西安理工大学

**实 习 答 辩 报 告**

项 目 名 称 影视助手

学 院（部） 计算机科学与工程

专 业 计算机科学与技术

组 长 顾晓倩

组 员 庞欣超、汶怡春、罗康中、刘中胜、侯升

组 名 幼儿园哈哈哈哈哈哈

**2020年7月31日**

目录

**[一．项目概述](#_Toc23298_WPSOffice_Level1)** **[3](#_Toc23298_WPSOffice_Level1)**

[1.1. 项目目的](#_Toc14643_WPSOffice_Level2) [3](#_Toc14643_WPSOffice_Level2)

[1.2. 爬取的数据](#_Toc30382_WPSOffice_Level2) [4](#_Toc30382_WPSOffice_Level2)

**[二．技术简介](#_Toc14643_WPSOffice_Level1)** **[4](#_Toc14643_WPSOffice_Level1)**

**[三．数据展示](#_Toc30382_WPSOffice_Level1)** **[5](#_Toc30382_WPSOffice_Level1)**

[热播排行](#_Toc3517_WPSOffice_Level2) [5](#_Toc3517_WPSOffice_Level2)

[关键字搜索](#_Toc7679_WPSOffice_Level2) [6](#_Toc7679_WPSOffice_Level2)

[在线观看](#_Toc24859_WPSOffice_Level2) [6](#_Toc24859_WPSOffice_Level2)

[下载](#_Toc8080_WPSOffice_Level2) [7](#_Toc8080_WPSOffice_Level2)

[影片详情](#_Toc25220_WPSOffice_Level2) [7](#_Toc25220_WPSOffice_Level2)

[界面](#_Toc7584_WPSOffice_Level2) [8](#_Toc7584_WPSOffice_Level2)

**[四． 数据的可视化](#_Toc3517_WPSOffice_Level1)** **[9](#_Toc3517_WPSOffice_Level1)**

[榜单词云](#_Toc9110_WPSOffice_Level2) [9](#_Toc9110_WPSOffice_Level2)

[电视剧Bar图](#_Toc25155_WPSOffice_Level2) [9](#_Toc25155_WPSOffice_Level2)

[电影Bar图](#_Toc4414_WPSOffice_Level2) [10](#_Toc4414_WPSOffice_Level2)

[电影热力图](#_Toc27960_WPSOffice_Level2) [10](#_Toc27960_WPSOffice_Level2)

[电视剧热力3D图](#_Toc24760_WPSOffice_Level2) [10](#_Toc24760_WPSOffice_Level2)

**[五． 生产实习总结](#_Toc7679_WPSOffice_Level1)** **[11](#_Toc7679_WPSOffice_Level1)**

**[六． 代码](#_Toc24859_WPSOffice_Level1)** **[12](#_Toc24859_WPSOffice_Level1)**

[DownloadGet.py](#_Toc26969_WPSOffice_Level2) [12](#_Toc26969_WPSOffice_Level2)

[ListGet.py](#_Toc31171_WPSOffice_Level2) [13](#_Toc31171_WPSOffice_Level2)

[MainUI.py](#_Toc3728_WPSOffice_Level2) [15](#_Toc3728_WPSOffice_Level2)

[OnlineGet.py](#_Toc8879_WPSOffice_Level2) [27](#_Toc8879_WPSOffice_Level2)

[RankingGet.py](#_Toc11038_WPSOffice_Level2) [28](#_Toc11038_WPSOffice_Level2)

[RankingView.py](#_Toc23068_WPSOffice_Level2) [30](#_Toc23068_WPSOffice_Level2)

一．项目概述

## 项目目的

当代大学生的业余生活，占据主导地位的无非就是游戏和影视。恰巧我们组都是电影爱好者，所以我们打算从影视方面开始，着手准备项目。确定了大致方向，便开始思考具体实施内容。在从网络上观看一部电影之前，往往需要很多准备工作，如查看电影评分，寻找高清资源。反复地查找过程中，便消磨了原本想看电影的兴致，我们的项目便是针对此问题而生。最终，在小组的协商之下，我们决定爬取优质电影资源，高分电影榜、热度榜可视化，帮助挑选优质电影，并且提供在线观看，云盘下载和vip视频迅雷下载功能。

## 爬取的数据

1. 爬取豆瓣：爬取了豆瓣的电视电影详情和海报，在搜索影视关键字时会帮助我们更好地了解这部电影值不值得我们观看。
2. 爬取影视榜单：这里获得的榜单我们实现了可视化，这样在无从选择时，可以帮助我们进行一个简单直接的筛选。
3. 爬取链接：我们搜集了大量的电影电视剧在线观看、云盘下载和迅雷下载链接，并且尽可能保证是优质资源，只需一键，便可获取。

二．技术简介

在此次项目中主要用了Urllib、BeautifulSoup、seleniumWebdriver，tkinter等技术。

Urllib：urllib 是 Python 标准库中用于网络请求的库。该库有四个模块，分别是：urllib.request、urllib.error、urllib.parse、urllib.robotparser。模拟浏览器发起一个 HTTP 请求，我们需要用到 urllib.request 模块。urllib.request 的作用不仅仅是发起请求， 还能获取请求返回结果。urllib.request 被用于打开和阅读URL，从而去获取我们想要的数据；urllib.parse是urllib中用来解析各种数据格式的模块，被用于解析URL，使用urllib获取HTML，来得到我们需要的内容。

BeautifulSoup：最主要的功能是从网页抓取数据。Beautiful Soup提供一些简单的、python式的函数用来处理导航、搜索、修改分析树等功能。它是一个工具箱，通过解析文档为我们提供需要抓取的数据。Beautiful Soup自动将输入文档转换为Unicode编码，输出文档转换为utf-8编码。BeautifulSoup是灵活又方便的HTML解析库，处理高效，支持多种解析器。利用它不用编写正则表达式即可方便地实现网页信息的提取。BeautifulSoup被用来来抓取我们想要的电影和电视剧相关的数据，取得了不错的效果

seleniumWebdriver：业界通用的测试框架，不仅是web测试的标准，同时在移动测试领域也是底层的核心驱动框架；可支持多种语言，用于操作浏览器的一套API；Webdriver按照Server-Client的经典设计模式设计（1）Server端：即Remote Server（远程服务器），可以是任意的浏览器，当脚本启动浏览器时，该浏览器就是Remote Server，它的职责是等待Client发送请求并做出响应（2）Client端：简单来说就是我们的测试代码，测试代码中的一些行为是以HTTP请求的方式发送给被测试浏览器——Remote Server，Remote Server接受请求，执行相应操作，并在Response中返回执行状态、返回值等信息；此次将它用来进行网页访问解析，从而获取我们需要的数据。

Tkinter 是 Python 的标准 GUI 库，Python 使用 Tkinter 可以快速的创建 GUI 应用程序进行界面设计。Tkinter包含了几种常用类型的控件，包括Label（标签，就是界面上显示的字）、Entry（输入框）、Button（按钮，可以绑定各种封装函数）、Radiobutton（单选框）、Checkbuttion（复选框）、messagebox（消息弹出框）、Text（文本编辑框）、Listbox（列表控件）、Scrollbar（滚条控件）等。此次我们的界面设计使用Tkinter库，这使得我们的所做的功能能够清楚地展现出来，同时可以方便我们的使用。

Pyecharts： Echarts 是百度开源的一个数据可视化 JS 库。主要用于数据可视化。pyecharts 是一个用于生成 Echarts 图表的类库。用 Echarts 生成的图可视化效果非常好，pyecharts 是为了与 Python 进行对接，方便在 Python 中直接使用数据生成图。我们用pyecharts来绘制图形生成html，来达到我们想要的结果。

