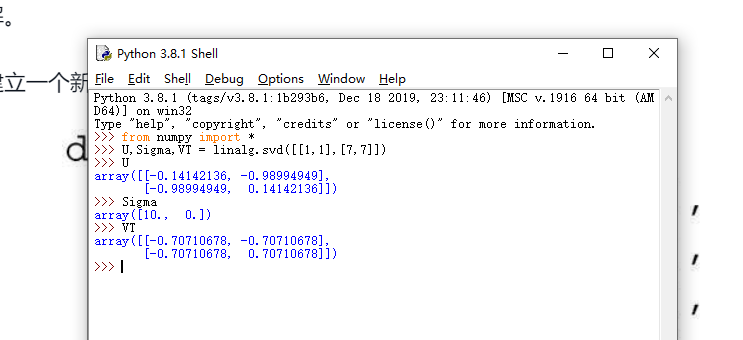
**实验一：基于协同过滤的推荐引擎实验报告**

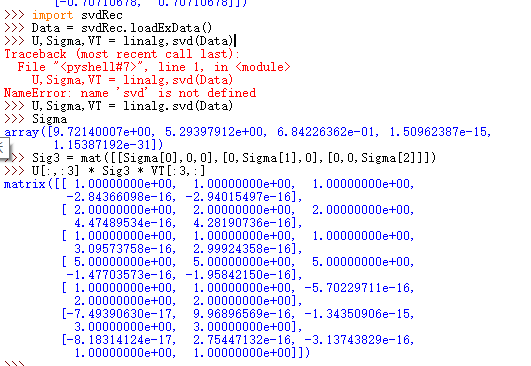
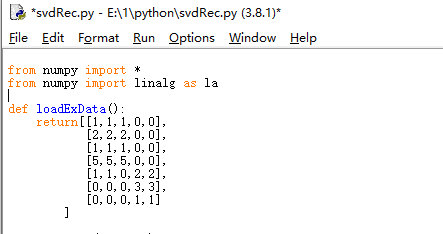
实验人员：杜先瑞

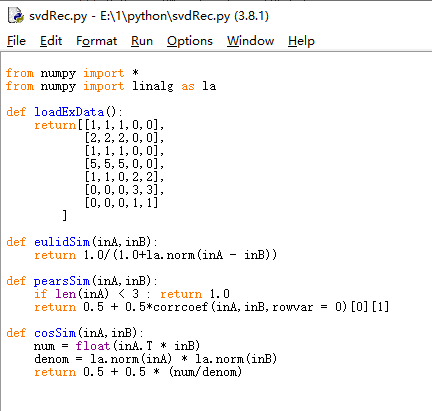
1. **实验原理**

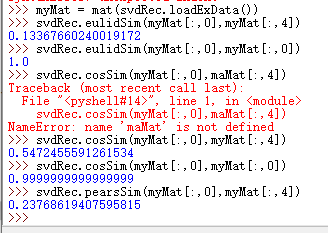
在处理数据的过程中，我们通常需要通过数据不同的特征将数据划分为不同的类别，通过对数据进行处理，我们可以发现数据背后的信息。使用SVD可以做到简化数据，去除噪声的作用。利用SVD实现，我们能够用小得多的数据集来表示原始数据集。SVD的一个应用方向就在推荐引擎。给出一个数据矩阵，使用SVD可以将矩阵分解为m行m列、m行n列和n行n列的U、Σ和VT三个矩阵，其中Σ矩阵是一个只有对角元素的矩阵，Σ中的对角元素为矩阵的奇异值，奇异值是矩阵的重要特征，奇异值是特征值的平方根，其余的元素都是噪声。使用python的numpy线性代数库可以实现SVD。

在对用户进行推荐时，一般有两种方法。一种是根据用户相似度推荐，一种是根据商品相似度推荐。计算相似度可以使用欧几里得距离，皮尔逊相关系数以及余弦相似度三种方法。将这三个系数归一化到0-1之间，数值越大相似度越高，从而可以判断出应该推荐哪种商品。

1. **实验过程**
2. 导入相关库，直接使用库函数中的SVD函数分解矩阵。
3. 经过SVD处理后矩阵变成三个，分别查看三个矩阵U,Sigma,V。可以发现，返回的第二个矩阵只有对角元素。
4. 重构原始矩阵。将矩阵U的前m行与Sigma相乘，再与V矩阵前n行相乘，就可以得到原始矩阵。



