

1 引言

1.1 课题研究背景和研究现状

1.1.1 课题研究的背景和目的

随着互联网技术日益成熟,越来越多的应聘者开始通过网络投简历,很多用人单位也通过招聘网站发布信息。为满足社会的需求,高校需要分析用人单位对人才的需求特点,有针对性地调整人才培养方案和设置相关课程,培养出更多的优秀人才。应聘者也需要从网络招聘信息平台获取信息,做好面试前的准备工作,因此从互联网上获取招聘信息变得十分重要^[1]。

1.1.2 课题研究的现状

作为搜索引擎技术核心元素之一,自 1993 年初 Matthew Gray's Wandering 在麻省理工学院开发出有史记载的第一个网络爬虫以来,爬虫技术历经 20 多年的发展,技术已日趋多样。为满足不同用户多种多样的需求,创建开发了类型众多的爬虫系统。

1.2 课题的研究意义和主要工作

1.2.1 课题的研究意义

在国内对高等教育改革正迈向更深层次的时候,各校的毕业生规模也逐年增加。临近毕业时,或多或少都会存在许多迷茫。而在招聘、应聘的过程中,互联网作为当下承载海量招聘信息的重要载体,则给毕业生的择业提供了一条便捷途径。只是互联网的信息检索中却会面临许多用户并不需要的信息,只有通过人工筛选、再经总结对比后,才能得到最终想要的信息。

为了帮助高校毕业生在择业时能够快速获取特定的需求信息,并且通过快速数据分析得到自身择业的准确定位,从而做出更好的选择,为此本文设计研发了一套针对于招聘就业的专用爬虫。这里即以 BOSS 直聘作为实例,对如何开发爬虫获取信息,及对获取的信息快速分析进行了深入探讨与研究。

1.2.2 课题的主要工作

爬虫系统的设计思路为:首先,需要获得所有包括岗位信息网页的源码;其次,在每一页的网页源码中寻找出与需求相匹配的信息,此时就需要连接爬虫系统和数据存储介质,将每次成功匹配到的信息均存入文件或者数据库中,直至所有网页检索完毕。在数据爬取的整个过程中,针对 BOSS 直聘的高度反爬,还要在各个模块中引入适当的反爬策略,这里我们主要使用 selenium,并且控制爬取频率,以此保证数据爬取的连续性。

1.2.3 课题的结构安排

确定研究对象、需求分析、系统设计、系统实现、系统测试和项目总结。

1.3 开发工具及开发环境

开发工具: PyCharm 2022.1.2 (Professional Edition)

Microsoft Edge 102.0.1245.44 (正式版本) (64 位)

开发环境: Windows11 专业版

1.4 本章小结

第一章为综述，旨在介绍理论背景与选题渊源。首先介绍了当代互联网发展的总趋势与技术前瞻；在讨论了基础必要性以及信息获取占有率之后，我们分析了信息源的存在方式为网站的网页形式；在锁定抓取对象之后，我们通过梳理爬虫技术发展进程以及概述爬虫技术发展困境的方式介绍了本课题需要应用的爬虫技术，其中，爬虫技术发展困境包括日渐高频且繁复的网站信息变化更新的状态、网站开发设计人员最原始的抗拒性阻隔、网站信息重复的、多次的进行无意义的抓取。进而，我们引出了本次课题的研究价值，帮助毕业生在择业时能够快速获取特定的需求信息，并且通过快速数据分析得到自身择业的准确定位，从而做出更好的选择。

2 需求分析

2.1 系统需求及分析

网络爬虫系统的开发是否成功取决于确保系统能够实现用户定制功能，达到预期设计目的。因此，在网络爬虫系统开发之前，就需要对该系统需求加以详尽分析，从而对整体的设计有一个清晰的思路。时下，普遍适用的爬虫系统都是模块化的，模块化的程序设计有利于代码块的测试与维护，而且也进一步增加了代码的适用性。在此基础上，只要对各个模块进行组合，就能够构建出一个完整的爬虫系统。本次研究即以 BOSS 直聘为例，开展模块化的编程设计。因为研究旨在通过爬虫系统对当前就业做出科学分析，故而针对此需求就要从 BOSS 直聘网站中获取全部的岗位信息，以及从每个岗位中获得包括各岗位名称、工作地点、薪水、公司规模性质、工作要求在内的各种关键信息。至此，在接下来的功能、模块设计中，就具备了较强的针对性。

