实验报告

课程名称： 《大数据分析与内存计算》

实验学期： 2019年至2020年第2学期

专 业： 网络工程

班 级： 网络17-1

姓 名： 吴利洪

学 号： 17034460134

指导教师： 黄志毅

### 实验7.1 **基本概念-本地向量、转换器、评估器、参数**

【实验目的】

理解本地向量、转换器、评估器、参数

【实验原理】

（1）本地向量：Mllib支持两种类型的本地向量:密集向量(dense)和稀疏向量(sparse)。密集向量只有一个浮点数组组成，而一个稀疏向量必须有索引和一个浮点向量组成。例如，(2.1,3.2,4.3)代表一个密集向量。(3，[1.1,2.3],[5.6,4.3,4.4])代表一个稀疏向量。

（2）Transformer：翻译成转换器，是一种可以将一个DataFrame转换为另一个DataFrame的算法。比如一个模型就是一个 Transformer。它可以把一个不包含预测标签的测试数据集 DataFrame 打上标签，转化成另一个包含预测标签的 DataFrame。 技术上，Transformer实现了一个方法transform()，它通过附加一个或多个列将一个DataFrame转换为另一个DataFrame

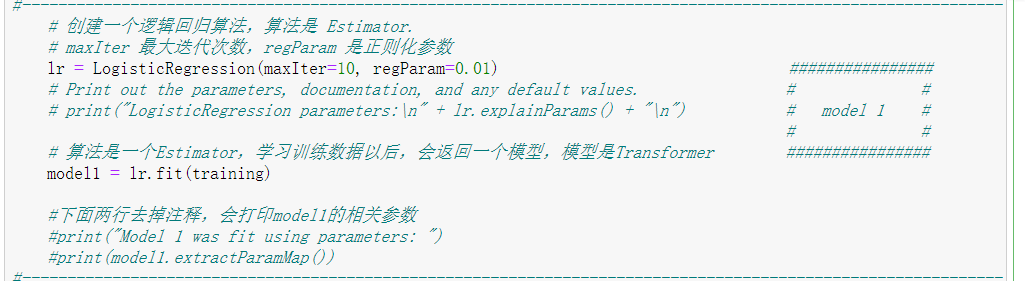
（3）Estimator：翻译成估计器或评估器，它是学习算法或在训练数据上的训练方法的概念抽象。在 Pipeline 里通常是被用来操作 DataFrame 数据并生成一Transformer。从技术上讲，Estimator实现了一个方法fit()，它接受一个DataFrame并产生一个转换器。比如，一个随机森林算法就是一个 Estimator，它可以调用fit()，通过训练特征数据而得到一个随机森林模型。

（4）Parameter：所有Transformers和Estimators现在共享一个用于指定参数的通用API

【实验步骤】

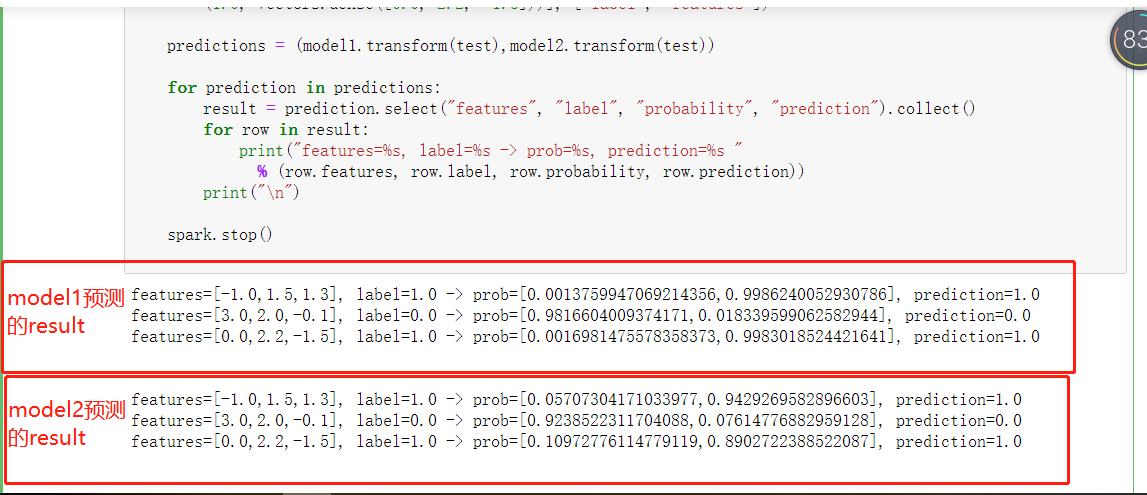
阅读并运行以下代码，请对输出label，prob，prediction的意义进行说明

