从0构建AI推荐系统（简单的电影推荐系统）

1. 前言

从0构建AI推荐系统好久没继续深入研究了，一方面因为工作内容较多，另一方面在准备task任务系列，从0开始一步步完成简单框架的搭建与应用。最近看到一篇文件介绍电影推荐系统，感觉非常有意思，抽半天的时间动手实践了一下，本文记录学习电影推荐系统的全过程。

另外，争取下个月能完成从0构建AI推荐系统（金融资讯）的内容。

本文代码已经上传至GitHub：

<https://github.com/lixj-zj/AI_recommendation_system/blob/master/re_system_movie.py>

1. 相关环境介绍

编程语言：python

编辑器：pycharm

第三方库：pandas、numpy、sklearn

数据集：<https://grouplens.org/datasets/movielens/> ml-100k.zip (size: 5 MB, checksum)

注：

1. Scikit-learn，简称sklearn，一个用于Python编程语言的免费软件机器学习库，支持分类、回归、降维和聚类四大机器学习算法，包含特征提取、数据处理和模型评估三大模块，支持k均值、支持向量机、随机森林、梯度增强等常见算法。

另外，由于sklearn建立在NumPy、Scipy和matplotlib库基础上，利用这几大模块的优势，可以大大提高机器学习的效率。

  sklearn拥有着完善的文档，上手容易，具有着丰富的API，在学术界颇受欢迎。同时sklearn内置了大量数据集，节省了获取和整理数据集的时间。

2. 本文使用的数据集较少且时间较早，但可以用来学习与练手，换作别的数据集后，实现的推荐方式大同小异。

1. 构建电影推荐系统（基于内容的协同过滤算法）
2. 推荐系统简介

使用推荐算法进行数据挖掘，发现用户可能需要的商品。

推荐系统一般由以下三部分构成：

1. 基础数据部分（数据的质量决定推荐系统的上限，各种推荐算法无限逼近这个上限）

2. 推荐算法系统（一般由多个算法组成的推荐模型）

3. 前端展示（直观、高效的展示效果往往更能引人入胜、说明问题）

常见的推荐算法包括以下：

1. 基于流行度的推荐算法（按热门推荐）

2. 协同过滤算法

基于用户的协同过滤算法（推荐和你类似的人喜欢的商品）

基于商品的协同过滤算法（推荐喜欢这个商品的人也喜欢的商品）

3. 基于内容的过滤算法（词权、word2vec、聚类）

4. 基于模型的推荐算法（多种机器学习算法）

5. 基于矩阵分解的推荐算法（LFM隐语义模型）

6. 混合算法（结合多种推荐算法）

1. 数据预处理

在数据集中，主要用到三种数据：

u.user：用户信息表（包括用户ID，年龄，性别，职业，邮编）

u.data：电影评分表（包括用户ID，电影ID，评分值，评分时间戳）

u.item：电影信息表（包括电影ID，电影名，上映时间，网址，电影类型）

首先，采用pandas.read\_csv()方法读取三种数据，并对应别设定每列的名称。

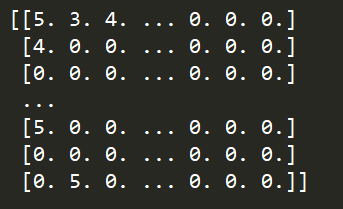
================init()代码=====================

1. 构造用户电影矩阵

这一步主要是构造一个矩阵，每一行代表一个用户，每一列代表一部电影，用户与电影对应的矩阵元素则是用户对该电影的评分。

================constructUserMovieMatrix ()代码=====================

执行结果：



注：

1. 第53、54行代码计算用户数与电影数，采用unique()方法去重；
2. Shape[0] 返回表示DataFrame维度的元组；
3. Line 的内容如下：

Pandas(Index=0, user\_id=196, movie\_id=242, rating=3, unix\_timestamp=881250949)

由于每行数据从1开始，存入下标为0的矩阵时要-1。

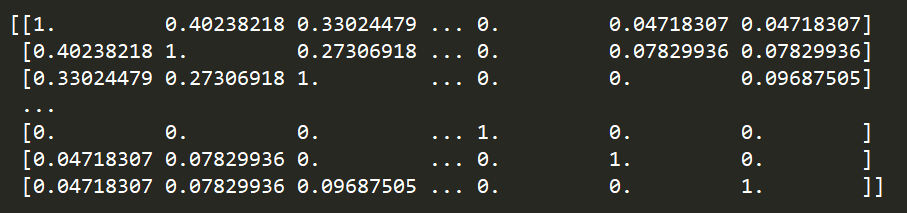
1. 构建电影之间相似度矩阵

采用计算余弦相似度的方式，计算电影之间的相似度矩阵。

================calculationSimilarity ()代码=====================

注：在此处，转置是计算电影之间的相似度矩阵，不转置则计算用户之间的相似度矩阵！！

执行结果：

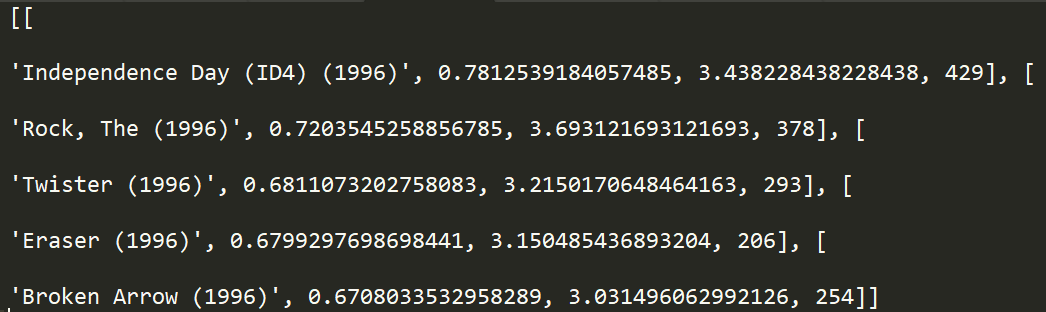


1. 计算推荐电影

利用电影数据集中电影的相似程度，输入一部电影名或部分电影名（主要利用contains()方法），可以返回相似电影的电影名、相似度、平均评分、评分数。

================rec\_sys ()代码=====================

执行结果：



1. 总结

此电影推荐系统相对来说较为简单，过程比较清晰，处理的数据也不是很复杂，重点在于理解基于内容的协同过滤算法。

当然，数据集中涵盖电影的分类信息，可以尝试采用聚类算法将相似的电影聚类，换一个新的角度提高推荐算法的准确性。

五、参考链接

<https://www.cnblogs.com/lianyingteng/p/7811126.html>

<https://blog.csdn.net/u014248127/article/details/78885180>

<https://mp.weixin.qq.com/s/b-BD0glGpTglqsJzkj8ecA>