1、配置本地Python环境

2、pycharm打开项目，安装Python依赖包：

（1）先安装这个： pip install -r requirements.txt -i https://pypi.mirrors.ustc.edu.cn/simple/

（2）可能需要补充的依赖：

pip install ujson

pip install django-grappelli

pip install pandas -i <https://pypi.mirrors.ustc.edu.cn/simple/>

conda install -c conda-forge scikit-surprise （这个安装会比较久）

pip install django==2.0 -i <https://pypi.mirrors.ustc.edu.cn/simple/>

pip install mysqlclient

3、启动项目

pycharm 的terminal命令行下执行：python manage.py runserver

浏览器输入 http://127.0.0.1:8000 访问

或者http://127.0.0.1:8001 访问

# # 运行说明

每次运行需打开pycharm软件

**-pycharm 的terminal命令行下执行：python manage.py runserver**

**- 浏览器输入 http://127.0.0.1:8000 访问**

**或者http://127.0.0.1:8001 访问**

---------------------------------------------------------

数据库迁移： 【运行前可能需要迁移数据库 】

python manage.py makemigrations

python manage.py migrate

python manage.py runserver

创建管理员：python manage.py createsuperuser

推荐系统核心文件:

/music/recommend.py文件

/music/views.py文件

1.将基于用户的协同过滤算法与用户画像相结合进行推荐，提高推荐列表数据的成熟度。

2.系统在Windows平台上搭建，采用Python3实现各项功能；采取sqlite进行数据的存储，通过Django框架连接系统的前、后端。

3.使用的数据集为kaggle平台上kkbox举办的—KKBox's Music Recommendation Challenge比赛的公开数据集，kkbox是亚洲领先的音乐流媒体服务提供商， 拥有世界上最全面的亚洲流行音乐库，拥有超过3000万首音乐曲目。

1. 针对数据集使用SVD矩阵分解进行相似相关度的计算分析，根据已有的评分情况， 分析出评分者对各个因子的喜好程度以及歌曲包含各个因子的程度，最后再反过来根据分析结果预测评分，根据评分的结果生成推荐列表。

# 协同过滤

协同过滤推荐是在信息过滤和信息系统中正迅速成为一项很受欢迎的技术。与传统的基于内容过滤直接分析内容进行推荐不同，协同过滤分析用户兴趣，在用户群中找到指定用户的相似（兴趣）用户，综合这些相似用户对某一信息的评价，形成系统对该指定用户对此信息的喜好程度预测。

协同过滤是迄今为止最成功的推荐系统技术，被应用在很多成功的推荐系统中。电子商务推荐系统可根据其他用户的评论信息，采用协同过滤技术给目标用户推荐商品。

协同过滤算法主要分为**基于启发式**和**基于模型式**两种。

其中，基于启发式的协同过滤算法，又可以分为**基于用户的协同过滤算法**（User-Based）和**基于项目的协同过滤算法**（Item-Based）。

* 启发式协同过滤算法主要包含3个步骤：  
  1）收集用户偏好信息；  
  2）寻找相似的商品或者用户；  
  3）产生推荐。

“巧妇难为无米之炊”，协同过滤的输入数据集主要是用户评论数据集或者行为数据集。这些数据集主要又分为**显性数据**和**隐性数据**两种类型。其中，显性数据主要是用户打分数据，譬如用户对商品的打分，五分制的1分，2分等。

但是，显性数据存在一定的问题，譬如用户很少参与评论，从而造成显性打分数据较为稀疏；用户可能存在欺诈嫌疑或者仅给定了部分信息；用户一旦评分，就不会去更新用户评分分值等。

而隐性数据主要是指用户点击行为、购买行为和搜索行为等，这些数据隐性地揭示了用户对商品的喜好。

隐性数据也存在一定的问题，譬如如何识别用户是为自己购买商品，还是作为礼物赠送给朋友等。

## 1.基于用户的协同过滤

用相似统计的方法得到具有相似爱好或者兴趣的相邻用户，所以称之为以用户为基础（User-based）的协同过滤或基于邻居的协同过滤(Neighbor-based Collaborative Filtering)。

