**广州城市理工学院**

**2020-2021学年度第2学期**

Python语言

网络爬虫实训报告

学 院： 计算机工程学院

专业班级： 19计科4班

姓　　名： 刘泽葶

学　　号： 201910089041

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 一级指标 | 二级指标 | 三级指标及分值 | | 分值 | 评分及成绩 |
| 得分 |
| 代码  （占60%） | 1.达到基本要求，功能点完整（355%） | 网页抓取 | 15 | 55 |  |
| 页面解析 | 15 |
| 数据意义 | 5 |
| 数据存储 | 20 |
| 4.运行测试（5%） | | | 5 |  |
| 项目报告（20%） | 1.内容与结构完整（10%） | | | 10 |  |
| 2.规格满足要求，文字描述规范，思路描述清晰（10%） | | | 10 |  |
| 答辩成绩 （20%） | 1.对实训设计技术和结构熟悉，能进行合理说明（10%） | | | 10 |  |
| 2.能较好的回答答辩所提问题，解释合理清晰（10%） | | | 10 |  |
| 合计（百分制） | | | | |  |

目录

[一、 爬取主题说明 1](#_Toc11354)

[1.1 背景分析 1](#_Toc7446)

[1.2 数据来源 1](#_Toc18888)

[1.3 数据项 1](#_Toc20332)

[二、 技术可行性分析 1](#_Toc16898)

[2.1 技术方案 1](#_Toc11178)

[2.2 依赖包说明 2](#_Toc1703)

[三、 详细设计及实现 3](#_Toc22853)

[3.1 程序流程图 3](#_Toc29792)

[3.2 思维导图 4](#_Toc28959)

[3.3 关键代码 4](#_Toc15327)

[四、 运行测试 6](#_Toc22297)

[五、 总结 8](#_Toc6063)

[参考文献 9](#_Toc20753)

# 爬取主题说明

## **1.1 背景分析**

主题：音乐（QQ音乐排行榜）

背景：近年来，随着用户对音乐的品味和要求逐步提升，各种类型的音乐也源源不断地涌现在各大音乐平台。音乐常常被视为大众的精神寄托，能较好地表达出用户此时此刻的心情以及较热门的歌曲会被选作为视频配乐等。而音乐排行榜单是最能反映歌曲受大众喜爱的程度体现。对于用户而言，通过一个歌曲排行榜，可以快速地了解到当前榜单上的收听量最多的热门歌曲，找到并欣赏当下的流行歌曲；对于歌手而言，一个榜单的排名可以体现出自己歌曲的收听率和歌曲普及率等信息，上榜的歌曲可以更好的获得听众，为自己的歌曲增加知名度等。

## **1.2 数据来源**

数据来源：QQ音乐中目前有四个较为经典的排行榜，可以基本体现出大众的听歌导向。在本次的爬取项目中，我对四种不同类型的排行榜进行了网页爬取，分别是QQ音乐中的流行榜、飙升榜、热门榜以及新歌榜。在榜单的爬取中，我用到的网站URL来自于QQ音乐排行榜代码中Network专门存储歌曲信息列表的网页，四个排行榜共使用到了4个URL网页信息。

网页：（URL过长，在此展示重定向后的简短版，使用的转换网站是https://0x3.me/）

流行榜：“https://0x9.me/E5Ofw”

飙升榜：“https://0x9.me/9BAj4”

热歌榜：“https://0x9.me/oG91Y”

新歌榜：“https://0x9.me/Doay1”

## **1.3 数据项**

在提取的歌曲信息列表中，我使用到的数据项共有4个，分别是歌曲排名（rank）、上升率（rankValue）、歌曲名称（title）和歌手名称（singerName），其中，上升率是表示该歌曲相较于前一天的排名的上升数，也表示收听率的增加量，歌曲排名是榜单中20首歌曲的排序序号。在取数据的过程中，我使用到了键值对匹配值的方法将需要的数据保存到歌曲列表和数据库当中。

# **技术可行性分析**

## **2.1 技术方案**

Ⅰ.运行环境

硬件：Windows10系统、AMD处理器；

软件：Python3.9、PyCharm2020以及数据库MySQL5.5 Command Line client；

展示：SQLyog和GUI界面

Ⅱ.使用到的库

数据库：mysql库、mysql-connector第三方库（用于数据库的连接）；

获取和解析网页：requests第三方库（模拟浏览器向服务器发送请求，破解网站反爬）、requests-exceptions库以及库中的RequestException（使用try-except测试网页爬取是否成功），bs4库中的BeautifulSoup（美化解析后的网页）；