多线程技术：python提供了thread 和threading两个模块来实现多线程。Thread类定义了以下常用方法与属性**：**Thread.getName() \Thread.setName():用于获取和设置线程的名称，官方建议用Thread.name替代Thread.ident:获取线程的标识符。只有在调用start()方法执行后才有效，否则返回None。Thread.is\_alive():判断线程是否是激活的。Thread.join([timeout]):调用Thread.join将会使主调线程堵塞，直到被调用线程运行结束或超时。参数timeout是一个数值类型，表示超时时间，如果未提供该参数，那么主调线程将一直堵塞到被调线程结束。我们界面使用了多线程技术可以分别去不同的服务器爬取所要获得的数据，这就大大提高了我们爬取数据的效率。

三．数据展示

**热播排行**

选择电视剧/电影查看今日观看指数，获取最新动态



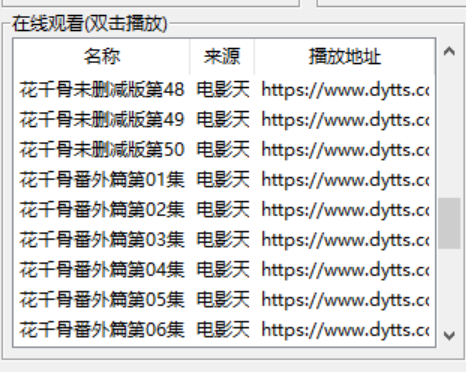
**关键字搜索**

通过输入的关键字爬取豆瓣网站数据，得到与该关键字相关的一些影视剧信息



**在线观看**

通过关键字同时爬取电影天堂网站在线播放数据，得到在线观看地址，双击直接观看



**下载**

通过关键字同时爬取电影天堂网站下载地址数据，得到视频下载地址

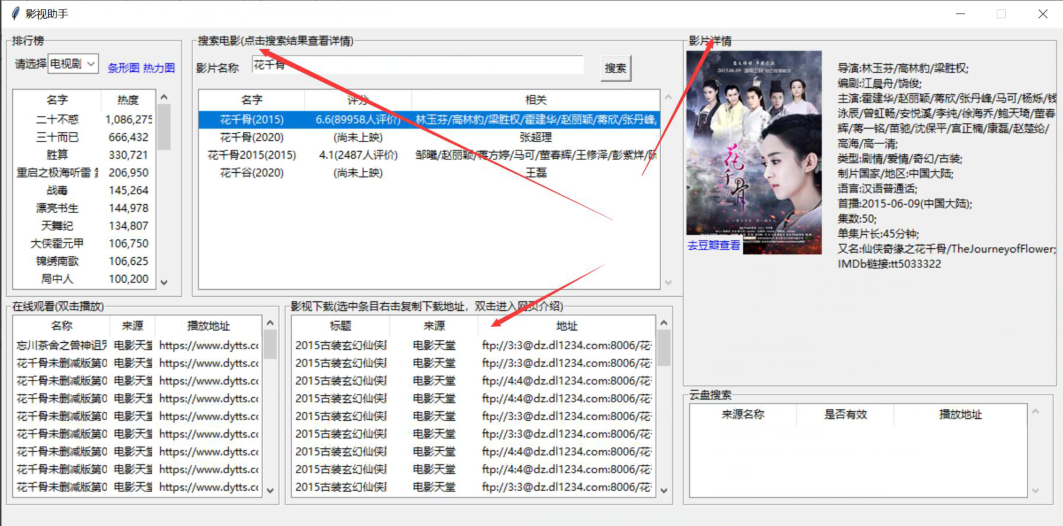


**影片详情**

对搜索结果的链接解析，再爬取该搜索结果的详细信息，得到影片的海报与一些属性信息



**界面**



1. 数据的可视化

主要旨在借助于图形化手段，清晰有效地传达与沟通信息。但是，这并不就意味着数据可视化就一定因为要实现其功能用途而令人感到枯燥乏味，或者是为了看上去绚丽多彩而显得极端复杂。为了有效地传达思想概念，美学形式与功能需要齐头并进，通过直观地传达关键的方面与特征，从而实现对于相当稀疏而又复杂的数据集的深入洞察。然而，设计人员往往并不能很好地把握设计与功能之间的平衡，从而创造出华而不实的数据可视化形式，无法达到其主要目的，也就是传达与沟通信息。

数据可视化与信息图形、信息可视化、科学可视化以及统计图形密切相关。当前，在研究、教学和开发领域，数据可视化乃是一个极为活跃而又关键的方面。“数据可视化”这条术语实现了成熟的科学可视化领域与较年轻的信息可视化领域的统一。

我们小组在可视化方面用的是Pyecharts库和Matplotlib库，最主要用的还是Pyecharts库。Pyecharts 是一个用于生成 Echarts 图表的类库。常规的Echarts 是由百度开源的一个数据可视化 JS 库，主要用于数据可视化。简单来说，Pyecharts是一款将python与echarts结合的强大的数据可视化工具。使用 Pyecharts 可以生成独立的网页，也可以在 flask , Django 中集成使用。

因为使用Pyecharts库有很多优点，所以我们组在可视化部分使用它去完成了电视剧和电影热度榜单的展现，并且将其做成了炫酷的动态形式。

本次我们组实现数据可视化部分的数据来源是360影视的电视剧和电影热度榜单<https://www.360kan.com/rank/dianying>。

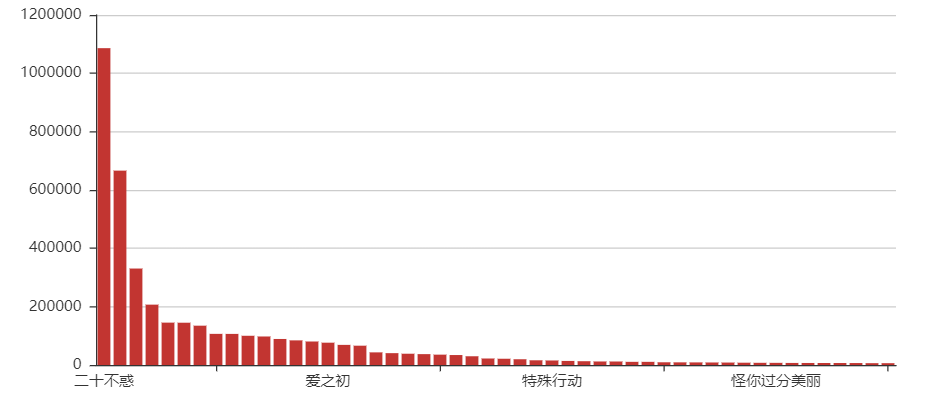
以下是我们的可视化成果展示：

**榜单词云（展示了最近热播电视剧和电影的情况，都是top10）**

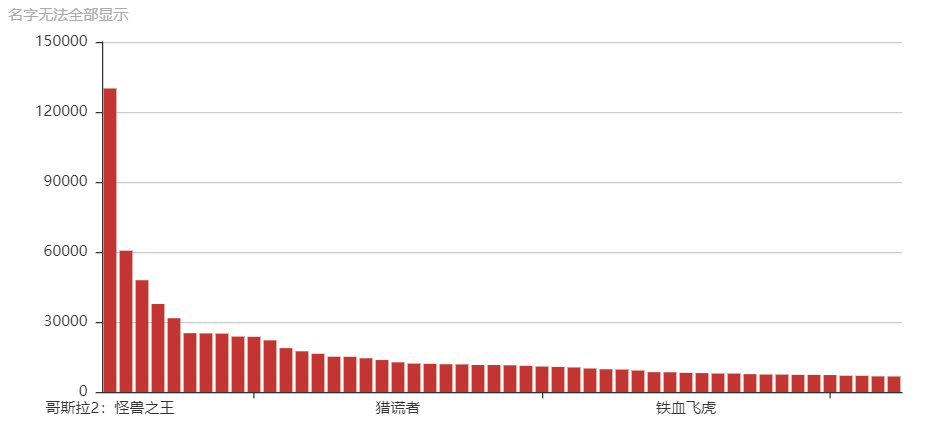
[](qq文件/词云.html)

