山东大学网络空间安全学院

Python高级程序设计课程实验报告

|  |  |  |  |
| --- | --- | --- | --- |
| 学号：202100460065 | 姓名：李昕 | | 班级：21级密码2班 |
| 实验题目：实验九. Numpy模块使用 | | | |
| 实验学时：2 | | 实验日期：2022/11/22 | |
| 实验目的：熟悉模块的用法 | | | |
| 硬件环境：  AMD Ryzen 7 5800H with Radeon Graphics 3.20 GHz  机带 RAM 16.0 GB (13.9 GB 可用) | | | |
| 软件环境：  系统：Windows11  编译器：IDLE | | | |
| 实验步骤与内容：  本次实验包括两个内容：  1、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。  2、根据顾客对菜品的打分，实现菜品推荐系统  【Numpy基本函数的使用】  （1）通过学习，我了解到创建不同维度的数组使用可以使用许多语句，如创建N维数组可以使用：   1. np.array（N）*#N为N维矩阵*   或者使用reshape()函数[reshape()函数可以实现给数组一个新的形状而不改变其数据，通过reshape()生成的新数组和原始数组共用一个内存，也就是说，假如更改一个数组的元素，另一个数组也将发生改变]，其常与arange()函数一起使用来构造多维数组：   1. x = np.arange(1,10).reshape(-1,4)   （2）访问数组与列表的访问方式类似, 索引数组可以通过下标的方式访问:   1. >>> a = np.array(np.arange(1,5)) 2. >>> a   或数组切片，切片得到的数组仍然指向原数组：   1. b=np.arange(0,10,2) 2. c = b[0:4]   或者使用整数索引和布尔索引这类高级索引，整数索引即每一个整数数组代表那一个维度(轴)上的索引值：   1. >>> a = np.array(np.arange(0,16).reshape(4,4)) 2. >>> a 3. >>> a[[0,1,2,3],[0,1,2,3]] 4. array([ 0, 5, 10, 15]) 5. *//a[x,y], x=[0,1,2,3] y=[0,1,2,3]* 6. *//表示取a的第0行0列* 7. *//第1行1列* 8. *//第2行2列* 9. *//第3行3列的元素*   当索引的方式是布尔数组的时候，会触发布尔索引：   1. >>> a = np.array(np.arange(0,16).reshape(4,4)) 2. >>>b = a[ :, [True,False,False,True]] 3. array([[ 0, 3], 4. [ 4, 7], 5. [ 8, 11], 6. [12, 15]])   （3）我同时学习了数组/矩阵运算，各类数学函数和功能函数：    (4) Numpy可用于计算矩阵相乘、分解矩阵、求解线性方程等线性代数问题:   1. >>> import numpy as np 2. >>> a = 3. np.mat(np.arange(1,7).reshape(2,3)) 4. >>> a 5. matrix([[1, 2, 3], 6. [4, 5, 6]]) 7. >>>b=np.linalg.pinv(a) *#广义逆矩阵* 8. >>>b 9. >>>a = np.array([[1., 2.], [3., 4.]]) 10. >>>b=np.linalg.inv(a) 11. >>>b 12. >>>A = np.array([[1,2],[2,-1]]) 13. >>>b = np.array([2,4]) 14. >>> x = np.linalg.solve(A,b) *#求解方程组*   具体完整代码见附加代码（1）  【菜品打分系统】  利用余弦相似度的方法来衡量两个菜谱间的相似程度。是否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，然后再对其进行余弦相似度的处理呢，这样就能避免上面描述的稀疏矩阵的一些不足。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，最终获取降维后的压缩矩阵。最后计算出该 userIndex 用户所有未打分菜品的预估分数值，将预估分数最高（或前 n高）的菜品推荐给他。  代码实现：   1. import numpy as np 2. def cosSim(vec\_1, vec\_2): ***#衡量菜谱之间的相似性，采用余弦相似度的方法*** 3. dotProd = float(np.dot(vec\_1.T, vec\_2)) 4. normProd = np.linalg.norm(vec\_1)\*np.linalg.norm(vec\_2) 5. return 0.5+0.5\*(dotProd/normProd) 6. def estScore(scoreData,scoreDataRC,userIndex,itemIndex): 7. simSum = 0 8. simSumScore = 0 9. for i in range(n): 10. userScore = scoreData[userIndex,i] 11. if userScore == 0 or i == itemIndex: 12. continue 13. sim = cosSim(scoreDataRC[:, i], scoreDataRC[:, itemIndex]) 14. simSum = float(simSum + sim) 15. simSumScore = simSumScore + userScore \* sim 16. if simSum == 0: 17. return 0 18. return simSumScore / simSum 19. scoreData= np.array([[5, 2, 1, 4, 0, 0, 2, 4, 0, 0, 0], 20. [0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0], 21. [1, 0, 5, 2, 0, 0, 3, 0, 3, 0, 1], 22. [0, 5, 0, 0, 4, 0, 1, 0, 0, 0, 0], 23. [0, 0, 0, 0, 0, 4, 0, 0, 0, 4, 0], 24. [0, 0, 1, 0, 0, 0, 1, 0, 0, 5, 0], 25. [5, 0, 2, 4, 2, 1, 0, 3, 0, 1, 0], 26. [0, 4, 0, 0, 5, 4, 0, 0, 0, 0, 5], 27. [0, 0, 0, 0, 0, 0, 4, 0, 4, 5, 0], 28. [0, 0, 0, 4, 0, 0, 1, 5, 0, 0, 0], 29. [0, 0, 0, 0, 4, 5, 0, 0, 0, 0, 3], 30. [4, 2, 1, 4, 0, 0, 2, 4, 0, 0, 0], 31. [0, 1, 4, 1, 2, 1, 5, 0, 5, 0, 0], 32. [0, 0, 0, 0, 0, 4, 0, 0, 0, 4, 0], 33. [2, 5, 0, 0, 4, 0, 0, 0, 0, 0, 0], 34. [5, 0, 0, 0, 0, 0, 0, 4, 2, 0, 0], 35. [0, 2, 4, 0, 4, 3, 4, 0, 0, 0, 0], 36. [0, 3, 5, 1, 0, 0, 4, 1, 0, 0, 0]]) 37. U,sigma,VT = np.linalg.svd(scoreData) 38. sigma\_K = np.mat(np.eye(6)\*sigma[:6]) 39. scoreDataRC = sigma\_K \* U.T[:6,:] \* scoreData 40. n = np.shape(scoreData)[1] ***#菜品总数*** 41. userlndex = 17 42. for i in range(n): 43. userScore = scoreData[userlndex, i] 44. if userScore != 0: 45. continue 46. print("index:{},score:{}".format(i, estScore(scoreData, scoreDataRC, userlndex, i)))   执行结果： | | | |
| 结论分析与体会：  通过本次练习，我熟悉了解了Python的numpy模块，巩固了知识点，包括创建、打印、访问n维数组，数组运算，副本与视图等。同时掌握了用余弦求相似度的方法以及稀疏数据矩阵的降维处理方法。 | | | |