****

****

**QG工作室数据挖掘组最终考核**

**题 目\_\_\_**实现电影推荐系统**\_\_**

**学 院\_\_\_\_**计算机学院**\_\_\_\_\_\_\_\_**

**专 业\_\_\_\_**计算机科学与技术**\_\_**

**年级班别\_\_\_\_**18级计科一班\_**\_\_\_\_\_**

**学 号\_\_\_\_**3218004712**\_\_\_\_\_\_\_\_**

**学生姓名\_\_\_\_**陆冰红\_**\_\_\_\_\_\_\_\_\_\_\_**

**2019年05月**

目录

**一、项目思路3**

1.整体思路3

2.模块分点4

**二、爬虫模块4**

1.完成情况4

2.爬取电影详细信息4

3.存储爬取数据5

**三、推荐算法5**

1.完成情况5

2.算法基本思想5

3.实现流程5

处理数据6

推荐算法6

算法评估7

**四、GUI界面8**

1.完成情况8

2.GUI界面9

**五、实验环境15**

**六、数据库16**

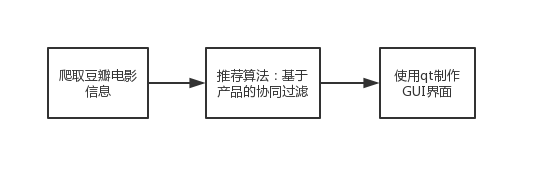
**七、系统可扩展性16**

**八、项目难点及解决16**

**九、心得体会16**

1. **项目思路**

1.整体思路



2．模块分点

spider：爬虫代码

movieSpider.py （爬取电影详细信息）

comment.py （爬取电影评论）

database：

database.py（存储爬取数据到数据库的代码）

stor.py （处理并存储其他数据文件到数据库）

algorithm：推荐算法

algorithm.py（由用户评分数据实现推荐算法

rec.py（由已存储的电影相关系数文件获取推荐列表

views:界面文件夹

mainUI.py (主界面)

detailsPage.py （电影详情界面）

sql.py（操作数据库相关函数）

modifyInfo.py (修改密码)

signin.py（注册界面）

src：存放数据文件

**二、爬虫模块**

1.完成情况：

①按照分类爬取电影详细信息并存储信息到数据库表movie，根据所爬取信息中的海报链接爬取海报，以电影id命名海报，存放到本地文件夹。

②从数据库中获取电影链接数据，构造短评链接，爬取电影短评存放到数据库中表comments。

不足：由于还没有足够清晰考核爬虫模块和算法模块之间的联系，开始时以为数据集是算法用的就没有先了解数据集的使用便开始动手爬虫，所以关于电影详情是直接按照豆瓣电影分类爬取，且爬取信息不足，以致后面直接使用师兄的数据做界面。

2.爬取电影详细信息

Ⅰ.基本思路

按照分类和页数构造网页链接，解析网页获取所需信息存储到数据库，根据电影信息爬取电影海报。

Ⅱ.爬取内容：

directors (导演)

id (电影id)

casts (演员信息)

rate（评分）

title（电影标题）

star （星级）

rate （评分）

url （电影详细信息链接）

cover （海报链接：用去获取电影海报）

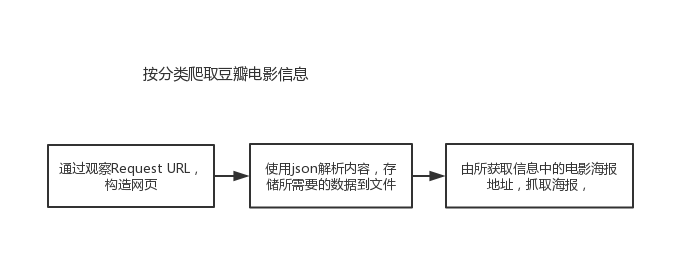
tag （类型标签）

以上信息作为字段名创建表movie存储到mysql

Ⅲ.使用库

Requests获取网页 json解析网页 panda,numpy处理数据 time设置time.sleep以防止被识别出爬虫

Ⅳ.步骤



1. 构造网页：

'https://movie.douban.com/j/new\_search\_subjects?sort=U&range=0,10&tags=%E7%94%B5%E5%BD%B1&start={}&genres={}'.format(page\_start,gen)