2.2 本章小结

分析网络爬虫系统的开发的需求，定制模块、确定信息。

3 方案系统设计

3.1 概要设计

分为三个模块：爬虫模块、数据分析处理模块、数据展示模块。

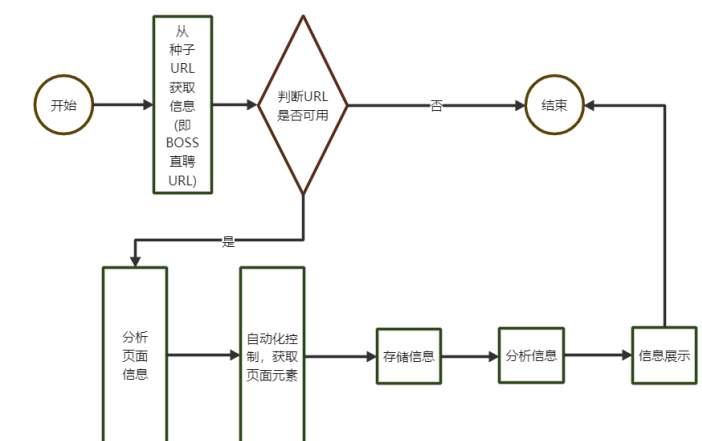


图 1 整体设计框图

3.2 详细设计

3.2.1 爬虫模块设计

3.2.1.1 主要技术选定

3.2.1.1.1 爬虫原理

爬虫就是获取网页并提取和保存信息的自动化程序。可以将爬虫总结为 4 个步骤：（1）通过一些接口，向网站的服务器发起请求，对方服务器给予响应，得到的响应为 html、css、js 的源码。（2）在获取源码后，通过 xpath、re 或者 bs4 等库进行解析，通常 re 正则表达式的提取效率是其他表达式的十倍以上，因此本文采用 re 正则表达式。（3）将数据进行持久化的存储，例如保存到 csv 文件，或者 mysql、Mongo DB 等数据库中。（4）对各种异常处理、错误重试等操作进行处理，自动化地完成网页访问、内容提取和持久化的操作^[2]。

3.2.1.1.2 Selenium 研究背景

Selenium 是一种浏览器测试自动化，通常用于 Web 爬虫程序中的手动操作自动化。这是一种基于网络的爬虫技术。它支持 Internet Explorer，谷歌浏览器，Safari，Opera 等等^[4]。Selenium 支持控制浏览器对象元素的浏览器驱动程序。Selenium 还支持许多语言，比如 Java、C、Ruby 和 python。Python + Se + chrome/firefox 组合通常用于解决现有网络爬虫绕过 JS 和滑块登录的问题。目前，Firefox 和 chrome 已经完全取代了之前的 phantomjs UI 免费浏览器，完成了所有工作并节省了内存。通过这种方式，chrome/firefox 呈现和解析 JavaScript，selenium 驱动浏览器和 python 之间的连接，python 处理后期处理。这三者构成了一个完整的轨道结构，可以实现点击链接、填写表单、点击鼠标、拖动鼠标等各种操作，给爬虫的操作带来了极大的方便。它可以解决一系列的手工操作问题，并为网络爬虫验证其登录信息提供了极大的方便。

3.2.1.1.3 基于 Selenium 的反爬虫流程

BOSS 直聘作为一个复杂、庞大的网站，通过查看 Robots 协议可以看见其并没有阻止用户对信息进行爬取，但在实际进行爬虫时，会发现其拥有着一系列的反爬虫机制。如在登录时的判断是人为还是机器自动控制。在获取商品列表时，若太过频繁会不断请求用户进行验证。若只是利用一般爬虫流程就会遇到好多阻碍，甚至连网站都无法登录。因此，需要针对 BOSS 直聘的反爬虫机制重新设置一套方案。

Selenium 的反爬虫流程，简单地归纳为以下几步：（1）通过 HTTP 库向目标站点发送 Request 请求，等待响应。（2）使用 Python 与 Selenium 配合用户一步操作，解决 BOSS 直聘动态 cookie 的反爬机制。（3）Selenium 进行自动化操作，自动链接跳转。（4）进行招聘信息的爬取，通过设置隐式等待，让 Selenium 躲过 BOSS 直聘的频繁获取验证，获取到对应的信息并离线保存到本地。（5）通过元素定位获取下一页的按钮并进行自动跳转。

[5]

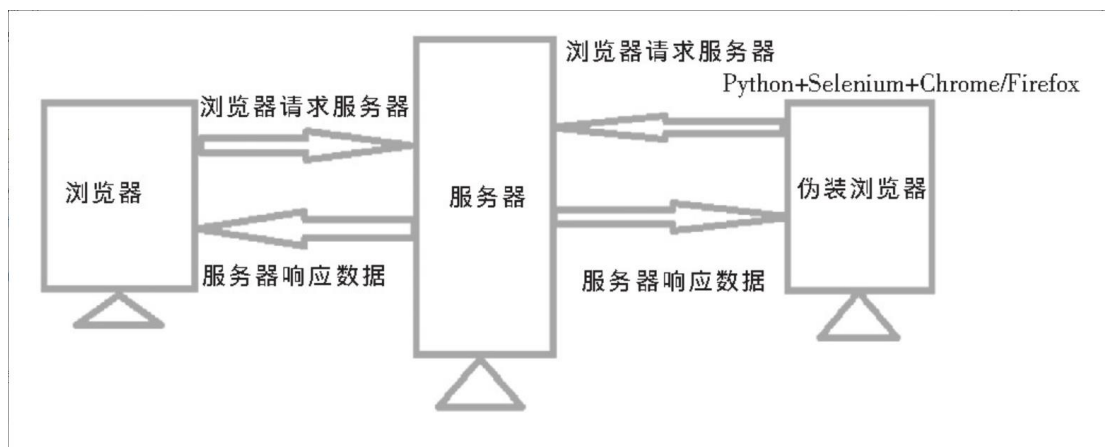


图 2 模拟浏览器进行请求数据^[6]

3.2.1.2 网页抓取模块

网页抓取模块作为爬虫系统中最重要的一部分，也是起始的模块。

首先是确定网址，通过分析，网址中我们需要的传去的属性是 query 和 city，query 指的是需要搜索的职位，city 后面的一串数字，如：100010000 代表全国、101010100 代表北京、101020100 代表上海，这样就可以直接通过访问网址访问，不需要特意在搜索框或者选择栏，手动选择应聘者需要的职位和城市。

