**分类技术——二分网络上的链路预测**

学号：18030100101 姓名：张帅豪

1. 实验内容

基于网络结构的链路预测算法被广泛的应用于信息推荐系统中。算法不考虑用户和产品的内容特征，把它们看成抽象的节点，利用用户对产品的选择关系构建二部图。为用户评估它从未关注过的产品，预测用户潜在的消费倾向。

**本次实验有两个任务：**

1. **根据已有的电影评价数据MoviesLens训练出一个可向用户推荐电影的推荐模型**
2. **评价该推荐模型的正确性**
3. 设计与分析

实验给定的数据类型为.csv，则使用Pandas库进行数据处理。

1. **将原始数据随机分为两部分，90%为训练集，10%为测试集。**
2. **构建“用户—电影”喜爱矩阵A**

由训练集数据构建矩阵A，该矩阵是一个m\*n的矩阵，其中m为用户人数，n为影片数量。矩阵中的值表示用户i对电影j喜爱与否，若=1，则表示用户i喜爱电影j,反之则不喜爱。

每一用户对其看过的电影都有打分，我们设定分值3为阈值，大于3分的即表示喜爱（即将不喜爱与未评价的电影不区分对待，值都等于0）。

1. **根据喜爱矩阵A计算资源分配矩阵W**

反映的是电影j与电影i的兴趣关联度。表示电影j愿意分配给电影i的资源配额，即喜爱电影j的用户又可能喜爱电影i的兴趣度，该值越大则该喜爱j电影的用户越容易喜爱电影i。所以，根据两电影关联兴趣度的概念，我们构造出了下面这个计算值的公式：



这个公式中，表示产品的度（即有多少用户喜爱电影j），表示每一个用户，表示每个用户的度（即用户喜爱多少部电影）。很显然，电影j与电影i的兴趣关联度与和成反比。形式化解释如下：

·当一部电影j被很多人喜爱（即很大），那么电影j就可能是一个泛化的热门电影，用户喜爱电影j就并不意味着他有强烈的喜爱电影i的倾向。

·如果一个用户喜爱非常多的电影（即很大），那么他喜爱电影j，就并不意味着他对电影j非常专注，相应的推荐度也就不那么高。

**W矩阵（n\*n）显然就是我们要用训练集数据训练的模型。**

1. **完成实验任务一：向用户做出电影推荐**

第三步我们完成了对资源分配矩阵W这一推荐模型的训练，现在我们可以对用户做出电影推荐，具体过程与解释如下：

目标用户选择向量表示该用户选择的电影的选择矩阵，则为一个n\*1的列向量，则各行i上的值表示电影i对用户的吸引程度，将用户所有未选择（没看过）的电影按照F中对应项的得分进行排序，推荐排序靠前的电影给该用户。

1. **完成实验任务二——模型准确性预测**

我们把资源分配矩阵与喜爱矩阵A的转置做矩阵相乘再转置运算得到一个m\*n的F矩阵。表示由我们推荐模型计算出来的j电影对i用户的吸引程度。有了矩阵F我们再用测试集数据去进行下面的模型评估算法。

对于一测试集中的用户i，假设其有Li个产品是不喜爱的（将不喜爱的与未选择归于一类），如果在测试集中用户i选择的电影j，而电影j依据向量F被排在第位，则计算其相对位置：