### 1.1方法步骤：

1.收集用户信息  
收集可以代表用户兴趣的信息。一般的网站系统使用评分的方式或是给予评价，这种方式被称为“主动评分”。另外一种是“被动评分”，是根据用户的行为模式由系统代替用户完成评价，不需要用户直接打分或输入评价数据。电子商务网站在被动评分的数据获取上有其优势，用户购买的商品记录是相当有用的数据。  
2.最近邻搜索(Nearest neighbor search, NNS)  
以用户为基础（User-based）的协同过滤的出发点是与用户兴趣爱好相同的另一组用户，就是计算两个用户的相似度。例如：查找n个和A有相似兴趣用户，把他们对M的评分作为A对M的评分预测。一般会根据数据的不同选择不同的算法，目前较多使用的相似度算法有Pearson Correlation Coefficient（皮尔逊相关系数）、Cosine-based Similarity（余弦相似度）、Adjusted Cosine Similarity（调整后的余弦相似度）。

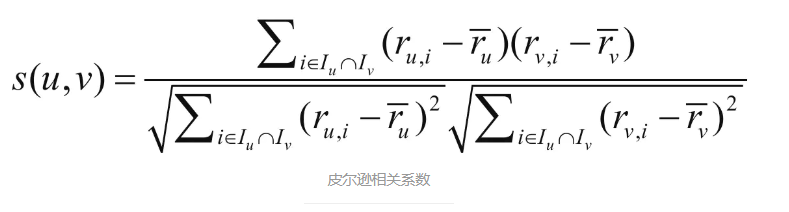
基于用户（User-Based）的协同过滤算法首先要根据用户历史行为信息，寻找与新用户相似的其他用户；同时，根据这些相似用户对其他项的评价信息预测当前新用户可能喜欢的项。

给定用户评分数据矩阵R，基于用户的协同过滤算法需要定义相似度函数s：U×U→R，以计算用户之间的相似度，然后根据评分数据和相似矩阵计算推荐结果。

### 1.2如何选择合适的相似度计算方法

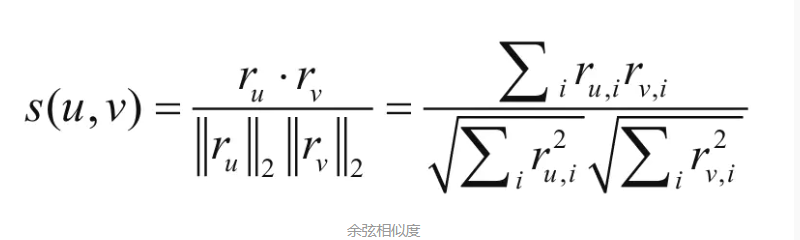
在协同过滤中，一个重要的环节就是如何选择合适的相似度计算方法，常用的两种相似度计算方法包括皮尔逊相关系数和余弦相似度等。**皮尔逊相关系数**的计算公式如下所示：

皮尔逊相关系数



其中，i表示项，例如商品；Iu表示用户u评价的项集；Iv表示用户v评价的项集；ru，i表示用户u对项i的评分；rv，i表示用户v对项i的评分；表示用户u的平均评分；表示用户v的平均评分。

另外，**余弦相似度**的计算公式如下所示：

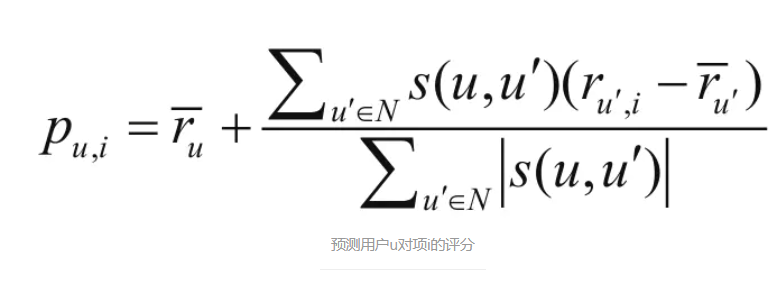


余弦相似度

### 1.3计算用户u对未评分商品的预测分值

另一个重要的环节就是计算用户u对未评分商品的预测分值。首先根据上一步中的相似度计算，寻找用户u的邻居集N∈U，其中N表示邻居集，U表示用户集。然后，结合用户评分数据集，**预测用户u对项i的评分**，计算公式如下所示：