另外，从实际爬取的情况来看，BOSS 直聘的反爬策略相对更复杂，采用了动态 cookie、IP 和 User-Agent 的检测、验证码等。因此，针对同一个 IP 在短时间内的多次爬取，会被网站屏蔽 IP 地址，所以在这里采用代理 IP 池的技术去访问。为了避免被对方发现，还需要加入 User-Agent 将自己伪装成代理服务器。通过构造代理 IP 池以及由众多用户代理组成的代理池，每次随机选择访问 IP 与用户代理的搭配，据此而将自己伪装成来自不同 IP 的用户访问，大大降低了被反爬虫的概率^[3]。

3.2.1.3 元素分析模块

打开 BOSS 直聘的网址，右键检查，分析当前文本，寻找用户需要的关键信息，根据用户的需求，还需要了解每一类工作的名称与对应网页链接，通过对 ELEMENTS 的寻找，发现名为 .job-primary 的 CSS 标签下有着所有我们需要的信息，包括：标题，地区，公司名，薪资，经验，公司类型，公司福利，详情页 URL，具体要求。

需要注意的是具体要求文本的获取，需要控制鼠标悬浮与公司名上，并等待一段时间才能获取到相关元素。

3.2.1.4 信息获取模块

主要借助 selenium 中的函数 find_element_by_css_selector()，获取到元素文本，存储进字典中，然后再将字典存入 CSV 文件，获取完一个页面的信息后可以定位元素 .next，并点击，实现翻页爬取。未登录的情况下，一次只能爬取十页，后续需要换城市或者换区来实现更多数据的获取。

3.2.2 数据分析处理模块设计

通过设计好的网络爬虫系统，从 BOSS 直聘网站上爬取了全国地区 500 条信息从招聘岗位、工资待遇、工作地点、工作要求、公司性质这几方面的信息，对全国范围的就业数据做出研究与分析，对广大择业人员可起到一个初步指导的作用。

对于就业数据分析来说，至关重要的就是薪资分析，将提取出来的 salary 列表，对每一个元素采用正则表达式匹配前两个数字，也就是这份工作的薪水上下限，求一个平均值，遍历整个列表，对薪水分布进行统计。同时，通过定位以及包含字符段的方法，可以将每个地区的工作以及相对应的薪水提取出来，再通过对全国各地区的工作岗位统计，对全国各地区的平均薪资做出分析。

3.2.3 数据展示模块设计

3.2.3.1 整体框架选择

- 使用 streamlit 作为整体的架构，在此基础上添加内容；
- 使用 wordcloud 分析技能需求，并做成词云；
- 使用 pyecharts 进行绘图。

3.2.3.2 展示原始数据

	标题	地区	公司名字	薪资	经验	公司类型	公司福利
0	数据分析师	福州 闽侯县-新西客站	扬腾创新	7-10K	1-3年本科	电子商务	节日福利，定期体检，带薪年假，股票期权，年终奖，五险一金，餐补，零食下午茶
1	数据分析师	北海 海城区-北京路	符号认知	3-6K	1年以内本科	广告/公关/会展	年终奖
2	数据分析师	广州	大牛教育	15-20K	3-5年本科	在线教育	年终奖，五险一金，员工旅游，节日福利，全勤奖
3	数据分析师	合肥 蜀山区-天乐	融沃科技	12-20K	3-5年本科	移动互联网	五险一金，股票期权，带薪年假，餐补，节日福利
4	数据分析师	天津 南开区-鼓楼	数能科技	4-8K	1-3年大专	互联网	五险一金，年终奖，节日福利，定期体检，带薪年假，零食下午茶，员工旅游
5	数据分析师	泉州 丰泽区-泉秀	品尚征信	10-15K	3-5年本科	互联网金融	员工旅游，五险一金，年终奖，定期体检，节日福利，带薪年假
6	数据分析师	天津 西青区-大寺	京创曙光	10-15K	3-5年本科	数据服务	住房补贴，员工旅游，五险一金，带薪年假，加班补助，节日福利，全勤奖，餐补，交
7	数据分析师	苏州 苏州工业园区-娄葑	英赛孚工业智能	12-24K	3-5年本科	计算机软件	带薪年假，员工旅游，零食下午茶，股票期权，节日福利，年终奖，五险一金
8	数据分析师	天津 西青区-西青大学城	福安易	8-9K	1-3年本科	医疗健康	带薪年假，五险一金，年终奖
9	数据分析师	扬州 广陵区-城东	万得扬州	4-8K	经验不限本科	互联网金融	带薪年假，节日福利，住房补贴，年终奖，股票期权，定期体检，补充医疗保险，餐补
10	数据分析师	上海 浦东新区-张江	叮咚买菜	20-40K	3-5年本科	互联网	五险一金，扁平化管理，节日福利，交通补助，餐补，零食下午茶
11	数据分析师	北京 朝阳区-三元桥	国美控股集团	15-25K	1-3年本科	互联网	五险一金，员工旅游，年终奖，定期体检，带薪年假，餐补，通讯补贴，节日福利，加
12	数据分析师	北京 昌平区-南口	三一石油	13-26K	3-5年本科	石油/石化	食堂，五险一金，带薪年假，节日福利，餐补，宿舍，年终奖
13	数据分析师-合肥	合肥 庐阳区-四里河	今日头条	12-20K	3-5年本科	移动互联网	加班补助，零食下午茶，全勤奖，节日福利，员工旅游，五险一金，交通补助，定期体
14	资深数据分析师	北京 朝阳区-太阳宫	BOSS直聘	25-45K	3-5年硕士	人力资源服务	交通补助，带薪年假，员工旅游，住房补贴，年终奖，补充医疗保险，餐补，股票期权
15	数据分析师（财务）	南京 建邺区	万得	4-6K	经验不限本科	数据服务	股票期权，节日福利，定期体检，餐补，带薪年假，住房补贴，补充医疗保险，五险一

