

**课程名称：**Python语言与数据分析

课 程 报 告

项目名称

班 级

学 号

姓 名

任课教师

开课学期： 20 至 20 学年 第 学期

完成时间： 20 年 月 日

**参考方向**

**可参考下面方向自定研究题目，本页在正式报告中请删除！**

**新冠肺炎疫情数据分析与可视化**

从网络获取新冠肺炎疫情相关数据（中外均可），展示疫情发展变化情况、区域差异，可结合地图展示

**中国大学排行榜**

从具有一定权威性的网站获取中国大学在某个方面（如录取分数、毕业生薪酬等）的排行，分析地域差异，展示排名随年度变化情况

**某大学招生录取数据分析**

从某大学招生就业处或研究生院网站获取近若干年招生分数和人数信息，分析各专业、各省份差异及年度变化情况

**全球国家基本情况分析**

分析全球主要国家国土面积、人口数量、GDP等信息，结合地图展示

**中国大学国内期刊论文发表情况分析**

获取中国大学在国内期刊论文发表数据，对学校、个人进行排名，分析研究主题热点随时间变化情况

**沪深股票数据分析**

利用Tushare等平台获取沪深股票相关数据进行分析和可视化

**全国电影票房与备案数据分析**

利用Tushare等平台获取全国电影票房与剧本备案数据进行分析和可视化

**新闻数据分析**

利用Tushare等平台获取新闻通讯、新闻联播文字稿等数据，进行分析和可视化

**天气数据分析**

对若干城市某段时间以来的天气、空气质量等数据进行分析和可视化

**40个微观数据库**

参考《40个微观数据库.docx》

**其他自选方向**

自己编写爬虫程序，或从在线数据集平台（Datatang、kesci、阿里云天池等）获取你感兴趣的数据集做你感兴趣的分析。

**注意：禁止抄袭、复制同学或网上已有数据分析报告，如发现将按0分处理。**

**《**项目名称**》数据分析报告**

**目 录**

# 1 概述

爬取中国新闻网最近的新闻稿件，获得新闻稿件的内容数据，对数据进行清洗后进行分词，情感分析，以此来获取最近时间段社会上的热点事件或现象，社会风气等。

# 2 数据描述

数据来自于中国新闻网（https://www.chinanews.com.cn）的滚动新闻栏，通过request请求页面，获取html，然后通过BeautifulSoup处理html，方便我们进行关键信息的提取，首先提取每个新闻稿件的url信息，存入一个列表中，之后对这个列表进行一个迭代，依次request对每个url，对返回的html提取其中的新闻内容部分，提取完以后创建一个正则表达式，用来过滤掉我们所不需要的信息，例如文末编者，以及一些无意义的链接等。最后将其存入内容列表中。