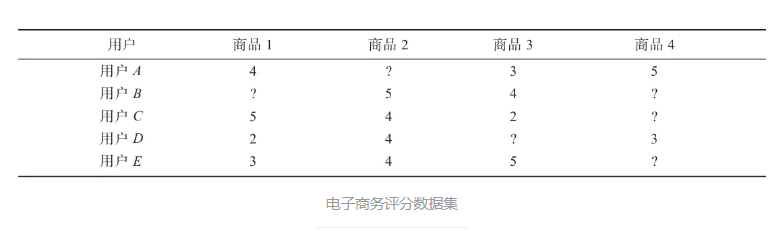
预测用户u对项i的评分



其中，s（u，u'）表示用户u和用户u'的相似度。

### 1.4通过例子理解

假设有如下电子商务评分数据集，预测用户C对商品4的评分。



电子商务评分数据集

表中？表示评分未知。根据基于用户的协同过滤算法步骤，计算用户C对商品4的评分，其步骤如下所示。  
（1）寻找用户C的邻居  
从数据集中可以发现，只有用户A和用户D对商品4评过分，因此候选邻居只有2个，分别为用户A和用户D。用户A的平均评分为4，用户C的平均评分为3.667，用户D的平均评分为3。

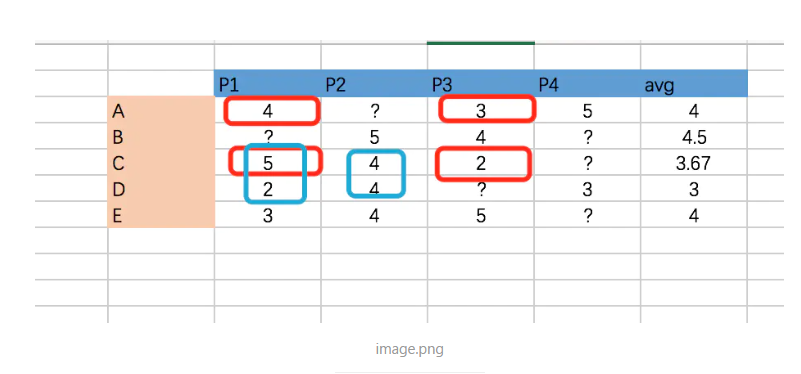
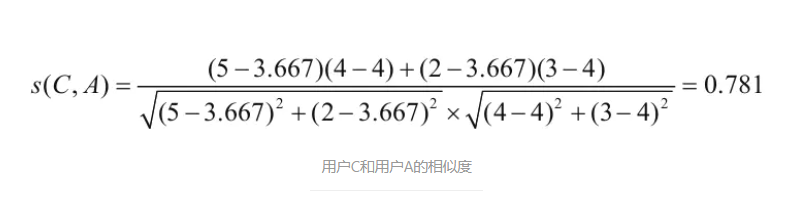


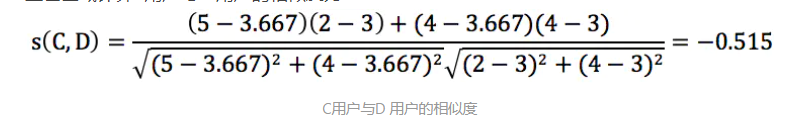
image.png

根据皮尔逊相关系数公式：  
红色区域计算C用户与A用户，用户C和用户A的相似度为：



用户C和用户A的相似度

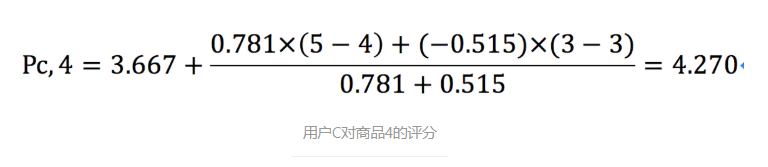
蓝色区域计算C用户与D 用户的相似度为:



C用户与D 用户的相似度

（2）预测用户C对商品4的评分  
根据上述评分预测公式，计算用户C对商品4的评分，如下所示：

用户C对商品4的评分



依此类推，可以计算出其他未知的评分。

## 2.基于项目的协同过滤

以用户为基础的协同推荐算法随着用户数量的增多，计算的时间就会变长，所以在2001年Sarwar提出了基于项目的协同过滤推荐算法(Item-based Collaborative Filtering Algorithms)。以项目为基础的协同过滤方法有一个基本的假设“能够引起用户兴趣的项目，必定与其之前评分高的项目相似”，通过计算项目之间的相似性来代替用户之间的相似性。

## 2.1方法步骤：

1.收集用户信息  
同以用户为基础（User-based）的协同过滤。  
2.针对项目的最近邻搜索  
先计算已评价项目和待预测项目的相似度，并以相似度作为权重，加权各已评价项目的分数，得到待预测项目的预测值。例如：要对项目 A 和项目 B 进行相似性计算，要先找出同时对 A 和 B 打过分的组合，对这些组合进行相似度计算，常用的算法同以用户为基础（User-based）的协同过滤。  
3.产生推荐结果  
以项目为基础的协同过滤不用考虑用户间的差别，所以精度比较差。但是却不需要用户的历史数据，或是进行用户识别。对于项目来讲，它们之间的相似性要稳定很多，因此可以离线完成工作量最大的相似性计算步骤，从而降低了在线计算量，提高推荐效率，尤其是在用户多于项目的情形下尤为显著。

基于项目（Item-Based）的协同过滤算法是常见的另一种算法。与User-Based协同过滤算法不一样的是，Item-Based协同过滤算法计算Item之间的相似度，从而预测用户评分。也就是说该算法可以预先计算Item之间的相似度，这样就可提高性能。Item-Based协同过滤算法是通过用户评分数据和计算的Item相似度矩阵，从而对目标Item进行预测的。

### 2.2相似度计算方法

和User-Based协同过滤算法类似，需要先计算Item之间的相似度。并且，计算相似度的方法也可以采用皮尔逊关系系数或者余弦相似度，这里给出一种电子商务系统常用的相似度计算方法，即基于物品的协同过滤算法，其中相似度计算公式如下所示：

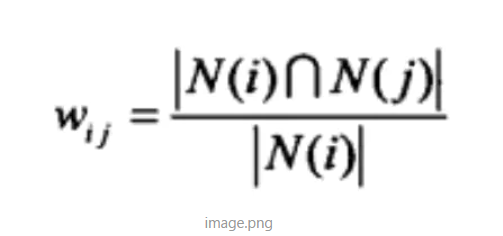
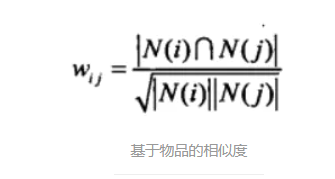


image.png

这里，分母|N(i)|是喜欢物品i的用户数，而分子|N(i)∩N(j)|是同时喜欢物品i和j的用户，但是如果物品j很热门，就会导致Wij很大接近于1。因此避免推荐出热门的物品，我们使用下面的公式：