建立mode1



建立mode2

把测试数据放放进模型里进行预测

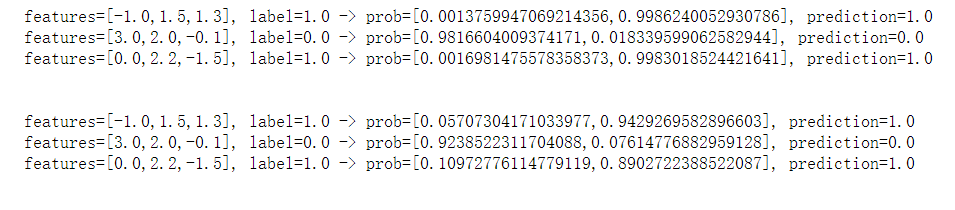
Label：标签，表示数据只有两种结果，要不为0，要不为1

features：特征向量，表示空间上的一个多维向量，即数据的分布位置

Prob：分别列出结果为0，与结果为1的概率

prediction：根据prob里对数据为0或者1的概率进行比较，作出预测结果

【实验结果】



### **实验7.2 ：逻辑回归算法**

【实验目的】

掌握Pipeline、逻辑回归的用法

【实验原理】

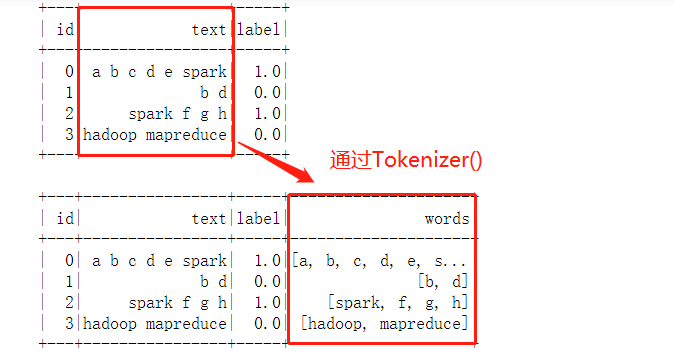
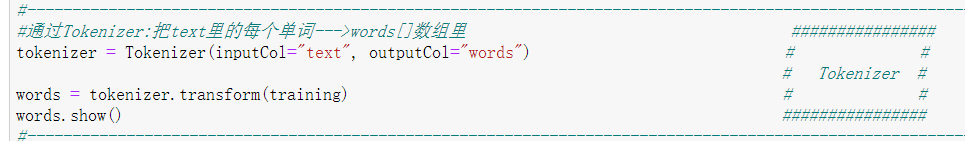
1）Pipeline：将Pipeline多个Transformers和Estimators 链在一起以指定ML工作流程。