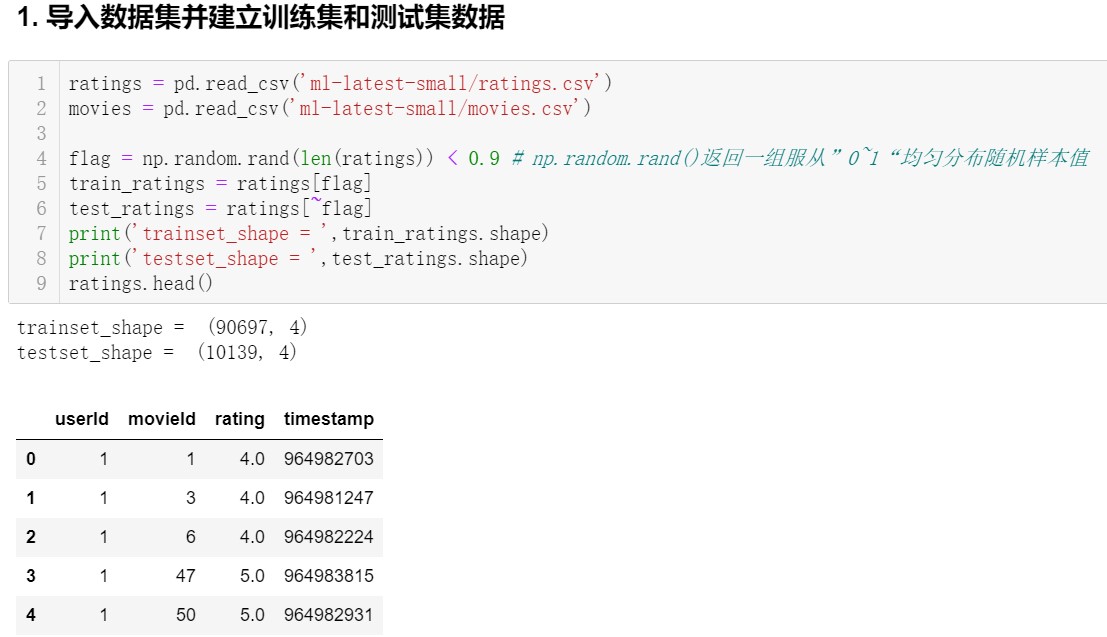
越精确的算法，给出的越小。对所有用户的求平均值来量化评价算法的精确度。

1. **可视化模型准确性——画出ROC曲线来度量预测方法的准确性**

计算相应的真阳性率（TPR）以及假阳性率（FPR），画出ROC曲线。

1. 实现详细

1. 导入数据集，并随机分为两部分，90%为训练集，10%为测试集。



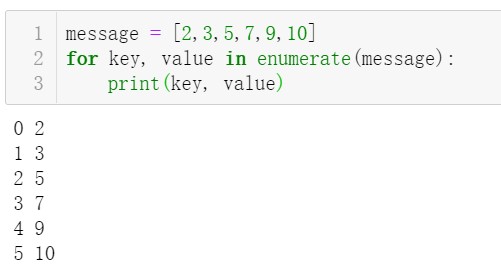
Pandas是围绕Series和DataFrame两种数据结构来处理数据的，DataFrame为二维表结构，Series相当于二维表中的一行。上图结果打印出了读取进来的ratings数据的一部分，是一个二维表。

用np.random.rand()返回一组服从0~1均匀分布的随机样本值，来实现将数据随机分为训练集和测试集两部分。

**2. 构建“用户—电影”喜爱矩阵A**

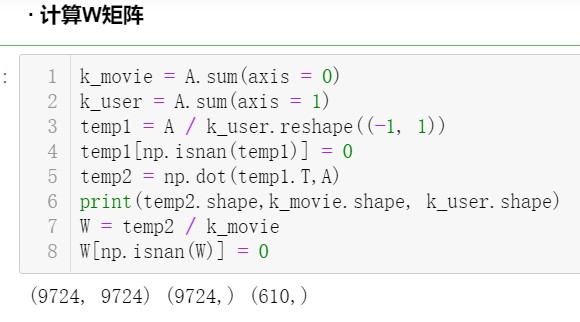


·首先离散化用户数据和电影数据：用enumerate()建立用户uid与其在矩阵A中的索引值id的字典关系{uid:id}，由于unique()函数返回的是由小到大的uid排列顺序，则最终将排好的uid与从0开始到用户数m的索引值id一一对应。对电影movieId也做同样的操作，进行离散化。enumerate()的功能示例如下：

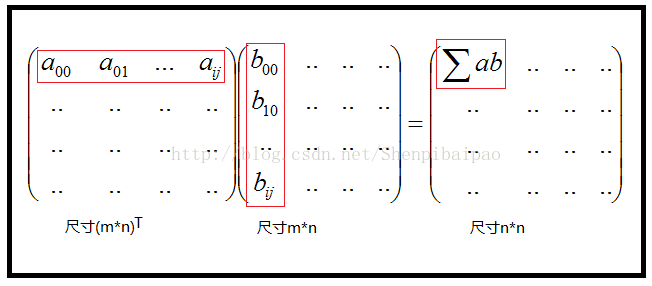


·由训练集数据建立喜爱矩阵A，由测试集数据建立目标用户电影喜爱矩阵A\_test。Pandas的DataFrame支持像train\_ratings[train\_ratings.rating>3]的操作来直接选择评分高于3的评价元组。