我们将网站上比较热门的搜索词汇爬取下来，做成词云的形式。这样词汇的热度大小就一目了然了。

**电视剧Bar图（显示了电视剧热力榜单前五十，昨日观看指数从多到少）**

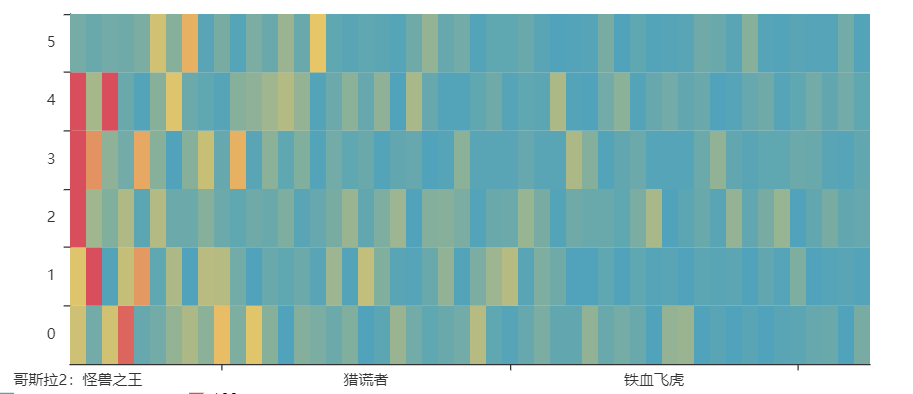
[](qq文件/电视剧Bar图.html)

**电影Bar图（显示了电影热力榜单前五十，昨日观看指数从多到少）**

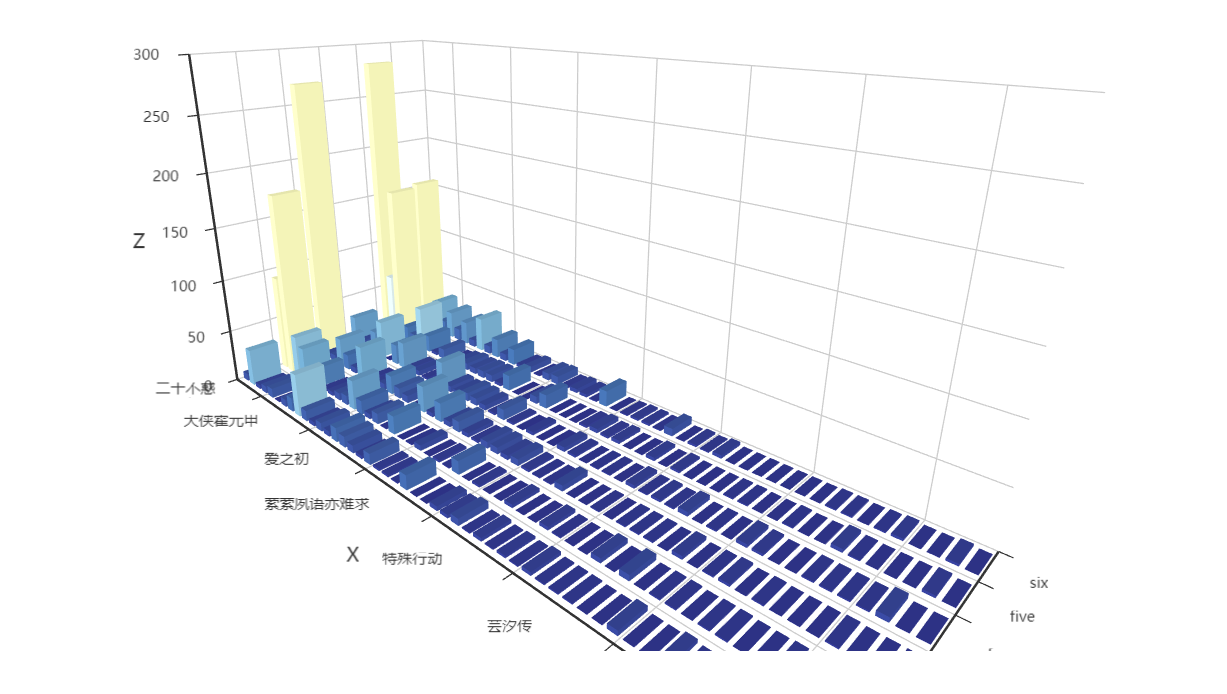
[](qq文件/电影Bar图.html)

这两张是我们根据爬取到的电视剧和电影数据做成的条形图，十分直观地展示了各电影或各电视剧的热度差别。

**电影热力图（为了感受每部电影的热门程度，将观看指数成比例缩小，一列代表一部电影，颜色越深，越受欢迎）**

[](qq文件/电影热力图.html)

**电视剧热力3D图（为了感受每部电视剧的热门程度，将观看指数成比例缩小，一列代表一部剧，颜色越深，越受欢迎，3d图更加直观有效，还设置了自动旋转）**

[](qq文件/电视剧热力3D图%20.html)

电视剧热度图我们想采取一种更加直观的方式来将其展现出来，于是在小组的商量之下做成了3D形式。做出来之后果然效果很好，达到了我们的要求。

1. 生产实习总结

**组长说**：

我们小组的优势便是每个人都乐于融入团体，并且有组员庞欣超提供大量的技术支持。缺点便是小组每个人都不善言谈，有时不能把自己的想法准确的表达出来，以后要努力提高自己的言语表达能力，提高学习能力和合作能力。在这次项目合作的过程中，我们组分工明确，每个人各司其职，争取把这个项目做得精良，我也是深切体会到了团队合作的好处，也体会到了团队合作的氛围所带来的乐趣和不一样的感觉。

我们的时间很紧张，所以不得不放下了手中其他的事情，全身心的投入到此次的项目之中。在有限的时间里，努力把自己负责的任务给做好，我们尽量先是自己独立完成，靠查找资料什么的，实在不行的就小组讨论，往往都能解决自己解决不了的问题。

最后，再次感谢这些天来认真负责的老师，感谢我的组员。

**每个小组成员的心得体会：**

**顾晓倩：**

整个项目只有短短的两天，尽管代码量不大，但是仍然需要一个字一个字地敲。有人开玩笑说“360行，行行出bug”，但也不无道理，很多时候就是一个小小的问题，报错却各式各样，所以我得到教训：写一点测试一点，否则代码过多，找错就麻烦了。项目中我完成的工作是热力榜数据的获取和可视化操作（Bar图、热力图、3D热力图、词云图），数据获取较容易，可视化部分热力图需要的数据是真的麻烦，通过不断查阅资料，最终完美的呈现。这两天，小组成员都“八仙过海，各显神通”，熬的了夜，写的了文档，敲的了代码，感谢各位的积极配合和认真负责。特别是庞欣超，技术没说的，很nice，还有汶怡春、刘中胜，给他们的任务都努力完成。

通过这些天的学习，让我对python的使用更加熟练，而且老师细致的讲解也解决了我不少的疑惑。感谢学校各位老师的认真负责，也感谢西安丝路软件有限责任公司各位老师的知识讲解和答疑解惑。

**庞欣超：**

本次实习学到了很多的python的应用，使用了很多的第三方库，学会了如何做出一个界面，如何把爬虫技术运用到实际的项目中，让我收获了很多，了解到了企业实际用到的一些技术，学到了很多很有意思的东西，使我对python技术的产生了浓厚的兴趣，十分感谢老师们的辛勤授课。