（2）逻辑回归：在线性回归增加了一个函数g(z)，能够把连续值映射到几个离散的数据，如：0、1等。

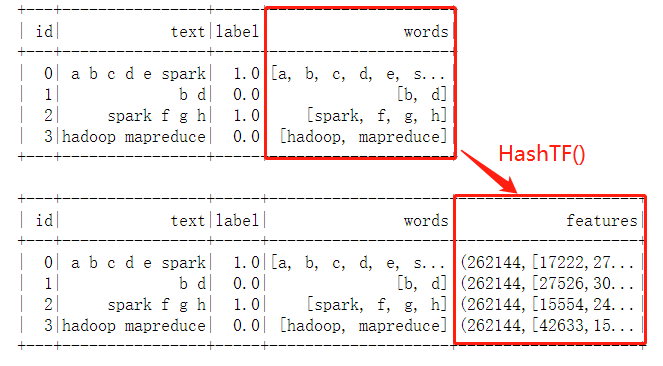
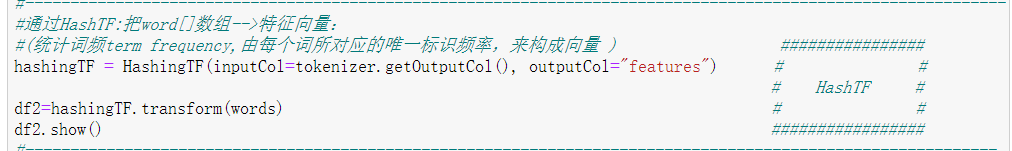
【实验步骤】

阅读并运行以下代码，请对代码输出的意义进行说明。

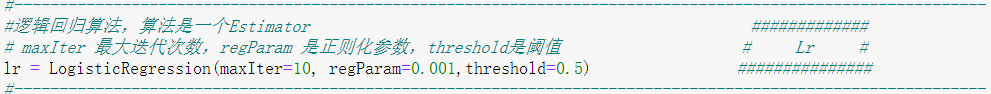
**1、Tokenizer**



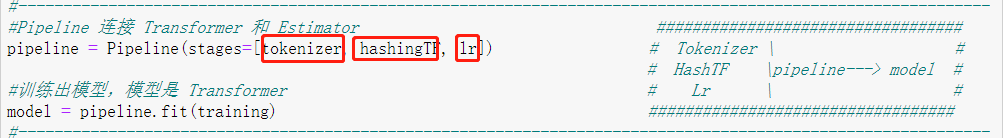
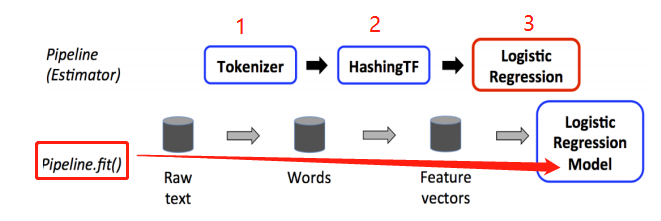
**2、HashTF**



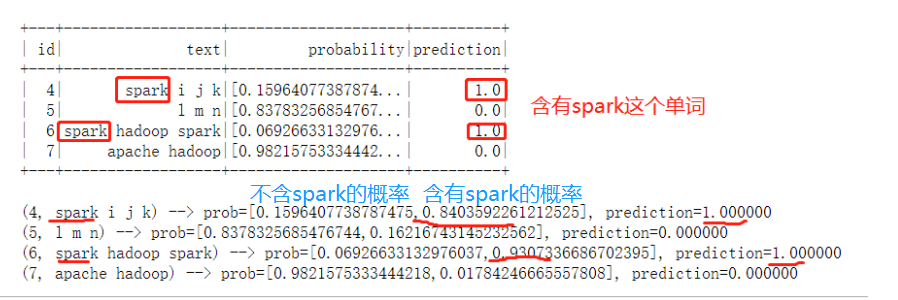
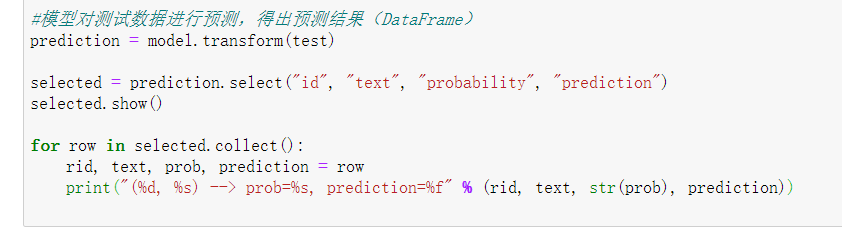
**3、Logistic Regression**

