# 网易云音乐评论情感分析

**作者**：刘泽宇，李强，阳松，王江兴，李亚玲，冯永

**单位**：重庆大学，计算机学院

**案例版权**：该案例归重庆大学计算机学院所有

**涉及的知识点**：Hadoop&Spark技术、分布式爬虫、深度学习、数据可视化

**案例来源及案例真实性情况**：该案例来源于重庆大学计算机学院专业硕士（电子信息）课程《大数据架构与技术》中的学生精选汇编课程设计。

**摘要：**随着互联网的快速发展，网络上的评论文本资源急速增长。面对海量的网络资源信息，如何应用计算机情感分析的技术进行自动文本资源分析，挖掘出评论文本中蕴含丰富价值的评论人的情感动态，是一个热点的研究问题。本案例以大数据下的网易云音乐评论情感分析为主题，利用大数据和机器学习的方法对网易云音乐评论进行数据分析。在具体实现上，首先使用了分布式爬虫技术，爬取一千一百万条评论数据，数据来源是热门歌手的前50首热门歌曲下的最新的700条评论数据加上该歌曲的热评数据，以及20万个发布了热门评论的用户，在数据爬取阶段采用Scrapy-redis分布式爬虫爬取评论和用户数据，存储在mongodb里面。之后采用snowNLP和自己训练的GRU,CNN模型来对数据进行情感分析和比较，其中GRU以及CNN是通过手工标注的数据利用tensorflow实现并训练得到。最后使用Spark框架来对数据进行整合和分析，将分析结果导出后使用Echarts将数据进行可视化展示，部署到云服务器上。

**关键词**：scarpy-redis，spark，tensorflow，网易云，情感分析

## 1 引言

该教学案例来源于重庆大学计算机学院专业硕士（电子信息）课程《大数据架构与技术》中的学生精选汇编课程设计。该案例的关键问题为大数据下的股票价格预测，需引导学生进行的主要内容有：（1）对网易云的数据进行分布式爬取，使用适当的方式进行数据预处理；（2）采用snowNLP，GRU和CNN模型对数据进行情感分析和比较；（3）使用Spark框架来对数据进行整合和分析，并对结果进行可视化展示。

## 2 背景介绍

随着信息技术和互联网的飞速发展，越来越多的互联网用户在各种网站上分享经验和发表评论，这些评论文本表达了评论者的观点和情感，用户也习惯于从互联网的各种评论中获取信息，为自己的某些决策寻找参考意见。分析用户发表在互联网上的评论，能够发现用户情感的演化规律，挖掘其中的潜在信息，具有巨大的应用价值。但是目前互联网上的信息量増长迅速，我们迫切需要借助自动化工具，在大量的信息来源中迅速获取我们真正需要的信息，从而应对信息爆炸带来的挑战。如何对这些包含大量信息的互联网评论文本进行情感分类、挖掘其中蕴含的情感信息是自然语言处理领域的研究难点，也己成为商业智能领域的关注热点，由此引发众多研究人员进行相关研巧，文本情感分析相关技术应运而生。

情感分析也被称为评论挖掘或者意见挖掘，指的是通过对文本内容的分析，挖掘评论者所表达的情感倾向。此技术可在现实生活中的很多方面得到应用，例如，商务智能、民意调查、舆论分析、企业管理等。利用互联网评论文本情感分析的结果，潜在的购买者可了解其他购买者对该产品的态度及使用体验，为自己是否购买提供决策依据；制造商和经销商可了解到消费者关于产品和服务的反馈信息，参照消费者对自己及竞争对手的评价意见，对产品及服务进行改进，从而获得竞争优势，因此对于网络评论进行情感分析具有较大的研究价值。

# 3 内容

本案例的主要内容为对网易云音乐评论进行情感分析，主要分为五个小节，分别为网易云数据的分布式爬取、情感分析，大数据框架搭建，基于Spark的数据处理，以及数据可视化。

# 3.1分布式爬虫

## 3.1.1 爬取对象

为了爬取热门歌手的歌曲信息，首先是进入网址[https://music.163.com/#/discover/artist](https://music.163.com/" \l "/discover/artist)，通过对于页面的分析获取每一个歌手的页面，以张惠妹为例，歌手主页是[https://music.163.com/#/artist?id=10559](https://music.163.com/" \l "/artist?id=10559)，针对收藏热门的前五十首歌曲，选择第一首歌进入[https://music.163.com/#/song?id=31311140](https://music.163.com/" \l "/song?id=31311140)，从歌手和歌曲的url规律来看，只需要获取歌手id以及歌曲id即可完成页面的获取，而歌手id可以通过在页面[https://music.163.com/#/discover/artist](https://music.163.com/" \l "/discover/artist)获得，而歌曲id可以通过歌手页面获得。通过scrapy-redis爬取包含歌手id的页面，提取歌手id，通过拼接url获取歌手热门歌曲页面，根据提取到的歌曲id，获取每一首歌曲每一个歌手的前五十首歌曲评论页面。通过对歌曲评论的页面进行分析，评论来自对<https://music.163.com/weapi/comment/resource/comments/get?csrf_token=>的请求。而在返回的数据中包含写出评论的用户的userId,通过对单个用户的网页进行分析，发现用户的数据是通过对<https://music.163.com/weapi/user/playlist?csrf_token=/>的进行请求获得。故根据该url可以获取评论用户的数据。最终获取了一千一百万评论数据，二十万热评用户数据。

## 3.1.2爬取方式

爬取网页，使用的工具包括scrapy-redis、mongodb。爬取评论信息的具体步骤归结如下：

1. 使用三台机器，一台是centos，两台是ubuntu，分别在两台机器上部署scrapy来进行分布式抓取网页。
2. 一台ubuntu用来作为redis的master端，其他的的机器作为slave,mongodb安装在一台ubuntu机器上。
3. 向redis中加入键值对，键是wangyiyun:start\_urls，值是<https://music.163.com/discover/artist>。
4. master的爬虫运行时会把提取到的url封装成request放到redis中的数据库中，并且从该数据库中提取request后下载网页，再从代表歌手热门歌曲的网页的内容中提取歌曲id信息，拼接url，将请求存放到redis的另一个数据库“wangyiyun:requests”中。
5. slave从master的redis中取出待抓取的request，下载完歌曲评论网页之后就把网页的内容发送回master的redis。
6. 继续从master的redis中取出待抓取的request，下载完评论信息，将评论格式化存储到mongodb中。
7. 重复上面的3、4、5，直到master的redis中的“wangyiyun:requests”数据库为空。
8. master里的reids还有一个数据“wangyiyun:dupefilter”是用来存储抓取过的url的指纹，是防止重复抓取的。

针对存入的mongdb的数据选择点赞数超过20个的评论，爬取做出评论的用户信息。步骤如下：

1. 机器的分布式架构与爬取评论的架构相同。
2. 在redis中设置键值对wangyiyun:start\_urls
3. 从mongodb中查询热评的userId,由于用户信息的接口url已经知道了，只需要将用户的userId等信息进行处理封装进请求的参数，封装成请求，提交给redis，存储在wangyiyun:requests中.
4. 从redis的wangyiyun:requests中获取请求下载用户页面信息，存储在mongodb中。
5. 不断重复3，4直至wangyiyun:requests为空。

## 3.1.3爬取结果

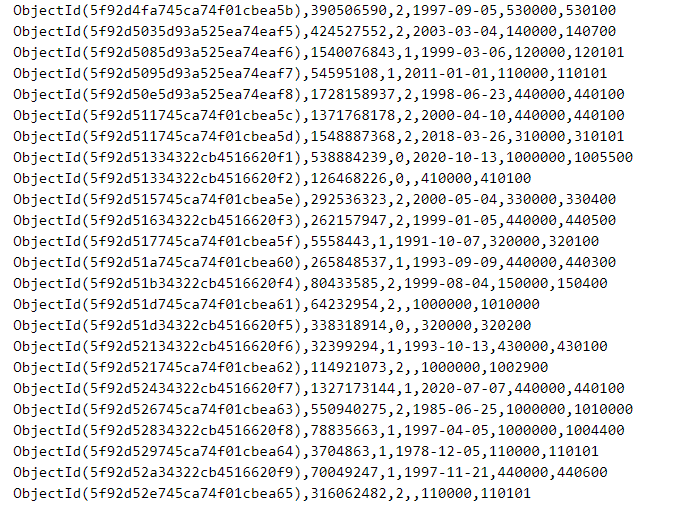
首先是评论数据，格式是id, 歌手名字，音乐名字，评论内容，点赞数，评论时间，评论者的用户id。