其中page\_start为当前页面起始电影编号，gen为分类值

1. 获取网页并解析

requests.get获取网页

json.load解析内容

1. 存储到文件

pd转为dataframe格式并调用to\_csv存储为csv文件

1. 存储到数据库

创建表movie将电影详细信息文件插入到表中

1. 由海报链接爬取海报存储到文件夹

从表中获取id和cover字段，get获取网页得到海报pic，pic.content写入文件夹中

3.爬取电影短评

Ⅰ.基本思路

从数据库中获取电影id和url,构造电影短评链接，使用xpath爬取信息存储到数据库中comments表

Ⅱ.爬取内容

电影评论

Ⅲ.使用库

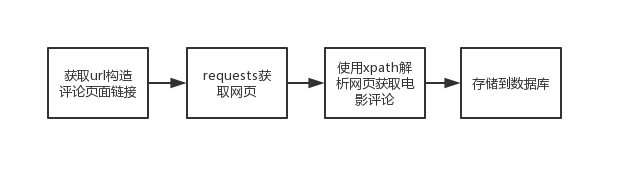
pymysql提取id,url

requests 获取网页

lxml.etree 解析

random，time 设置随机时间间隔

Ⅳ.步骤



1. 从数据库中提取id，url，构造电影短评链接
2. 获取网页，使用xpath解析
3. 处理所获取信息
4. 每爬取100个存入数据库

4.爬取结果

爬取电影3000+详细信息

根据已有电影信息爬取电影评论

**三、推荐算法**

1.完成情况

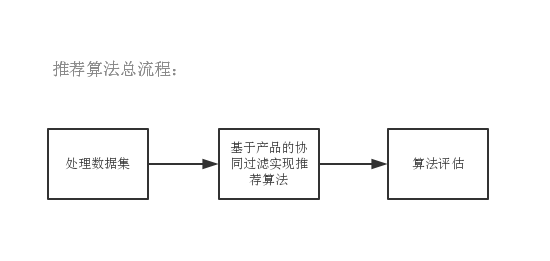
包括两个文件：其一为包括划分，获取用户电影矩阵，计算电影相关系数，返回推荐列表，评估算法的完整算法。用以得到用户-电影矩阵和电影相关度矩阵。

另一部分依照相同思想，使用已保存的用户-电影矩阵和电影相关度矩阵中获取数据进行计算获取推荐列表。

2.基本思想

基于产品的协同过滤思想：根据已有的用户电影评分数据，计算电影之间的相关系数，再根据用户历史行为生成推荐列表。

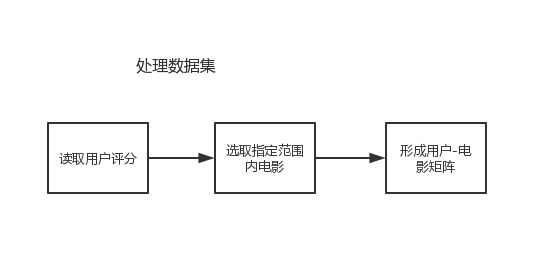
3.实现过程



1. 处理数据：

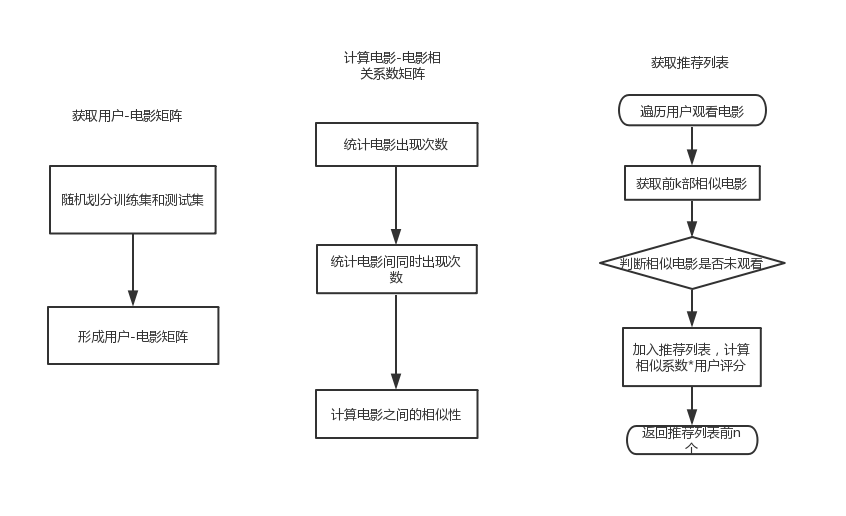
从所给数据集中提取前两百个用户用以实现推荐算法。

主要步骤：



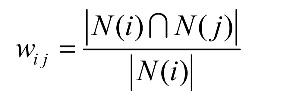
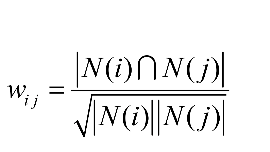
1. 推荐算法

