****

**2020—2021学年第2学期**

**《大学计算机》课程报告**

题目： 豆瓣图书的爬取与数据分析

|  |  |
| --- | --- |
| 学生姓名： |  |
| 学 号： |  |
| 专业班级： |  |

|  |  |
| --- | --- |
| **评价指标** | **教师评分** |
| 1）文档：结构完整，有条理；格式规范，排版好；语言通顺，错误少（20%）  2）数据获取与清洗：过程清晰，方法得当，描述清楚准确（30%）  3）数据处理、数据可视化：图表选择合理，方法恰当，描述清晰；界面美观、效果好；数据处理方法科学有效，描述完整、清晰（30%）  4）数据分析：分析合理，逻辑性好；结论、观点有说服力（20%） |  |
| **教师评语**  教师签名：  2021年 月 日 | |

**说 明**

1. 要求学生把自己实现的实验结果展示出来，重点描述对某个领域数据进行获取、分析、可视化的方法和过程，形成分析结果和结论。因此，课程报告主要包括任务要求、数据获取、分析、可视化展示等数据处理过程的内容。
2. 课程结束后，需要提交**课程报告（电子版）、相应数据文件和程序源代码**。未提交相关资料者不能参加成绩评定。
3. 课程报告要求参考本报告模板撰写，可根据具体情况适当调整，但文档结构要完整、格式排版美观、条理清晰、论述准确。
4. **文档撰写不符合要求，或者抄袭他人作品，记零分**。

**豆瓣图书的爬取与数据分析**

**一、任务描述（根据自己选择的网站，进行的数据处理分析等，对自己工作进行任务描述）**

使用python编写爬虫程序爬取豆瓣图书的数据，内容包括书的标题，作者，译者，出版社，价格，评分，简介以及详细链接。在爬取的时候对数据进行筛选处理，然后爬取数据并存入csv文件。然后使用plotly库对数据进行可视化，散点图是前十本图书和评分分布，柱状图是出版社以及其数目，饼状图是书本的评分状况。

**二、数据获取与预处理（描述清楚数据来源，使用工具，操作步骤，程序结果）**

1．数据获取

目标网站:

<https://book.douban.com/tag/%E4%BA%A4%E4%BA%92%E8%AE%BE%E8%AE%A1?start=%7Bi%7D&type=T>

通过手动翻页来构造url

对于每个网页都有多种书籍，通过分析网页的源代码，并使用xpath工具来获取想要的内容。





2.编写程序

程序的具体代码在文件的最后

1. 爬取数据：爬取的代码都在get\_urls\_book函数中，通过requests库以及lxml库一起实现，并加以错误处理。
2. 数据保存：通过python的csv库来对数据保存成csv格式的文件，方便后面的数据可视化。

3.结果



**三、数据分析**（**详细描述清楚你的数据分析工作步骤及结果，包含程序实现、运行结果）**

1.译者

有些书是国内的书，有些书是国外的书，这就导致有的书籍没有译者，这就需要特别处理，如果datas的长度是5说明有译者，为4就是没有译者。

2.评分

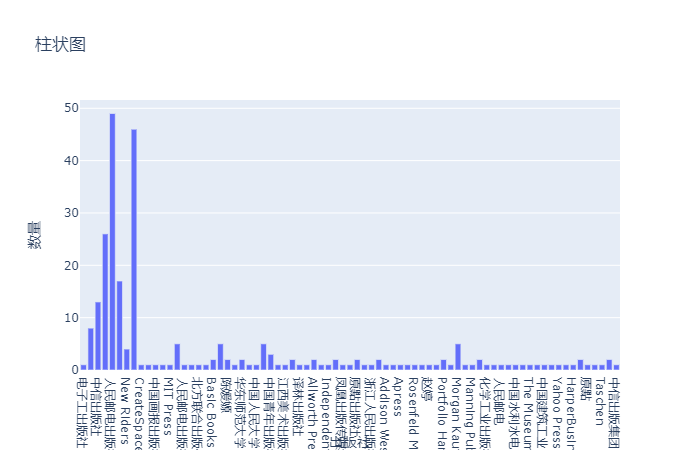
有的书本有评分，但有的书由于评价的人数太少就没有评分，爬取不到的时候会发生IndexError，这时需要错误处理，如果发生错误则为特定的文本。

