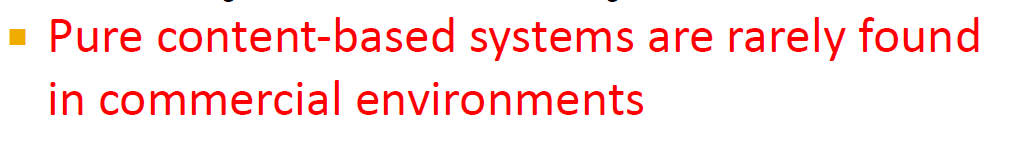
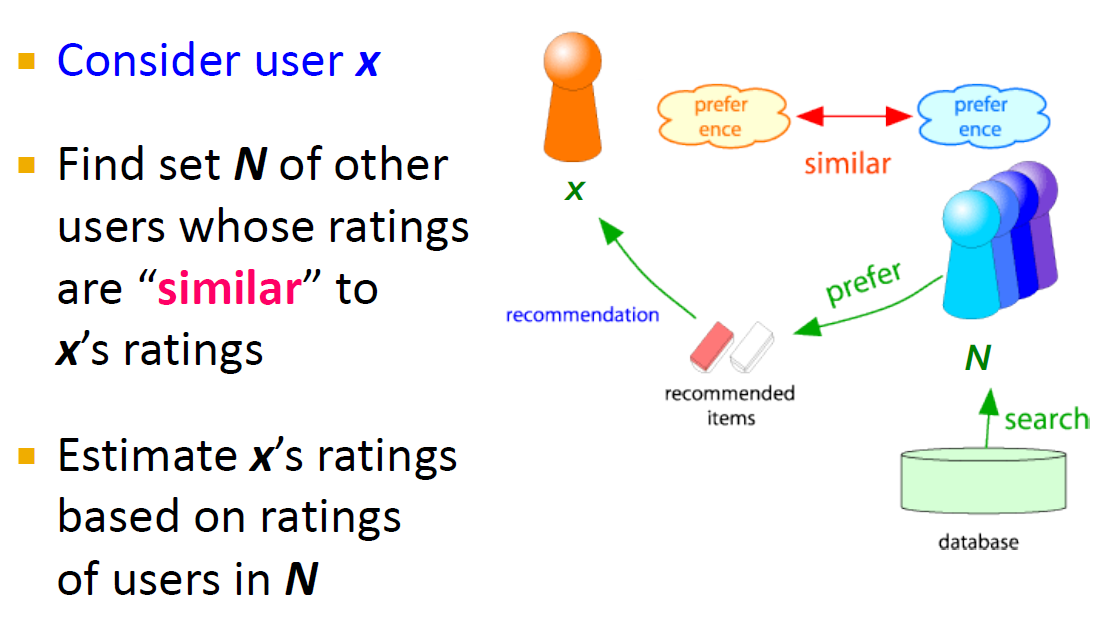
**电影推荐算法实现思路**

1. 算法选择

上课的时候，老师的PPT上提到，在实际的商业系统应用中，纯粹的Content-based的推荐方式十分少见，因此我选择实现Collaborative Filtering (CF) 的推荐方式。



