

人工智能原理及应用

实验报告

实验名称：《乘风破浪的姐姐》爬虫与数据分析

学 号： 202205490501

姓 名： 蔡雨彤

专业班级： 自动化2202

学 院： 信息工程学院

指导教师： 付明磊

目录

[一、实验目的 3](#_Toc22536)

[二、实验设备 3](#_Toc25573)

[三、实验内容 3](#_Toc5296)

[四、实验过程 3](#_Toc6684)

[五、实验结果分析 3](#_Toc17310)

[六、实验小结 3](#_Toc22075)

[七、其它 3](#_Toc31657)

## 一、实验目的

本实验通过《乘风破浪的姐姐》百度百科界面，让我们熟悉python基本代码的编写，并在此基础上学习基础的爬虫操作，尤其熟悉requests库和beautifulsoup库的函数调用。

同时，在实验的过程中，会不可避免地遇到反爬机制，这将促使我们不断深入思考改进代码，如何避免反爬。

## 二、实验设备

thinkbook14+；pycharm；anaconda自建环境安装pandas、numpy、requests、matplotlib、beautifulsoup等python库；python3

## 三、实验内容

在实验的过程中，由于我习惯性使用pycharm软件编程，并且更倾向于利用conda在本地新创建一个python环境，避免包的版本冲突问题，所以我打算继续使用pycharm本地环境。

同时，为了更好的理解代码的含义，更好的理解作者的代码风格，我选择将每一模块的代码阅读理解后，再敲一遍，并反复思考它的含义。

在本次实验中，我们主要分为四个部分，首先是浪姐个人信息的爬取，然后是解析寻找到我们需要的内容，接着是将信息以json数据的格式存储到本地，最后一步是利用matplotlib进行可视化分析。

## 四、实验过程

1.百度百科信息的爬取

利用requests.get()函数向浏览器发送请求，返回值是response，状态码200表示成功接收。requests有两个参数，url代码发送请求的网址，headers是请求头，代表本次请求是浏览器在访问，防止反爬。

得到response之后，使用beautifulsoup解析响应。

这里使用的find\_all()函数寻找到所有标签为div，data-module-type为table的所有html信息。find\_previous()函数返回距离最近的包含指定标签的html。

在这里我发现AIstudio上的源代码有误，我当时尝试了很久发现根本没有按照要求解析成功，后来发现是因为find\_all()函数和之后的previous搭配错误，previous的指定值根本没有迭代到“\*（按姓氏首字母排序）”。在不断查看网页html界面并且更改了find\_all()的参数后，此bug解决。

因为我们不需要整个页面的信息，因此我们只需要找到所有女嘉宾的信息即可。在页面上我们可以发现所有参赛女嘉宾的信息是以table形式罗列的，并且在列表的正上方有一个此页面唯一的语句“\*（按姓氏首字母排列）”，因此我们根据这两条线索检索到我们需要的信息，并作为返回值返回。

