山东大学网络空间安全学院

Python高级编程 课程实验报告

|  |  |  |  |
| --- | --- | --- | --- |
| 学号：202100450069 | 姓名：孟庆丹 | | 班级：网安21.2 |
| 实验题目： | | | |
| 实验学时：2 | | 实验日期：2022.11.22 | |
| 实验目的：  学习Numpy基本函数的使用：创建不同维度的数组、利用索引访问数组元素、数组切片、整数索引、布尔索引、数组的基本运算函数、数学函数、创建矩阵、矩阵运算、排序函数、条件筛选函数、线性代数相关函数。通过SVD算法实现推荐系统。 | | | |
| 硬件环境：  处理器 AMD Ryzen 5 5600H with Radeon Graphics 3.30 GHz  机带 RAM 16.0 GB (13.9 GB 可用) | | | |
| 软件环境：  Windows 11 家庭中文版 21H2  Python 3.10.7 | | | |
| 实验步骤与内容：   1. Numpy基本函数的使用   学习Numpy基本函数的使用：创建不同维度的数组、利用索引访问数组元素、数组切片、整数索引、布尔索引、数组的基本运算函数、数学函数、创建矩阵、矩阵运算、排序函数、条件筛选函数、线性代数相关函数。   1. 根据顾客对菜品的打分，实现菜品推荐系统   我们的菜品推荐系统通过SVD算法进行  （1）首先我们进行对稀疏矩阵的处理，我们要求参与相似度计算的分数向量的每个元素都必须非零，且来自于相同的几个顾客。  我们需要依据数据矩阵的实际打分情况，按行对原始打分矩阵进行压缩降维，将其处理成一个低维的矩阵，我们通过行压缩的方式，对矩阵进行行压缩，在行压缩的基础上，推荐算法中通常还需要再乘以奇异值方阵，赋予其对应的权重值，最终获取降维后规模为6×11行压缩矩阵。   1. 计算菜品相似度。我们对两个菜品的分数向量进行分析比较，计算它们的余弦相似度。 2. 根据相似的进行评分估计，我们通过相似度加权的方式，通过该顾客已经评出的菜品分数，推测为评分的菜品分数。   代码实现：  import numpy as np  # 初始化矩阵  # 18 \* 11  table = '''5 2 1 4 0 0 2 4 0 0 0  0 0 0 0 0 0 0 0 0 3 0  1 0 5 2 0 0 3 0 3 0 1  0 5 0 0 4 0 1 0 0 0 0  0 0 0 0 0 4 0 0 0 4 0  0 0 1 0 0 0 1 0 0 5 0  5 0 2 4 2 1 0 3 0 1 0  0 4 0 0 5 4 0 0 0 0 5  0 0 0 0 0 0 4 0 4 5 0  0 0 0 4 0 0 1 5 0 0 0  0 0 0 0 4 5 0 0 0 0 3  4 2 1 4 0 0 2 4 0 0 0  0 1 4 1 2 1 5 0 5 0 0  0 0 0 0 0 4 0 0 0 4 0  2 5 0 0 4 0 0 0 0 0 0  5 0 0 0 0 0 0 4 2 0 0  0 2 4 0 4 3 4 0 0 0 0  0 3 5 1 0 0 4 1 0 0 0'''  scoreData = table.split('\n')  for i in range(len(scoreData)):      scoreData[i] = scoreData[i].split()  for i in range(len(scoreData)):      scoreData[i] = list(map(int, scoreData[i]))  scoreMat = np.matrix(scoreData)  # 数据降维  def dataDimensionReduction(scoreData):      U, sigma, VT = np.linalg.svd(scoreData)      sigmaK = np.mat(np.eye(6)\*sigma[:6])      scoreDataRC = sigmaK \* U.T[:6, :] \* scoreData      return scoreDataRC  # 求余弦相似度  def cosSim(vector1, vector2):      dot = float(np.dot(vector1.T, vector2))      norm = np.linalg.norm(vector1) \* np.linalg.norm(vector2)      Sim = dot / norm      return 0.5 + 0.5\*Sim  # 估计userIndex对itemIndex的评分  def estScore(scoreData, scoreDataRC, userIndex, itemIndex):      n = np.shape(scoreData)[1]      simSum = 0      simSumScore = 0      for i in range(n):          userScore = scoreData[userIndex, i]          if userScore == 0 or i == itemIndex:              continue          sim = cosSim(scoreDataRC[:, i], scoreDataRC[:, itemIndex])          simSum = float(simSum + sim)          simSumScore = simSumScore + userScore \* sim      if simSum == 0:          return 0      return simSumScore / simSum  for i in range(18):      a = list()      for j in range(11):          if scoreMat[i, j] == 0:              a.append([estScore(                  scoreMat, dataDimensionReduction(scoreMat), i, j), j])      a = np.array(a)      print('给', i, '推荐第', a[a[:, 0].argmax(), 1], '个菜，评分是', max(a[:, 0]))  我们对所有没有被评分的菜品进行评分，推荐结果如下： | | | |
| 结论分析与体会：  通过本次实验，学习了Numpy基本函数的使用：创建不同维度的数组、利用索引访问数组元素、数组切片、整数索引、布尔索引、数组的基本运算函数、数学函数、创建矩阵、矩阵运算、排序函数、条件筛选函数、线性代数相关函数。通过SVD算法实现了推荐系统。 | | | |