**3. 根据喜爱矩阵A计算资源分配矩阵W**



用矩阵乘法来实现公式，这种方法称为向量化并行计算。观察下面的两个矩阵的乘法：



前两个矩阵红框中的计算结果会成为结果矩阵的一个元素值，则的计算可以直接用矩阵相乘，最终得到一个n\*n的资源分配矩阵W，从而训练出分类模型。

代码中由于可能会出现除零操作而产生NAN，因此需要对这些数据做出处理。 **4. 完成实验任务一——做出推荐**

做出推荐没有在在程序中展现，因为我们训练出了推荐模型W，我们就可以很容易得对一个用户做出电影推荐。由模型做出推荐的过程在“第二部分——设计与分析 第4点”中已做过说明。至于推荐的效果我们在下面进行模型准确性的评价。

**5. 完成实验任务二——模型准确性预测**



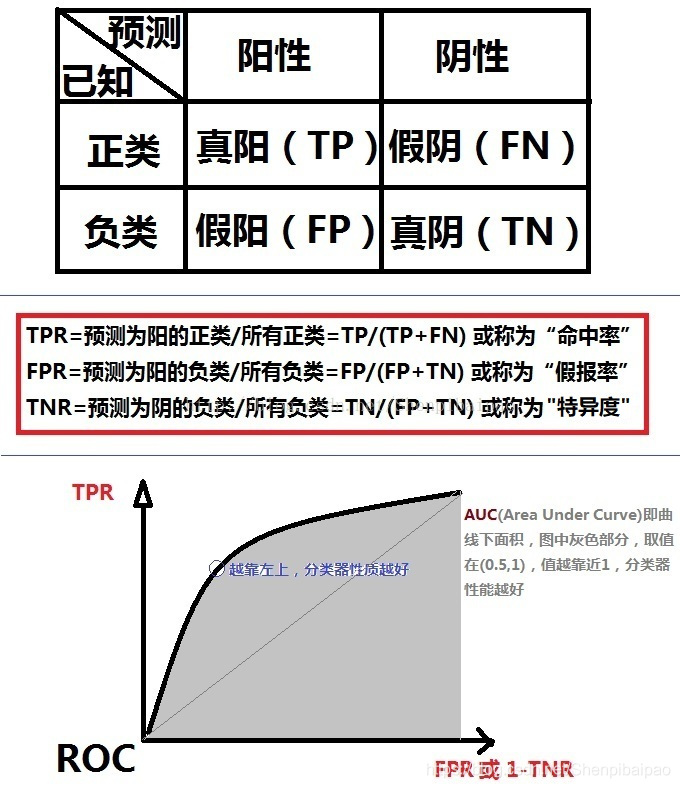
资源分配矩阵与喜爱矩阵A的转置做矩阵相乘再转置运算得到一个m\*n的F矩阵。F\_sort矩阵是一个(m\*n）的矩阵，值表示对于用户i，电影j在所有电影中的吸引度排名名次。np.argsort()函数返回的是数组值从小到大的索引值，则通过F\_sort\_index数组我们可以构造出数组F\_sort，即该推荐算法的评价模型。接下来由公式计算平均推荐相对位置r来评价算法精确度。先对每个人i计算每部喜爱影片j对应的相对位置，再计算该人的平均相对位置，最后计算所有用户的平均相对位置。

最终计算得到的平均推荐相对位置r=0.00010763274233963475

1. **可视化模型准确性——画出ROC曲线来度量预测方法的准确性**



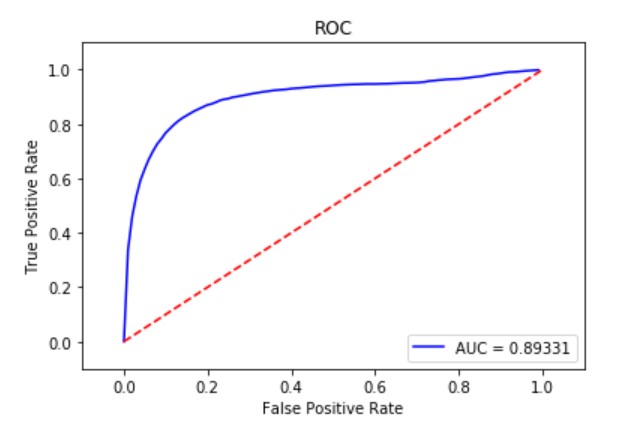
ROC曲线是用来验证一个分类器（二分）模型的性能的。原理是，给出一个模型，输入已知正负类的一组数据（测试集数据），通过对比模型对该组数据进行的预测，衡量这个模型的性能。