GUI展示：tkinter库以及库中的ttk（将数据使用树结构展示成列表形式），PIL库以及库中的ImageTk和Image（在界面中插入图片）。

Ⅲ.运用到技术

①将爬取到的歌曲信息数据保存到数据库中；

②使用键值对匹配获取所需要的数据；

③使用BeautifulSoup优化解析后的网页；

④使用tkinter实现GUI界面并且用ttk树结构储存展示歌曲信息；

⑤将键值对匹配获取的信息以列表的形式保存，使用append()将数据添加至列表中；

⑥使用实参传递形参的方法进入到IF的选择判断功能中调用解析函数，读取并展示用户需查看的榜单信息（通过参数的不同来匹配对应榜单的URL）；

⑦使用ImageTk-PhtotImage()函数来插入图片美化界面等。

Ⅳ.功能函数

本次程序爬取使用的功能函数共有6个，分别是爬取和解析网页数据功能：get\_soup(URL)、读取信息至列表功能：save\_list(URL)、保存信息至数据库功能：save\_database(LST)、子窗口展示功能：sub\_pages(URL)、选择URL使用IF判断功能：case(string)、首页展示功能：music\_index()，另外还要1个主函数main()以及mydb模块中的创建数据库和创建表等功能。

## **2.2 依赖包说明**

Requests\_exceptions包：在获取网页数据时，使用try-except功能检查数据是否获取成功。若获取失败，即情况为RequestException，返回空值。

PIL包：在向GUI界面中插入图片时，需要导入PIL包。

Bs4包：解析页面时，为美化和优化页面结构，通常使用Bs4包中的BueatifulSoup库。

Mysql包：在数据库的连接和使用时，为使数据保存至数据库中，需要导入Mysql包。

# **详细设计及实现**

## **3.1 程序流程图**

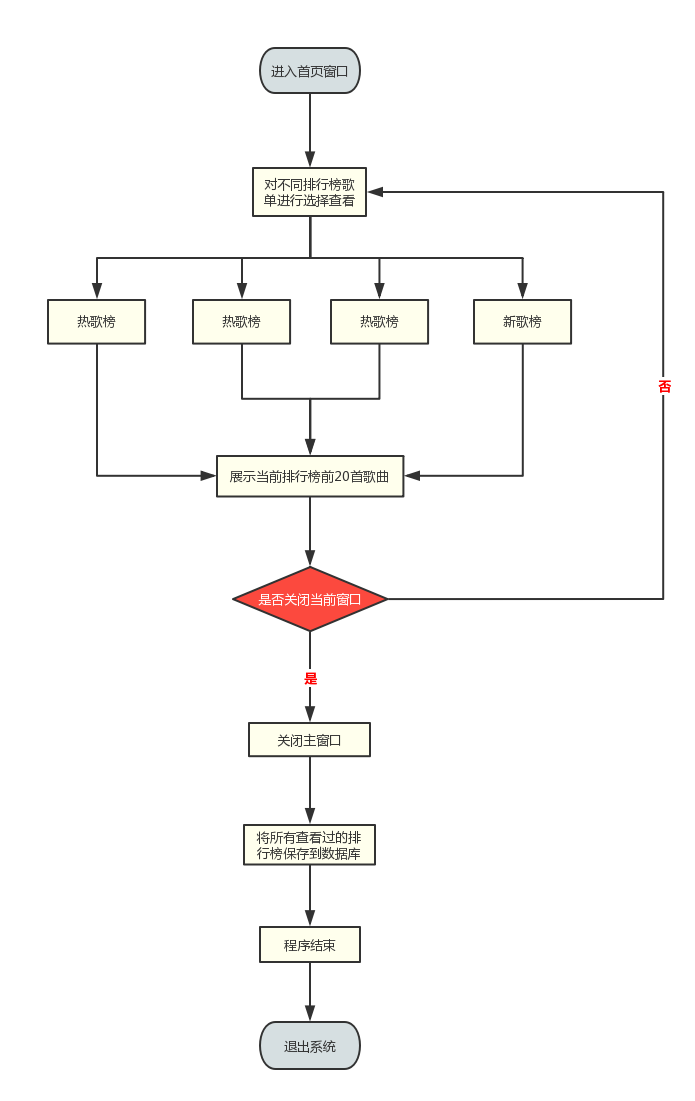


图3-1 QQ音乐排行榜程序流程图

上图为QQ音乐排行榜程序流程图，对用户进入程序后程序执行的任务进行大致的解析，程序共有两个窗口，分别为主窗口和子窗口，主窗口供用户选择排行榜的类别，子窗口用于展示该排行榜的20首歌曲，只有用户关闭了全部子窗口并且关闭了主窗口（即退出程序），用户查看的榜单数据才会被保存到数据库中。