3.2.3.3 展示技能需求

将详情信息用 jieba 库切割，并用空格连接，作为词云对象创建的输入，反复过滤不需要的高频词，留下的技能用来做成词云。



图 3 岗位职责

3.2.3.4 展示各地区岗位分析

读取文件，对各城市的招聘人数进行统计，然后由城市计数转为省份计数，将每个省份的招聘人数以地图的方式展示出来。

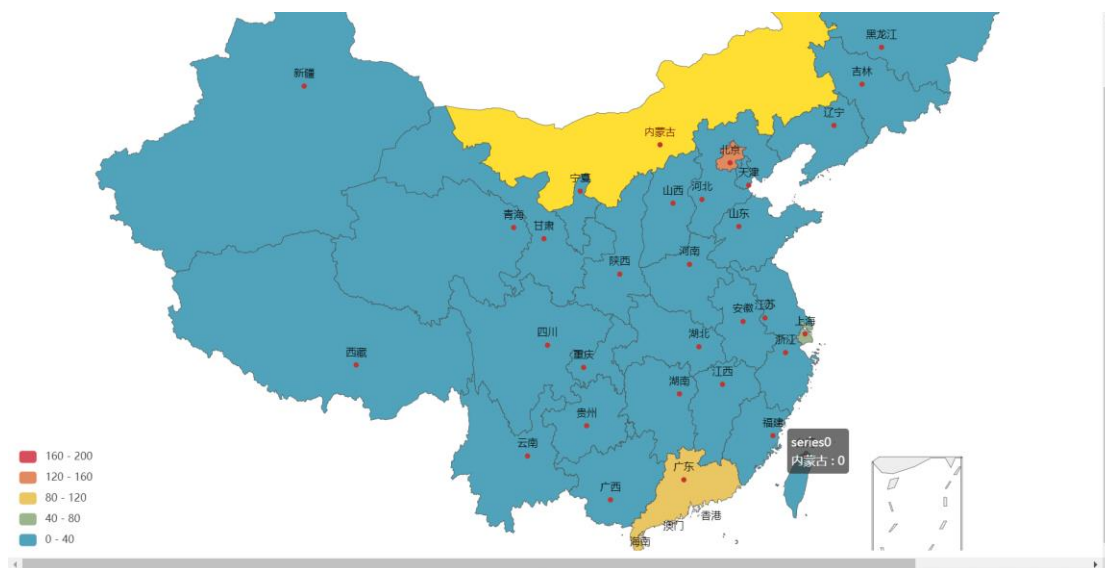


图 4 招聘岗位全国分布情况

3.2.3.5 展示薪资分析

读取数据，将薪资区间中的 k 去掉，并提取最小和最大薪资，将以城市为单位的数据转换为以省份为单位，分别计算最小和最大薪资的平均值，进而确定平均薪资。最后，在地图上展示均值，柱状图上展示薪资区间。