Ⅰ.流程图



Ⅱ.相似度计算

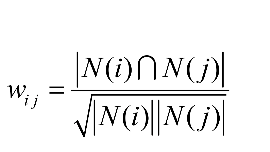
从定义“观看该电影的用户同时也观看的其他电影”，使用公式

计算相似度。但该公式会造成任何电影都和热门电影相似，进而改用公式，惩罚j的权重，减轻热门电影和许多电影相似的可能性。

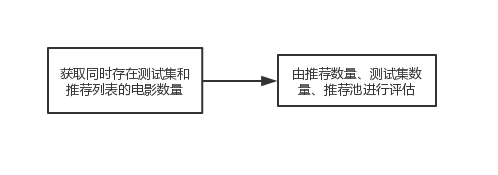
Ⅲ.详细步骤

1. 随机划分训练集和测试集，得到用户-电影相似度矩阵

使用pd.read\_csc从文件ratings总获取用户评分信息，由随机数分别形成训练集、测试集，以字典{‘user’,{‘mov’,’rating’}} 形式存储用户评分信息，转化为dataframe形式存储到csv文件。

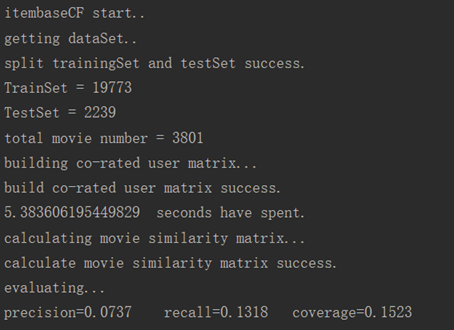
1. 计算电影相似度
   1. 遍历用户与用户已评分的所有电影统计电影用户数。
   2. 对用户已评分电影，遍历当前用户已评分的其他电影，统计电影之间同时出现的次数，形成电影相关矩阵。
   3. 遍历电影与相关电影列表，由已得数据和公式计算电影相似度。
2. 获取推荐列表
   1. 遍历用户已评分电影，获取与当前电影相似度最接近的前k个，计算评分与相关度乘积，存入推荐列表。
   2. 遍历完成后，对值降序排列返回前n个作为推荐列表。
3. 算法评估

通过准确率，召回率，覆盖率进行评估



1. 覆盖率：推荐电影 / 推荐池
2. 召回率：同时存在于推荐列表和测试集中的电影数量 / 测试集中电影数量（样本
3. 准确率：同时存在于推荐列表和测试集中的电影数量 / 推荐电影数量