**汶怡春：**

经过15天的云实习特别是后面的小组项目作业，我从老师，小组成员身上学到了不少东西，也学会了如何从对应网站上爬取自己所需要的信息，并且将这些信息收集起来表示在图片中，即可视化，使其更加形象具体。在本次小组项目作业中我做的是在线观看地址的爬取，刚开始收集资料的时候，想着可以在网上找点资料参考一下，但网上有很多类似的但是又和我的爬取思路不太一样，最后只能硬着头皮自己写了，由于对python还没有到熟练运用的地步，刚开始写感觉挺头疼的不知道如何下手，最后就按着平常老师授课教的方法慢慢一步一步向下爬取，最终得到了结果，然而并不是特别完美，还有很多小问题，但幸好有组长顾晓倩和组员庞欣超的帮助，使我可以顺利完成任务；在这里也非常感谢所有实习老师的辛勤付出，俗话说，“授人以鱼不如授人以渔”，这15天我肯定地说我学到的是学习方法，另外，也非常感谢我们组长和所有成员，在最后的项目中我们明确分工，积极配合得以完成了最终的任务，也是他们让我体会到团队合作的重要性；最后，我想说，在今后的学习生活中，我仍会继续积极学习、树立远大目标，不断加强自己的动手实践能力，成为更好的自己。

**刘中胜：**

通过本次实习的不断学习，我对Python有了全新且又全面的认识，再次被其简单强大所折服，前面的基础过程，在卫老师的认真教学之下，让我对python的理解又上了一个台阶，从卫老师那里学了python的基础知识，卫老师很有耐心，讲解的也很细致，授课节奏很舒服，这一周使用python完成了一些作业，如：做一个简单的计算器，使用python中  
的关系运算符等生成2020年的月份天数等。了解到了python的一些特性，简洁，超丰富的库等。这短短的一周，使我获益匪浅。  
 后面小组合作完成了一个爬虫项目，通过这个项目的合作完成，我真的学到了很多，小组合作很重要，有些事情不是一个人就能完成的，现代社会是建立在分工的基础上的，科研项目也必然是要求分工明确的。        
  沟通交流也必不可少，要善于倾听，善于总结别人的话，这样才能从优秀的人身上学到更多的东西；同时更要学会表达，自己的想法要让别人清晰易懂，这样团队效率就会大大提高。  
  逝者如斯夫，不舍昼夜，转眼间15天的项目实习就宣告结束，也再次感谢各位老师的辛苦付出

**罗康中：**

在这次的项目中，我学到了很多东西。在做自己的项目中，因为对python还不是很熟练，对一些库掌握的还不够，导致编程的时候出现好多错误，还有一些功能不能准确的完成。不过在组长和组员的帮助下，我最终还是解决了这些问题。同时，在这次项目中，我还学会了团队协作的精神。一个人好不算好，整个团队好才是真的好。合作不仅对于这次项目，更对于后来的工作，乃至生活的方方面面都很重要。  
  从这次的项目中，我看到了自己的问题，对于基础掌握的还不够牢固，还缺少将书本转化到实际中的能力。不过我相信在这次项目之后，我会逐渐解决自己的问题然后成长起来！

**侯升：**

本次项目我主要负责撰写项目文档报告，同时对代码进行相应的学习。在书写项目文档时可以发现许多新的问题和知识，加深了自己对项目的理解。python书写更加简洁方便，几行代码提供的功能远大于其他语言。使得我对python语言的简洁性有了更好的认识和了解。为今后python语言的学习奠定了良好的基础。

1. 代码

**DownloadGet.py**

import urllib.request

from bs4 import BeautifulSoup

from selenium import webdriver

from selenium.webdriver.common.by import By

from selenium.webdriver.common.keys import Keys

from selenium.webdriver.support import expected\_conditions as EC

from selenium.webdriver.support.wait import WebDriverWait

from selenium.webdriver.chrome.options import Options

header = 'Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/63.0.3239.132 Safari/537.36'

chrome\_options = Options()

chrome\_options.add\_argument('--headless')#设置为无头，即不显示浏览器窗口

chrome\_options.add\_argument('user-agent='+header)

chrome\_options.add\_experimental\_option('excludeSwitches',['enable-automation']) #设置为开发者模式，防止被识别出来使用了Selenium

chrome\_options.add\_experimental\_option("prefs",{"profile.managed\_default\_content\_settings.images": 2}) # 不加载图片

domain='https://www.dy2018.com'

def getUrl(title,detailurl):

request = urllib.request.Request(url=detailurl)

request.add\_header('user-agent', header)

html = urllib.request.urlopen(request).read()

# print(html)

soup = BeautifulSoup(html, 'lxml')

href = soup.select('td a')

lst=[]

for i in href:

if i['href'][:7]=='magnet:' or i['href'][:4]=='ftp:':

lst.append([title,'电影天堂',i.get\_text(),i['href'],detailurl])

if len(lst)>0:

return lst

else:

return [[]]

def getDownloadUrl(keywords):

browser = webdriver.Chrome(r'E:\chromedriver.exe', options=chrome\_options)

try:

browser.get(domain)

wait = WebDriverWait(browser,10) # 等待

wait.until(EC.presence\_of\_element\_located((By.NAME, 'keyboard'))) # 等待元素加载

lst=browser.find\_elements\_by\_name('keyboard')

if len(lst)>0:

lst[0].send\_keys(keywords) # 向元素里发送（敲入）Python

lst[0].send\_keys(Keys.ENTER) # 敲回车

wait = WebDriverWait(browser, 5) # 等待

browser.switch\_to.window(browser.window\_handles[1])#切换为新页面

html=browser.page\_source

soup=BeautifulSoup(html, 'lxml')

urlLst=soup.select('.ulink')

result=[]

for i in urlLst:

result=result+getUrl(i['title'],domain+i['href'])

finally:

browser.close()

if len(result)>0:

return result

else:

return [['未找到资源!','',None,None,None]]

**ListGet.py**

import urllib.request,json

from bs4 import BeautifulSoup

from selenium import webdriver

from selenium.webdriver.common.by import By

from selenium.webdriver.common.keys import Keys

from selenium.webdriver.support import expected\_conditions as EC

from selenium.webdriver.support.wait import WebDriverWait

from selenium.webdriver.chrome.options import Options

header = 'Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/63.0.3239.132 Safari/537.36'

chrome\_options = Options()

chrome\_options.add\_argument('--headless')#设置为无头，即不显示浏览器窗口

chrome\_options.add\_argument('user-agent='+header)

chrome\_options.add\_experimental\_option('excludeSwitches',['enable-automation']) #设置为开发者模式，防止被识别出来使用了Selenium

chrome\_options.add\_experimental\_option("prefs",{"profile.managed\_default\_content\_settings.images": 2}) # 不加载图片

def getList(keywords):

browser = webdriver.Chrome(r'E:\chromedriver.exe', options=chrome\_options)

url = 'https://search.douban.com/movie/subject\_search?search\_text=' + urllib.parse.quote(keywords) + '&cat=1002'

browser.get(url)

try:

wait = WebDriverWait(browser,10) # 等待

wait.until(EC.presence\_of\_element\_located((By.ID, 'root'))) # 等待元素加载

# browser.find\_elements\_by\_class\_name()

lst=browser.find\_elements\_by\_class\_name('item-root')

classList=['title-text','rating','abstract\_2']

result=[]

for i in lst:

record = []

for j in range(0,len(classList)):

txt=i.find\_elements\_by\_class\_name(classList[j])

if len(txt)>0:

record.append(txt[0].text.replace(' ',''))

else:

record.append('无')

record.append(i.find\_element\_by\_tag\_name('a').get\_attribute('href'))

result.append(record)

finally:

browser.close()

return result

def getDetails(url):

request = urllib.request.Request(url=url)

request.add\_header('user-agent', header)

html = urllib.request.urlopen(request).read().decode('utf-8')

soup = BeautifulSoup(html, 'lxml')

img=soup.select('#mainpic img')