3.简介

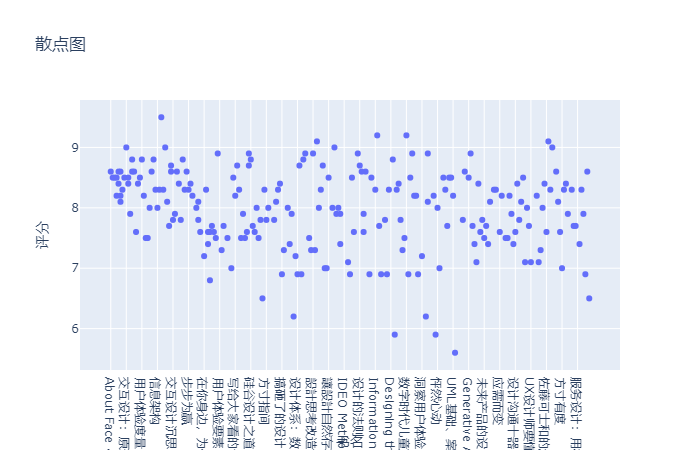
简介也是同理，有的没有简介，则需要抓取错误并作说明

**四、数据可视化展示**（**详细描述清楚数据可视化的工作步骤及结果，包含程序实现、运行结果，得出分析结论）**

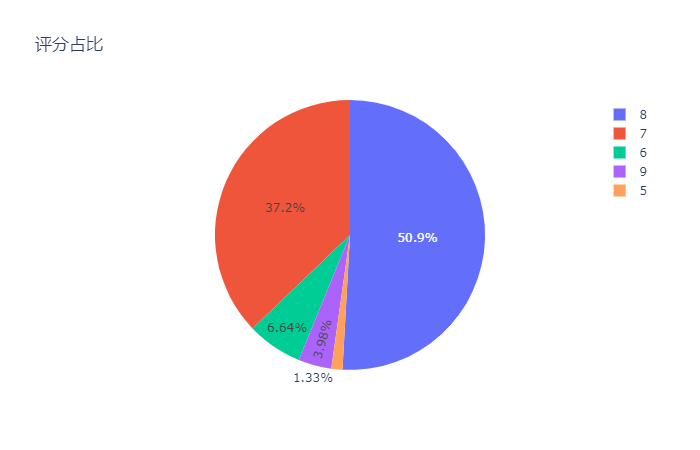
数据可视化任务由python的plotly库实现，



通过柱状图可以看出，人民日报出版社出版的书籍的数目最多，其次是电子工业出版社



可以看出评分的分布比较均匀，在6-9之间



从图中可以看出8-9分的占比最多，最少的是5-6分，说明平均分值还是很高的。

**五、遇到的问题与解决方法（详细说明遇到的问题、问题分析过程、及最终解决办法）**

**1、**爬取的时候发生错误，发现有的书本没有译者，于是加了一层判断来解决这个问题。

**2、**第二次发生错误是因为有的没有评分，发生了IndexError，于是加了个错误处理，让没有评分的变成了评分少于10人。

**3、** 最后一个错误就是简介问题，有的没有简介的就给说明没有简介。

**六、总结（需与课程相关内容总结，避免空泛）**

本次报告使用了python语言对豆瓣图书的数据进行了分析并可视化，其中用到了requests lxml pandas等库，方便了代码的编写。通过本次课程，大大提高了数据爬取以及分析的能力。并且增强了数据可视化的能力。

在编写的过程中为了方便阅读，对代码进行了注释。代码格式符合pep8规范。

数据可视化使用了plotly库，图标的外表比较美观。

总之，在这次报告编写的过程中收获满满，提高了自己的能力，同时也有许多可以改进的地方，比如可以引入面向对象来提高代码的可拓展性。

**附、完整代码（此处粘贴代码，并对每一部分功能作必要说明，不要截图）**

# 导入库

import requests

import csv

from lxml import etree

# 文件名

file\_name = 'datas.csv'