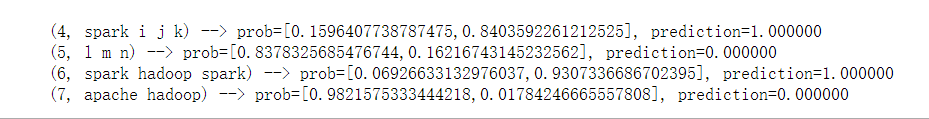


有了这3个模块后，再用pipeline把它们串起来



把测试数据放进模型里预测

【实验结果】



## **实验7.3 ：K-Means 算法**

【实验目的】

掌握K-Means的用法

【实验原理】

K-Means

选择K个点作为初始质心

repeat

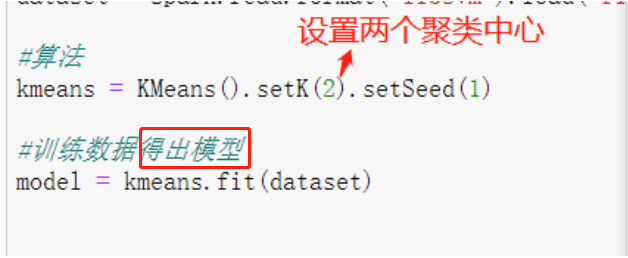
* 将每个点指派到最近的质心，形成K个簇
* 重新计算每个簇的质心

until 簇不发生变化或达到最大迭代次数。

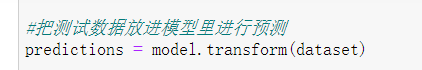
【实验步骤】

阅读并运行以下代码，请对代码输出的意义进行说明。

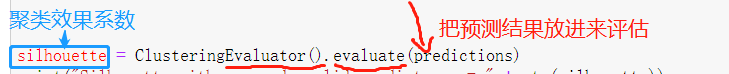
设置聚类中心，创建model



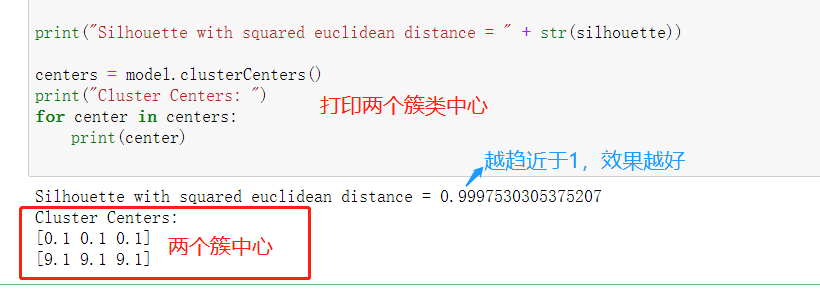
进行预测



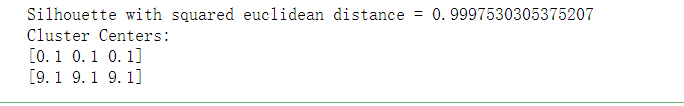
评估聚类效果系数



打印聚类效果系数、两个簇中心



【实验结果】



## **实验7.4 ：协同过滤算法**

【实验目的】

掌握基于Spark MLLib的协同过滤。

* 基于用户（User-Based）的协同过滤
* 基于物品（Item-Based）的协同过滤
* 基于ALS协同过滤

【实验原理】

（1）基于用户的CF（User CF） 基于用户的 CF 的基本思想相当简单，基于用户对物品的偏好找到相邻邻居用户，然后将邻居用户喜欢的推荐给当前用户。计算上，就是将一个用户对所有物品的偏好作为一个向量来计算用户之间的相似度，找到 K 邻居后，根据邻居的相似度权重以及他们对物品的偏好，预测当前用户没有偏好的未涉及物品，计算得到一个排序的物品列表作为推荐。图 2 给出了一个例子，对于用户 A，根据用户的历史偏好，这里只计算得到一个邻居 - 用户 C，然后将用户 C 喜欢的物品 D 推荐给用户 A。