1. 算法实现



协同过滤算法的关键步骤大概有两个：第一个是找到与被推荐用户口味相近的“邻居”，第二个是根据邻居对不同电影的打分为被推荐用户推荐电影。

1. 用户打分矩阵构建

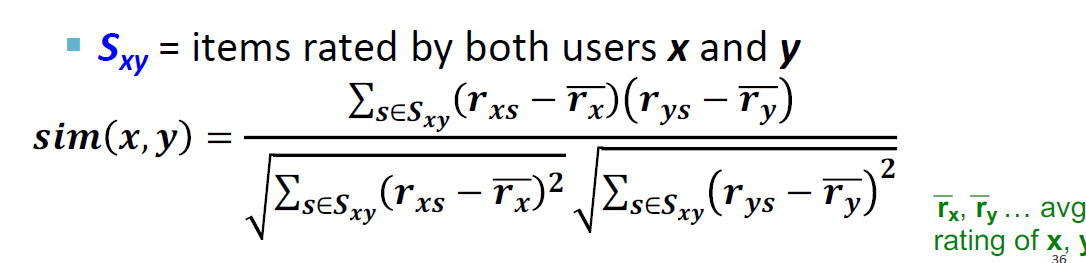
由于用户的打分矩阵是稀疏的，因此无法严格意义地使用numpy.ndarray来构建矩阵，因此我采用字典地方式，将每个用户ID作为键，对应的值是一个子字典，键是电影ID，值是对应的打分。

代码如下：

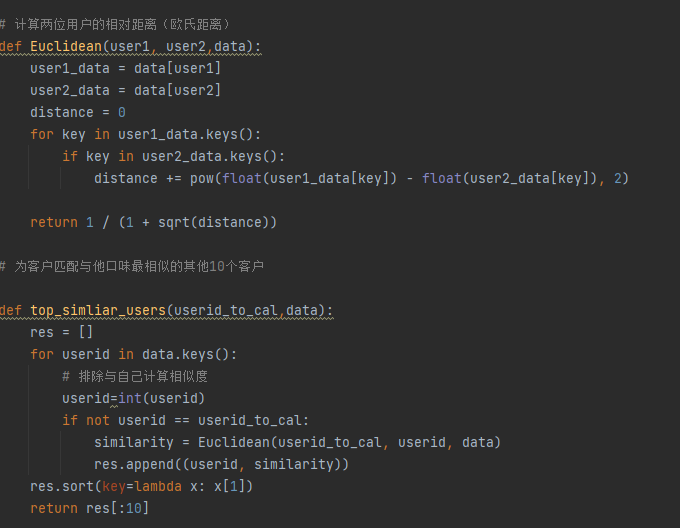


1. 相似度计算

对于不同用户之间的相似性，常常通过简单的欧式距离进行计算，在这一步中，我选择了皮尔逊相关系数来作为相似度。对于不同的用户，person系数取值在[-1,1]之间，如果person系数接近于1，则表示两个用户之间具有强相关性，两者对电影具有相同的个人喜好。当person系数接近于-1时，两个用户间完全负相关，及表示两人对电影的喜好完全对立。Person系数为0时，表示两个用户对电影的个人喜好并没有太多的联系。公式如下：

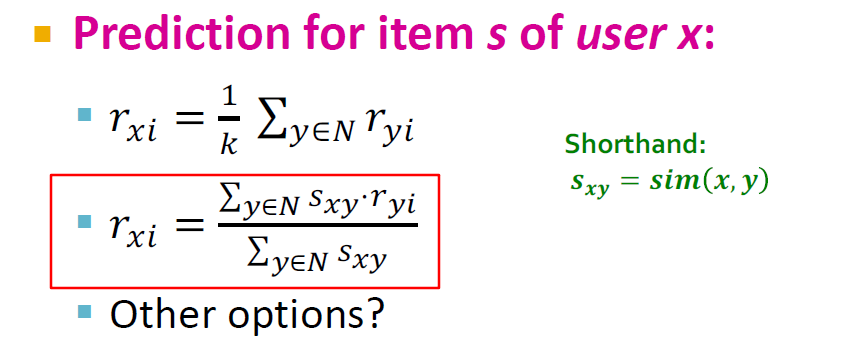


代码如下：



1. 预测打分

在获取了被推荐用户与其他剩余用户的相似度后，就可以对这个列表进行排序，得到相似度最大的前K个用户（邻居）。这里我取K=10。之后，使用下列公式，预测每个被其他邻居用户打分过的电影对于被推荐者的可能得分。



之后，把这些得分进行排序，为被推荐用户推荐得分最高的前五个电影。

代码如下：



重复上述过程，就可以完成推荐。