

[4]高榕岭.Bootstrap在前端开发中的优势[J].计算机光盘软件与应用,2015,18(01):74+76.

[5]傅翠玉,王少茹,洪秀金.Bootstrap框架在响应式WEB开发中的应用[J].电脑知识与技术,2018,14(21):85-86.DOI:10.14004/j.cnki.ckt.2018.2309.

[6]王子毅,张春海.基于ECharts的数据可视化分析组件设计实现[J].微型机与应用,2016,35(14):46-48+51.DOI:10.19358/j.issn.1674-7720.2016.14.015.

[7]崔蓬.ECharts在数据可视化中的应用[J].软件工程,2019,22(06):42-46.DOI:10.19644/j.cnki.issn2096-1472.2019.06.011.

[8]唐敏,宋杰.嵌入式数据库SQLite的原理与应用[J].电脑知识与技术,2008(04):600-603.

[9]刘雨珂,王平.基于Python+Pandas+Matplotlib的学生成绩数据统计与图形输出实现[J].福建电脑,2017,33(11):104-106+142.DOI:10.16707/j.cnki.fjpc.2017.11.053.

[10]何春燕,王超宇.基于python+pandas的数据分析处理应用[J].数码世界,2018(07):386.

[11]安子建. 基于Scrapy框架的网络爬虫实现与数据抓取分析[D].吉林大学,2017.

[12]姜杉彪,黄凯林,卢昱江,张俊杰,曾志高,刘强.基于Python的专业网络爬虫的设计与实现[J].企业科技与发展,2016(08):17-19.

[13]陈伟,孙梦蝶.基于网络爬虫技术的大数据审计方法研究[J].中国注册会计师,2018(07):76-80.DOI:10.16292/j.cnki.issn1009-6345.2018.07.014.

[14]黄秀丽,陈志.基于JSON的异构Web平台的设计与实现[J].计算机技术与发展,2021,31(03):120-125.

[15]赵志升,靳晓松,温童童,梁俊花.基于Python-Snownlp的新闻评论数据分析[J].科技传播,2018,10(18):104-105.DOI:10.16607/j.cnki.1674-6708.2018.18.052.