（2）基于物品的CF（Item CF） 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。

（3）基于ALS协同过滤推荐

ALS 是交替最小二乘 （alternating least squares）的简称。 ALS算法是2008年以来，用的比较多的协同过滤算法。它已经集成到Spark的Mllib库中，使用起来比较方便。 从协同过滤的分类来说，ALS算法属于User-Item CF，也叫做混合CF。它同时考虑了User和Item两个方面。 用户和商品的关系，可以抽象为如下的三元组：<User,Item,Rating>。其中，Rating是用户对商品的评分，表征用户对该商品的喜好程度。 假设我们有一批用户数据，其中包含m个User和n个Item，则我们定义Rating矩阵，其中的元素表示第u个User对第i个Item的评分。 在实际使用中，由于n和m的数量都十分巨大，因此R矩阵的规模很容易就会突破1亿项。这时候，传统的矩阵分解方法对于这么大的数据量已经是很难处理了。 另一方面，一个用户也不可能给所有商品评分，因此，R矩阵注定是个稀疏矩阵。矩阵中所缺失的评分，又叫做missing item。

ALS算法举例说明：

(1) 下面的矩阵R表示：观众对电影的喜好，即：打分的情况。注意：实际情况下，这个矩阵可能非非常庞大，并且是一个稀疏矩阵。 矩阵R

(2) 这时，我们可以把这个大的稀疏矩阵R，拆分成两个小一点的矩阵：U和V。通过U和V来近似表示R，如下图

其中：  U矩阵代表：用户的特征，包括三个维度：性格，文化程度，兴趣爱好

 V矩阵代表：电影的特征，也包括三个维度：性格，文化程度，兴趣爱好

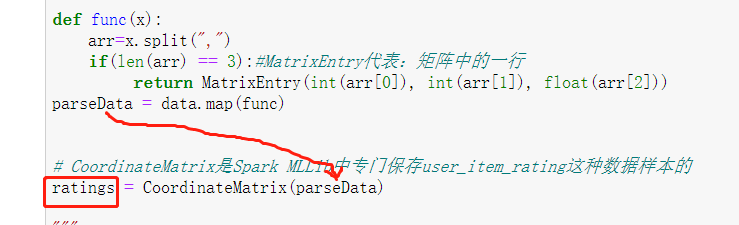
(3) 这样，U和V的乘积，近似表示R。

(4) 但是，这样的表示是存在误差的，因为对于一个U矩阵来说，我们并不可能说（性格，文化程度，兴趣爱好）这三个属性就代表着一个人对一部电影评价全部的属性，比如还有地域等因素。这个误差，我们用RMSE（均方根误差）表示。