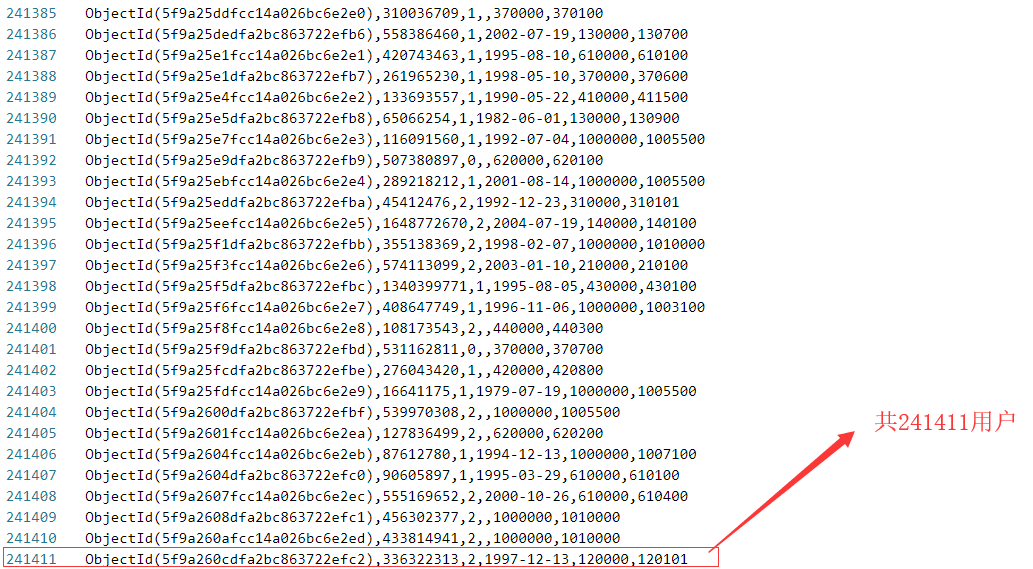
共有11899705条数据



针对用户数据，数据格式是id，性别，生日，省份，城市。



总共爬取了241411位用户数据。



# 3.2 情感分析

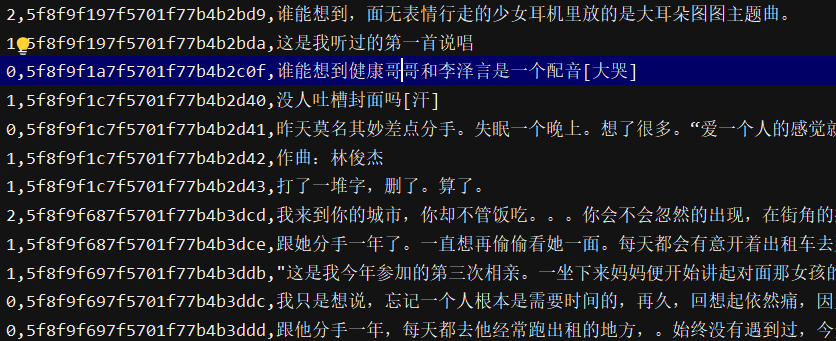
接下来将阐述如何使用机器学习以及深度学习的方法利用这些数据。

## 3.2.1数据预处理

对于爬取到的数据，在使用深度学习方法时，需要进行分词、去停用词，词语向量化等操作。

首先是分词，实验中采用的是jieba分词的精确模式，针对分词结果存在大量的无意义词，比如：了，今天，是的等等，需要对这些停用词进行去除，实验中用的是百度的停用词词典。最后词语的向量化是通过word2vec的CBOW模型实现的，需要注意的是，我们的数据集包含一千一百多万的数据可供word2vec训练，故为了比较数据集规模对word2vec在情感分析中的影响，我们的CBOW模型有两种，一种是一千一百万数据训练的300维词向量，一种是300万训练的50维词向量。

另外，由于实验中使用的模型需要训练数据集，我们标注了1144条热评数据，用作情感分析，下图中0代表消极，2代表积极。



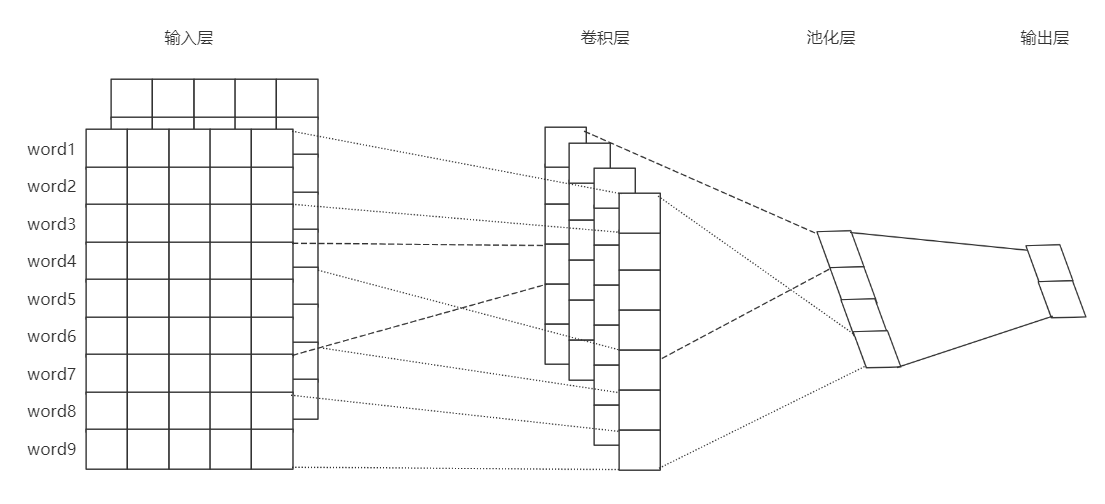
## 3.2.2使用到的模型

（一）SnowNLP

SnowNLP是一个python写的类库，可以方便的处理中文文本内容，并且自带了一些训练好的字典。可以直接对中文数据进行情感分析。不过缺点是，该工具的训练数据是电商评论数据对其他领域的准确性并不是特别理想。

（二）TextCNN模型

TextCNN模型是使用卷积神经网络来处理NLP问题的模型。相比较自然语言处理中传统的RNN、LSTM等模型,CsNN能更加高效的提取重要特征,这些特征在分类中占据着重要位置.该模型的模型结构如图所示：



在TextCNN模型中使用到了嵌入层、卷积层、池化层、全连接层等神经网络基本组成构件，下面对TextCNN模型进行详细介绍：

1. 嵌入层

嵌入层是模型的输入层，常用于将输入的多维数据进行降维，在文本处理领域，主要作用是将文本中出现的词语转换成向量表示，作为后续神经网络层的输入。

1. 卷积层

卷积层由若干个卷积核组成，卷积核中的参数是通过反向传播学习所得到的，在文本处理领域，卷积层主要是对向量化后的文本进行特征提取和特征映射，从低级特征中迭代选择复杂特征。

1. 池化层

池化层是用于特征降维，对卷积层从输入层提取到的特征进行降采样操作，压缩数据和参数的数量，减小过拟合，同时提高模型的容错性。

1. 全连接层

全连接层常处于最后一层，输入是池化后的高级特征，作用是对池化后向量所表示的分布式表示特征映射到样本标记空间。

（三）GRU模型

GRU是循环神经网络的一种。和LSTM模型一样，也是为了解决长期记忆和反向传播中的梯度等问题而提出来的。与LSTM不同的是，在 GRU模型中只有两个门控结构，分别为更新门和重置门。更新门用于控制前一时刻的状态信息被带入到当前状态中的程度，更新门的值越大说明前一时刻的状态信息带入越多。重置门用于控制忽略前一时刻的状态信息的程度，重置门的值越小说明忽略得越多。GRU网络单元的结构如下图4.8所示。



图4.8 GRU网络单元结构图

GRU单元中的更新过程如下所示，其中一般为sigmoid函数，可以将数据变换为取值范围为(0，1)的数值，从而来充当门控信号：

计算更新门，如公式（4.9）所示。其中是时刻GRU单元的输入，是上一时刻GRU单元的输出，和分别对应输入数据和上一时刻GRU单元输出的权值。



计算重置门，如公式（4.10）所示。



计算候选输出，如公式（4.11）所示。候选输出的计算需要利用当前时刻的输入数据、上一时刻LSTM单元输出值以及当前时刻计算出的重置门。



计算当前时刻GRU单元的输出，如公式（4.12）所示。



## 3.2.3 使用模型进行情感分析

（一）SnowNLP

使用SnowNLP进行情感分析，只需要引入该模块，之后直接使用函数SnowNLP(j)，这里的j代表句子，返回的对象调用sentiments属性即可获得情感分数，这里的情感分数是0到1的一个小数，越小代表情感是消极的概率越大。

（二）TextCNN模型

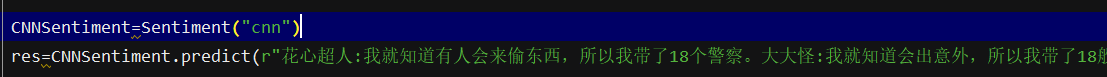
TextCNN模型输入的是一个评论的所有词向量，卷积层进行一维卷积提取特征，卷积核大小分别是1，2，3，数目为256，激活函数是relu，dropout参数选取0.2，经过最大池化后，三种卷积结果拼接在一起，送入全连接层。输出是sigmoid激活的全连接层，输出的是消极评论的概率。

（三）GRU模型

使用GRU进行情感分析，输入的是一个评论经过分词、向量化后的所有词向量，每输入一个词向量，会得到一个当前时刻gru的输出，选择输入在最后时刻获得的输出作为评论的表示，送入全连接层输出消极评论的概率。在实验中，我们选用了word2vec实现词语的向量化，网络结构是单层的32个GRU单元,dropout概率设置为0.5，激活函数采用tanh。输出层是sigmoid激活的全连接层，unit单元数是1。

TextCNN以及GRU模型都是依赖tensorflow实现的，实验剔除了中性标注数据，使用二元交叉熵函数作为损失函数，选用adam优化器，实验中的训练集是400条数据，测试集是200条，batch size是50，TextCNN的学习率是0.00006,gru学习率是0.0015。

