电影推荐系统

大数据分析技术期中开题报告

杨淳瑜，王海天，陈可豪，王熙同

2025 年 4 月 24 日

目录

加权评分方法

相似性推荐

关联规则推荐

加权评分方法

加权评分方法

为什么要定义一种加权的评分方法？

- 只考虑评分：小众电影评分人少但分高，不准确
- 只考虑评分人数：烂片评分人多但分低，不准确
- 需要一种加权的评分方法，既能反映评分质量，又能反映评分人数

数据集基础统计

统计项	值
电影总数	45,463
年份范围	1874 - 2020
最高产年份	2014.0
平均评分	5.62 / 10
评分中位数	6.00 / 10
评分范围	0.0 - 10.0
平均投票次数	109.9
投票次数中位数	10.0
最高投票次数	14,075.0
平均时长	94.1 分钟
时长范围	0 - 1256 分钟

基础评分公式

$$WR = \left(\frac{v}{v+m}\right) \cdot R + \left(\frac{m}{v+m}\right) \cdot C$$

参数说明：

- C：全平台电影平均分（基准线）
- m：最小有效投票阈值，例如下 95% 的位置
- v：当前电影实际投票数
- R：当前电影原始平均分

参数分析

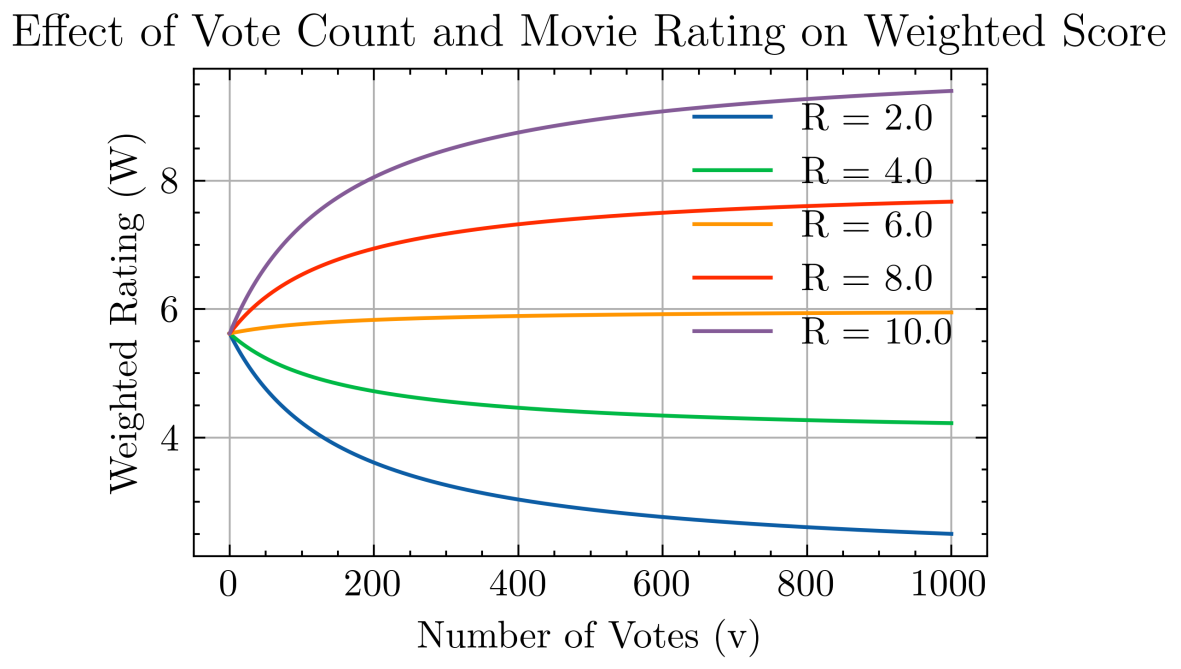

图 1: 参数分析

由于均值是评分 5.62:

- 当 R 小于 5.62 的时候，评分人数越多，WR 越小
- 当 R 大于 5.62 的时候，评分人数越多，WR 越大

相似性推荐

相似性推荐

核心思路：对剧情文本进行清洗，然后向量化，然后计算余弦相似度。

向量化的方法由简单到复杂，我们计划使用：

- TF-IDF
- BERT

向量化：TF-IDF

核心步骤：

1. 使用 TF-IDF 算法进行向量化
2. 计算剧情向量间的余弦相似度
3. 基于相似度排序，生成 Top-K 推荐

核心算法：

- TF-IDF **权重计算**：

$$w_{ij} = \mathrm{TF}(t_j, d_i) \times \log \frac{N}{\mathrm{DF}(t_j)}$$

- **余弦相似度**：**与文本长度无关**, Scaling Invariant

$$\cos(\theta) = \frac{\vec{A} \cdot \vec{B}}{\|\vec{A}\| \|\vec{B}\|}$$

向量化：BERT

技术路线：

1. 采用 Huggingface 上的轻量级 RoBERTa 预训练模型
2. 对电影剧情文本进行语义向量编码
3. 使用余弦相似度计算电影语义相似性
4. 生成基于深度语义的推荐结果

关联规则推荐

FP-Growth 算法流程

实现步骤：

1. 转换用户评分行为数据为事务数据集
2. 构建 FP-Tree 压缩数据结构
3. 递归挖掘频繁项集
4. 生成高质量关联规则
5. 根据置信度和加权评分的综合分数进行筛选和排序

有效规则定义

应用策略：

- 最小支持度：0.06
- 最小置信度：0.3
- 规则排序：置信度优先

FP-Tree 离线生成，用户输入电影名称，我们查询挖掘到的规则中，以该电影为前件的规则，然后根据置信度排序，生成 Top-K 推荐。

更进一步，我们可以允许前件有多个电影，用户的输入也可以是多个。