# print(img[0]['src'])

if len(img)>0:

urllib.request.urlretrieve(img[0]['src'], filename='images/details.png')

info=soup.select('#info')

if len(info)>0:

txt=info[0].get\_text()

txt = txt.replace(' ', '').strip('\n').split('\n')

# print('\n'.join(txt))

return ';\n'.join(txt)

else:

return '无'

**MainUI.py**

from ListGet import \*

from RankingGet import hot\_search

from DownloadGet import getDownloadUrl

from ClipboardOP import setClip

from OnlineGet import getOnlineUrl

# from RankingView import \*

from PIL import Image, ImageTk

from tkinter import messagebox

from tkinter import Tk

from tkinter import ttk

from tkinter import font

from tkinter import LabelFrame

from tkinter import Label

from tkinter import StringVar

from tkinter import Entry

from tkinter import END

from tkinter import Button

from tkinter import Frame

from tkinter import RIGHT,BOTTOM

from tkinter import NSEW

from tkinter import NS

from tkinter import NW

from tkinter import N

from tkinter import Y,X

from tkinter import DISABLED

from tkinter import NORMAL

from threading import Thread

from webbrowser import open

# import os

# import ssl

# ssl.\_create\_default\_https\_context = ssl.\_create\_unverified\_context #关闭SSL证书验证

"""重新定义带返回值的线程类"""

class MyThread(Thread):

def \_\_init\_\_(self, func, args):

super(MyThread, self).\_\_init\_\_()

self.func = func

self.args = args

def run(self):

self.result = self.func(\*self.args)

def get\_result(self):

try:

return self.result

except Exception:

return None

def thread\_it(func, \*args):

# 创建

t = Thread(target=func, args=args)

# 守护

t.setDaemon(True)

# 启动

t.start()

# print('end')

def handlerAdaptor(fun, \*\*kwds):

'''事件处理函数的适配器，相当于中介，那个event是从那里来的呢，我也纳闷，这也许就是python的伟大之处吧'''

return lambda event, fun=fun, kwds=kwds: fun(event, \*\*kwds)

def resize(w\_box, h\_box, pil\_image):

"""

等比例缩放图片,并且限制在指定方框内

:param w\_box,h\_box: 指定方框的宽度和高度

:param pil\_image: 原始图片

:return:

"""

f1 = 1.0 \* w\_box / pil\_image.size[0] # 1.0 forces float division in Python2

f2 = 1.0 \* h\_box / pil\_image.size[1]

factor = min([f1, f2])

# print(f1, f2, factor) # test

# use best down-sizing filter

width = int(pil\_image.size[0] \* factor)

height = int(pil\_image.size[1] \* factor)

return pil\_image.resize((width, height), Image.ANTIALIAS)

class uiObject:

def \_\_init\_\_(self):

self.doubanUrl=""

def open\_ranking(self):

# print(self.ranking\_type.get())

self.clear\_tree(self.treeview\_ranking)

self.add\_tree([['正在更新...', '', '']], self.treeview\_ranking)

lst=hot\_search(self.ranking\_type.get())

self.clear\_tree(self.treeview\_ranking)

self.add\_tree(lst, self.treeview\_ranking) # 将数据添加到tree中

def open\_Online(self):

self.clear\_tree(self.treeview\_play\_online)

self.add\_tree([['正在努力搜索...', '', '']], self.treeview\_play\_online)

resultOnline = getOnlineUrl(self.T\_vote\_keyword.get())

self.clear\_tree(self.treeview\_play\_online)

self.add\_tree(resultOnline, self.treeview\_play\_online)

def open\_Download(self):

# print(self.ranking\_type.get())

self.clear\_tree(self.treeview\_bt\_download)

self.add\_tree([['正在努力搜索...','','']], self.treeview\_bt\_download)

resultDownload=getDownloadUrl(self.T\_vote\_keyword.get())

self.clear\_tree(self.treeview\_bt\_download)

self.add\_tree(resultDownload, self.treeview\_bt\_download)

def open\_ranking\_event(self, event):

self.open\_ranking()

def show\_detail\_img(self, file\_name):

img\_open = Image.open(file\_name) #读取本地图片

pil\_image\_resized = resize(160, 230, img\_open) #等比例缩放本地图片

img = ImageTk.PhotoImage(pil\_image\_resized) #读入图片

self.label\_img.config(image=img, width = pil\_image\_resized.size[0], height = pil\_image\_resized.size[1])

self.label\_img.image = img

def center\_window(self, root, w, h):

"""

窗口居于屏幕中央

:param root: root

:param w: 窗口宽度

:param h: 窗口高度

:return:

"""

# 获取屏幕 宽、高

ws = root.winfo\_screenwidth()

hs = root.winfo\_screenheight()

# 计算 x, y 位置

x = (ws/2) - (w/2)

y = (hs/2) - (h/2)

root.geometry('%dx%d+%d+%d' % (w, h, x, y))

def clear\_tree(self, tree):

'''

清空表格

'''

x = tree.get\_children()

for item in x:

tree.delete(item)

def add\_tree(self,list, tree):

'''

新增数据到表格

'''

i = 0

for subList in list:

tree.insert('', 'end', values=subList)

i = i + 1

tree.grid()

def keyboard\_T\_vote\_keyword(self, event):

thread\_it(self.searh\_movie\_in\_keyword)

#def refresh\_Online(self):

def thread\_view(self,fname,tree, args):

# 创建

t = MyThread(func=fname, args=args)

# 守护

t.setDaemon(True)

# 启动

t.start()

t.join()

result = t.get\_result()

self.add\_tree(result,tree)

# print(result)

# self.add\_tree(result,tree)

def searh\_movie\_in\_keyword(self):#从关键字中搜索符合条件的影片信息

if self.T\_vote\_keyword.get()!='':

# 按钮设置为灰色状态

self.clear\_tree(self.treeview) # 清空表格

# self.clear\_tree(self.treeview\_bt\_download)

# self.clear\_tree(self.treeview\_play\_online)

self.B\_0\_keyword['state'] = DISABLED

self.T\_vote\_keyword['state'] = DISABLED#s搜索的文本

self.B\_0\_keyword['text'] = '正在努力搜索'

thread\_it(func=self.open\_Online)

thread\_it(func=self.open\_Download)

self.add\_tree([['正在努力搜索...', '', '']], self.treeview)

# self.thread\_view(fname=getList, tree=self.treeview, args=(self.T\_vote\_keyword.get(),))

# self.thread\_view(fname=getOnlineUrl, tree=self.treeview\_play\_online, args=(self.T\_vote\_keyword.get(),))

# self.thread\_view(fname=getDownloadUrl, tree=self.treeview\_bt\_download, args=(self.T\_vote\_keyword.get(),))

result=getList(self.T\_vote\_keyword.get())

self.clear\_tree(self.treeview) # 清空表格

self.add\_tree(result,self.treeview) # 将数据添加到tree中

# resultOnline = getOnlineUrl(self.T\_vote\_keyword.get())

# self.add\_tree(resultOnline, self.treeview\_play\_online)

#

# resultDownload=getDownloadUrl(self.T\_vote\_keyword.get())

# self.add\_tree(resultDownload, self.treeview\_bt\_download)

# 按钮设置为正常状态

self.B\_0\_keyword['state'] = NORMAL

self.T\_vote\_keyword['state'] = NORMAL

self.B\_0\_keyword['text'] = '搜索'

else:

messagebox.showerror('什么都没有输入ｂ（￣▽￣）ｄ', '输入后再来搜索吧(^\_−)☆')

def open\_details(self):

item = self.treeview.selection()

if len(item)>0:

detail\_url=self.treeview.item(item, "values")[-1]