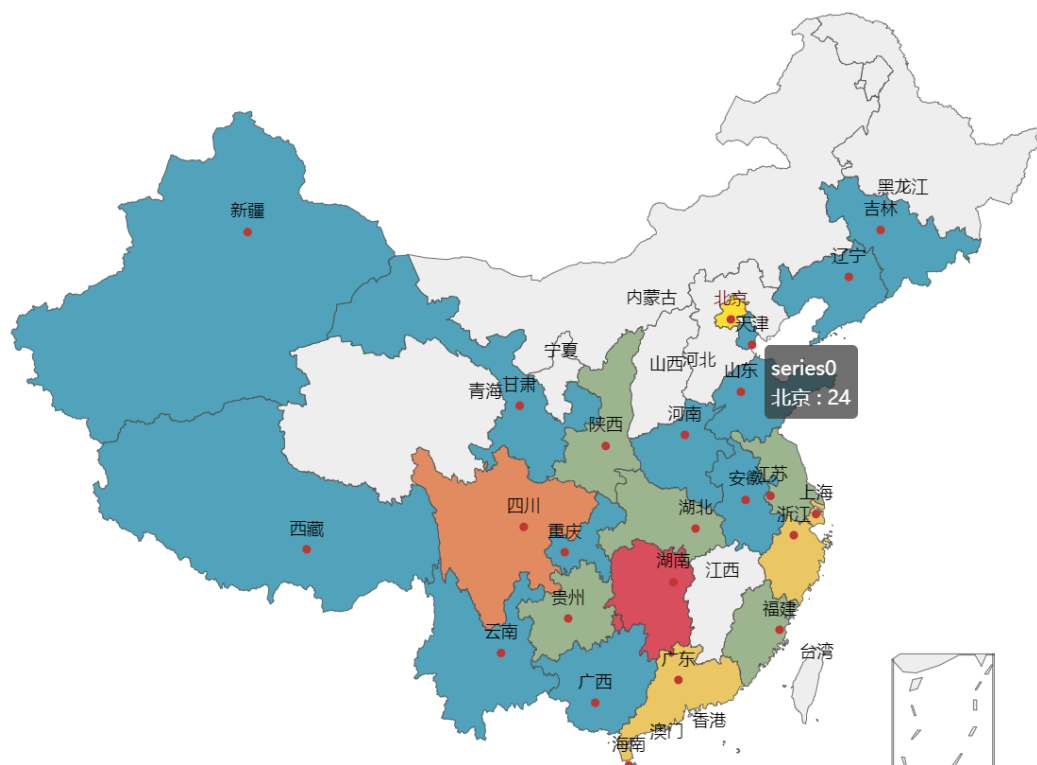


图 5 平均薪资地图

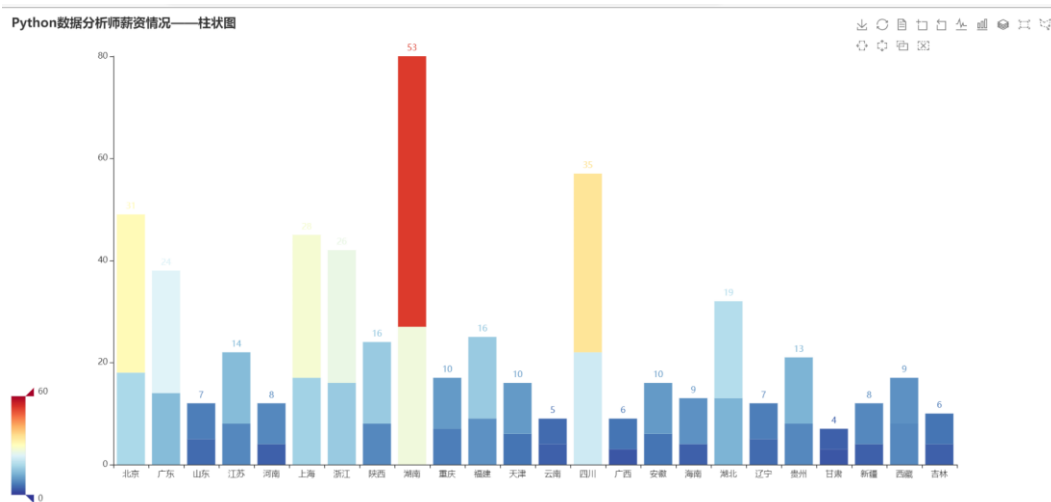


图 6 薪资区间柱状图

3.3 本章小结

第三章由大至小，探讨了整个方案系统设计

4 系统主要代码展示

4.1 主要代码实现

4.1.1 selenium 爬取信息模块

- (1) 实例化 driver 对象。使用 EdgeOptions()实例化 driver 对象。使用 add_argumentt 可以添加浏览器相关设置，如：设置浏览器窗口大小、添加 user-agent 等^[7]。关键代码如下：


```
chrome_options = EdgeOptions() # 添加参数
chrome_options.use_chromium = True
chrome_options.add_argument('--user-agent=' + UA + '"') # 添加 USER_AGENTS
chrome_options.add_argument('--start-maximized') # 窗口最大化
chrome_options.add_argument('--incognito') # 无痕
driver = Edge(executable_path=webdriver_path, options=chrome_options)#浏览器
驱动对象
```

(2) 确定网址。发起请求。关键代码如下：

```
keyword = input('请输入你想要搜索的岗位名字：')
driver.get(f'https://www.zhipin.com/job_detail/?query={keyword}&city=100010000&industry=&position=')
```

(3) 获取具体招聘信息。获取 html 页面内容可以通过 re 或者 xpath 等库实现。经过对比发现，CSS 是最简洁简单的方式，所以本文通过 CSS 定位元素，从而获取商品页面内容。关键代码如下：

```
need = li.find_element_by_css_selector('.job-area') # 具体要求
ActionChains(driver).move_to_element(need).perform()
time.sleep(5)
need =
li.find_element_by_css_selector('.detail-bottom .detail-bottom-text').text
# 具体要求
area = li.find_element_by_css_selector('.job-area').text # 地区
title = li.find_element_by_css_selector('.job-name a').text # 工作名称
company_name = li.find_element_by_css_selector('.name a').text # 公司名字
money = li.find_element_by_css_selector('.job-limit.clearfix span').text #
薪资
exp = li.find_element_by_css_selector('.job-limit p').text # 经验学历
company_type = li.find_element_by_css_selector('.false-link').text # 公司类
型
company_boon = li.find_element_by_css_selector('.info-desc').text # 公司福利
href = li.find_element_by_css_selector('.job-name a').get_attribute('href')
# 详情页
```

(4) 保存爬取数据。将数据持久化到 csv 中进行保存。关键代码如下：

```
with open(f'{keyword}data.csv', mode='a', encoding='utf-8', newline='') as f:
    csv_writer = csv.DictWriter(f, fieldnames=['标题', '地区', '公司名字', '薪
资', '经验', '公司类型', '公司福利', '详情页', '具体要求'])
    csv_writer.writerow(dit)
```

(5) 翻页。继续爬取数据。关键代码如下：

```
driver.find_element_by_css_selector('.next').click()# 下一页
```