Ⅳ.测试结果



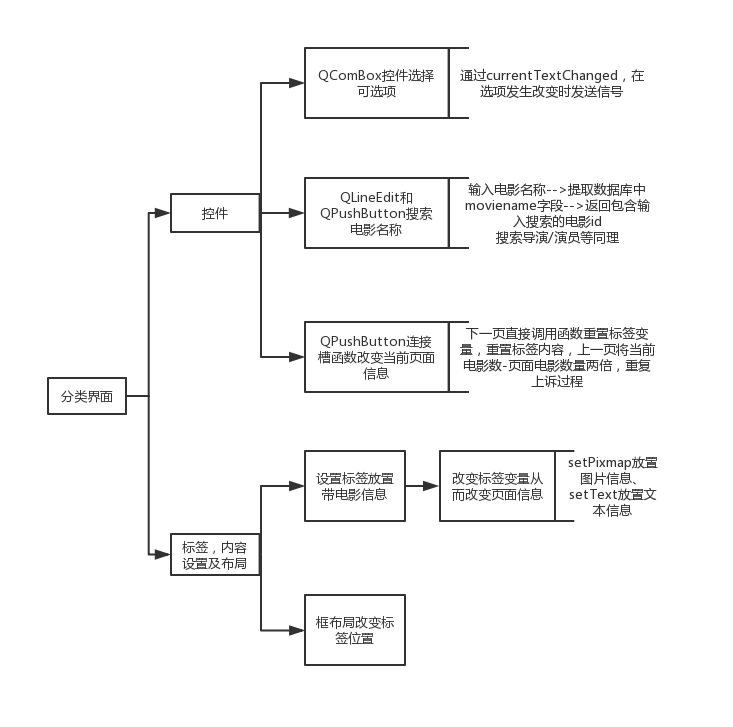
**四、GUI界面**

1. 使用的库：pyqt5
2. 完成情况

完成分类搜索界面，用户主界面，电影详情界面，简单的修改信息和注册界面。

主界面为QTabWidget，选项卡包括分类搜索和用户主界面。电影详情、修改信息和注册界面为新界面弹出。

1. GUI界面
   1. 分类界面



程序运行默认显示分类搜索界面。分类界面界面包括QComBox控件选择地区/类型，搜索文本框进行搜索电影名称和六个电影标签。

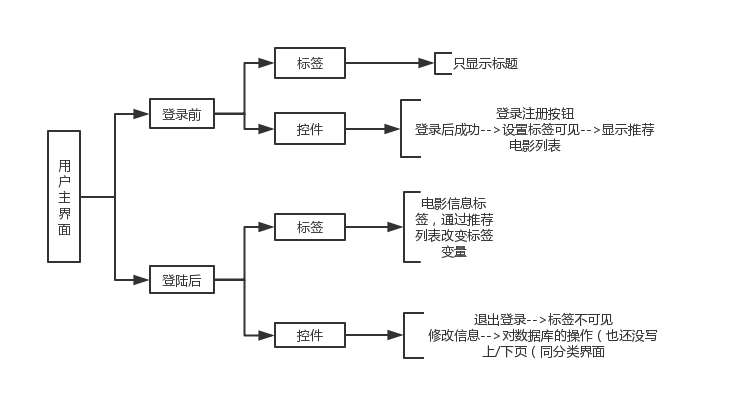
选择类型/地区的实现：通过currentTextChange发送信号，传递所选类型，提取数据库中id和type字段，返回type与选择类型相同的id构成的列表。(地区返回place字段所匹配的id列表，年代通过判断years字段内容是否所选范围)

搜索：moviename字段中包含搜索内容的对应电影id列表。

电影标签设置标签变量，改变类型/搜索/翻页时改变标签变量内容实现标签信息的改变。



* 1. 用户主界面



默认只显示登录框和注册按键。登录成功后显示用户推荐列表，修改信息和退出登录按键。

登录用户为数据集中所提取的前两百个，编号为用户id,密码5824。修改信息界面实际上只有修改密码。注册界面填入userid，密码和所喜爱的电影类型，新用户推荐列表为所喜欢的类型的按降序排列的列表。