## **3.2 思维导图**

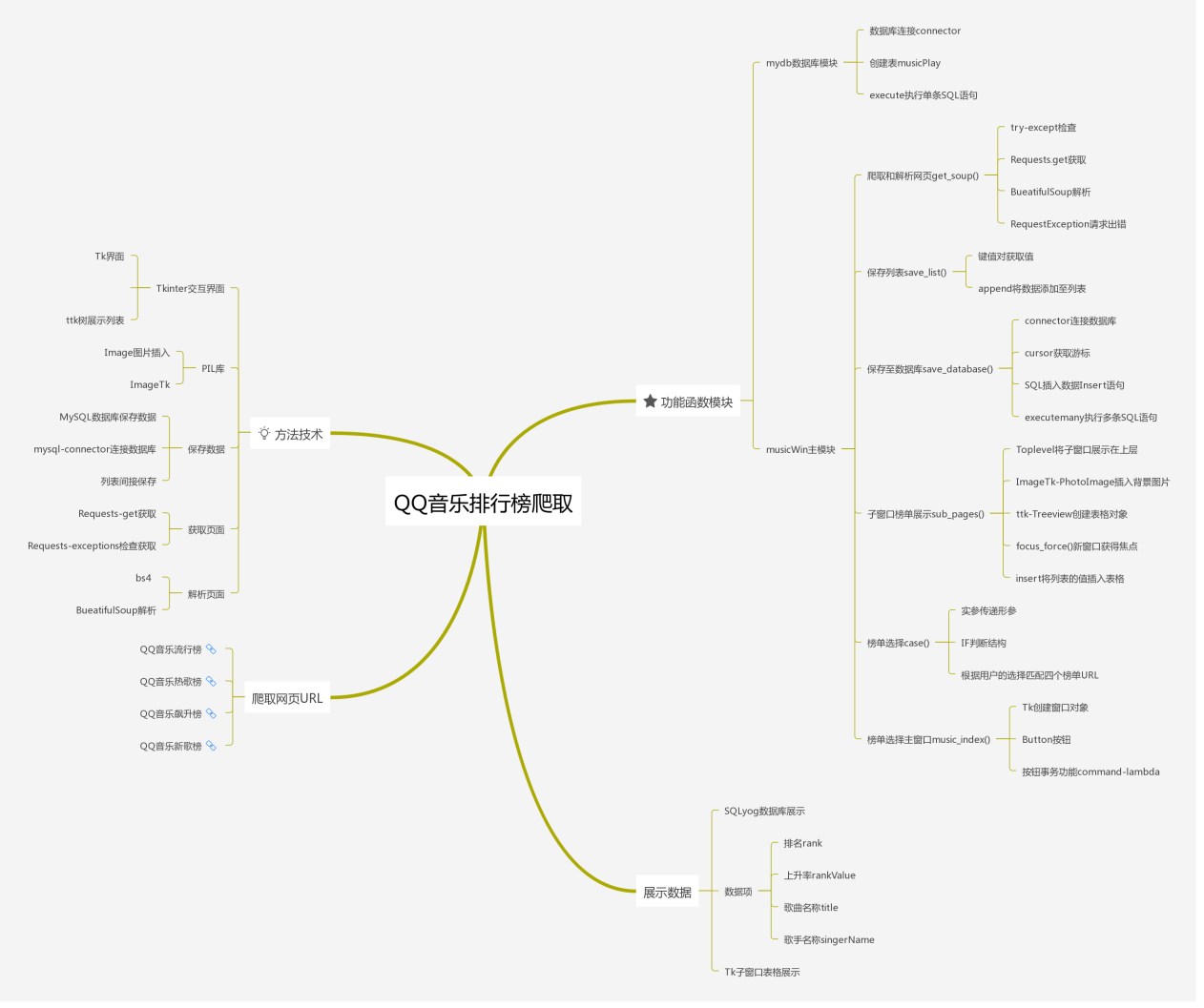


图 3-2 QQ音乐排行榜思维导图

上图为QQ音乐排行榜爬取的思维导图，大致可分为四个板块，分别是使用到的方法技术、功能函数模块、需要爬取的网页URL以及展示数据板块。其中，重要的板块为功能函数模块中的musicWin主模块，包含了使用到的主要实现功能的函数。在网页URL中，因需要爬取的URL过长，不便添加展示，因此在此处插入了对应的URL短链接，使用到的转换网站为0x3.me短网址（‘https://0x3.me/’）

## **3.3 关键代码**

①网页抓取

try:  
 # 添加头部信息  
 headers = {  
 'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.114 Safari/537.36'  
 }  
 resp = requests.get(url, headers=headers) # resp是响应结果

在Requests.get获取网页数据时，使用try-except结构检测网页是否获取成功，并且添加头部信息headers破解网站反爬，可以模拟浏览器向服务器发送请求，将爬取的结果保存到resp中。

②页面解析

if resp.status\_code == 200:  
 resp.encoding = "utf-8"  
 music = resp.text  
 soup = BeautifulSoup(music, "html.parser") # 解析网页

使用IF结果判断读取到的网页状态是否正常响应，将读取到的信息以“utf-8”编码的方式写成文本，再用BueatifulSoup对数据进行解析。

③将需要的数据存进列表

# 从字典当中根据detail的键获取值(获取到歌曲信息)  
list\_song = soup\_json['detail']['data']['data']['song']  
  
# 遍历列表获取每首歌曲的字典(序号，歌曲名，歌手名),并把这些歌曲数据元组加到列表中  
for item in list\_song:  
 lst.append((item['rank'], item['rankValue'], item['title'], item['singerName']))

在json中将获取到的字典以键值对的方式查找到所需要提取的值，从字典中根据detail的键获取值，使用for in遍历列表，将这些歌曲数据元组依次通过append添加到列表中。

④保存数据库

# 编写sql语句（事先创建好的一个数据库）  
sql = 'insert into musicPlay values(%s,%s,%s,%s)' # 向表中插入数据(排名，上升率，歌曲名，歌手名)  
# 获取储存歌曲的列表  
val = lst  
