

实验三

实验名称：电影推荐

实验目的：

- 1) 熟练掌握字典操作
- 2) 熟练掌握列表推导式
- 3) 熟练掌握 lambda 函数
- 4) 熟练掌握导入模块，txt 文件读写
- 5) 学会 JSON 文件读写

实验内容：

必做内容：

V1.0

假设已有若干用户名字及其喜欢的电影清单，现有某用户，已看过并喜欢一些电影，现在想找个新电影看看，又不知道看什么好。

思路：根据已有数据，查找与该用户爱好最相似的用户，也就是看过并喜欢的电影与该用户最接近，然后从那个用户喜欢的电影中选取一个当前用户还没有看过的电影，进行推荐

代码采用字典嵌套集合（看过并喜欢）来存放数据，格式为{用户 1:{电影名称 1, 电影名称 2, ...}, 用户 2: {...} }

代码：

```
from random import randrange
# 模拟已有历史数据，{用户名:{喜欢的电影名单}}
data = {'user'+str(i):{'film'+str(randrange(1, 10)) for j in range(randrange(15))} for
i in range(10)}
# 待测用户曾经看过并感觉不错的电影
user = {'film1', 'film2', 'film3'}
# 查找与待测用户最相似的用户和 Ta 喜欢看的电影
similarUser, films = max(data.items(), key=lambda item: len(item[1]&user))
print('历史数据： ')
for u, f in data.items():
```

```

    print(u, f, sep=':')
print('和您最相似的用户是： ', similarUser)
print('Ta 最喜欢看的电影是： ', films)
print('Ta 看过的电影中您还没看过的有： ', films-user)

```

任务 1：读懂以上 V1.0 代码

V2.0

根据各个不同的用户给电影的评分完成电影的推荐

已有大量用户（随机产生）对若干电影的打分数据，现有某用户（随机产生）也看过一些电影并进行过评分，要求根据已有打分数据为该用户进行推荐。

增加的功能：

代码采用基于用户的协同过滤算法，也就是根据用户喜好来确定与当前用户最相似的用户，然后再根据最相似用户的喜好为当前用户进行推荐

最相似的用户及其对电影打分情况即：两个用户共同打分的电影最多，并且这些所有的共同电影打分差值的平方和最小

代码采用字典嵌套字典来存放数据，格式为{用户 1:{电影名称 1: 打分 1, 电影名称 2: 打分 2, ...}, 用户 2: {...} }

运行结果格式如下：

```

=====know data=====
2:9:('user0', {'film12': 3, 'film10': 1, 'film3': 2, 'film9': 4, 'film8': 3, 'film4': 1})
0:0:('user1', {'film7': 4, 'film10': 5, 'film8': 2, 'film5': 4})
1:9:('user2', {'film14': 5, 'film3': 1, 'film11': 5, 'film1': 2, 'film5': 2, 'film7': 1, 'film8': 1,
'film2': 5})
1:4:('user3', {'film11': 3, 'film9': 3, 'film10': 5, 'film2': 3})
2:2:('user4', {'film8': 2, 'film1': 4, 'film7': 2, 'film9': 2, 'film2': 3, 'film5': 1, 'film13': 1})
1:4:('user5', {'film5': 2, 'film7': 4, 'film14': 4})
1:9:('user6', {'film8': 3, 'film3': 5, 'film6': 4, 'film2': 3})
2:8:('user7', {'film14': 4, 'film9': 3, 'film5': 2, 'film8': 1, 'film2': 5})
1:4:('user8', {'film4': 3, 'film12': 4, 'film1': 4, 'film11': 5, 'film7': 3, 'film3': 3, 'film8': 1})
3:1:('user9', {'film4': 1, 'film13': 2, 'film1': 5, 'film14': 1, 'film10': 2, 'film12': 5, 'film8':
3})

```

```

=====current user=====
{'film14': 2, 'film6': 1, 'film13': 2, 'film9': 1, 'film4': 1}
=====most similar user and his films=====
user9:{'film4': 1, 'film13': 2, 'film1': 5, 'film14': 1, 'film10': 2, 'film12': 5, 'film8': 3}
=====recommend film=====
film12

```

任务 2：请根据 V1.0 来编写 V2.0 的 Python 代码

V3.0

每个用户有对电影打分，根据用户喜好来确定与当前用户最相似（多维欧几里得距离）的用户，然后再根据最相似用户的喜好为当前用户进行推荐

采用字典嵌套字典来存放数据，格式为{用户 1:{电影名称 1: 打分 1, 电影名称 2: 打分 2, ...}, 用户 2: {...} }.

3.0 版本新增功能：

将用户、不同用户看过的电影以及对电影的评分等原始数据存入文件。

追加一部分新用户以及他们看过的电影及评分，并加到文件中。

根据录入的用户及相关信息找出观影习惯最接近的用户，并给出推荐的电影。

确定功能：

设计一个预处理程序：

预处理：随机生成用户原始打分数据, 存入 JSON 文件(文件名: history_list.json)

history_list.json 文件举例如下所示：

```

{
    "user0": {
        "film10": 5,
        "film7": 2,
        "film3": 5,
        "film14": 4,
        "film2": 1
    },
    "user1": {
        "film2": 1,

```

```
        "film4": 4,
        "film9": 4,
        "film3": 2,
        "film7": 2,
        "film1": 5
    },
    "user2": {
        "film3": 5,
        "film1": 5,
        "film12": 3,
        "film5": 2,
        "film14": 5
    },
    "user3": {
        "film4": 2,
        "film11": 3,
        "film10": 5,
        "film7": 5,
        "film2": 5,
        "film12": 4
    },
    "user4": {
        "film7": 5,
        "film9": 2,
        "film13": 4,
        "film12": 4,
        "film5": 5,
        "film3": 1,
        "film8": 2
    }
}
```

设计一个推荐程序：

输入：原始 JSON 文件

处理：读入原始文件，添加若干新的打分数据（随机生成），根据录入的用户及相关信息找出观影习惯最接近的用户，并给出推荐的电影。

输出：1) 添加数据后，写回 JSON 文件。

2) 在屏幕打印出如版本 2.0 的结果，并将该结果写入 txt 文件 (recommendmoive.txt)

任务 3：学习 JSON 文件的读和写，然后根据 V2.0 来编写 V3.0 的 Python 代码

选做内容：

设计图形界面交互

需要提交的文件：

- 1) 任务 2 和任务 3 的源代码（里面要有注释,包括版本和功能注释）请分成两个.cpp 文档
- 2) 实验报告（实验名称、作者、实验目的、实验内容（简述）、实验流程图，实验代码、实验结果（截图）、实验心得（请认真写心得）），两个版本放在一个报告（请注意格式的美观）
- 3) 运行视频（带个人电脑特点的运行视频）