推荐列表的获得：由algorithm.rec文件中推荐算法获取当前用户电影推荐列表。

登录：用户id和密码都输入后设置登录按键可用，选择登录后从数据库中提取匹配信息，存在则登录成功，获取推荐列表；不存在匹配信息则清空文本框，重新输入。

修改密码：输入原始密码和新密码，检查旧密码与用户id是否匹配，匹配则检查新密码输入是否相同，相同则更新数据库中当前用户的密码信息，修改成功。

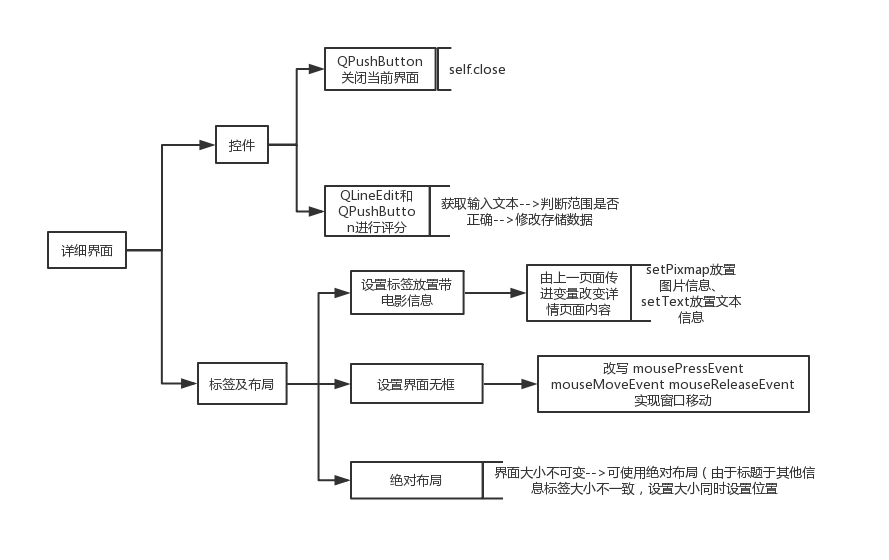


1登录前

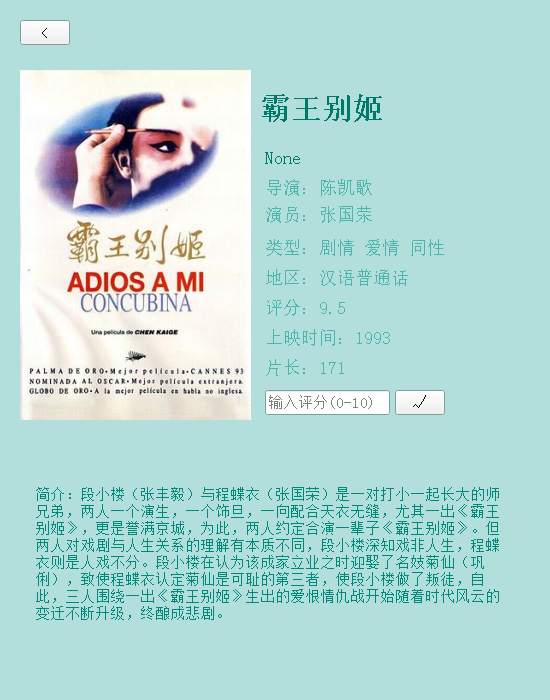


2登陆后

* 1. 电影详细界面

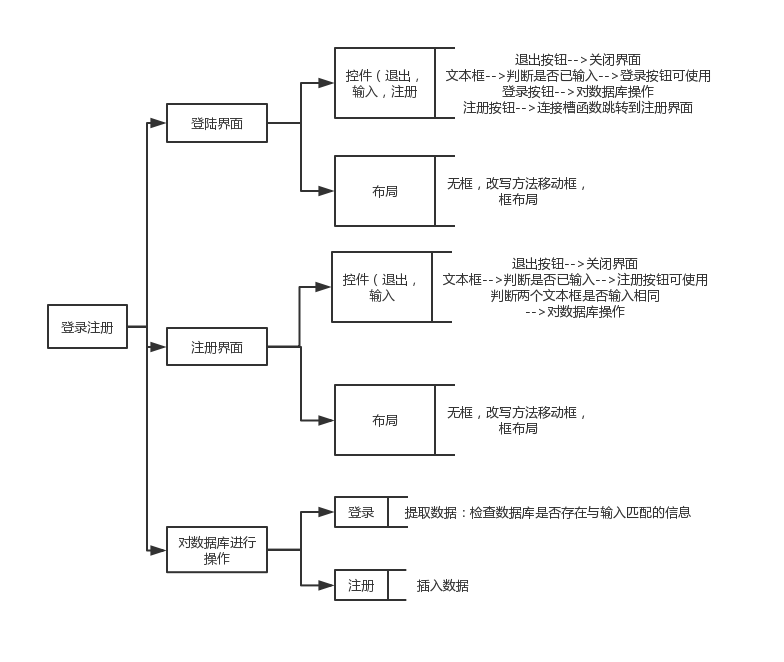


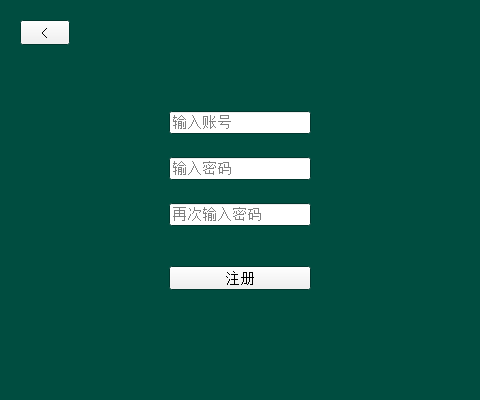
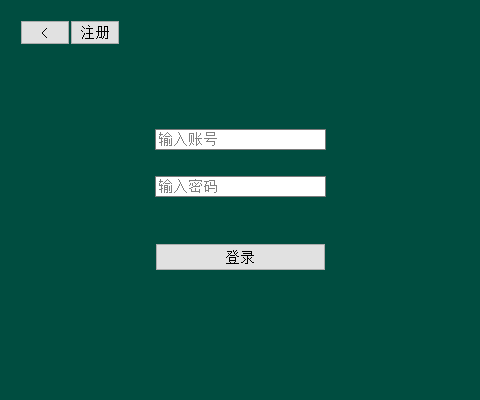
由新窗口呈现所选择的电影的详细信息。界面无框，添加按键选择退出，改写方法实现窗口移动。详细界面使用绝对定位放着各个信息标签。同样通过改变标签变量改变界面电影信息。