1. 计算相似度。将相似度计算方法写成函数。

分别使用三种相似度计算方法计算相似度

1. **实验遇到的问题及解决方法**

问题一：矩阵与书上一样，但是使用欧式距离计算的相似度与书上结果不一样。

解决方式：手工计算了相似度，发现与自己的结果相同，与书上不同。

问题二：在得到推荐结果的过程中用到的是相似度，并没有用到前面的SVD，这两者之间有何关系？

解决方式：先保留疑问，也许在后面的学习中可以找到答案

问题三：书上的计算结果都是整数，但是我的结果都是保留很多位的小数。

可能的原因：python版本问题

1. **总结**

通过这次实验，我明白了推荐算法的一种最简单的实现方式，可以通过计算相似度来确定是否推荐。同时学会了使用SVD分解矩阵，简化运算的方法。

**实验二：餐馆菜肴推荐引擎**

1. **实验原理**

菜肴推荐引擎的目的是通过用户已经评价过的菜品和菜品与其他菜的相似度来预测其他用户可能喜欢的菜品。为了实现这一目标，首先我们需要找出所有用户没有评价过的菜肴，然后预测用户可能的评分，将所有评分放到一个数组里面，取出排名靠前的几个推荐给用户。这一方法的重点在于预测用户对菜品的评分。通过将用户已经评价过的菜与其他为评价的菜相比较，计算出两者相似度。计算相似度需要比较所有用户对这一菜品的评价，可以使用上一实验中的欧几里得距离，皮尔逊相关系数以及余弦相似度三种方法来进行计算。得出相似度后将相似度与用户评分相乘然后累加求和，最后除以相似度累加求和得到预测用户评分。还可以使用SVD对矩阵简化，提高推荐效果。找出几个可以达到总能量90%的奇异值，然后构建一个对应的对角矩阵，降低矩阵维度。最后还是使用前面的相似度推荐方法找出评分最高的几个菜品。

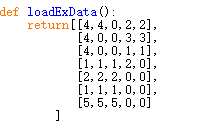
1. **实验步骤**

1、根据估计评分原理编写利用物品相似度评分的函数



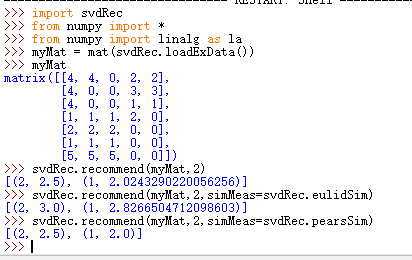
函数standEst的作用是估计物品评分，函数recommend的作用是找到评分靠前的物品推荐给用户。

1. 构造一个用户与菜品矩阵

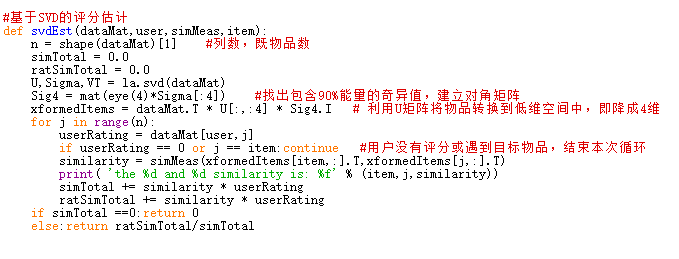


其中每一行代表用户，每一列代表一个菜品，数字代表用户评分。

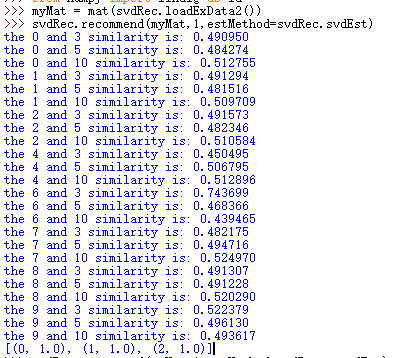
1. 加载矩阵，观察推荐结果



默认推荐方式为采用余弦相似度计算，为编号为2的用户推荐。输出结果为预测评分最前的两个菜品。随后还可以尝试欧几里得距离，皮尔逊相关系数来进行计算。

4、使用SVD提高推荐效果。先编写使用SVD推荐的评分函数。

随后与前面一样进行测试



测试结果如图

1. **实验遇到的问题及解决方法**

问题一：numpy库中nonzero函数的使用方法

解答：nonzero函数是numpy中用于得到数组array中非零元素的位置（数组索引）的函数。它的返回值是一个长度为a.ndim(数组a的轴数)的元组，元组的每个元素都是一个整数数组，其值为非零元素的下标在对应轴上的值。

问题二：numpy中eye函数的作用

解答：numpy.eye（N，M，k，dtype，order）作用是返回一个对角线diagonal上全是1，而其他位置全为0的一个二维数组。N是数组行数，M是列数，K是对角线偏移的距离。