# self.label\_details['text'] = '加载中......'

self.douban['text']='加载中......'

self.label\_details['text']=getDetails(detail\_url)

self.show\_detail\_img(r'images/details.png')

self.doubanUrl=detail\_url

self.douban['text'] = '去豆瓣查看'

def open\_details\_event(self,event):

thread\_it(func=self.open\_details)

def copyURL(self,event):

item = self.treeview\_bt\_download.selection()

if (item):

item\_text = self.treeview\_bt\_download.item(item, "values")

url = item\_text[3]

setClip(url)

messagebox.showinfo('复制链接成功', '链接：'+url)

def open\_in\_browser(self, event):

item = self.treeview\_bt\_download.selection()

# print('hello')

# print(item)

if(item):

item\_text = self.treeview\_bt\_download.item(item, "values")

url = item\_text[4]

open(url)

def open\_in\_browser\_online(self,event):

item = self.treeview\_play\_online.selection()

# print('hello')

# print(item)

if (item):

item\_text = self.treeview\_play\_online.item(item, "values")

url = item\_text[2]

open(url)

def ui\_process(self):

root = Tk()

self.root = root

# 设置窗口位置

root.title("影视助手")

self.center\_window(root, 1200, 565)

root.resizable(0, 0) # 框体大小可调性，分别表示x,y方向的可变性Ranking

#################################################################################################排行榜

ranking\_frame = LabelFrame(root, width=200, height=300, text="排行榜")

ranking\_frame.place(x=5, y=5)

label\_select=Label(ranking\_frame, text='请选择')

label\_select.place(x=5, y=5)

comvalue = StringVar()

ranking\_type = ttk.Combobox(ranking\_frame, width=5, textvariable=comvalue, state='readonly')

movieList = ['电视剧','电影']

ranking\_type["values"] = movieList # 初始化

ranking\_type.current(0) # 选择第一个

ranking\_type.place(x=45, y=5)

self.ranking\_type = ranking\_type

# label\_bar = Label(ranking\_frame, text='条形图')

# label\_bar.place(x=110, y=5)

label\_bar = Label(ranking\_frame, text="条形图", fg="blue", cursor="hand2")

label\_bar.place(x=110, y=10)

label\_bar.bind("<Button-1>", lambda event: open(r'E:/python projects/project\_an/htmls/'+self.ranking\_type.get()+r'Bar图.html'))

label\_reli = Label(ranking\_frame, text="热力图", fg="blue", cursor="hand2")

label\_reli.place(x=150, y=10)

label\_reli.bind("<Button-1>", lambda event: open(r'E:/python projects/project\_an/htmls/'+self.ranking\_type.get()+r'热力图.html'))

label\_reli = Label(ranking\_frame, text='热力图')

frame\_root\_ranking = Frame(ranking\_frame)

frame\_l\_ranking = Frame(frame\_root\_ranking)

frame\_r\_ranking = Frame(frame\_root\_ranking)

self.ranking\_frame = ranking\_frame

self.frame\_l\_ranking = frame\_l\_ranking

self.frame\_r\_ranking = frame\_r\_ranking

columns = ("名字", "热度")

treeview\_ranking = ttk.Treeview(frame\_l\_ranking, height=10, show="headings", columns=columns)

treeview\_ranking.column("名字", width=100, anchor='center') # 表示列,不显示

treeview\_ranking.column("热度", width=60, anchor='center')

treeview\_ranking.heading("名字", text="名字") # 显示表头

treeview\_ranking.heading("热度", text="热度")

# 垂直滚动条

vbar\_ranking = ttk.Scrollbar(frame\_r\_ranking, command=treeview\_ranking.yview)

treeview\_ranking.configure(yscrollcommand=vbar\_ranking.set)

treeview\_ranking.pack()

self.treeview\_ranking = treeview\_ranking

vbar\_ranking.pack(side=RIGHT, fill=Y)

self.vbar\_ranking = vbar\_ranking

# 框架的位置布局

frame\_l\_ranking.grid(row=0, column=0, sticky=NSEW)

frame\_r\_ranking.grid(row=0, column=1, sticky=NS)

frame\_root\_ranking.place(x=5, y=45)

#################################################################################################搜索电影

# 容器控件

labelframe = LabelFrame(root, width=650, height=300, text="搜索电影(点击搜索结果查看详情)")

labelframe.place(x=215, y=5)

self.labelframe = labelframe

# 电影搜索

# 影片名称

L\_vote\_keyword = Label(labelframe, text='影片名称')

L\_vote\_keyword.place(x=0, y=10)

#L\_vote\_keyword.grid(row=0,column=0)

self.L\_vote\_keyword = L\_vote\_keyword

# 文本框

T\_vote\_keyword = Entry(labelframe, width=53)

T\_vote\_keyword.delete(0, END)

T\_vote\_keyword.insert(0, '')

T\_vote\_keyword.place(x=66, y=7)

self.T\_vote\_keyword = T\_vote\_keyword

# 查询按钮

#lambda表示绑定的函数需要带参数，请勿删除lambda，否则会出现异常

#thread\_it表示新开启一个线程执行这个函数，防止GUI界面假死无响应

B\_0\_keyword = Button(labelframe, text="搜索")

B\_0\_keyword.place(x=460, y=7)

frame\_root = Frame(labelframe, width=400)

frame\_l = Frame(frame\_root)

frame\_r = Frame(frame\_root)

self.frame\_root = frame\_root

self.frame\_l = frame\_l

self.frame\_r = frame\_r

self.B\_0\_keyword = B\_0\_keyword

columns = ("名字", "评分", "相关")

treeview = ttk.Treeview(frame\_l, height=10, show="headings", columns=columns)

treeview.column("名字", width=120, anchor='center') # 表示列,不显示

treeview.column("评分", width=120, anchor='center')

treeview.column("相关", width=280, anchor='center')

treeview.heading("名字", text="名字") # 显示表头

treeview.heading("评分", text="评分")

treeview.heading("相关", text="相关")

# 垂直滚动条

vbar = ttk.Scrollbar(frame\_r, command=treeview.yview)

treeview.configure(yscrollcommand=vbar.set)

# rbar = ttk.Scrollbar(frame\_r, command=treeview.xview)

# treeview.configure(xscrollcommand=rbar.set)

treeview.pack()

self.treeview = treeview

vbar.pack(side=RIGHT, fill=Y)

self.vbar = vbar

# rbar.pack(side= BOTTOM, fill=X)

# self.rbar = rbar

# 框架的位置布局

frame\_l.grid(row=0, column=0, sticky=NSEW)

frame\_r.grid(row=0, column=1, sticky=NS)

frame\_root.place(x=5, y=45)

#################################################################################################详情

labelframe\_movie\_detail = LabelFrame(root, text="影片详情")

labelframe\_movie\_detail.place(x=770, y=5)

self.labelframe\_movie\_detail = labelframe\_movie\_detail

# 框架布局，承载多个控件

frame\_left\_movie\_detail = Frame(labelframe\_movie\_detail, width=160,height=380)

frame\_left\_movie\_detail.grid(row=0, column=0)

self.frame\_left\_movie\_detail = frame\_left\_movie\_detail

frame\_right\_movie\_detail = Frame(labelframe\_movie\_detail, width=260,height=380)

frame\_right\_movie\_detail.grid(row=0, column=1)

self.frame\_right\_movie\_detail = frame\_right\_movie\_detail

#影片图片

self.label\_img = Label(frame\_left\_movie\_detail, text="", anchor=N)

self.label\_img.place(x=0,y=0) #布局

self.label\_img = self.label\_img

self.douban = Label(frame\_left\_movie\_detail, text="", fg="blue", cursor="hand2")

self.douban.place(x=0, y=210)

self.douban.bind("<Button-1>", lambda event: open(self.doubanUrl))

ft = font.Font(size=8, weight=font.BOLD)

label\_details = Label(frame\_right\_movie\_detail, text='',justify='left',anchor=NW,wraplength=250)

label\_details.place(x=10, y=10)

