**推荐算法功能点主要介绍**：

首先，通过个人登录页面跳转到推荐主页，我们的推荐算法主要应用在此页面，接下来我主要介绍一下该页面实现的具体功能：

推荐页面主要对食物种类进行分类，同时可以按名搜索食物，整个页面下方是前100名美食的推荐图片，我们可以随意打开推荐的美食图片对其进行评分，当评分低时，推荐排名自动排到后边，右侧是我的最近邻居的展示，最近邻居指和自己口味最相似的一个用户。

**接下来是我们推荐算法技术点的应用以及相应的代码**：

一、首先在推荐算法中，我们在获取数据方面用到了爬虫这一技术，

我们使用的是requests+正则来匹配网页内容，对于数据量较多的采用了多线程抓取的方法，共2个例子，分别是抓取美团受欢迎前一百美食和美团的美食数据。这2个例子采用的方法大同小异。

   1、首先选择想要爬取的网站

   2、确定要用的模块，requests,json,re三个模块，如果想加快爬取速度可以加一个Pool

   3、 网页请求，先得到整个页面，需要加一个headers来进行请求，否则会被网站拦截

   4、格式化整个页面，通过patter的正则来匹配，找出我们需要的内容，

   5、   获取数据，findall，然后通过yield将数据返回，yield 是一个类似 return 的关键字，迭代一次遇到yield时就返回yield后面(右边)的值