问题三：如何将物品相似度用于估计用户评分？

解答：物品的相似度是计算的列向量的相似度，也就是结合了所有用户评分结果来得出的两个物品的相似度。得到相似度后，相似度越高，说明这两件物品重合度越高，越相似。如果一个物品与用户评分高的物品相似度高，说明这个物品被用户喜爱的概率就大，反之亦然。因此，评估用户对某一物品喜爱程度时，与用户已评价物品的相似度就是一个重要参考。

1. **总结**

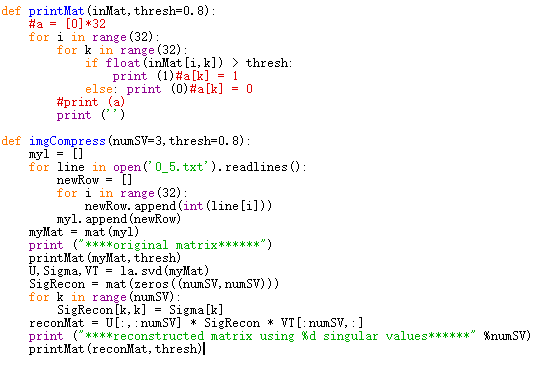
这次实验使用到了SVD来优化推荐引擎，这也解决了前面一个问题，即SVD在推荐系统中的用处。使用SVD可以降低矩阵维度，简化运算。在这次实验中，我学会了使用一些之前没有学习过的numpy函数。同时，在使用这些函数的过程中，让我对矩阵以及对矩阵的操作有了更深刻的认识，特别是多维矩阵很容易让人感到混乱。另外，也学到了用函数处理矩阵的技巧，学会如何从矩阵中拿到自己想要的信息。最后，了解了基于协同过滤的推荐引擎的原理及实现方法。

**实验三：基于SVD的图像压缩**

1. **实验原理**

前面两个实验已经验证了使用SVD可以将矩阵分解压缩，使用很少的信息就可以表示一个很大的矩阵，还可以将分解压缩的矩阵还原。利用这个原理，我们可以将一个32\*32像素的矩阵压缩，只用很少的像素就能表示这个图像。

1. **实验步骤**
2. 编写使用SVD压缩图像的函数。



函数printMat输出矩阵，参数thresh为阈值，超过这个阈值为1，没有超过为0。函数imgCompress将读取文件，然后压缩，输出，再还原，输出。

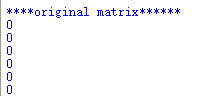
2、运行程序，观察结果

1. **实验中的问题及解决方法**

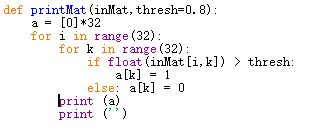
问题一：函数readlines的使用方法及作用

解答：readlines方法作为列表返回文件中的所有行，其中每一行都是列表对象中的一项。

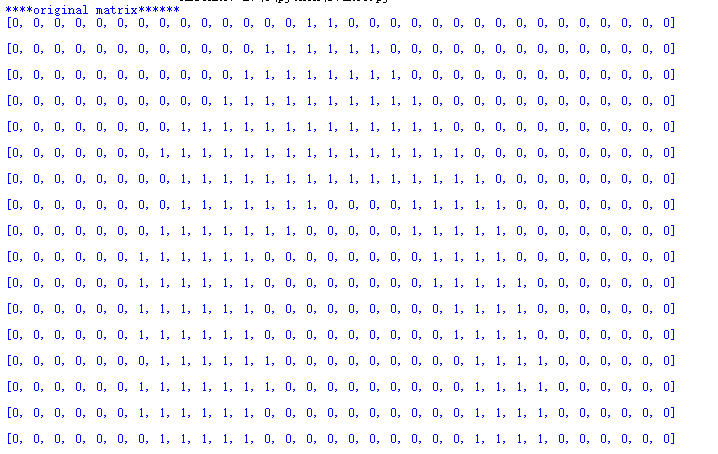
问题二：函数输出的矩阵为一个数字一行的输出，而不是书上每行32个数输出。



解答：造成这种问题的原因是前面按行读取内容后，后面又将每一行内容一个个的防到了myl数组中，因此输出时也是一个一个输出，而不是一行一行输出。可以将程序修改一下



如图，每32个元素放到一个数组中，然后每次将这个数组输出。



输出结果。

1. **总结**

这次实验使用了SVD对图像像素压缩，展现了SVD在实际应用中的一个方向。通过这次实验，更加熟悉了SVD的用法，也增加了遇到问题时解决问题的能力。