【实验步骤】

1. 代码输出的意义

ratings矩阵

将ratings转为matrix



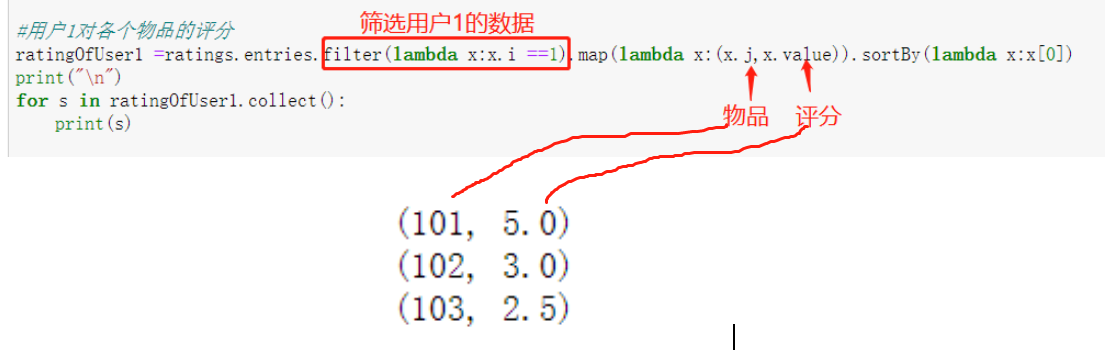
matrix大致为下图

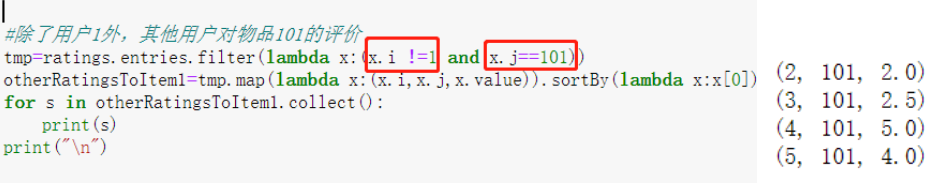


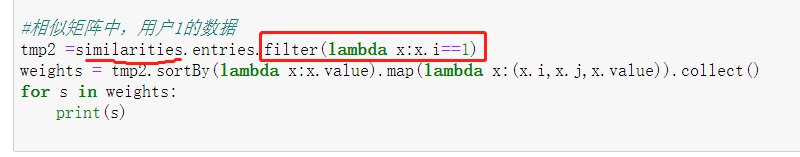
有了如上图的matrix，就可以用余弦相似度公式，

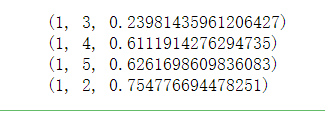
即调用columnSimilarities()方法算出用户之间的相似度



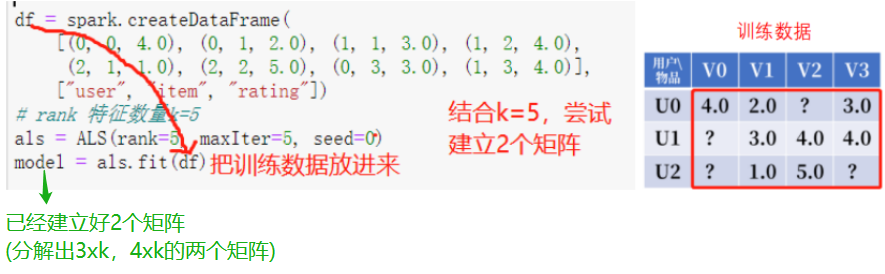
用户1对各个物品的评分用户1的所有评分的平均值

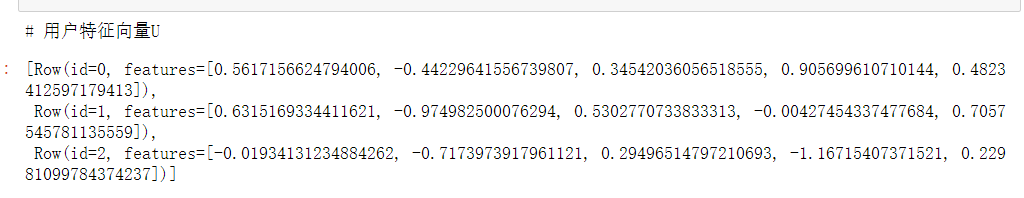
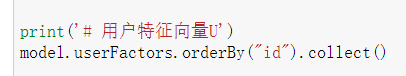
除了用户1外，其他用户对物品101的评价相似矩阵中，用户1的数据