为了更好的将实现的模型提供给别的模块调用，实验中将模型的训练好的模型保存，并另写了一个模块将模型封装成类似于SnowNLP一样方便调用的模式。



如图所示，根据函数Sentiment的参数选择模型，根据predict获取传入的参数的情感分数。

# 3.3大数据框架搭建

## 3.3.1 搭建环境及配置介绍

### 1 环境介绍

操作系统：Deepin 20

图形界面： Deepin DE

语言环境：Python3.7，Java8

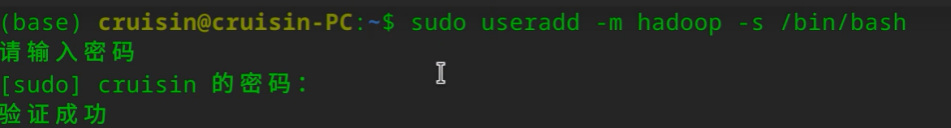
相关软件：Hadoop 2.8.5、Spark2.4.7

### 2 配置介绍

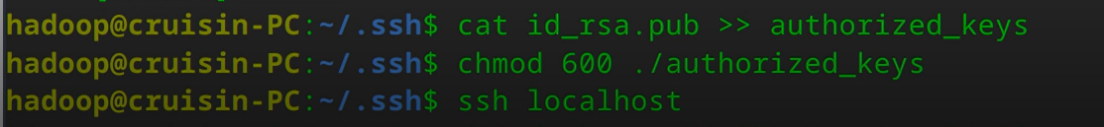
|  |  |  |  |
| --- | --- | --- | --- |
| 主机名 | IP | 分配内存数 | 分配核心数 |
| master | 172.24.56.2 | 15GB | 6 |
| slave01 | 172.24.56.32 | 5.8GB | 4 |
| slave02 | 172.24.138.43 | 10GB | 4 |
| slave03 | 172.24.138.42 | 10GB | 4 |

## 3.3.2 Hadoop大数据架构搭建

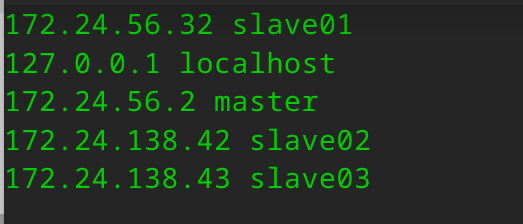
首先在机器上创建用户hadoop：



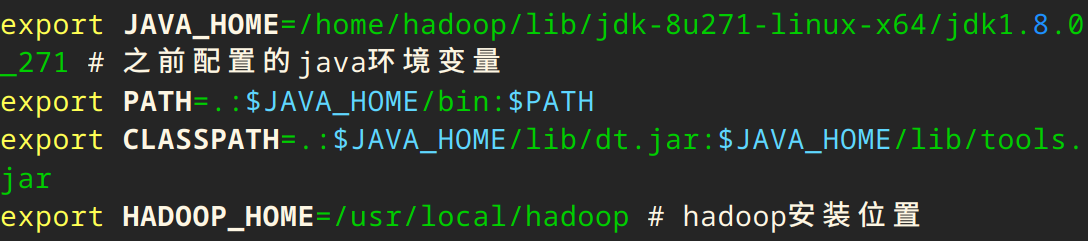
设置免密登录localhost和slave



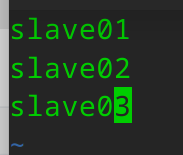
之后修改hostname为各自对应的名字(master和slave)，然后在hosts文件中来配置：



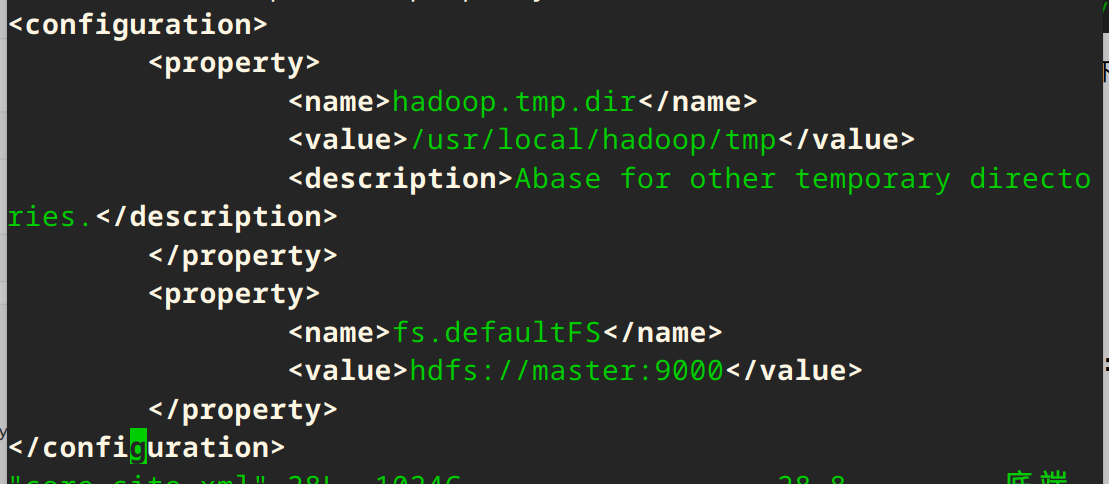
将下载好的Hadoop解压后存放在/usr/local/hadoop下并将所有者改为Hadoop用户，之后开始设置环境变量：



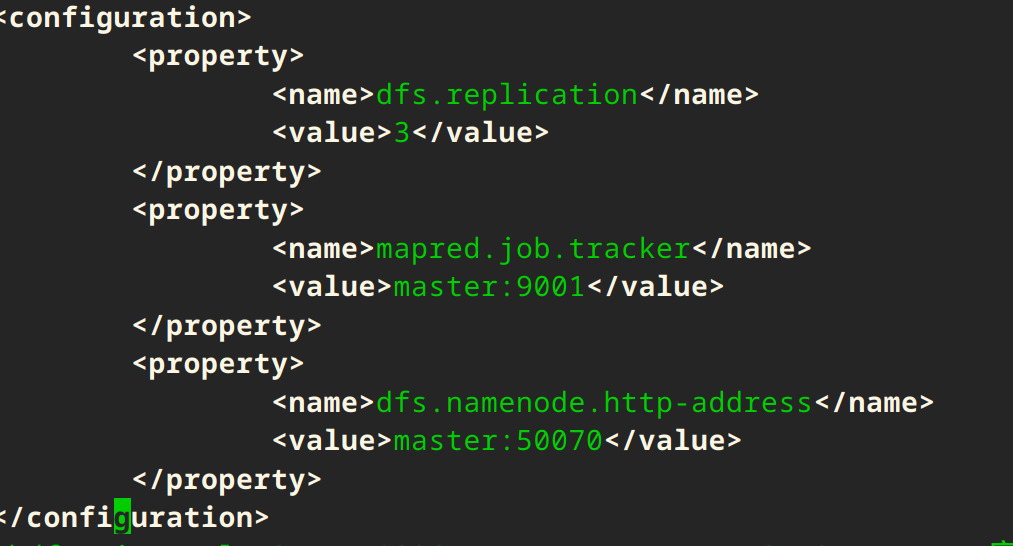
之后开始配置hadoop目录下的slave文件，添加slave01,slave02,slave03作为DataNode：



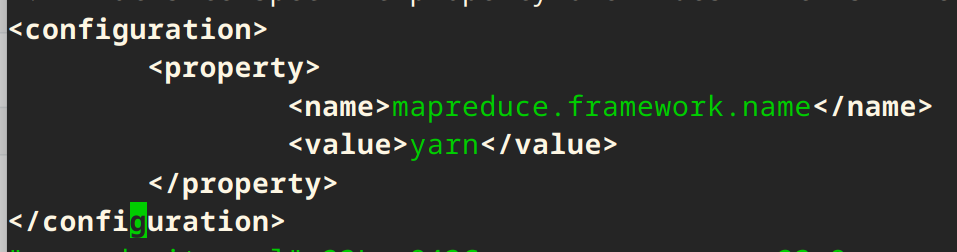
然后修改core-site.xml文件，配置临时目录和默认文件系统：



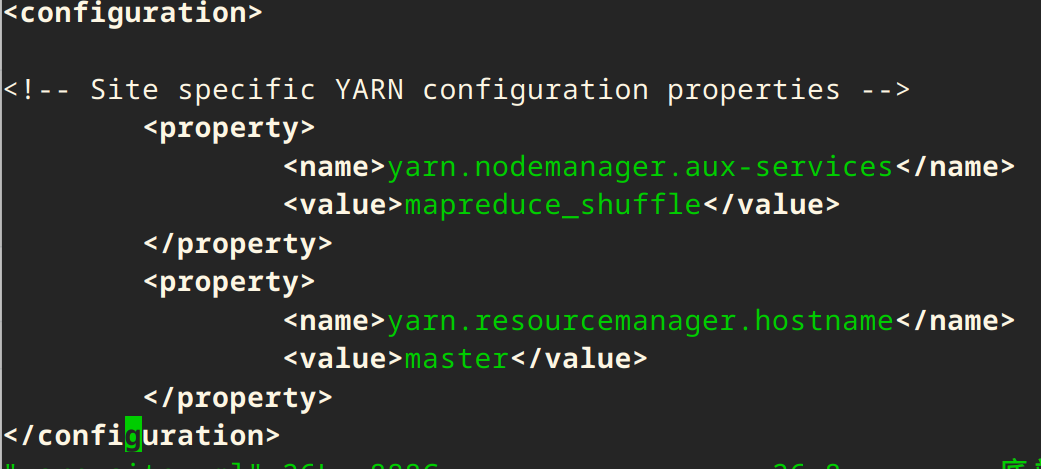
然后修改hdfs-site.xml文件,设置3个副本，job tracker的交互端口，namenode的端口：