   6、遍历获取到的数据

   7、保存到相应的文档中

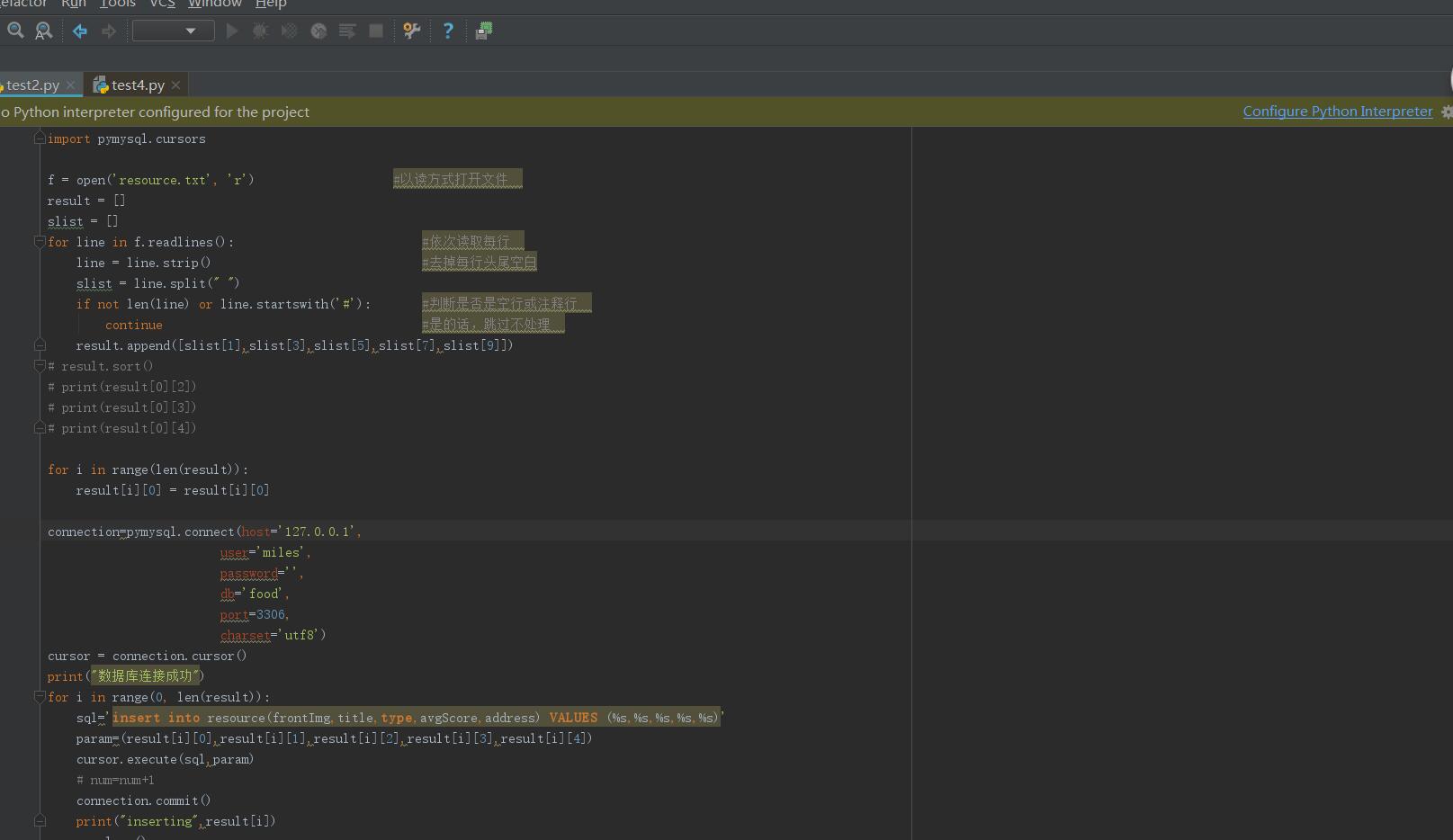
   8、关闭文档，

   9、提示数据保存成功。