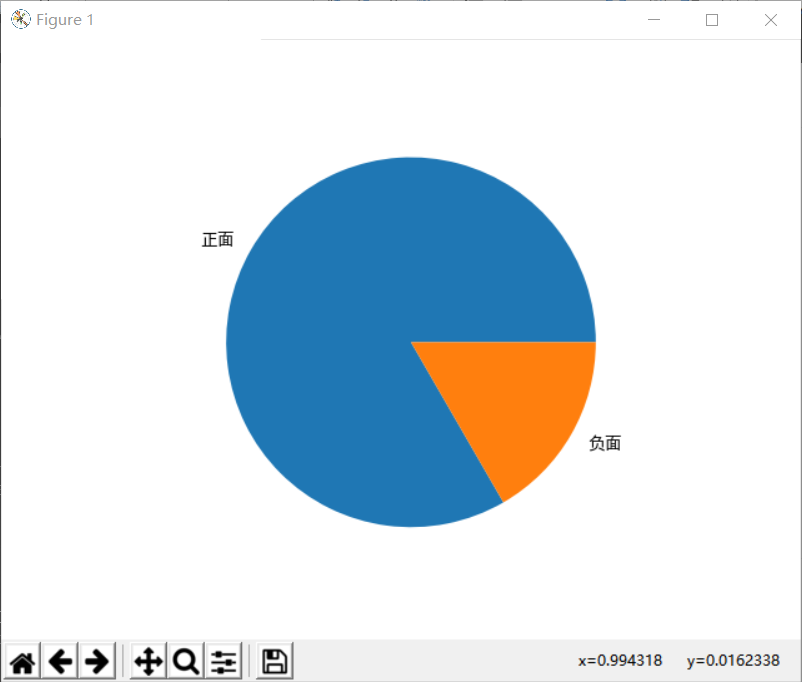
# 3 数据分析内容

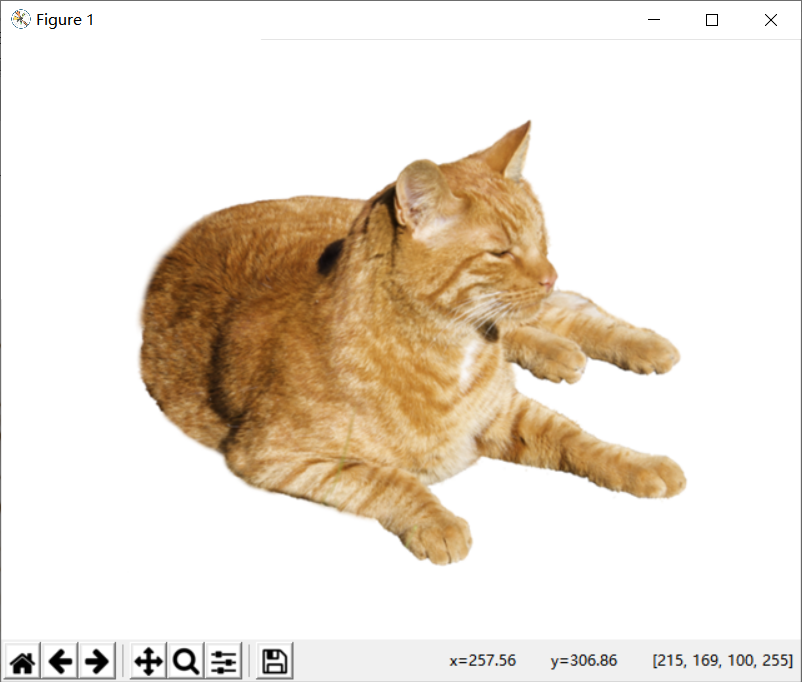
首先对上述取得的数据进行分句，以。 ？ ！等标点符号作为分割，就得到了以句子为单位的数据列表，之后利用自然语言中文处理库snownlp进行情感正负值以及主客观正负值的判断，以主客观为权重，情感正负值为得分计算出每一篇新闻稿件的情感正负值（0-1）将大于0.5的判定为正面新闻事件，将小于0.5的判断为负面新闻事件。

其次，对每一句话进行jieba分词，就得到了以词语为单位的数据列表，之后再将其中的停用词给去除，避免对结果精确度造成影响。得到目标列表后，利用collection系统库中的Counter函数对列表进行词频统计，统计出每个词语出现的频率，并按照频率降序，生成词频排序列表，利用plot画图库，画出词频较高的柱状图。

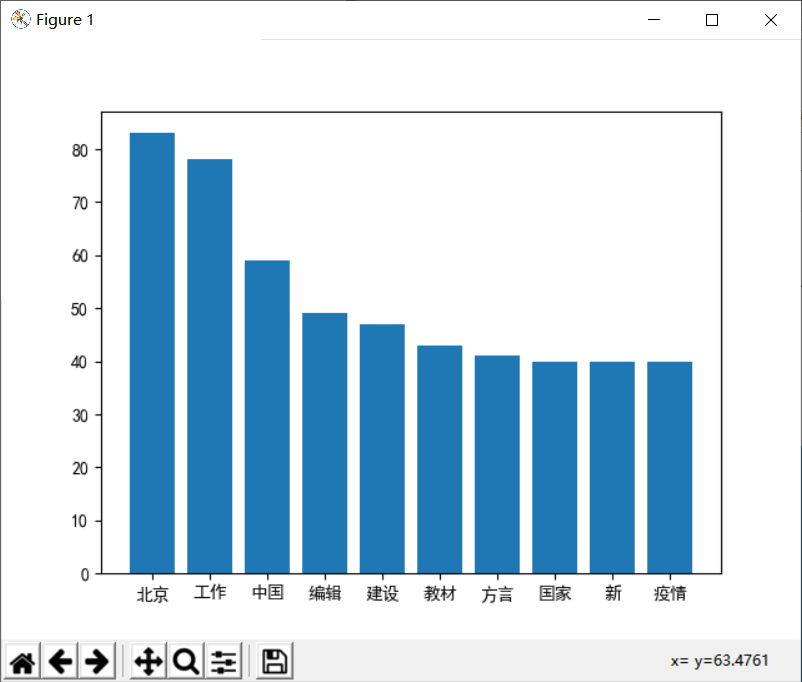
最后，将词频列表转换为字典的格式，利用wordcloud词云库处理，设置相对应的参数，生成词云图片，更直观的看出当下社会热点词汇。

# 4 数据分析图表









# 5 数据分析结果

不难看出最近社会上......