Mapreduce使用yarn，修改mapred-site.xml文件为：

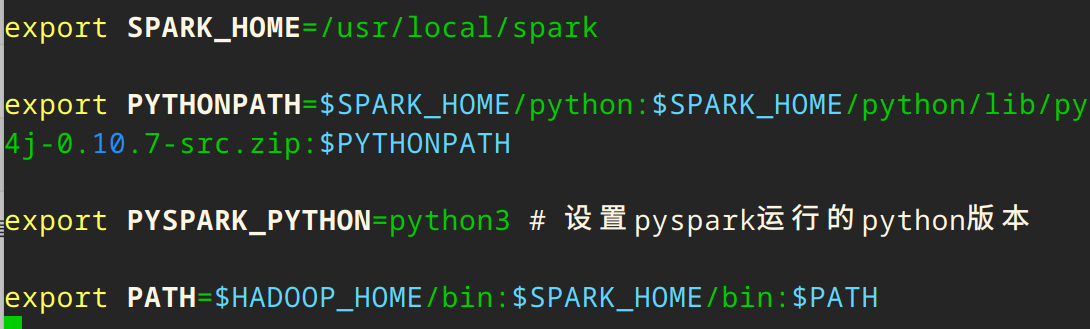


修改yarn-site.xml为：

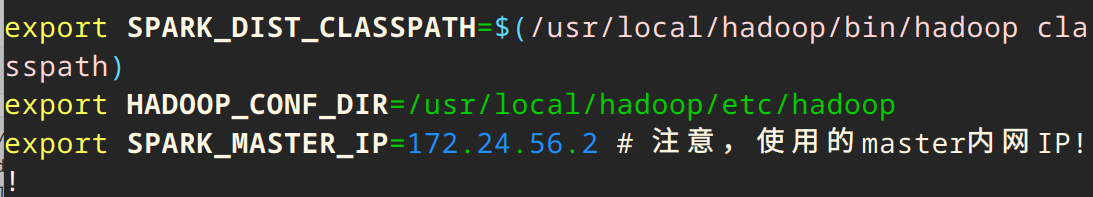


## 3.3.3 Spark引擎搭建

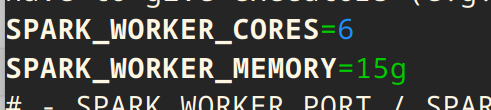
将下载好的spark解压缩到/usr/local/spark之后，修改环境变量：



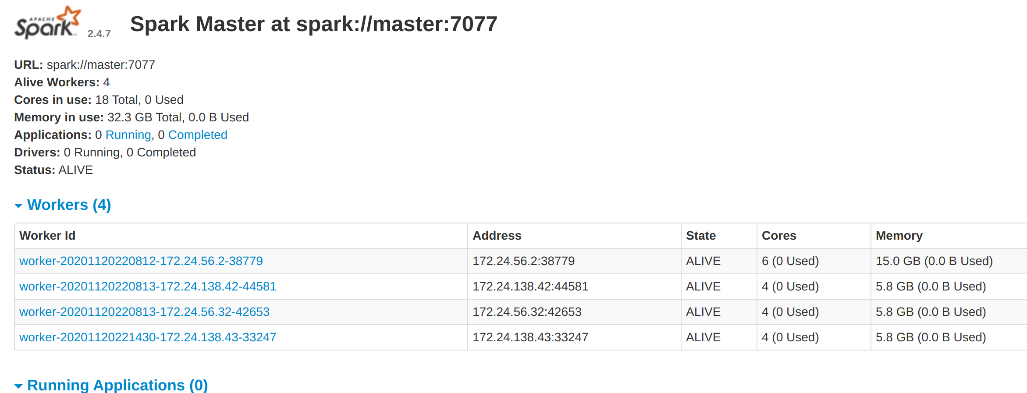
将slave的几个节点添加到slave文件中后之后编辑spark-env.xml文件



设置核心数和内存：



## 3.3.4 运行结果



# 3.4基于Spark的数据处理

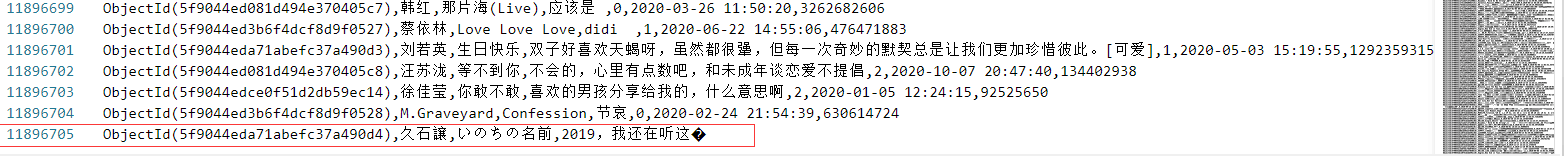
这部分将mongodb存储的数据进行处理，生成可视化展示所需要的数据。数据处理基于以下5个任务展开：

1. 基于用户地区分类，分析全国各个地区用户活跃度，生成地区热力图。
2. 基于评论的情感分数，分析某个省份用户的心情等级（该地区下每个用户的心情等级取均值）。
3. 基于评论的情感分数，根据歌曲进行聚合 分析某首歌下的评论倾向。
4. 基于评论的情感分数，根据歌手进行聚合 分析某个歌手下的评论倾向。
5. 基于爬取的所有歌曲评论，展示关键词。

## 3.4.1数据预处理

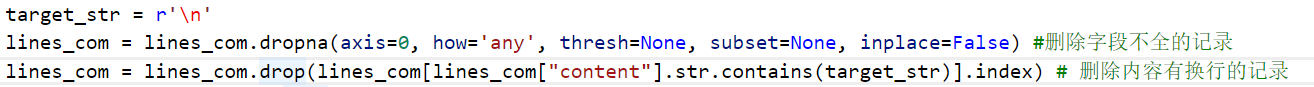
1. 数据清洗

从mongodb导出的数据有两个表，分别是comments All（以下简称评论表），和ueserAll（以下简称用户表）,其中评论表包含9个字段，主要为歌手名字，歌曲名字，评论内容，评论时间和用户ID。用户表包含6个字段，主要为用户ID，用户所在省份和城市。其中部分记录出现字段不全和评论内容为乱码的情况，我们预先将这部分记录给删掉。