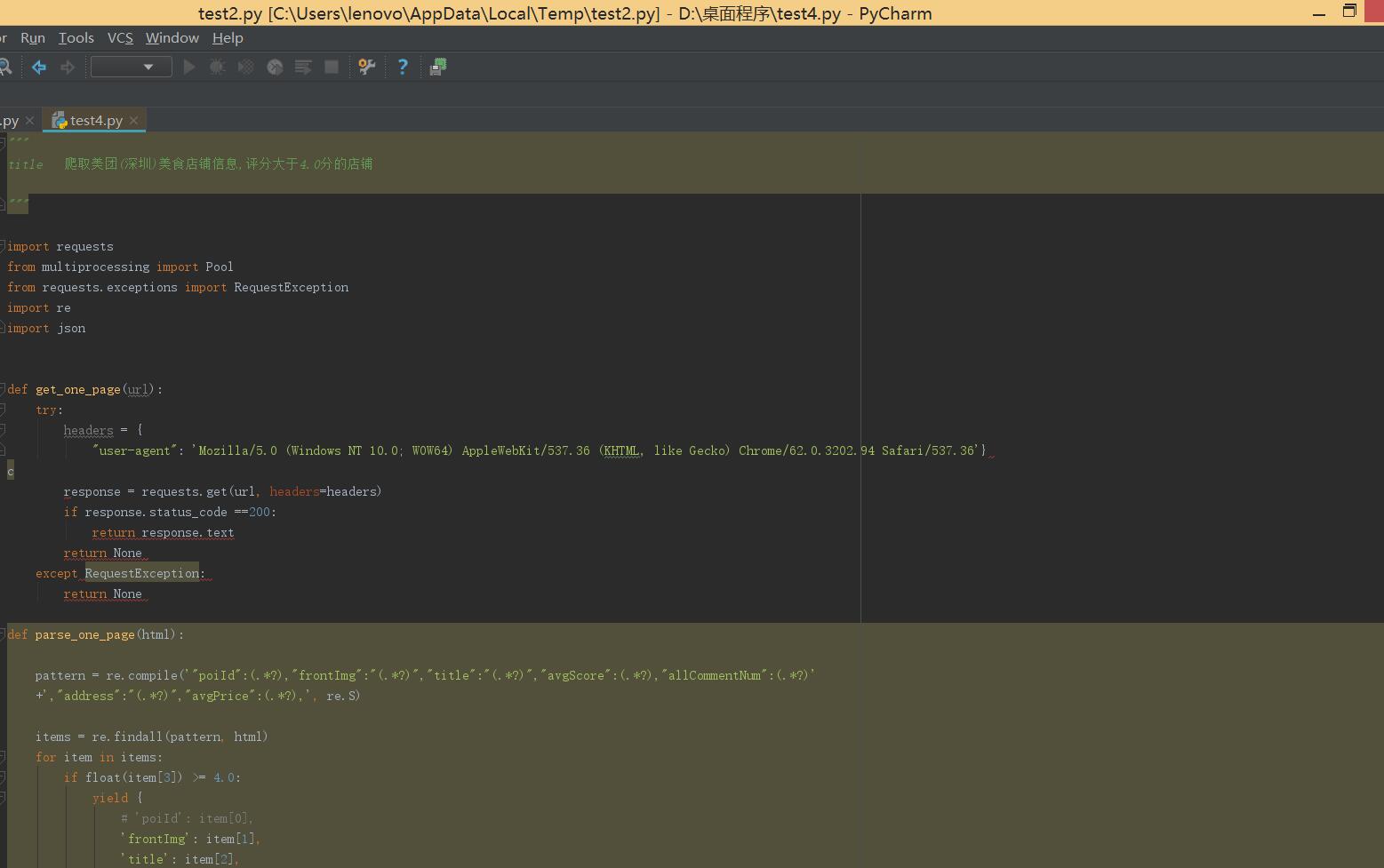
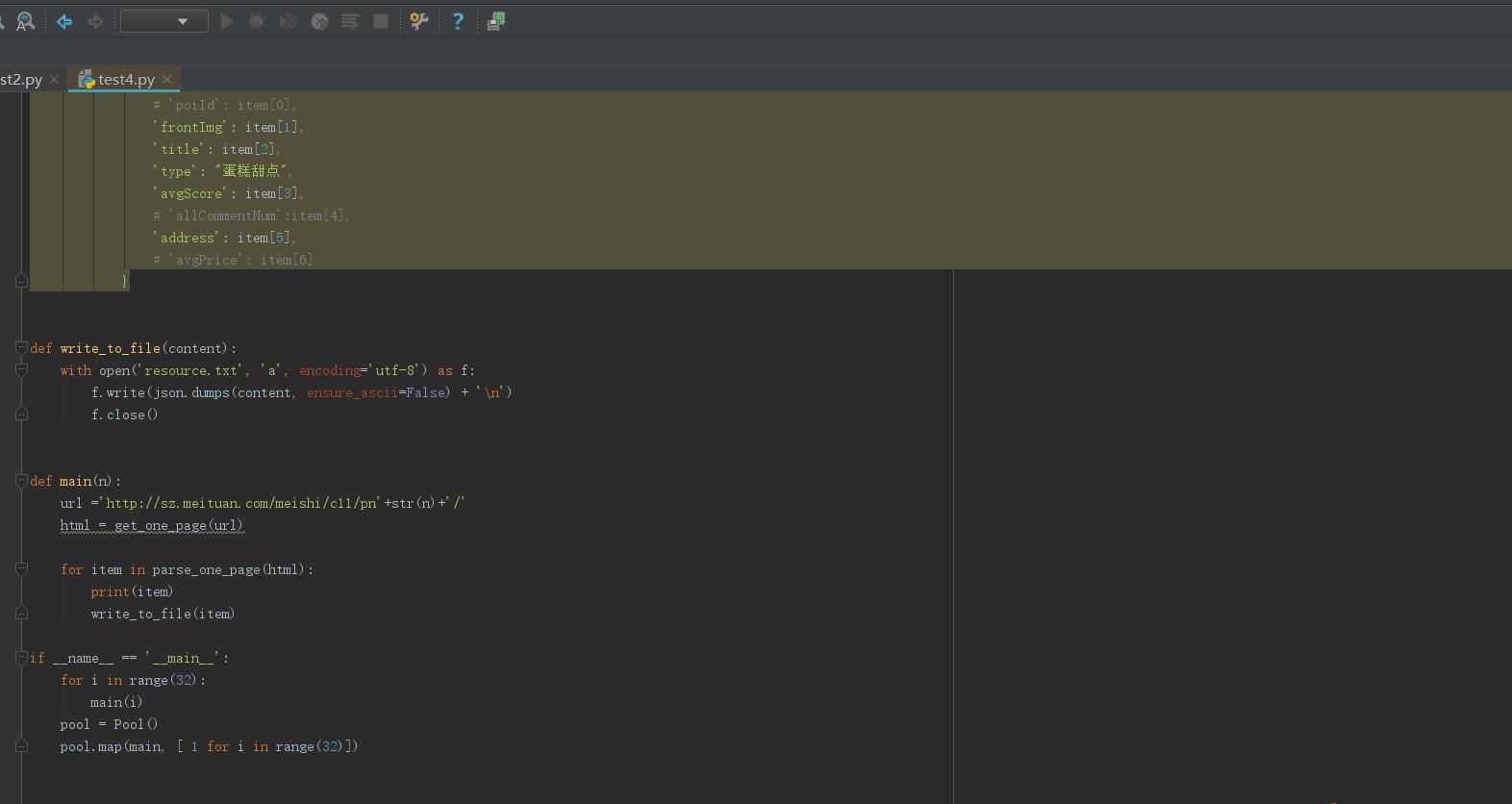
1. 爬取美团前100名美食的数据：