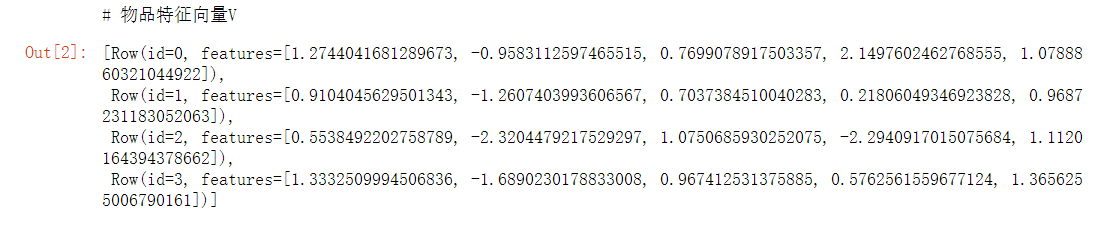
1. ASL过滤算法代码，请对代码输出的意义进行说明。

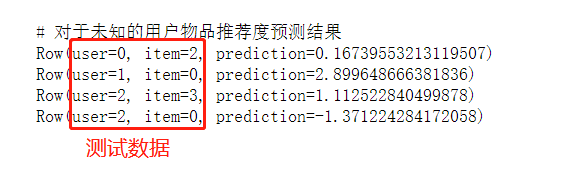
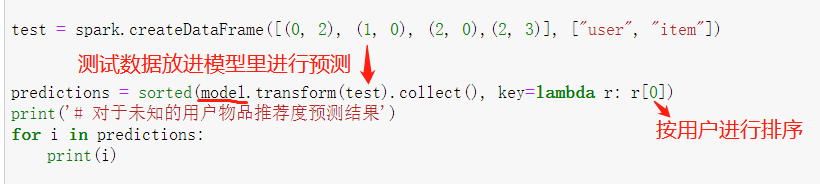
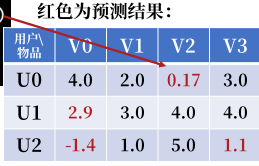
用ALS()，尝试建立两个矩阵用户特征矩阵U：3行x k列

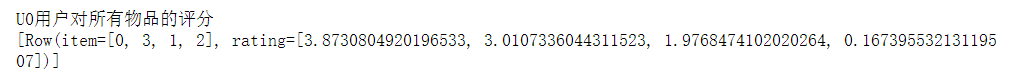
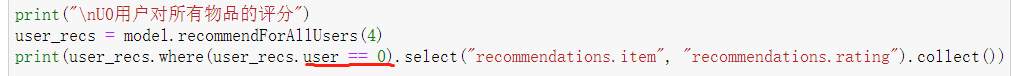


物品特征矩阵V: 4行 x k列

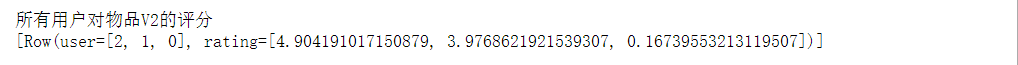
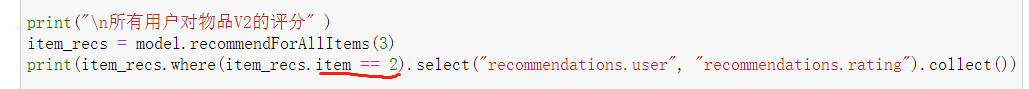
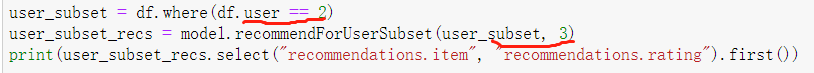


预测结果

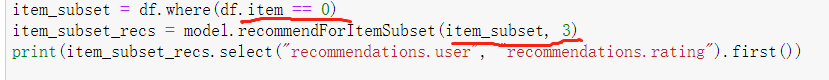
 

**U0用户**对所有物品的评分 

所有用户对**物品V2**的评分

根据mode1分解的两个矩阵，对**用户U2**子集中3个物品，进行评分预测，然后与源数据对比 

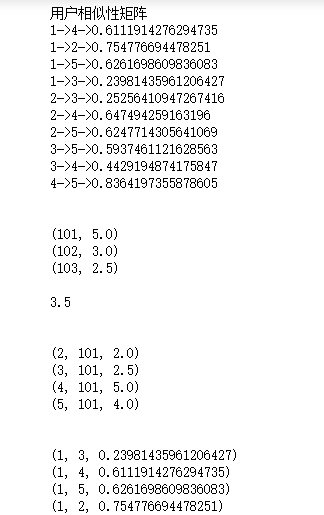
根据mode1分解的两个矩阵，对**物品V0**子集中3个用户，进行评分预测，然后与源数据对比





【实验结果】

一、用户协同过滤算法和物品协同过滤算法



1. ASL过滤算法