记录字段不全

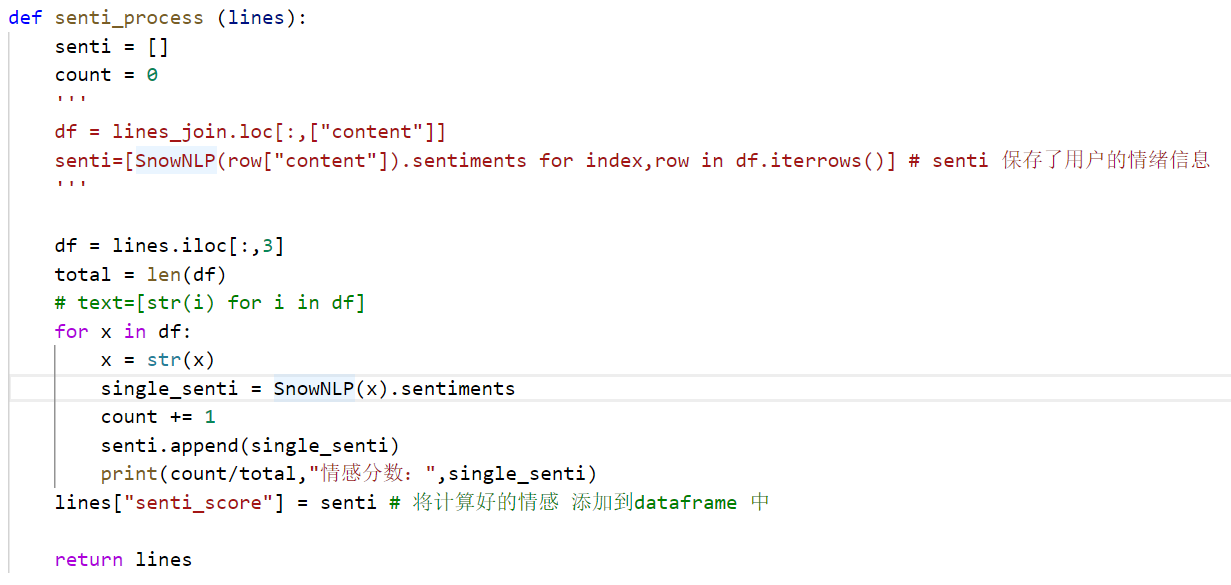
数据清洗代码主要部分如下



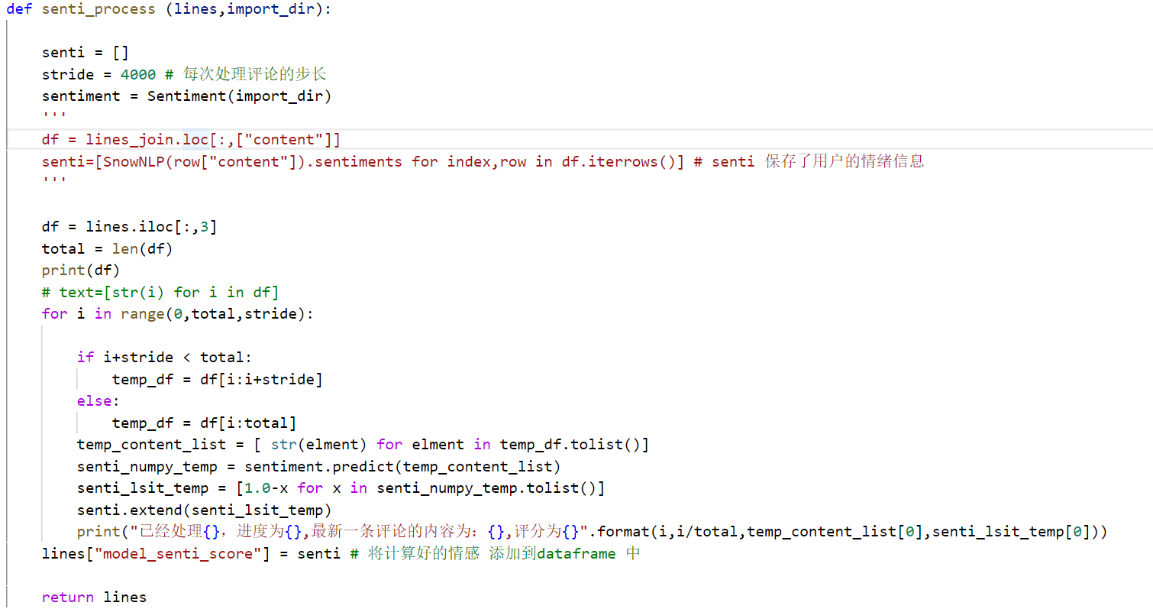
数据清洗代码

1. 情感分析预处理

我们将清洗过的数据，分别送入snownlp，cnn模型，gru模型中进行情感分析。

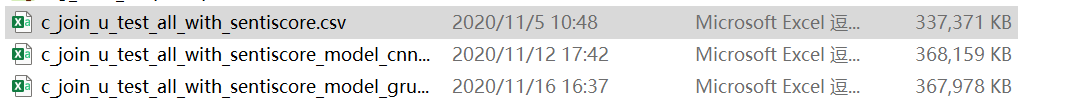


snownlp情感分析

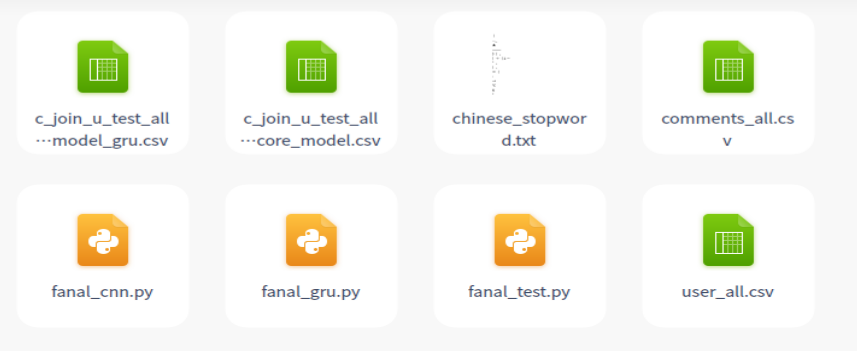


gru/cnn情感分析

情感处理过后生成如图2.5的3个文件c\_join\_u\_test\_all\_with\_sentiscore.csv为snownlp打分后的文件，\*\_cnn.csv文件为cnn模型打分后的文件,\*\_gru.csv为gru模型打分后的文件。



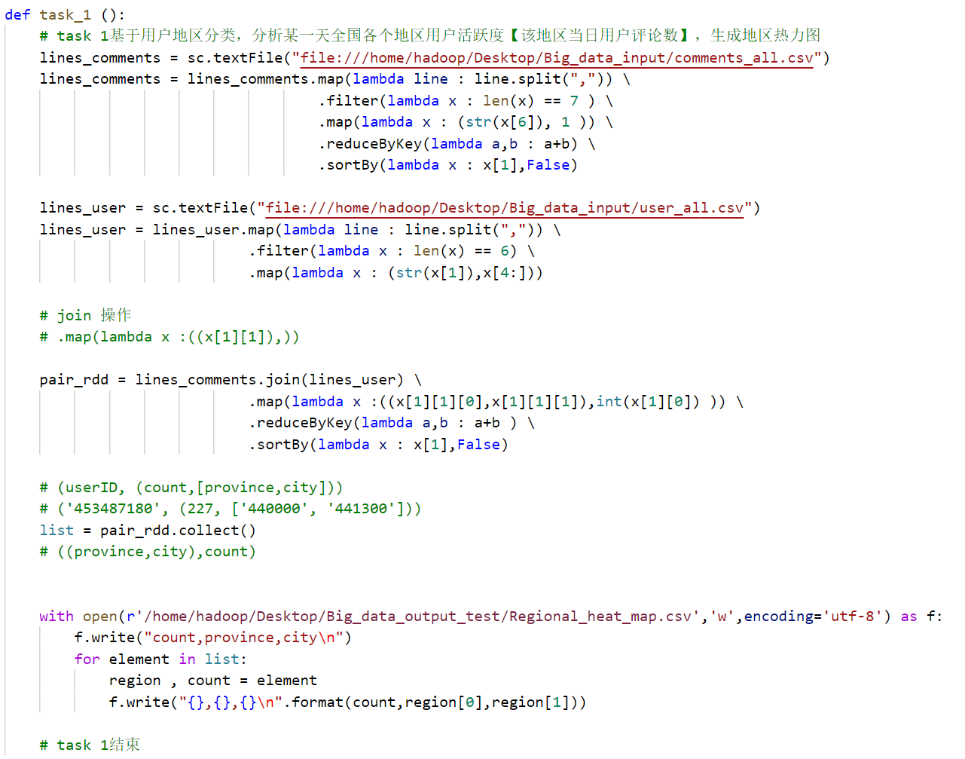
sentiscore文件



预处理过后的所有文件

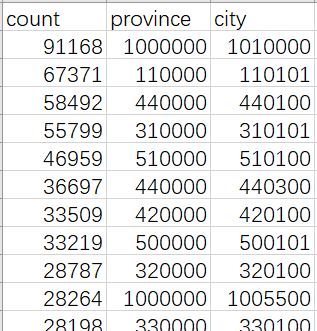
## 3.4.2 spark数据处理

（一）分析全国各个地区用户活跃度



task\_1处理代码文件

生成Regional\_heat\_map\_sorted.csv文件的内容如图2.6



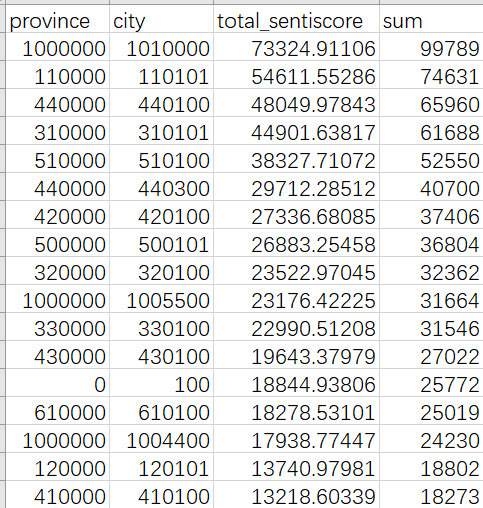
Regional\_heat\_map\_sorted.csv文件

（二）某个省份用户的心情等级



task\_2处理代码文件

生成regional\_user\_mood\_level\_model\_gru.csv文件的内容如下图



regional\_user\_mood\_level\_model\_gru.csv文件

(三)某首歌曲下的情感等级



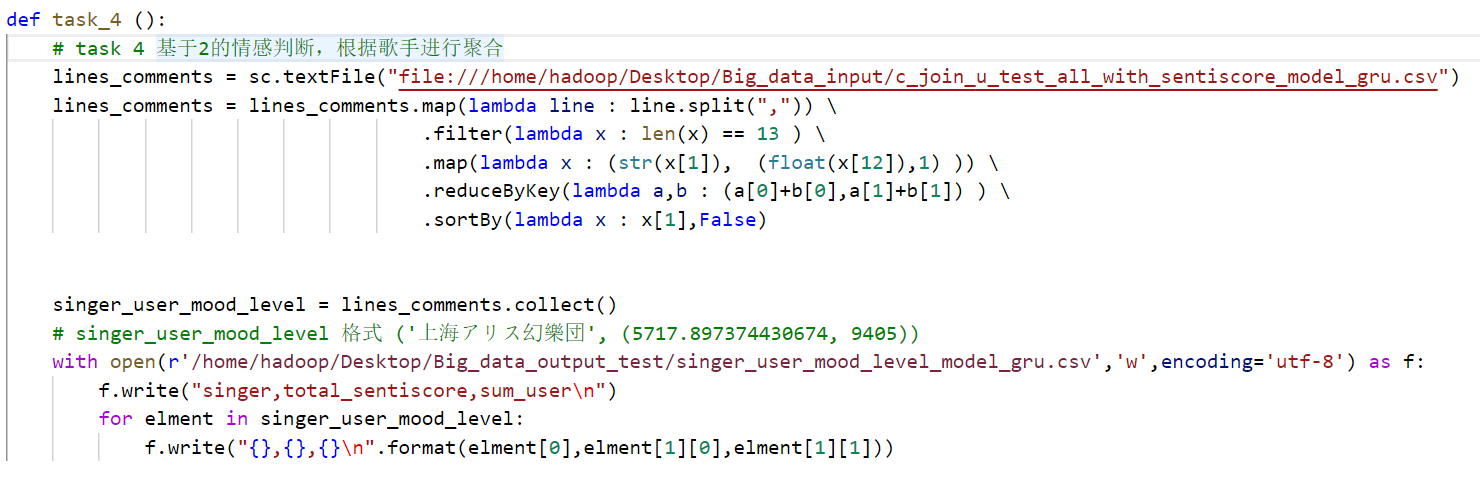
task\_3处理代码文件

生成song\_user\_mood\_level\_model\_gru.csv文件的内容如下图



singer\_user\_mood\_level\_model\_gru.csv文件

（四）某个歌手下的情感等级



task\_4处理代码文件

生成singer\_user\_mood\_level\_model\_gru.csv文件的内容如下图



singer\_user\_mood\_level\_model\_gru.csv文件

（五）某个歌手下的情感等级



task\_4处理代码文件

所有歌曲评论，生成words\_frequency.csv关键词文件



words\_frequency.csv文件

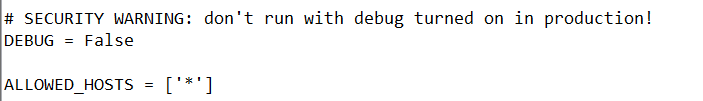
# 3.5数据可视化

可视化对比的算法是snownlp,cnn,gru,地图实现了三级下钻能够到区/县一级。以下图片中“情感总得分”是某个歌曲(或歌手）所爬取评论情感分析后所有用户情感值的加和，基于该值和用户总数得到以下图片的“情感心情等级值”。限于篇幅，例如全国地区心情等级部分，底图相同，数据不同就不予以展示。