基于物品的相似度

从上面的定义可以看出，在协同过滤中两个物品产生相似度是因为他们共同被很多用户喜欢，也就是说每个用户都可以通过他们的历史兴趣列表给物品“贡献”相似度。

计算物品相似度首先建立用户-物品倒排表（即对每个用户建立一个包含他喜欢的物品的列表），然后对每个用户，将他物品列表中的物品两两在共现矩阵中加1。

### 2.3用户u对于物品j的兴趣

得到物品之间的相似度后，可以根据如下公式计算用户u对于物品j的兴趣：

用户u对于物品j的兴趣

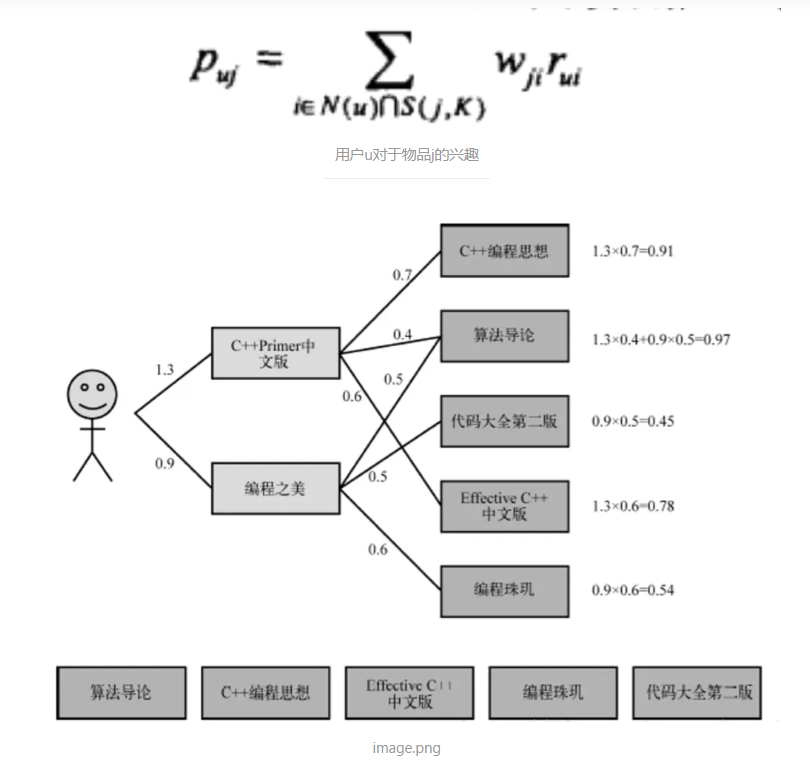
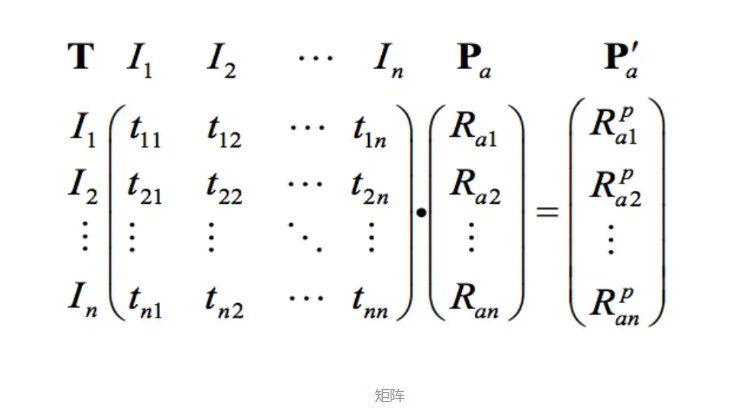


image.png

这里N(u)是用户喜欢的物品的集合，S(j,K)是和物品j最相似的K个物品的集合，wji是物品j和i的相似度，rui是用户u对物品i的兴趣。（对于隐反馈数据集，如果用户u对物品i有过行为，即可令rui=1。）该公式的含义是，和用户历史上感兴趣的物品越相似的物品，越有可能在用户的推荐列表中获得比较高的排名。

当我们看到这里的时候很可能由于自己功底不足，很难看懂公式中的i∈N(u)∩S(j,K)。

我们来看另外一种计算方式：



矩阵

其中，Pa为新用户对已有产品的向量，T为物品的共现矩阵，得到的P`a为新用户对每个产品的兴趣度。

那么就可以理解上述公式的i∈N(u)∩S(j,K)，我们在下面的例子中详细讲解。