上海理工大学光电信息与计算机工程学院

**《推荐系统》实验二**

****

|  |  |
| --- | --- |
| **学　　院** | 光电信息与计算机工程学院 |
| **专　　业** | 数据科学与大数据技术 |
| **学生姓名** | 罗震宇 |
| **学　　号** | 2135060321 |
| **指导教师** | 艾均 |
| **成 绩** |  |

**目录**

[实验一 User-based Cosine-CF算法实现 5](#_Toc152052417)

[1. 实验要求与环境 5](#_Toc152052418)

[1.1 实验要求 5](#_Toc152052419)

[1.2 实验环境 5](#_Toc152052420)

[2. 实验过程 5](#_Toc152052421)

[2.1 数据输入 5](#_Toc152052422)

[2.2 用户间相似性计算与排序 5](#_Toc152052423)

[2.3 用户评分预测 6](#_Toc152052424)

[2. 实验结果 6](#_Toc152052425)

[4. 个人收获 8](#_Toc152052426)

实验一 User-based Cosine-CF算法实现

## 实验要求与环境

### 1.1 实验要求

1. 选择MovieLens small 数据集；
2. 实现基于用户Cosine相似性的协同过滤（CF）个性化推荐算法；
3. 读入Ratings数据并适当保存和索引；
4. 数据按折n验证分割成训练集和测试集；
5. 计算用户间相似性；
6. 用户邻居按照相似性进行排序；
7. 针对测试集中预测目标，预测用户在k个邻居时的可能评分；
8. 计算k个邻居时User-based Pearson-CF的平均绝对误差并画出误差曲线；

### 1.2 实验环境

处理器 AMD Ryzen 7 5800H with Radeon Graphics

内存 16 GB

系统 Windows 11

IDE PyCharm Professional 2023.2.4

## 2. 实验过程

### 2.1 数据输入

使用pandas库，从指定文件地址读取电影评分数据和电影数据并随机打乱电影评分数据。将打乱后的电影评分数据采用K折交叉验证法进行划分，将数据分成5个折叠。返回这些折叠、原始的电影评分数据、电影数据以及用户ID。

### 2.2 用户间相似性计算与排序

读取用户的电影评分数据和电影信息后，将评分数据转换成字典数据结构。基于Cosine相关系数计算用户之间的相似度，为每一对用户生成一个相似度分数。

Cosine相关系数计算公式：

接下来为每个用户确定与其相似度最高的用户，并将其他用户按相似度降序排列，获得指定数量的邻居。

### 2.3 用户评分预测

设计函数用于对一组用户-物品对进行评分预测。它接受数据集的折叠、用户-物品双重字典、物品-评分数据双重字典、所有用户的平均评分数据双重字典、用户配对信息和指定邻居数量。

在函数中，首先计算目标用户的平均评分（如果存在），然后根据邻居的评分和相似度计算出预测评分。如果计算结果不是一个数字（计算时有None），则返回目标用户的平均评分。最后，根据相似用户的邻居信息进行评分预测，将不同邻居数量要求下的预测结果存储在一个双重字典中。

基于Cosine相似性的协同过滤评分预测公式：

## 实验结果



Fig.1 Line Chart of MAE vs. Neighbors

如表 1和Fig.1所示，在从5至100的20个不同邻居数量下，模型的MAE逐渐下降。在Neighbors从5增加到100的过程中，MAE的下降趋势逐渐减小。邻居数量达到55后，MAE基本不再减小，开始在0.679800左右波动。

从上述实验中，我们可以得出以下结论：在一定范围内增加Neighbors数量会提高模型的性能，降低MAE。然而，增加Neighbors的数量对改进模型性能的效果在某个点开始减弱，而且随着Neighbors数量的增加，性能的提升幅度逐渐减小。当Neighbors为60时，MAE达到0.679843，而当Neighbors为65时，MAE略微升高为0.679870，这说明在这个数据集中，Neighbors数量的微小变化可能对性能没有显著影响，或者MAE已经收敛到一个较低的值。

总之，从这组数据中可以得出，增加Neighbors数量可以改善模型的性能，但在某个点后性能提升会减小，因此需要权衡性能和计算成本，选择适当数量的Neighbors来进行建模

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **表 1**  MAE on different numbers of *Neighbors* | | | | | | | |
| Neighbors | MAE | Neighbors | MAE | Neighbors | MAE | Neighbors | MAE |
| 5 | 0.716177 | 30 | 0.681233 | 55 | 0.679869 | 80 | 0.679734 |
| 10 | 0.692826 | 35 | 0.680735 | 60 | 0.679843 | 85 | 0.679716 |
| 15 | 0.685904 | 40 | 0.680543 | 65 | 0.679870 | 90 | 0.679687 |
| 20 | 0.683612 | 45 | 0.680375 | 70 | 0.679824 | 95 | 0.679695 |
| 25 | 0.682083 | 50 | 0.680131 | 75 | 0.679818 | 100 | 0.679712 |

## 4. 个人收获

本次题目的目的是设计并实现一个基于协同过滤的推荐系统，利用用户的评分数据，为用户提供个性化的推荐服务，使用Cosine函数来计算相似性。我对本次题目的完成情况给予了96分。在完成本次题目的过程中，有以下几个方面的收获：

在理论学习方面，最开始的时候没有理解原理便开始做实验，效果不好，后来我深入掌握了协同过滤的核心思想和算法流程，学习了如何使用皮尔逊相关系数来衡量用户之间的相似度，以及如何使用k折交叉验证法来评估模型的准确性和稳定性，再写代码就快很多了。在划分好了训练集和测试集之后，并没有根据训练集来计算用户之间的相似性，然后用测试集来计算用户的预测得分。后来将训练集和测试集分别传入到相似性的计算中和预测得分的计算中。通过这个部分，我感受到了计算机科学的严谨性和创新性，激发了我对科研的兴趣和热情。

在编程实现方面，最开始并没有很清晰的思路，都是想到哪写到哪，没有太注意规范。后来经过同学指教，我采用了模块化的编程思想，将每一个步骤封装成一个函数，然后在主程序中调用，实现了代码的高效和可维护。我还注意了代码的规范和注释，提高了代码的可读性和可理解性。

在报告撰写方面，我遵循了上海理工大学论文的格式要求，详细地阐述了实验的背景、原理、过程和结果，较好地反映了本次实验的目的和意义。实验结果具有可重复性和科学性，可以作为推荐系统研究的一个参考案例。

整个过程中还是遇到了很多问题，例如每次运行代码时间都很长，导致debug速度较慢；对debug工具不够数量，前期浪费了很多时间等，希望自己能够汲取这次的经验教训，下次更快更好的完成实验。