（一）前期准备

项目部署：

* 1. 安装python3
  2. 安装和项目一致的包（Django=2.2等）
  3. 安装uwsgi
  4. 修改django项目配置允许所有ip访问



Django项目配置

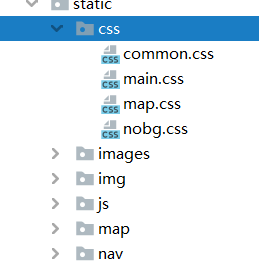
* 1. 配置uwsgi，在项目文件里创建 uwsgi.ini文件,编辑文件 ，设置uwsgi属性。
  2. 安装nginx并配置。nginx一般默认安装好的路径为/usr/local/nginx。在/usr/local/nginx/conf/中先备份一下nginx.conf文件，以防意外nginx一般默认安装好的路径为/usr/local/nginx 在/usr/local/nginx/conf/中先备份一下nginx.conf文件，以防意外，然后打开nginx.conf，把原来的内容删除，对里边任务进行修改。
  3. 将项目迁移好后 通过命令 启动 uwsgi--ini uwsgi.ini/nginx 启动
  4. 输入公网ip 47.112.139.168 访问

mtv

访问可视化结果流程

Echarts环境准备：

为了生成ECharts 图表我们需要创建一个 HTML 页面，引入 echarts.min.js，为 了能够在页面上显示需要为ECharts 准备一个具备高宽的 DOM 容器，从BootStrap中文模板网下载相应的CSS依赖，将其修改为我们适合的样式。

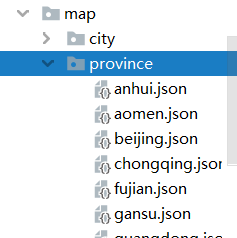


项目静态文件CSS文件目录树

（二）存在的问题

**坐标缺失**：爬取的网易云评论有的地区会到（区、县）一级，绘制区域活跃散点图的时候需要这些地方的经纬坐标，但现在的echarts版本只提供中国大地图和有限的市坐标，为了能够尽量展示区域活跃程度，需要在网上寻找第三方的数据。

**解决方法**：在Github上搜集相应的第三方数据，获得省份地图json 33个，市地图json 344个。

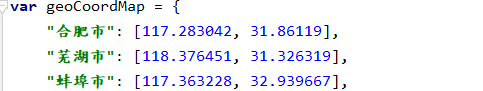


地图JSON数据

因为要画散点图，需要在大地图上标注省（市）的位置，根据这些json文件的组织形式，可以发现，里边"CP"字段有相应的坐标，通过python字典操作读取获得第三方的json，获得省（市）的名称和属性，将结果以文本文档的形式写入。部分代码如下：

1. **import** json
2. **import** os
3. **import** numpy as np
4. file\_path="map/cityJsonMap/"
5. file\_list = os.listdir(file\_path)
6. **for** province **in** file\_list:
7. file\_list2 = os.listdir(file\_path+province)
8. **for** filename **in** file\_list2:
9. filename=file\_path+province+'/'+filename
10. f\_obj=open(filename,'rb')
11. data=json.load(f\_obj)
12. f = open('coordxian.txt', mode='a+')
13. **for** i **in** data['features']:
14. f.write('\"'+i['properties']['name']+'\"'+':'+str(i['properties']['cp'])+','+'\n')
15. f.close()

将结果写入到变量geocoord中

写入坐标

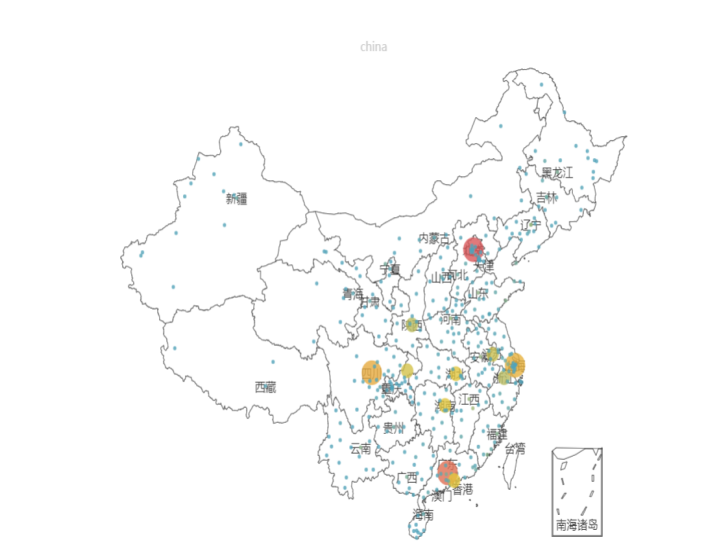
根据Echarts提供的API的数据格式，就能在地图上绘制原来没有城市的散点图。

**图表功能**：Echarts提供了相应的toolbox，有一些基础的功能，但为了满足我们的应用需要定义自己的toolbox。

**解决方法：**在图表设置里边设置我们自己的方法，在option里边的toolbox里的feature设置my开头的自定义图表工具。工具的ICON图标在阿里云[Iconfont-阿里巴巴矢量图标库](https://www.baidu.com/link?url=MChEgDR0e8qEFECbCV9VpGAKX2vZ-2zSkXgGOPdFnKB8xDFv1sObT9Mi0peowbL9&wd=&eqid=9360aca800035757000000065fb3eb16" \t "https://www.baidu.com/_blank)[官方](http://trust.baidu.com/vstar/official/intro?type=gw" \t "https://www.baidu.com/_blank)网站进行下载，将其转换为base64编码写入js脚本，这样我们就实现了相应的正态分布等功能。下面是还原图表原来分布的js代码。

1. myTool2: {
2. show: **true**,
3. title: '还原',
4. icon: 'image://../static/images/icon-r.png',
5. onclick: **function** (){
6. **var** oydata={......}
7. **var** oxdata={......}
8. option.xAxis.data = oxdata;
9. option.series[0].data = oydata;
10. myChart.clear();
11. myChart.setOption(option,**true**);
12. }
13. }