# 请求头

headers = {

    'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.88 Safari/537.36'

}

#开始的页数

start\_page = 0

# 要爬取的页数

end\_page = 260

# 构造urls

urls = [

    f'https://book.douban.com/tag/%E4%BA%A4%E4%BA%92%E8%AE%BE%E8%AE%A1?start={i}&type=T' for i in range(start\_page, end\_page+1, 20)]

# csv文件配置

fp = open(file\_name, 'wt', *newline*='', *encoding*='utf-8')

writer = csv.writer(fp)

writer.writerow(('标题', '作者', '译者', '出版社', '价格', '评分', '简介', '详细链接'))

# 获取数据并保存到文件

def get\_urls\_book(*u*):

    global authors, price, press

    res = requests.get(*u*, *headers*=headers)

    sele = etree.HTML(res.text)

    lis = sele.xpath(r'//\*[@id="subject\_list"]/ul/li')

    for li in lis:

        book\_url = li.xpath('div[2]/h2/a/@href')[0]  # 书本的具体信息地址

        title = li.xpath('div[2]/h2/a/text()')[0].strip()  # 书的标题

        datas = li.xpath('div[2]/div[1]/text()')[0].strip("\n")  # 书的多种数据

        datas = datas.split('/')

        translator = ""  # 有的书有译者，有的书没有译者

        if len(datas) == 5:  # 有译者

            authors = datas[0].strip()

            translator = datas[1].strip()

            press = datas[2].strip()

            price = datas[4].strip()

        elif len(datas) == 4:  # 无译者

            authors = datas[0].strip()

            press = datas[1].strip()

            price = datas[3].strip()

        try:

            score = li.xpath('div[2]/div[2]/span[2]/text()')[0].strip()  # 书的评分

        except IndexError:  # 如果发生索引错误，说明评价人数少于10人

            score = "评价人数少于10人"

        try:

            introduction = li.xpath('div[2]/p/text()')[0].strip()  # 书的简介

        except IndexError:  # 如果发生索引错误，说明没有简介。

            introduction = "暂无简介"

        if translator != "":  # 有译者

            writer.writerow((title, authors, translator, press,

                             price, score, introduction, book\_url))

        elif translator == "":  # 没有译者

            writer.writerow((title, authors, " ", press,

                             price, score, introduction, book\_url))

# 主函数

if \_\_name\_\_ == '\_\_main\_\_':

    for url in urls:

        get\_urls\_book(url)

        print("完成-----------------")

    print("所有工作已完成！")  # 结束

柱状图.py

import plotly.graph\_objects as go

import pandas as pd

from collections import Counter

# 读取爬取的数据

datas = pd.read\_csv("datas.csv", *encoding*="ANSI")

press = datas["出版社"]

counter = Counter(press)

bar = go.Bar(*x*=list(counter.keys()), *y*=list(counter.values()))

figure = go.Figure(bar)

figure.update\_layout(

*title*="柱状图",

*yaxis\_title*="数量"

)

figure.show()

散点图.py

import plotly.graph\_objects as go

import pandas as pd

# 读取爬取的数据

datas = pd.read\_csv("datas.csv", *encoding*="ANSI")

scatter = go.Scatter(*x*=datas["标题"][:10],*y*=datas["评分"][:10],*mode*="markers")

figure = go.Figure(scatter)

figure.update\_layout(

*title*="散点图",

*xaxis\_title*="书名",

*yaxis\_title*="评分"

)

figure.show()

饼状图

import plotly.graph\_objects as go

import pandas as pd

from collections import Counter

# 读取爬取的数据

datas = pd.read\_csv("datas.csv", *encoding*="ANSI")

scores = datas["评分"]

last\_scores = []

for s in scores:

    if s != "评价人数少于10人":

        last\_scores.append(float(s))

counter = Counter(map(int,last\_scores))

pie = go.Pie(*labels*=list(counter.keys()), *values*=list(counter.values()))

figure = go.Figure(pie)

figure.update\_layout(

*title*="评分占比"

)

figure.show()