部分代码如下：

连接数据库代码如下:



爬取数据代码如下：



结果：将爬取的数据存放到文本文件中，

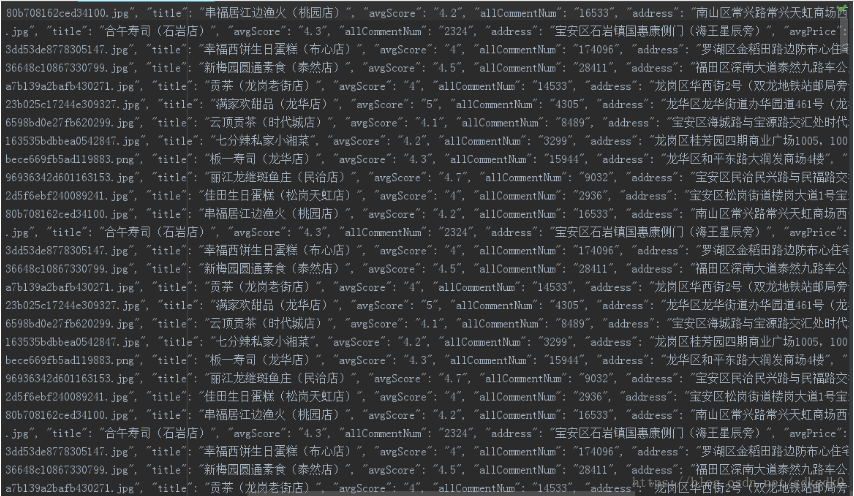
因为我这边采用的是线程池爬取的，所以有时候是不按顺序进行存储的，如果采用非多线程方式，就会按照顺序进行存储。

2. 爬取美团（深圳）美食店铺信息，评分大于4.0分的店铺

做为一名吃货，想知道我所在城市的美食店，所以爬取评分较高的店铺信息：

美团的这个网页的不同之处在于，全部是通过js渲染生成的，所以我是拿到页面后，在js里面查找到的数据，然后用正则来匹配。

结果如下：



对于后期，可以选择把这个数据落库，常用的可以放在mongodb或者mysql数据库中进行存储。

二：对于推荐算法的使用：

我们实现的是和自己口味类似的几个人选择菜品的推荐以及受欢迎菜品的推荐功能。

**基本原理**：

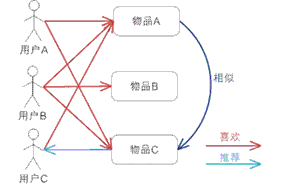
基于物品的 CF 的原理和基于用户的 CF 类似，只是在计算邻居时采用物品本身，而不是从用户的角度，即基于用户对物品的偏好找到相似的物品，然后根据用户的历史偏好，推荐相似的物品给他。从计算的角度看，就是将所有用户对某个物品的偏好作为一个向量来计算物品之间的相似度，得到物品的相似物品后，根据用户历史的偏好预测当前用户还没有表示偏好的物品，计算得到一个排序的物品列表作为推荐。图 3 给出了一个例子，对于物品 A，根据所有用户的历史偏好，喜欢物品 A 的用户都喜欢物品 C，得出物品 A 和物品 C 比较相似，而用户 C 喜欢物品 A，那么可以推断出用户 C 可能也喜欢物品 C。