## 3.5.1基于用户地区分类生成地区热力分布散点图

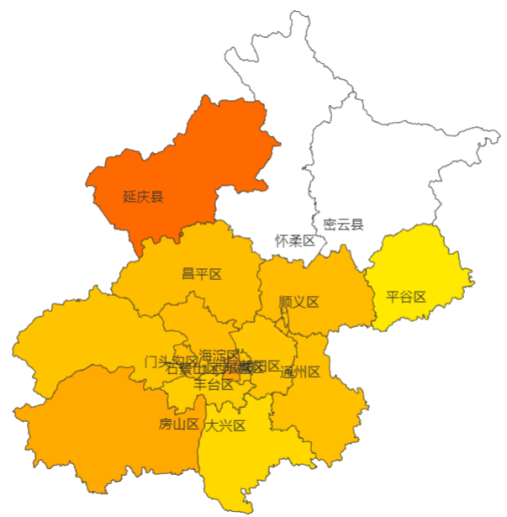


全国活动热力散点图

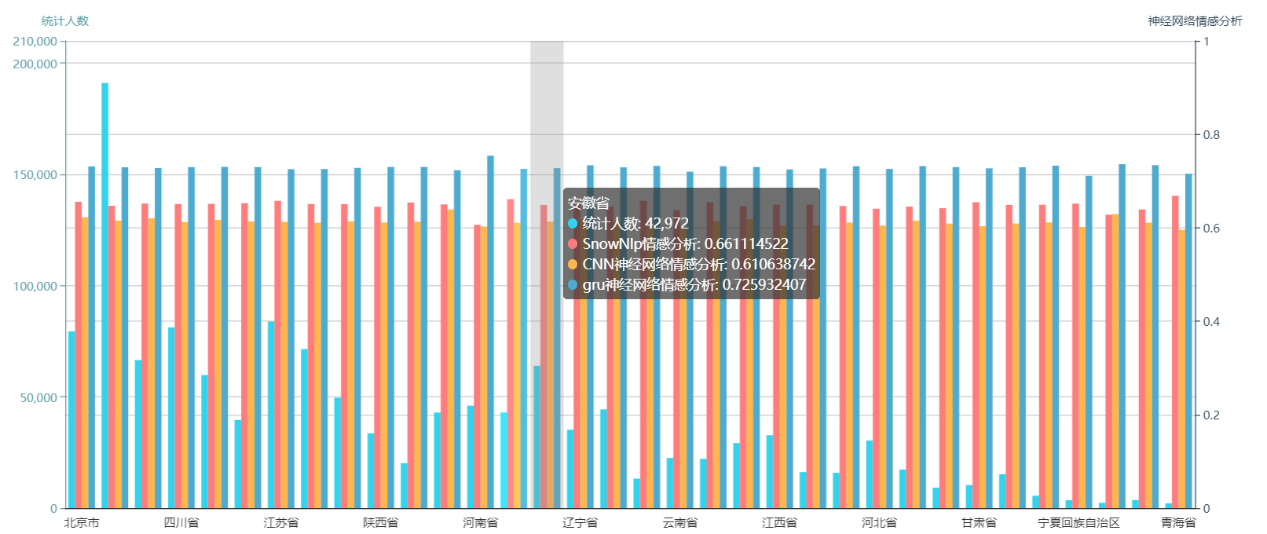
## 3.5.2 基于情感判断，分析某个地区用户的心情等级



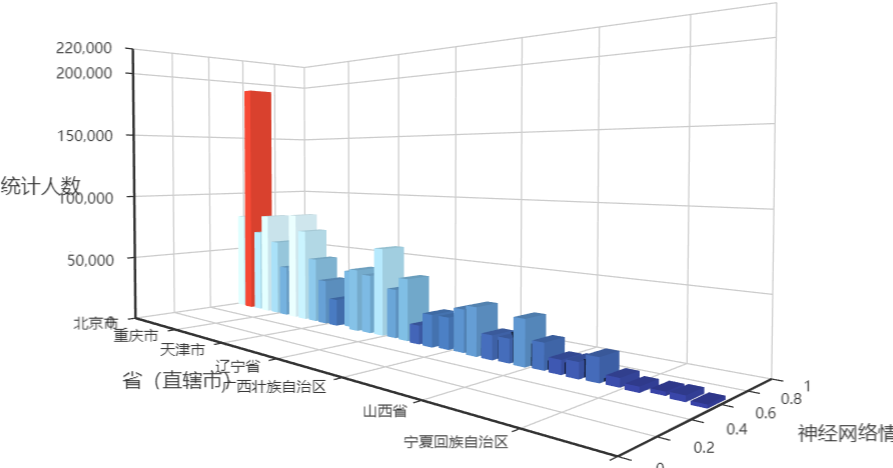
全国大地图地区心情等级



地图下钻北京市心情等级

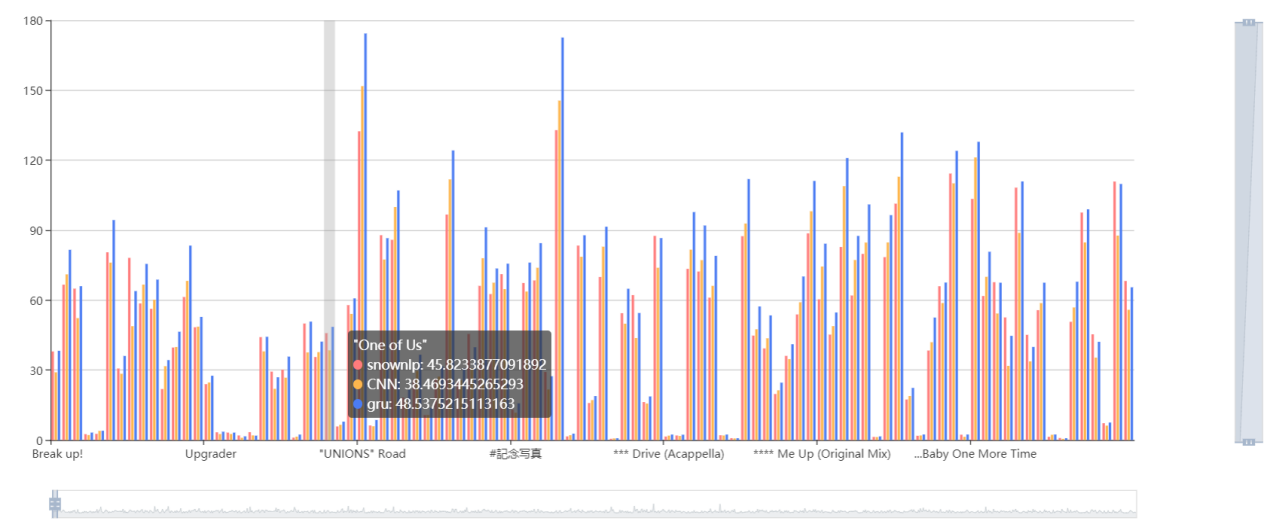


按省份分类算法对比柱状图

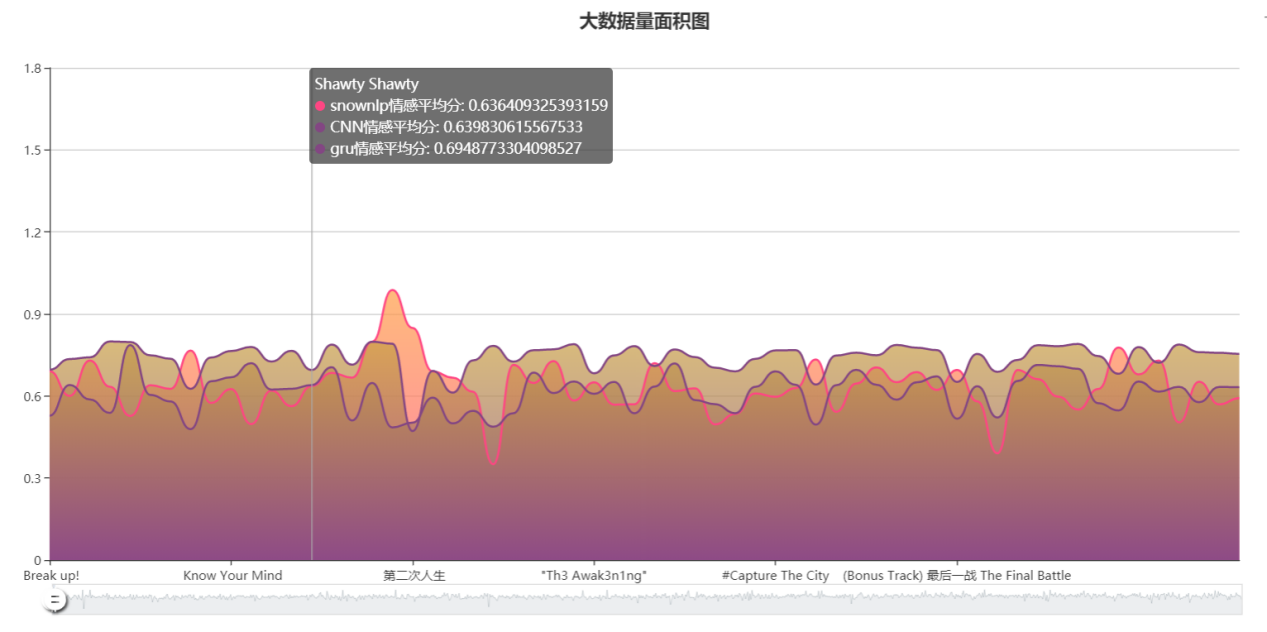


按省份分类心情等级柱状图（CNN 3D)

**3.5.3基于歌曲下的所有评论，进行情感分析**

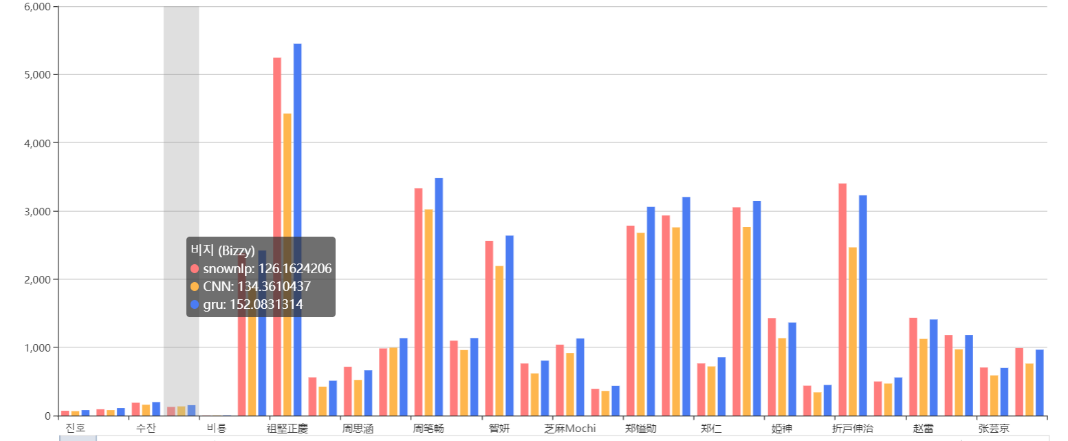


歌曲总得分算法对比柱状图（67533 data)