4.1.2 数据分析模块

- (1) 数据清洗。将薪资中的 k 和-去掉并提取出最大最小薪资，并且将城市归入省份一下，以省份为单位进行后续的步骤。关键代码如下：

```
#处理薪资
pd_data['地区'] = [i[0:2] for i in pd_data['地区']]
pd_data['薪资'] = [i.replace('K', '') for i in pd_data['薪资']]
salary = [i.split('-') for i in pd_data['薪资']]
city_salary_min = [[i[0], int(i[1][0])] for i in zip(pd_data['地区'], salary)]
city_salary_max = [[i[0], int(i[1][1])] for i in zip(pd_data['地区'], salary)]

# 将由城市计数转为省份计数
for cs_min, cs_max in zip(city_salary_min, city_salary_max):
    for province, city in province_city.items():
        for c in city:
            if cs_min[0] in c:
                cs_min[0] = province
                province_salary_min.append(cs_min)
            if cs_max[0] in c:
                cs_max[0] = province
                province_salary_max.append(cs_max)

# 分别对最低薪、最高薪和平均薪资计算平均值
province_city_min_avg = [[pvs, round(sly_sum / count[pvs])] for pvs, sly_sum
in province_salary_min_sum.items() if count[pvs]]

province_city_max_avg = [[pvs, round(sly_sum / count[pvs])] for pvs, sly_sum
in province_salary_max_sum.items() if count[pvs]]

province_city_avg_avg = [[pc_min_a[0], round((pc_min_a[1] + pc_max_a[1]) / 2)]
for pc_min_a, pc_max_a in zip(province_city_min_avg, province_city_max_avg)]
```

- (2) 分词。对数据进行词频分析，并且将无用的词过滤。关键代码如下：

```
# jieba 库分割字符串，并用空格连接
word_ls = jieba.lcut(text)
space_separate_text = ' '.join(word_ls)
# 将列表中字典的相同信息串起来,同时返回 str 对应的停用词
for info in specific_ls:
    try:
        text += info[str]
    except:
        pass
stop_word_dict = {
    '岗位职责': ['数据', '业务', '的', '和', '与', '用户', '为', '等', '并', '']
```

```

    任职'],
    '任职要求': ['和', '优先', '者', '的', '良好的', '数据', '及', '以上学历', '业务']
}

```

(3) 绘制词云。关键代码如下：

```

mask = np.array(image.open('python.png'))
# 词云对象生成
wordcloud = WordCloud(background_color=None, width=400,
height=400, max_words=80, max_font_size=80, mask=mask,
mode='RGBA', stopwords=stop_word,
font_path='C:\Windows\Fonts\msyh.ttc').generate(space_separate_text)
# 从图片中生成颜色
image_colors = ImageColorGenerator(mask)
wordcloud.recolor(color_func=image_colors)

```

4.1.3 数据展示模块

(1) 构建各省岗位地图和平均薪资薪资地图。关键代码如下：

```

salary_map = (
    Map(init_opts=opts.InitOpts(width="1500px", height="700px",
page_title="salary_map"))
    .add(
        series_name="",
        data_pair=province_city_avg_avg,
        maptype="china"
    )
    .set_global_opts(
        title_opts=opts.TitleOpts(title="Python 数据分析师薪资情况—全国分布情况"),
        visualmap_opts=opts.VisualMapOpts(is_piecewise=True, max_=40),
    )
    .render("salary_map.html")
)

```

(2) 构建薪资柱状图。绘制各省份的最大最小薪资柱状图。关键代码如下：

```

bar = (
    Bar(init_opts=opts.InitOpts(width="1500px", height="700px",
page_title="salary_bar"))
    .add_xaxis(province_x)
    .add_yaxis("最小", min_salary, stack="stack1")
    .add_yaxis("最大", max_salary, stack="stack1")
    .set_global_opts(
        title_opts=opts.TitleOpts(title="Python 数据分析师薪资情况—柱状图"),
    )
)

```

```

        toolbox_opts=opts.ToolboxOpts(),
        legend_opts=opts.LegendOpts(is_show=False),
        visualmap_opts=opts.VisualMapOpts(
            max_=60,
            range_color=[
                "#313695",
                "#4575b4",
            ]
        )
    )
    .render("salary_bar.html")
)

```

4.2 本章小结

展示各个模块主要代码。

5 测试

×

导航栏

-----数据分析师-----

● 招聘信息

○ 数据分析师任职要求

○ 数据分析师岗位职责

○ 招聘岗位分布地图

○ 薪资统计柱状图

○ 招聘岗位薪资情况分布地图

正在等待 api.segmentio...

数据分析师

招聘信息:

	标题	地区	公司名字	薪资	经验	公司类型	公司福利
0	数据分析师	福州 闽侯县·新西客站	扬腾创新	7-10K	1-3年本科	电子商务	节日福利
1	数据分析师	北海·海城区·北京路	符号认知	3-6K	1年以内本科	广告/公关/会展	年终奖
2	数据分析师	广州	大牛教育	15-20K	3-5年本科	在线教育	年终奖
3	数据分析师	合肥·蜀山区·天乐	融沃科技	12-20K	3-5年本科	移动互联网	五险一金
4	数据分析师	天津·南开区·鼓楼	数能科技	4-8K	1-3年大专	互联网	五险一金
5	数据分析师	泉州·丰泽区·泉秀	品尚征信	10-15K	3-5年本科	互联网金融	员工旅游
6	数据分析师	天津·西青区·大寺	京创曙光	10-15K	3-5年本科	数据服务	住房补贴
7	数据分析师	苏州·苏州工业园区·娄葑	英赛孚工业智能	12-24K	3-5年本科	计算机软件	带薪年假
8	数据分析师	天津·西青区·西青大学城	福安易	8-9K	1-3年本科	医疗健康	带薪年假