# 开始执行sql语句(sql语句和值value)  
mycursor.executemany(sql, val) # 同时执行多条语句

编写SQL插入语句Insert INTO table values()，并通过executemany()同时执行多次SQL语句，将列表中的数据依次插入到数据库中的musicPlay表。

⑤子窗口表格展示（仅展示主要部分）

sub\_win = tkinter.Toplevel() # 创建window子窗口

tree = ttk.Treeview(sub\_win) # 创建表格对象  
tree["columns"] = ("排名", "上升率", "歌曲名", "歌手名") # 定义列

sub\_win.focus\_force() # 新窗口获得焦点

for i in range(20):  
 tree.insert('', 'end', values=lst1[i]) # 插入Tk树

为使点击后出现新的子窗口，需要使用到Toplevel顶级窗口让子窗口展现在上层，并且使用focus\_force()使得新窗口获得焦点。窗口展示数据列表用到树结构Ttk-Treeview()创建表格对象，使用insert()将列表数据依次插入到Tk树中。

⑥数据库建表

# 获取连接对象  
conn = mysql.connector.connect(host='localhost', port=3306, user='root', passwd='12345', database='music\_play', charset="utf8") # 主机名，账号和密码  
  
# 获取cursor（游标）的对象  
cursor = conn.cursor()

# 建立存放歌曲榜单数据的表musicPlay  
sql = """create table musicPlay(  
 rank int(10),  
 rankValue char(10),  
 songName varchar(60),  
 singerName varchar(100)  
 )DEFAULT CHARSET=utf8;"""  
  
cursor.execute(sql) # 执行一次

在mydb.py模块中使用SQL创建表语句Create table创建存放数据的musicPlay表，调用execute函数执行一次语句进行表创建，之后无需再执行。连接数据库对象需要使用connector.connect来创建连接，建立cursor游标进行获取对象。

# **运行测试**

①界面（主窗口）测试



图4-1 主窗口界面

在创建好数据库后，进入程序主界面，可以看到选择排行榜类型的提示信息，下方设有四个按钮，分别是流行榜、热歌榜、飙升榜和新歌榜。在每个按钮绑定的事件中，可以获得相应榜单的URL页面，随后即进入榜单前20首歌曲的列表展示。