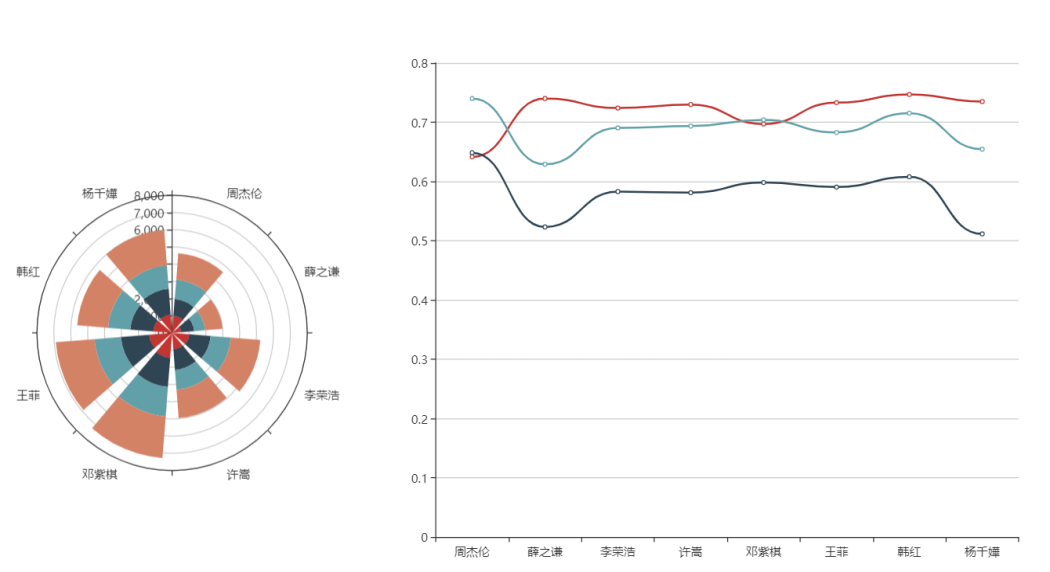


歌曲心情等级算法对比折线图

## 3.5.4基于歌手，进行情感分析

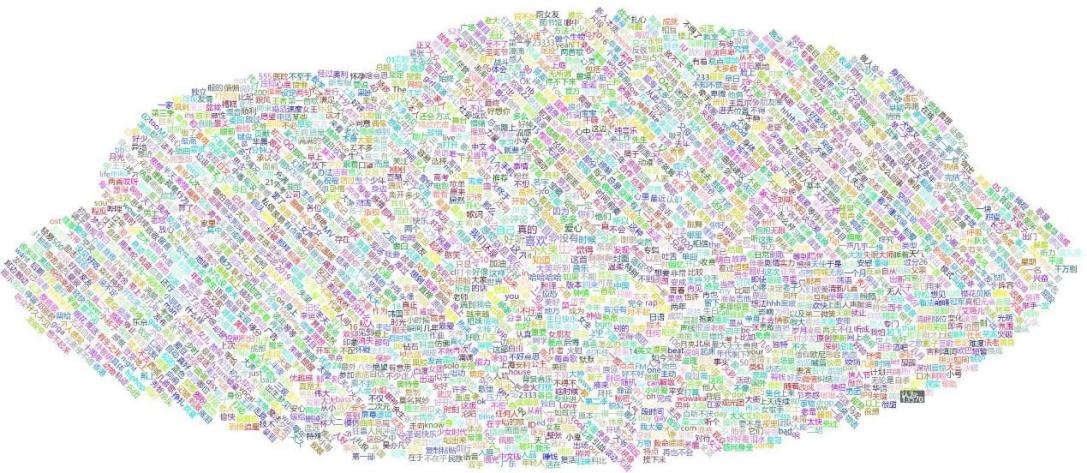


歌手情感总得分算法对比柱状图

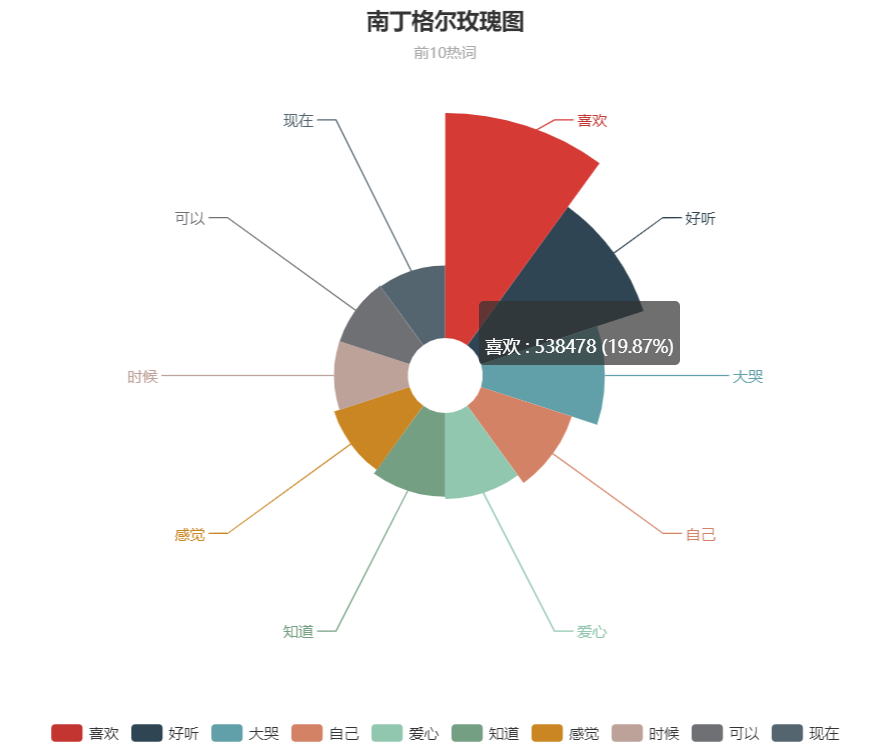


具体歌手的心情等级算法对比极坐标图，折线图

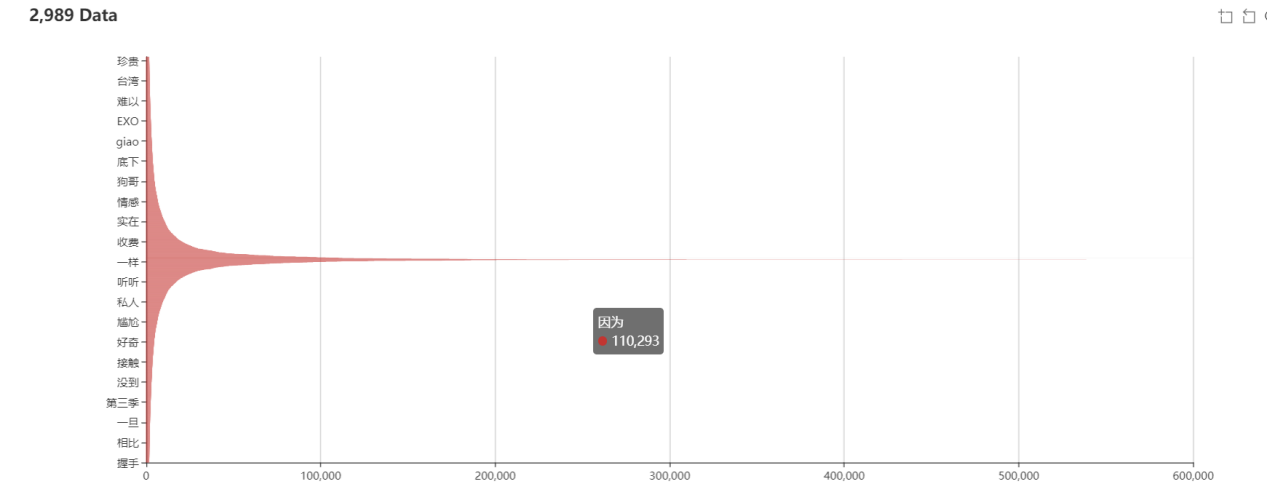
**3.5.5所有评论，词云展示**



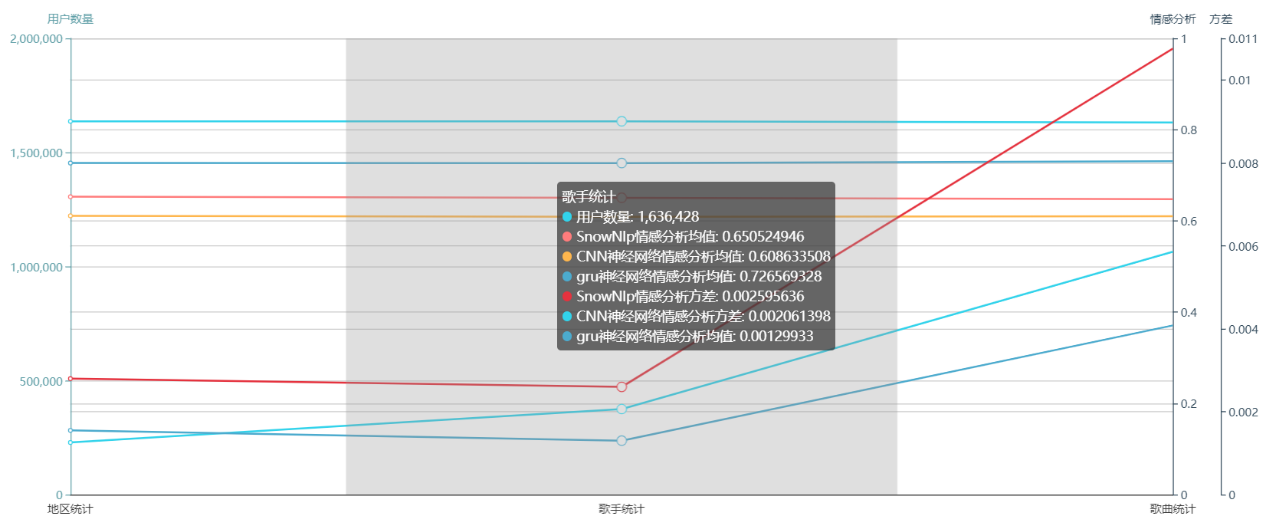
评论词云展示



南丁格尔玫瑰图（前10热词）



词频统计 正态分布图



最后的总结分析折线图

# 4 小结

本案例以网易云音乐评论情感分析问题作为主题，基于Hadoop&Spark集群环境，使用分布式爬虫爬取到的数据作为实验数据，结合snowNLP，GRU,CNN模型来对数据进行情感分析和比较，其中GRU以及CNN是通过手工标注的数据利用tensorflow实现并训练得到。主要内容包括原始数据集的爬取与预处理、网易云评论情感分析、数据可视化。该案例主题新颖，结合了经济社会现实需求与大数据分析与挖掘的多种理论与技术，可以充分增强学生的实践能力与理论基础。另外，本案例的内容仅为指导性的过程，在实际教学中，可保持基本研究内容不变，鼓励学生引入其它的数据预处理、数据挖掘、深度学习方法完成任务。

## 附录

1. 本案例提供配套的PPT、视频、数据集与代码等，发布于Github，链接为：https://github.com/Wanghui-Huang/CQU\_bigdata。

2. 本案例涉及到数据预处理以及多种机器学习算法，建议使用python语言进行编写，推荐的工具包有pandas（数据读取与预处理库），scikit-learn（机器学习算法库），Echarts（可视化绘图库）。

3. 本案例参考文献如下：

[1]吴宇鹏.分布式网络爬虫技术的研究与实现[J].电脑编程技巧与维护,2020(11):9-10+19.DOI:10.16184/j.cnki.comprg.2020.11.003.

[2]张胜敏,王爱菊.基于Python的分布式多主题网络爬虫的研究与设计[J].开封大学学报,2021,35(01):93-96.

[3]史国举.数据可视化技术在大数据分析领域的应用及发展研究[J].无线互联科技,2021,18(18):96-97.