·如何画ROC曲线？

1. 给定一个初始阈值（从0开始）
2. 根据分类结果写出混淆矩阵，然后算出“命中率”TPR和“假报率”FPR，由TPR和FPR可以确定一个点，在图中画出。
3. 给一个新的阈值（如每次增加0.01），并重复步骤2、3，直到阈值取到1
4. 若干个点最后组成了ROC曲线。

对于本实验很显然测试集数据为我们已知用户喜爱电影与否的数据，F\_sort是我们得到的用于评价该算法的模型，通过以上的分析，测试数据属于正类即用户喜爱电影的总数为T=np.sum(A\_test)，反之不喜爱的电影总数为F=np.sum(A\_test==False)。由于F\_sort衡量的是每部电影的排名，由n\*阈值和排名的关系我们可以确定i用户是否喜爱j电影（n为电影数量），那么可以得到矩阵F\_out，则真阳数TP=np.sum(A\_test&F\_out)，假阳数FP=np.sum(~A\_test&F\_out)。这样对于每一个阈值，我们可以计算出TPR和FPR确定一个点，最终绘制出ROC曲线。

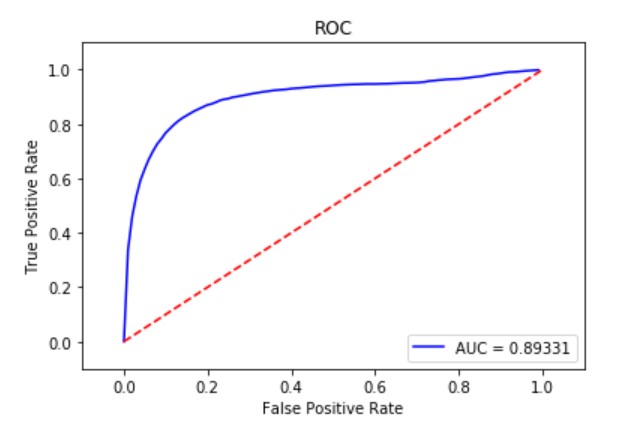


·如何计算AUC值？

AUC值即为ROC曲线下方的面积，这里取每个阈值下的正阳率TPR值之和再取平均值作为AUC值来量化该推荐模型的性能，最终计算出AUC在0.893附近，所以训练得到的推荐模型它具有较好的性能。

1. 实验结果

实验代码和注释最先在Jupyter上完成，可以查看每一步的结果。最终的.py代码在PyCharm编辑器中完成，最终完成了两个实验任务，得到了一个基于MovieLens数据集的电影推荐模型（即资源分配矩阵W）并评价了该模型的准确性，画出了ROC曲线并计算出了模型的AUC值，可以看出该模型的推荐准确性还不错。



1. 心得与反思
2. 完成实验之前，我学习了Python Pandas数据分析包。认识了Pandas是基于Series和DataFrame两种数据结构来处理数据的、学会了如何选择数据、赋值、读取与写入文件等操作。
3. 掌握了基于网络结构的链路预测算法的算法思想，该算法不考虑用户和产品的内容特征，把它们看成抽象的节点，利用用户对产品的选择关系构建二部图。之后在此二部图的基础上构造出一个数学模型来解决问题。对于本题，我们要得到的模型是各部电影之间的兴趣关联度关系以便根据目标用户已选电影推荐他未选的潜在的电影，于是我们构造出了公式，这就是我们要训练的模型。
4. 熟悉了评价模型性能的算法思路。训练集用来训练模型，测试集用来评价模型的准确性。根据资源分配矩阵与喜爱矩阵A的转置做矩阵相乘再转置运算得到的m\*n的F矩阵元素值表示由我们分配模型计算出来的j电影对i用户的吸引程度，构造出评价该模型的相对距离函数——，并计算出r的平均值来评价推荐模型的准确性。
5. 学会了根据问题分析并画出ROC曲线。理解了在特定的问题背景下ROC曲线的作图方法与结果分析。
6. 反思：这次实验中的推荐模型和算法评价的数学函数在题目中都已给出，我只是动手实现了模型的构建和评价过程，但算法核心之处在于如何将实际问题建模成可训练的数学函数。自己应该努力学习数学模型和数学函数去逐渐掌握数学建模能力。