# label\_details.pack()

self.label\_details = label\_details

#################################################################################################在线播放

labelframe\_movie\_play\_online = LabelFrame(root, width=310, height=235, text="在线观看(双击播放)")

labelframe\_movie\_play\_online.place(x=5, y=305)

self.labelframe\_movie\_play\_online = labelframe\_movie\_play\_online

# 框架布局，承载多个控件

frame\_root\_play\_online = Frame(labelframe\_movie\_play\_online, width=324)

frame\_l\_play\_online = Frame(frame\_root\_play\_online)

frame\_r\_play\_online = Frame(frame\_root\_play\_online)

self.frame\_root\_play\_online = frame\_root\_play\_online

self.frame\_l\_play\_online = frame\_l\_play\_online

self.frame\_r\_play\_online = frame\_r\_play\_online

# 表格

columns\_play\_online = ("名称", "来源","播放地址")

treeview\_play\_online = ttk.Treeview(frame\_l\_play\_online, height=9, show="headings", columns=columns\_play\_online)

treeview\_play\_online.column("名称", width=110, anchor='center')

treeview\_play\_online.column("来源", width=50, anchor='center')

treeview\_play\_online.column("播放地址", width=120, anchor='center')

treeview\_play\_online.heading("名称", text="名称")

treeview\_play\_online.heading("来源", text="来源")

treeview\_play\_online.heading("播放地址", text="播放地址")

#垂直滚动条

vbar\_play\_online = ttk.Scrollbar(frame\_r\_play\_online, command=treeview\_play\_online.yview)

treeview\_play\_online.configure(yscrollcommand=vbar\_play\_online.set)

treeview\_play\_online.pack()

self.treeview\_play\_online = treeview\_play\_online

vbar\_play\_online.pack(side=RIGHT, fill=Y)

self.vbar\_play\_online = vbar\_play\_online

# 框架的位置布局

frame\_l\_play\_online.grid(row=0, column=0, sticky=NSEW)

frame\_r\_play\_online.grid(row=0, column=1, sticky=NS)

frame\_root\_play\_online.place(x=5, y=0)

#################################################################################################下载

labelframe\_movie\_bt\_download = LabelFrame(root, width=440, height=235, text="影视下载(选中条目右击复制下载地址，双击进入网页介绍)")

labelframe\_movie\_bt\_download.place(x=320, y=305)

self.labelframe\_movie\_bt\_download = labelframe\_movie\_bt\_download

# 框架布局，承载多个控件

frame\_root\_bt\_download = Frame(labelframe\_movie\_bt\_download, width=324)

frame\_l\_bt\_download = Frame(frame\_root\_bt\_download)

frame\_r\_bt\_download = Frame(frame\_root\_bt\_download)

self.frame\_root\_bt\_download = frame\_root\_bt\_download

self.frame\_l\_bt\_download = frame\_l\_bt\_download

self.frame\_r\_bt\_download = frame\_r\_bt\_download

# 表格

columns\_bt\_download = ("标题", "来源","地址")

treeview\_bt\_download = ttk.Treeview(frame\_l\_bt\_download, height=9, show="headings", columns=columns\_bt\_download)

treeview\_bt\_download.column("标题", width=110, anchor='center')

treeview\_bt\_download.column("来源", width=100, anchor='center')

treeview\_bt\_download.column("地址", width=200, anchor='center')

treeview\_bt\_download.heading("标题", text="标题")

treeview\_bt\_download.heading("来源", text="来源")

treeview\_bt\_download.heading("地址", text="地址")

#垂直滚动条

vbar\_bt\_download = ttk.Scrollbar(frame\_r\_bt\_download, command=treeview\_bt\_download.yview)

treeview\_bt\_download.configure(yscrollcommand=vbar\_bt\_download.set)

treeview\_bt\_download.pack()

self.treeview\_bt\_download = treeview\_bt\_download

vbar\_bt\_download.pack(side=RIGHT, fill=Y)

self.vbar\_bt\_download = vbar\_bt\_download

# 框架的位置布局

frame\_l\_bt\_download.grid(row=0, column=0, sticky=NSEW)

frame\_r\_bt\_download.grid(row=0, column=1, sticky=NS)

frame\_root\_bt\_download.place(x=5, y=0)

#################################################################################################云盘

labelframe\_movie\_save\_cloud\_disk = LabelFrame(root, width=420, height=135, text="云盘搜索")

labelframe\_movie\_save\_cloud\_disk.place(x=770, y=405)

self.labelframe\_movie\_save\_cloud\_disk = labelframe\_movie\_save\_cloud\_disk

# 框架布局，承载多个控件

frame\_root\_save\_cloud\_disk = Frame(labelframe\_movie\_save\_cloud\_disk, width=420)

frame\_l\_save\_cloud\_disk = Frame(frame\_root\_save\_cloud\_disk)

frame\_r\_save\_cloud\_disk = Frame(frame\_root\_save\_cloud\_disk)

self.frame\_root\_save\_cloud\_disk = frame\_root\_save\_cloud\_disk

self.frame\_l\_save\_cloud\_disk = frame\_l\_save\_cloud\_disk

self.frame\_r\_save\_cloud\_disk = frame\_r\_save\_cloud\_disk

# 表格

columns\_save\_cloud\_disk = ("来源名称", "是否有效", "播放地址")

treeview\_save\_cloud\_disk = ttk.Treeview(frame\_l\_save\_cloud\_disk, height=4, show="headings",columns=columns\_save\_cloud\_disk)

treeview\_save\_cloud\_disk.column("来源名称", width=120, anchor='center')

treeview\_save\_cloud\_disk.column("是否有效", width=110, anchor='center')

treeview\_save\_cloud\_disk.column("播放地址", width=150, anchor='center')

treeview\_save\_cloud\_disk.heading("来源名称", text="来源名称")

treeview\_save\_cloud\_disk.heading("是否有效", text="是否有效")

treeview\_save\_cloud\_disk.heading("播放地址", text="播放地址")

# 垂直滚动条

vbar\_save\_cloud\_disk = ttk.Scrollbar(frame\_r\_save\_cloud\_disk, command=treeview\_save\_cloud\_disk.yview)

treeview\_save\_cloud\_disk.configure(yscrollcommand=vbar\_save\_cloud\_disk.set)

treeview\_save\_cloud\_disk.pack()

self.treeview\_save\_cloud\_disk = treeview\_save\_cloud\_disk

vbar\_save\_cloud\_disk.pack(side=RIGHT, fill=Y)

self.vbar\_save\_cloud\_disk = vbar\_save\_cloud\_disk

# 框架的位置布局

frame\_l\_save\_cloud\_disk.grid(row=0, column=0, sticky=NSEW)

frame\_r\_save\_cloud\_disk.grid(row=0, column=1, sticky=NS)

frame\_root\_save\_cloud\_disk.place(x=5, y=0)

# 保存到云盘布局结束

#绑定事件

treeview.bind('<ButtonRelease-1>',self.open\_details\_event)#搜索结果单击绑定详情

treeview\_bt\_download.bind('<ButtonRelease-3>', self.copyURL)#下载列表绑定选中右击复制和双击打开网页

treeview\_bt\_download.bind('<Double-1>', self.open\_in\_browser)

treeview\_play\_online.bind('<Double-1>', self.open\_in\_browser\_online)#在线播放绑定左键双击播放

# treeview\_save\_cloud\_disk.bind('<Double-1>', self.open\_in\_browser\_cloud\_disk) # 表格绑定左键双击事件

ranking\_type.bind('<<ComboboxSelected>>',self.open\_ranking\_event)#排行榜combox绑定修改事件