②榜单表格（子窗口）测试



图4-2-1 热歌榜 图4-2-2 飙升榜

点击按钮后，进入排行榜单展示，左图为点击按钮‘热歌榜’后的子窗口，右图为点击按钮‘飙升榜’后的子窗口，可以发现两个榜单的上升率一栏中数据类型不同，差了一个‘%’，通过返回网页页面介绍我们可以知道，‘热歌榜’榜单中的上升率是歌曲排名相较于上一周排名的差值，而‘飙升榜’榜单中的上升率是歌曲搜索播放量相较于前一天数量的上升占比。结合榜单的意义和作用，可以得出入榜后的音乐搜索播放量明显提升的结论。

③关闭主程序后保存信息至数据库（展示器：SQLyog）



表4-3-1 数据库歌曲数据信息表

数据保存成功后程序提示：

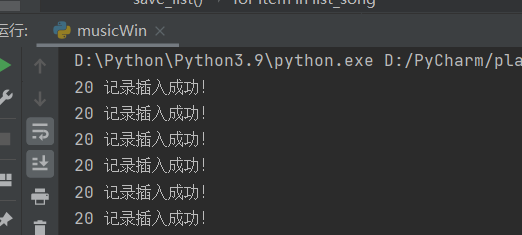


图4-3-2 结果成功保存截图

当用户查看榜单结束后，关闭程序主窗口，函数返回save\_database()的值，该函数将数据插入数据库后，输出行数并打印“记录插入成功！”，将用户查看的列表数据全部通过列表插入值的方式保存至数据库里，最终结果可以在SQLyog中点击‘查看表’查看。

# **总结**

通过本次的实操爬取网页数据，我对网页框架结构的认识更为地清晰。总的来说，这次的爬取并不是我最满意的一种方法。在检查网页的同时曾尝试使用正则表达式匹配Element框架中歌曲信息，发现爬取并不能出现body中的数据，于是我尝试了另一种方法——爬取Network中存放歌曲信息的URL页面，发现在这个页面可以直接使用键值对匹配值，于是使用了列表来保存字典中所需键对应的值。连接数据库时，需要事先在MySQL command中创建一个数据库，再通过执行创建语句完成建表。在编写代码时，我对部分掌握得不够好的知识进行了复习，并通过网上找资料来获取问题的解决方法，在这个寻找的过程中我收获颇多。例如，在Tkinter图形界面设计中，我涉及的知识不是特别地深，我简单地使用了按钮调用事务处理函数command，但是却发现四个按钮的事件在还未按下按键时就已经开始执行，查阅过后发现问题原因是command函数带有参数造成的，Tkinter要求由按钮（或者其它的插件）触发的控制器函数不能含有参数若要给函数传递参数，需要在函数前添加lambda。在页面数据信息查找中，我发现需要爬取的网页URL过长，为使URL较为简短，我使用到了重定向缩短网址（即通过重定向的方式把长网址转化成一个简短的网址），转换的网站为https://0x3.me/。本次实操对自己来说非常有益，既使我发现了自己的弱项，也通过查阅书籍、资料、博客等学会了很多解决方法。让我在不断的“跌倒”和“碰壁”中走出了一条可行的道路，由于时间较为仓促，本次爬取项目较为粗糙，但我相信通过往后的锻炼，我会有更有信心把这件事做得更加符合我的预期。（在此，由衷地感谢我的任课老师。Thank!）

# **参考文献**

网站CSDN

https://blog.csdn.net/weixin\_44677806/article/details/100107889

<https://blog.csdn.net/qq_40084026/article/details/113860440>

<https://blog.csdn.net/congcong7267/article/details/95626960>

<https://blog.csdn.net/ysblogs/article/details/88530124>

<https://blog.csdn.net/weixin_39833509/article/details/88743267>

<https://blog.csdn.net/qq_41556318/article/details/85597698>

教材

《Python程序设计应用教程》 夏敏捷 张广西 中国铁道出版社

第25页 序列的数据结构

第35页 Python控制语句

第51页 Python函数与模块

第114页 Tkinter图形界面设计

第160页 Python数据库应用

第256页 BueatifulSoup库

第262页 网络爬取实战

第265页 Python图像处理