* 1. 注册界面

用户id，密码都输入后注册按钮可用，选择注册按钮后检查数据库中是否存在相同id，不存在则检查两次密码是否输入相同，相同则注册成功，将注册信息插入数据库中存放用户信息的表中。



**五、实验环境**

1.实验环境：PyCharm

2.Python环境：python3.7

3.第三方库

* requests
* sys
* pyqt5
* pandas
* numpy
* pymysql
* json
* time

**六、数据库**

1. 使用到数据库的模块：
2. 存储爬取信息（对数据库的插入操作）

Movie 存放电影详细信息

Comments 存放电影短评

Ⅱ.登录注册

登录 （提取操作，检查是否存在用户id，是否于password匹配

注册 （提出操作，检查是否以存在用户id，不存在时进行插入操作

修改信息（更新操作）

Ⅲ.评分

电影评分矩阵Dataframe.to\_sql存储到数据库，评分时对数据库进行更新操作

**七、参考文献**

《python3网络爬虫实战开发》崔庆才

《推荐系统实践》项亮

《pyqt5快速开发与实战》王硕 孙洋洋

谷歌以及csdn，博客园等

**九、难点及解决**

1.数据库的操作：爬取信息无法存入数据库，更新数据库信息出现的问题

2.界面交互：登录信息的返回，主界面信息的更换，新用户的推荐

**八、心得体会**

1、存在问题

* 1. 不懂得要怎么并行就先爬虫再算法，所以也没有先了解好数据集怎么用，，没有按数据集的内容爬取而是直接按分类随便爬的电影信息，没有将两个模块连接起来。
  2. 没有考虑好界面交互，控件功能后面要补充的时候发现前面一些限制好像没法直接改所以做不完整。
  3. 登录或各种切换页面可能是在计算推荐列表程序时常处于未响应状态
  4. 着急下手，各个方面都没有学得系统，做一半才发现有其他方法可以好一点的实现就重新来过浪费了很多时间，以致也没有按时间规划进行，后面文档ppt就又太赶了虽然师兄多给了我们一天

2.一些感想

至少还是做到了最后。真快阿就这么快要结束了。

这些天一直赶着时间边学边做，更多时候都是需要用到什么查什么然后也没记住什么，至少还是做出来一点东西。起码知道了做一个项目大概是怎么操作的现在。

还是遇到不少困难。首先做爬虫从爬虫基础开始看起看博客爬了top250再看博客按分类爬其他所有，但都是没什么目的性的，一开始没认为数据集跟爬虫是有关系的，做爬虫的无目的性导致没用自己爬的东西做界面也没有爬取评论。做算法没有先进行数据清洗，没有考虑这些数据后续有什么用途，直接用整个数据集计算相似度和进行评估，然后跑了一个晚上起来发现电脑发烫黑屏…直到最后一天做用户主界面要用到推荐列表才想起先处理一下数据比较好。

然后界面这里，大致扫了qt教程要做的时候还是啥都不懂，第一次做了很多个界面跳得自己都搞不清怎么改，后面发现QTabWidget就删掉写了三两天的东西重新开始，但是后面登录注册详细信息什么的又回到弹出新界面那里，都是什么都没学系统不直到该怎么实现各种功能，交互性也没有考虑好，以为有一点思路就写得出来，后面还是像写算法，看懂基本思想也写得真是艰难。重写界面得时候剩下时间很不够简直一直崩溃边缘，起码终于还是写出一点界面~

大概还是有很多的想法没有实现，如果再重来一定可以更好一点。考核下来还是积攒了一些的经验，也学到了不少东西，从第一次接触爬虫数据库机器学习算法一切都让我很懵逼到现在能说是有一点模糊的认识。

总之很感谢qg给的这样一个学习机会~还是好希望好希望好希望不是止于此，但是无论如何以后也会继续努力~