B\_0\_keyword.configure(command=lambda:thread\_it(self.searh\_movie\_in\_keyword)) #按钮绑定单击事件

T\_vote\_keyword.bind('<Return>', handlerAdaptor(self.keyboard\_T\_vote\_keyword)) # 文本框绑定回车

# label\_bar.bind('<Double-1>',open(r'E:\python projects\project\_an\htmls\电影Bar图.html'))

# self.open\_ranking()#初始化排行榜

thread\_it(self.open\_ranking)

root.mainloop()

ui = uiObject()

ui.ui\_process()

**OnlineGet.py**

import urllib.request

from bs4 import BeautifulSoup

from urllib.parse import quote

domain='https://www.dytts.com'

def getPlayUrl(title,url):

request = urllib.request.Request(url=url)

html = urllib.request.urlopen(request).read().decode('utf-8')

soup = BeautifulSoup(html, 'lxml')

urlLst = soup.select('#playlist1 li a')

result=[]

for i in urlLst:

result.append([title+i.get\_text(),'电影天堂',domain+i['href']])

if len(result)>0:

return result

else:

return [[]]

def getOnlineUrl(keywords):

res = quote(keywords)

url = domain+"/so/-------------.html?wd="+res+"&submit="

s = urllib.request.urlopen(url)

soup = BeautifulSoup(s, 'lxml')

title = soup.select('.detail h3 a')

result=[]

for i in title:

result+=getPlayUrl(i.get\_text(),domain+i['href'])

if len(result)>0:

return result

else:

return [['未找到资源!','','']]

**RankingGet.py**

import requests

from bs4 import BeautifulSoup

from RankingView import \*

#定义函数获取并检测页面信息

def Text(url):

# 爬虫请求头信息

headers = {

'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/84.0.4147.105 Safari/537.36',

}

try:

respone = requests.get(url,headers=headers,timeout=30)

respone.raise\_for\_status()

#解决中文字符编码问题

respone.encoding = respone.apparent\_encoding

return respone.text

except:

return ""

def bang(url):

html = Text(url)

#使用BeautifulSoup解析页面信息

soup = BeautifulSoup(html, "html.parser")

name,zhishu,xingxi = [],[],[]

#开始获取作品名称信息

allname = soup.find\_all('p',class\_ = 'm-title')

for x in allname:

name.append(x.string) #使用列表方式 添加信息

#获取作品昨日指数信息

allzhishu = soup.find\_all("li",class\_='m-item-playcount')

for y in allzhishu:

zhishu.append(y.span.string)

#使用中文字符的空格填充

# print("{0}\t{1}\t{2:^40}".format("排名","作品名称","昨日指数",chr(12288)))

#用range函数获取前50个数据

for i in range(50):

# print("{0}\t{1}\t{2:^40}".format(i+1, name[i], zhishu[i],chr(12288)))

# xingxi.append([i+1, name[i], zhishu[i]])

xingxi.append([name[i], zhishu[i]])

return xingxi

def hot\_search(mode='电视剧'):

url=['https://www.360kan.com/rank/dianshi','https://www.360kan.com/rank/dianying']

if mode=='电视剧':

k=0

elif mode=='电影':

k=1

else:

return [['未找到!','','']]

list = bang(url[k])

if k==0:

dainshi\_bang(list)

reli2(list)

if k==1:

dainying\_bang(list)

reli(list)

return list

**RankingView.py**

from pyecharts import Bar

from pyecharts import HeatMap

import random

from pyecharts import WordCloud

from pyecharts import Bar3D

import random

def dainying\_bang(x):

pian,performance=[],[]

for i in x:

pian.append(i[0])

Str =''.join(i[1].split(','))

performance.append(int(Str))

# print(pian,performance)

bar = Bar("这是电影的热度指数", "名字无法全部显示")

bar.add("电影",pian, performance)

# bar.show\_config()

bar.render('htmls/电影Bar图.html')

def dainshi\_bang(x):

pian,performance=[],[]

for i in x:

pian.append(i[0])

Str =''.join(i[1].split(','))#注意这里i[2]是一个对象

performance.append(int(Str))

# print(pian,performance)

bar = Bar("这是电视剧的热度指数", "名字无法全部显示")

bar.add("电视剧",pian, performance)

# bar.show\_config()

bar.render('htmls/电视剧Bar图.html')

bar.width=1500

def reli(t):

x\_axis, y\_axis, y = [], [], []

for i in t:

x\_axis.append(i[0])

Str = ''.join(i[1].split(',')) # 注意这里i[2]是一个对象

y.append(int(int(Str)/200))

# print(x\_axis, y)

y\_axis = [

"one", "two", "three", "four", "five", "six"]

da, data = [], []

for i in y:

# 生成7个随机数，是他们的和为i

while 1:

n1 = random.randint(0, i)

n2 = random.randint(0, i)

n3 = random.randint(0, i)

n4 = random.randint(0, i)

n5 = random.randint(0, i)

n6 = random.randint(0, i)

su = n1 + n2 + n3 + n4 + n5 + n6

if su == i:

listRandom = n1, n2, n3, n4, n5, n6

break

da.append(listRandom)

for i in range(50):

for j in range(6):

x = [i, j, da[i][j]]

data.append(x)

# print(data)

heatmap = HeatMap()

heatmap.add(

"电影热力图",

x\_axis,

y\_axis,

data,

is\_visualmap=True,

visual\_text\_color="#000",

visual\_orient="horizontal",

)

heatmap.render('htmls/电影热力图.html')

def reli2(t):

bar3d = Bar3D("3D 柱状图示-热播电视剧", width=1200, height=600)

x\_axis, y\_axis, y = [], [], []

for i in t:

x\_axis.append(i[0])

Str = ''.join(i[1].split(',')) # 注意这里i[2]是一个对象

y.append(int(int(Str) / 1500))

x\_axis = x\_axis

# print(x\_axis, y)

y\_axis = ["one", "two", "three", "four", "five", "six"]

da, data = [], []

for i in y:

# 生成7个随机数，是他们的和为i

while 1:

n1 = random.randint(0, i)

n2 = random.randint(0, i)

n3 = random.randint(0, i)

n4 = random.randint(0, i)

n5 = random.randint(0, i)

n6 = random.randint(0, i)

su = n1 + n2 + n3 + n4 + n5 + n6

if su == i:

listRandom = n1, n2, n3, n4, n5, n6

break

da.append(listRandom)

for i in range(6):

for j in range(50):

x = [i, j, da[j][i]]

data.append(x)

# print(data)

range\_color = ['#313695', '#4575b4', '#74add1', '#abd9e9', '#e0f3f8', '#ffffbf']

bar3d.add(

"",

x\_axis,

y\_axis,

[[d[1], d[0], d[2]] for d in data],

is\_visualmap=True,

visual\_range=[0, 80],

visual\_range\_color=range\_color,

grid3d\_width=200,

grid3d\_depth=80,

grid3d\_shading="lambert",

is\_grid3d\_rotate=True,

# grid3d\_rotate\_speed=180

)

bar3d.render('htmls/电视剧热力图.html')

def word\_cloud(t):

name,value=[],[]

x,y=t[0],t[1]

for i in range(15):

# print(x[i][1],y[i][1])

name.append(x[i][1])

name.append(y[i][1])

Str =''.join(x[i][2].split(','))#注意这里i[2]是一个对象

value.append(int(Str))

Str =''.join(y[i][2].split(','))#注意这里i[2]是一个对象

value.append(int(Str))

wordcloud =WordCloud(width=1300, height=620)

wordcloud.add("", name, value, word\_size\_range=[20, 100])

wordcloud.show\_config()

wordcloud.render('htmls/词云.html')