作业 4Readme

191250111 裴为东

作业内容：

根据movielens数据集，使用已学习的推荐算法完成一个简单的推荐系统。具体结果呈现形式为：给每个用户(userId)推荐其可能喜好的电影(movieId)

处理过程：

1. 数据分析：

   数据集中有效数据包有3个文件，分别是movies.csv、ratings.csv、tags.csv。movies.csv中含有电影的id、标题和类型标签；ratings.csv中含有用户id、电影id、对应电影的用户评分和时间；tags.csv中含有用户id、电影id、对应电影的标签和时间。本次推荐系统采用了基于用户的协同过滤算法（UserCF），有效利用的数据集为ratings.csv

2. 推荐系统算法：

   本次使用了基于用户的协同过滤算法，其基本思想是首先找到与目标用户相似的用户群，然后在这个用户群所喜欢的电影集合中找到目标用户可能最感兴趣的电影推荐给目标用户。

   第一步：计算用户之间的相似度

本次实现采用了余弦相似度方法来计算用户间的相似度

```
#计算用户相似度
userSim = np.zeros((len(ratingValue),len(ratingValue)),dtype=np.float32)
for i in range(len(ratingValue)-1):
    for j in range(i+1,len(ratingValue)):
        userSim[i,j] = Cosine(ratingValue[i],ratingValue[j])
        userSim[j,i] = userSim[i,j]
```

ratingValue是用户对电影评分的数组，ratingValue[i][j]中i为用户id，j为电影id，其值为用户评分。userSim是用户之间相似度的数组，userSim[i][j]中i、j都为用户id，其值为这两个用户的相似度。

```
#选取前10个最相似的用户
userMostSim = dict()
for i in range(len(ratingValue)):
    userMostSim[i] = sorted(enumerate(list(userSim[i])),key = lambda x:x[1],reverse=True)[:10]
```

选取前10个最相似的用户作为下一步所需要的用户群

第二步：计算用户对电影的兴趣分

$$p(u,i)=\sum_{v\in S(u,K)\cap N(i)} w_{uv} r_{vi}$$



其中，$S(u, K)$包含和用户u兴趣最接近的$K$个用户，$N(i)$是对物品$i$有过行为的用户集合，$w_{uv}$是用户u和用户v的兴趣相似度，$r_{vi}$代表用户v对物品i的兴趣，因为使用的是单一行为的隐反馈数据，所以所有的$r_{vi}$=1。

```
#计算用户对每部电影的兴趣分
userRecValue = np.zeros((len(ratingValue),len(ratingValue[0])),dtype=np.float32)
for i in range(len(ratingValue)):
    for j in range(len(ratingValue[i])):
        if ratingValue[i][j] == 0:
            v = 0
            for (user,sim) in userMostSim[i]:
                v += (ratingValue[user][j] * sim)
            userRecValue[i,j] = v
```

计算用户对没看过的电影的兴趣分，userMostSim数组中有相似用户群的id和相似度，userRecValue数组存放用户对电影的兴趣分

```
#选取前3个最感兴趣的电影
userRecDict = dict()
for i in range(len(ratingValue)):
    userRecDict[i] = sorted(enumerate(list(userRecValue[i])),key = lambda x:x[1],reverse=True)[:3]
```

将兴趣分最高的前3部电影推荐给用户

3. 输出结果：

   输出结果movie.csv位于RecSys/data/movie.csv