# 6 总结

文字总结整个数据分析过程、收获和体会

# 附录-数据分析代码

import requests

import wordcloud

import jieba

import re

from bs4 import BeautifulSoup

from collections import Counter

import matplotlib.pyplot as plt

import numpy

from PIL import Image

import csv

from snownlp import SnowNLP

url = 'https://www.chinanews.com.cn'

n = 1

#创建一个模拟浏览器的代理头，防止被反爬

headers = {

'User-agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/80.0.3987.163 Safari/537.36',

}

pos = 0

neg = 0

#通过request请求网页，获取html

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

response.encoding = 'UTF-8'

#通过BeautifulSoup处理html，方便我们进行关键信息的提取

soup = BeautifulSoup(response.text,'lxml')

link = str(soup.find(class\_='nav\_navcon').find(name='a')['href'])

url = 'https://www.chinanews.com.cn/' + link

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

response.encoding = 'UTF-8'

href = []

#创建一个正则匹配模式，方便后面进行数据的清理

pattern = re.compile(u'\t|\n|\.|-|:|;|\)|\(|\?|"')

#对新闻目录进行爬取，获得每个新闻的链接

for i in BeautifulSoup(response.text,'lxml').find(class\_='content\_list').find\_all(name='li'):

try:

href.append(['https://www.chinanews.com.cn' + i.find(class\_='dd\_bt').find(name='a')['href']])

except:

pass

# print(href)

content = []

#对爬取到的所有新闻链接进行内容的爬取，爬满10条为止

# print(href)

for i in href:

if len(content) == 30:

break

response = requests.get(i[0], headers=headers)

response.encoding = 'UTF-8'

#新闻页面共有2种html形式，若其中一种获取失败则换另一种进行获取

try:

try:

soup = BeautifulSoup(response.text,'lxml')

s = soup.find(class\_='left\_zw').text.replace('\u3000','')

s = re.sub(pattern,'',s)

if SnowNLP(s).sentiments > 0.5:

pos += 1

else:

neg += 1

content.append(s)

except:

soup = BeautifulSoup(response.text, 'lxml')

s = soup.find(class\_='content\_desc').text.replace('\u3000', '')

s = re.sub(pattern, '', s)

if SnowNLP(s).sentiments > 0.5:

pos += 1

else:

neg += 1

content.append(s)

except:

pass

#将爬取到的新闻内容对应上前面爬取的新闻链接

for i in range(len(content)):

href[i].append(content[i])

#只取前面30条方便我们进行分析

href = href[:30]

object\_list = []

for i in href:

#对每一条新闻内容，进行jieba分词处理，模式选择精确分词模式

seg\_list\_exact = jieba.cut(i[1], cut\_all=False, HMM=True)

with open('停用词库.txt', 'r', encoding='UTF-8') as meaninglessFile:

stopwords = set(meaninglessFile.read().split('\n'))

stopwords.add(' ')

#打开停用词文件，将分词成果去掉所有的停用词

for word in seg\_list\_exact:

if word not in stopwords:

object\_list.append(word)

sentiment = [pos,neg]

x = ['正面','负面']

plt.rcParams['font.sans-serif']=['SimHei'] #用来正常显示中文标签

plt.pie(x=sentiment,labels=x)

plt.show()

#打开我们预先选择好的词云背景图

mask = numpy.array(Image.open(str(n) + '.png'))

plt.axis('off')

plt.imshow(mask)

#设置词云的一些参数，字体，背景，词语数量以及词语大小等

wc = wordcloud.WordCloud(

font\_path = 'C:/Windows/Fonts/simfang.ttf',

background\_color='white',

mask = mask,

max\_words = 80,

max\_font\_size = 80

)

#利用Counter对词频进行统计

res = Counter(object\_list)

rows = []

for i,j in res.items():

rows.append([i,j])

#设置词云参数，生成词云图像

# print(object\_list)

wc.generate\_from\_frequencies(Counter(object\_list))

wc.recolor(color\_func=wordcloud.ImageColorGenerator(mask))

plt.figure('词云')

plt.subplots\_adjust(top=0.99,bottom=0.01,right=0.99,left=0.01,hspace=0,wspace=0)

plt.imshow(wc, cmap=plt.cm.gray, interpolation='bilinear')

plt.axis('off')

plt.show()

head = ['词语','频率']

#对词频进行排序，方便我们后续有序的写入文件中

rows.sort(key=lambda x:x[1],reverse=True)

print(rows)

bx = []

by = []

for i in rows:

bx.append(i[0])

by.append(i[1])

bx = bx[:10]

by = by[:10]

plt.bar(bx,by)

plt.show()

#写入csv文件

with open('词频分析.csv','w',newline='') as f:

f\_csv = csv.writer(f)

f\_csv.writerow(head)

for i in rows:

try:

f\_csv.writerow(i)

except:

pass