2.浪姐个人信息的采集

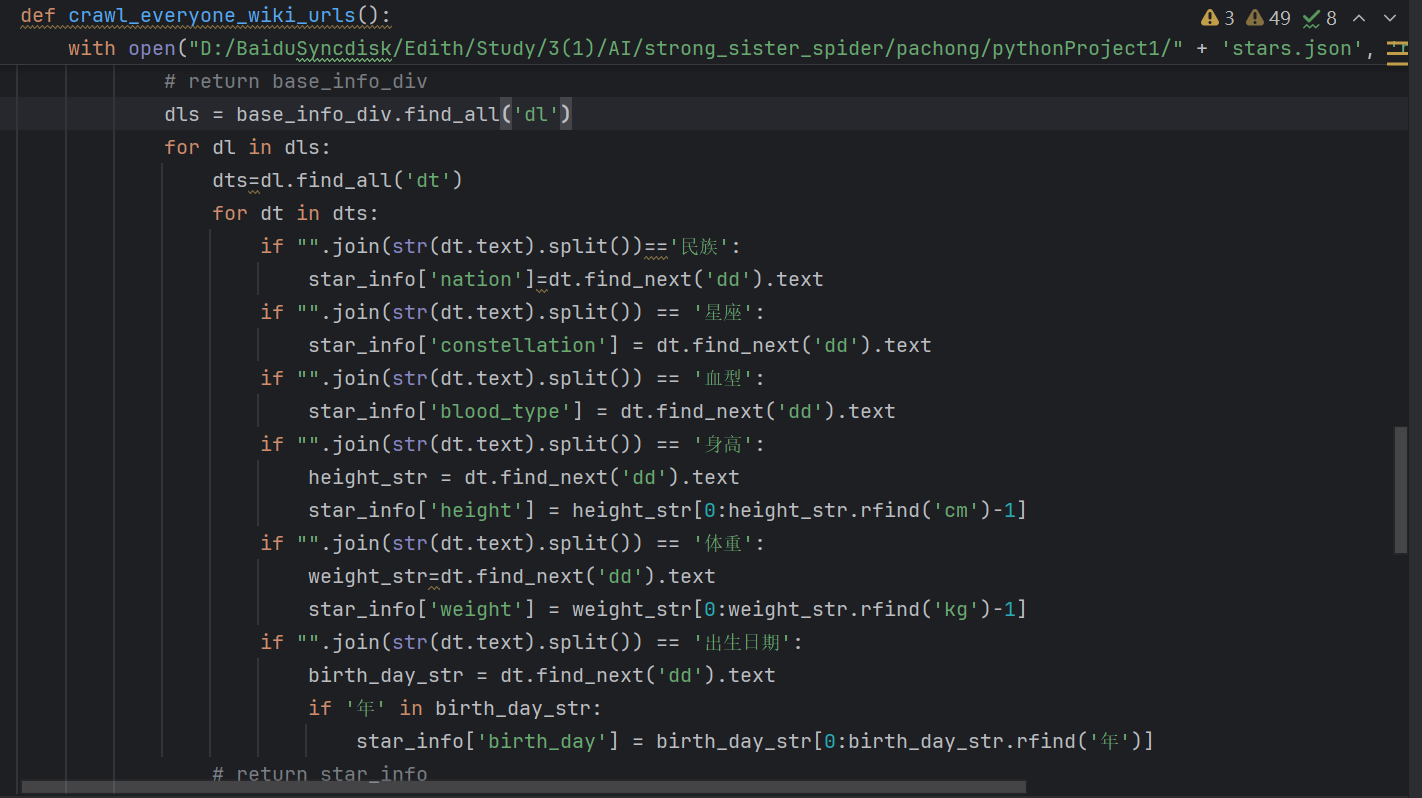


由于浪姐页面只有姓名的罗列，对于每位女选手的个人信息没有详细的介绍。但是好在该界面提供了URL可供跳转至每位参赛选手的个人信息界面。

因此在此段代码中，我们收集了每位姐姐的姓名和对应的个人界面链接，每一个人的信息都以字典的形式存储，所有人的信息将合并成一个列表结构。最后将这个列表存储在本地。

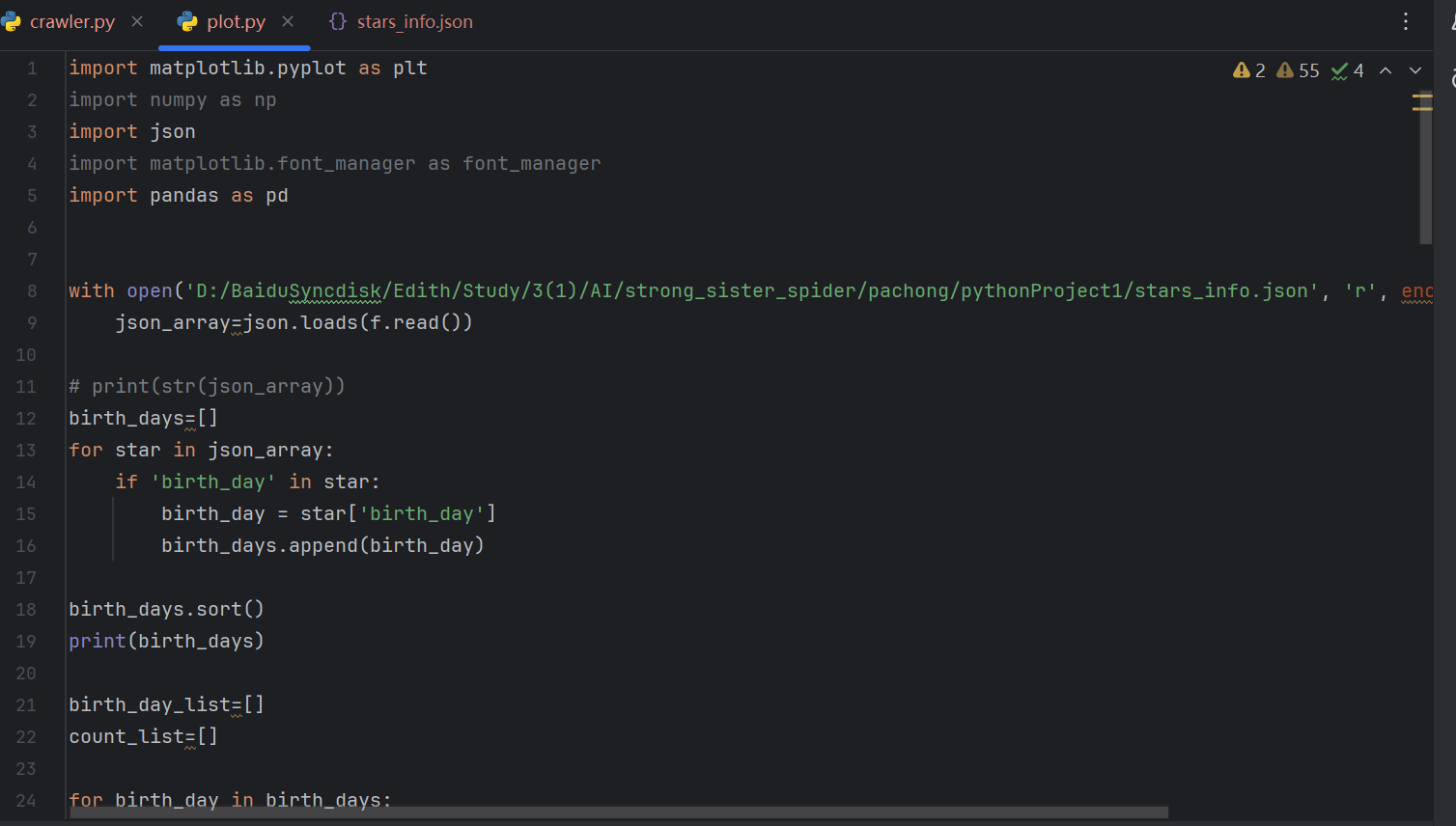
3.对每位女嘉宾的个人信息进行整合





我们通过访问之前已经存在本地的信息，然后使用requests对每位浪姐的个人百度百科界面发起请求，并且通过观察网页的html界面提取出浪姐的民族、星座等个人信息。每位朗杰的信息汇总成一个字典元素，所有字典再汇总成列表，保存在本地。

1. 可视化分析
2. 年龄柱状图





我们首先在本地存储的信息文件中提取所有的年龄值，并存储在列表中进行排序。接着我们利用列表的count()函数对每一个身高值进行统计。这段代码我个人认为很值得学习的是他对于这种计数统计分类的设计。我之前可能会先直接使用for循环，但是这里它使用了

## 五、实验结果分析

内容

## 六、实验小结

内容

# 七、其它

内容