×

导航栏

-----数据分析师-----

○ 招聘信息

● 数据分析师任职要求

○ 数据分析师岗位职责

○ 招聘岗位分布地图

○ 薪资统计柱状图

○ 招聘岗位薪资情况分布地图

数据分析师

数据分析师任职要求:

应用

软件

责任心

团队合作

沟通能力

抗压

基础

Excel

数据库

SQL

工具

逻辑思维

计算机

统计

建模

算法

数学

统计学

金融

互联网

系统

问题

数据

分析

Python

办公

软件

经验

学历

本科

硕士

敏感

度

导航栏

-----数据分析师-----

- ☐ 招聘信息
- ☐ 数据分析师任职要求
- ☒ 数据分析师岗位职责
- ☐ 招聘岗位分布地图
- ☐ 薪资统计柱状图
- ☐ 招聘岗位薪资情况分布地图

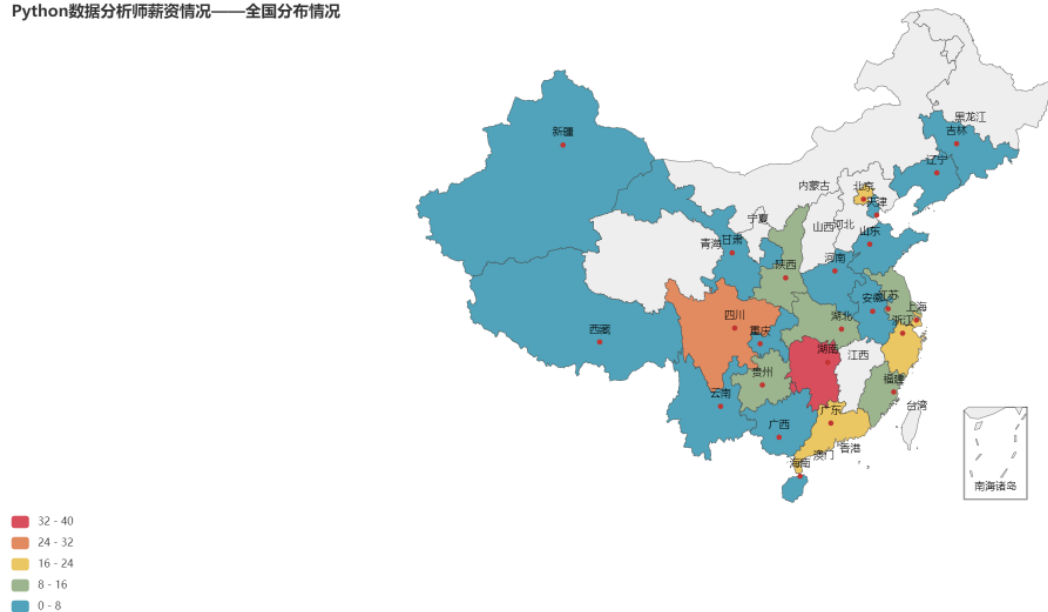


数据分析师

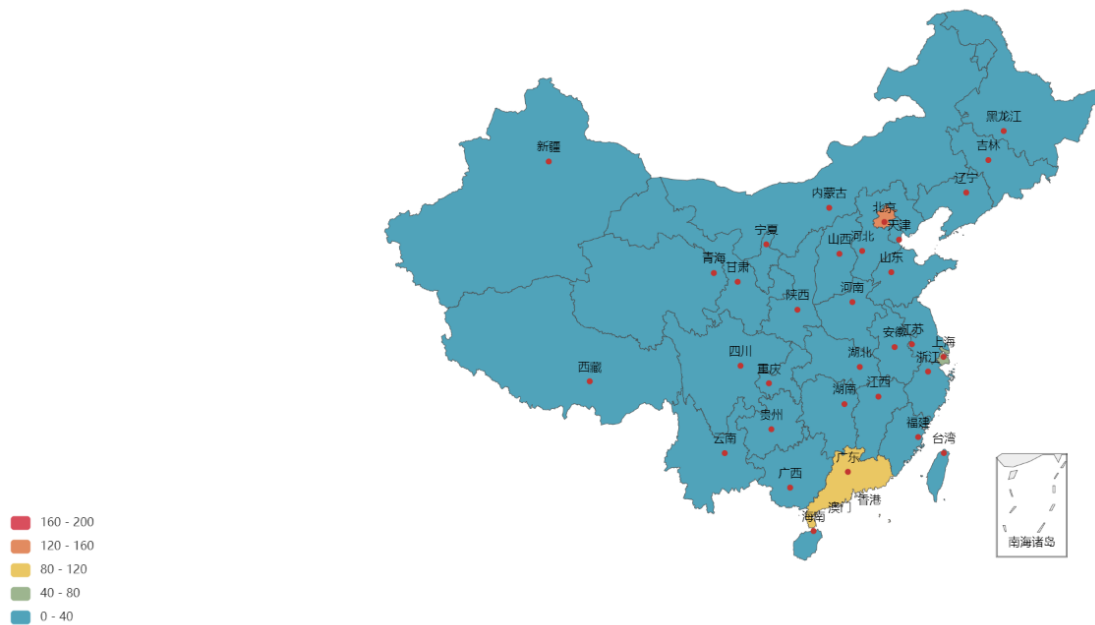
数据分析师岗位职责:



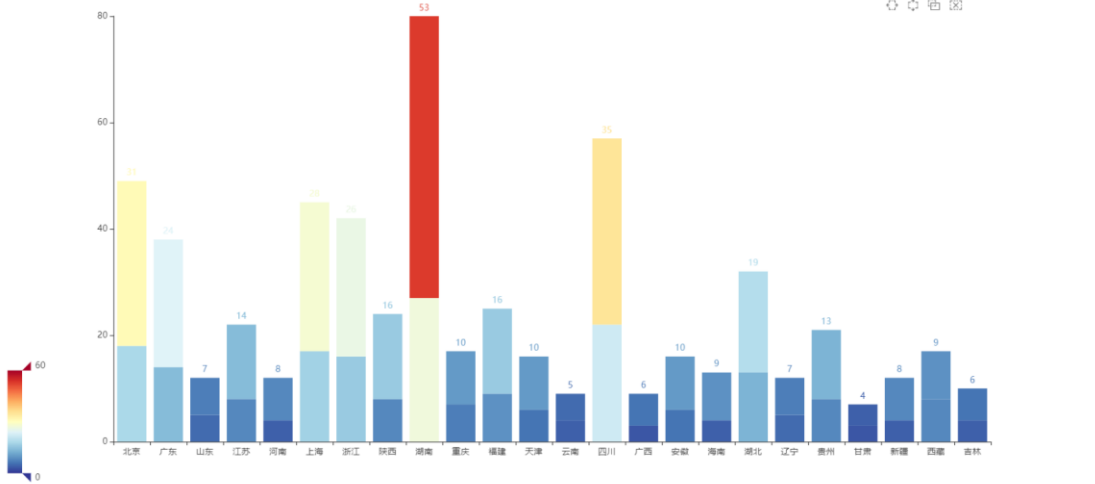
Python数据分析师薪资情况——全国分布情况



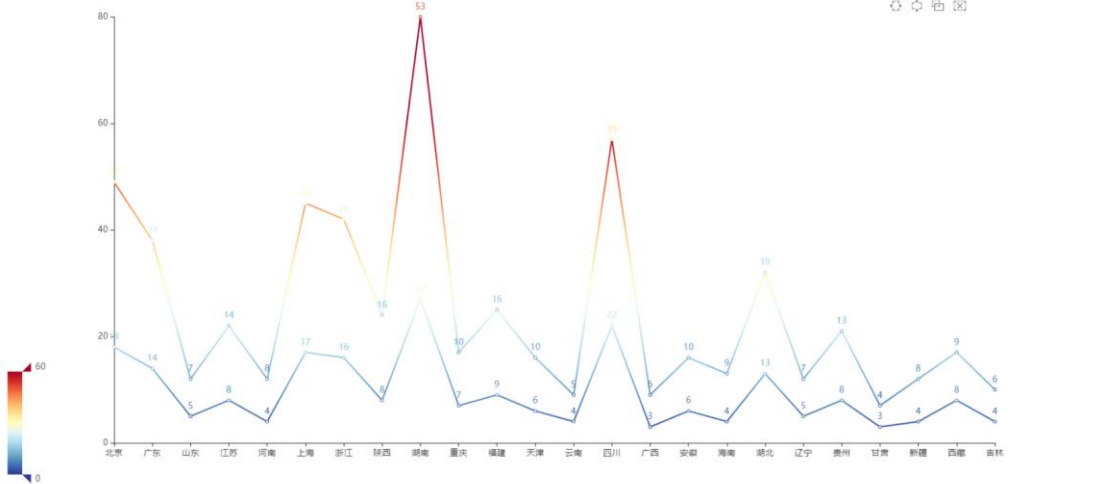
Python数据分析师招聘岗位——全国分布情况



Python数据分析师薪资情况——柱状图



Python数据分析师薪资情况——柱状图



6 总结

本文通过 Python 加上 selenium 的配置，创建了一个基于 BOSS 直聘网站的网络爬虫数据收集分析系统，该系统能够打开 BOSS 直聘，并获取页面信息，分析页面中的元素，将用户获取到的数据存储到持久化介质，在此基础上将对数据进行深层次的挖掘，也就是运用一系列的数据分析手段，获得关于全国各地区、各岗位的薪资待遇、招聘需求等一系列重要信息，为广大的就业人员提供有益的借鉴与参考。

7 参考文献

- [1] 白雪丽.浅析基于 Python 爬虫技术的特性及应用[J].山西科技,2018,186(2):58-60.
- [2] 樊涛,赵征,刘敏娟.基于 Selenium 的网络爬虫分析与实现.电脑编程技巧与维护,2019(09):155-156,170.
- [3] 刘石磊.对反爬虫网站的应对策略[J].电脑知识与技术,2017,13(15):19-21+23.
- [4] [印度]冈迪察. U(Unmesh, Gundecha,).金鑫,熊志男,译. Selenium 自动化测试—基于 Python 语言[M].人民邮电出版社, 2018:15-28.
- [5]许景贤,林锦程,程雨萌. Selenium 框架的反爬虫程序设计与实现[J]. 福建电脑,2021,37(01):26-29.
- [6]樊涛,赵征,刘敏娟. 基于 Selenium 的网络爬虫分析与实现[J]. 电脑编程技巧与维护,2019,(09):155-156+170.
- [7] 陈晴光.电子商务数据挖掘可视化系统模型研究及应用.计算机工程与应用,2007(05):242-245,248

8 附录

- 图 1 整体设计框图
- 图 2 模拟浏览器进行请求数据^[6]
- 图 3 岗位职责
- 图 4 招聘岗位全国分布情况
- 图 5 平均薪资地图
- 图 6 薪资区间柱状图