我们通过之前的爬虫技术已经获得了大量数据，对于大量数据进行加工，我们选择了mahout的itemcf算法

1.itemcf算法原理

##### 如图：基于物品的 CF 的基本原理





基于用户对某种食物的偏好找到相似口味的食物，然后根据用户的历史偏好，推荐相似的食物给他。

2．算法实现

首先计算食物之间的相似度，然后根据食物的相似度和用户的历史行为给用户生成推荐列表

步骤如下：

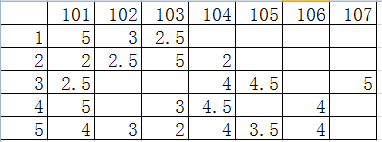
1. 建立物品的同现矩阵  
   2. 建立用户对物品的评分矩阵  
   3. 矩阵计算推荐结果

拿一个例子说明：

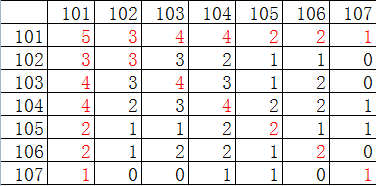
测试数据：

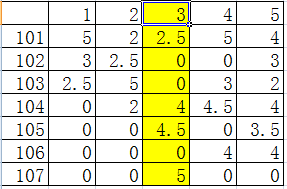
1,101,5.0  
1,102,3.0  
1,103,2.5  
2,101,2.0  
2,102,2.5  
2,103,5.0  
2,104,2.0  
3,101,2.5  
3,104,4.0  
3,105,4.5  
3,107,5.0  
4,101,5.0  
4,103,3.0  
4,104,4.5  
4,106,4.0  
5,101,4.0  
5,102,3.0  
5,103,2.0  
5,104,4.0  
5,105,3.5  
5,106,4.0  
数据说明：用户ID,物品ID,评分

转化为矩阵：

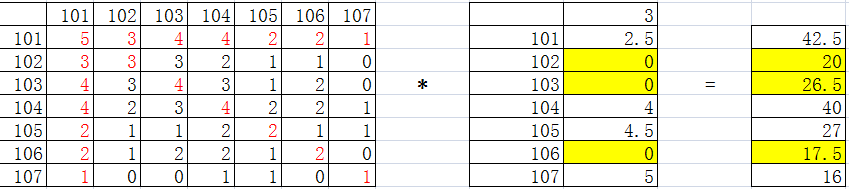


步骤1：建立物品的同现矩阵  
按用户分组，找到每个用户所选的物品，单独出现计数及两两一组计数。即统计任一两个物品同时被评分的**人的个数**，首行首列和对角线值相同代表物品单独出现的次数，例如对于101商品1、2、3、4、5用户都有评分即其本身出现了5次所有（101，101）为5，（101，102）代表同时对101和102评分的用户个数即1、2、5号用户共三个人所以（101，102）为3.

****  
步骤2：建立用户对物品的评分矩阵

  
对于3号这个用户就是标黄了的

步骤3：矩阵计算推荐结果 其中对3号用户的计算为同现矩阵与3号用户对物品的评分矩阵相乘



对于3用户推荐其没有评分的的102、103、106中的103，103的26.5值最高，和计算结果一致

1 [104:1.280239,106:1.1462644,105:1.0653841,107:0.33333334]  
2 [106:1.560478,105:1.4795978,107:0.69935876]  
3 [103:1.2475469,106:1.1944525,102:1.1462644]  
4 [102:1.6462644,105:1.5277859,107:0.69935876]  
5 [107:1.1993587]