



# 大数据处理综合实验课程(2025)

课程设计1 - 个性化图书兴趣分析与推荐系统

南京大学 计算机学院







## 1. 课程设计目标

本课程设计聚焦基于Hadoop平台的个性化图书推荐系统开发,贯通数据预处理与特征提取、用户画像建模与聚类分析、图书多维度相似度计算、协同过滤推荐服务实现的全流程。通过MapReduce分布式编程与推荐算法实践,掌握海量数据处理的工程化方法,强化团队协作与系统架构设计能力,最终形成大数据分析技术在推荐场景下的系统性理解与应用能力。

## 2. 学习技能

通过本课程设计,可以熟悉或掌握以下大数据处理和推荐系统开发相关技能:

- 利用 MapReduce 完成图书数据预处理;
- 利用 MapReduce 进行图书相似度计算;
- 利用 MapReduce 对用户画像建模、根据特征对于用户进行聚类;
- 理解基于用户聚类和实体相似度的推荐算法基本原理(兴趣画像、协同过滤)。







## 3. 任务描述

在数字化图书平台中,用户行为(如借阅、评分、浏览)记录蕴含了丰富的兴趣信息。通过对这些数据进行建模和分析,可以实现个性化推荐,提高用户满意度。本课程设计的任务是基于 MapReduce 技术,构建一个完整的图书兴趣分析与推荐系统。如未特殊说明,要求所有任务均使用 MapReduce 程序完成,Reducer的数量要求为2~4个。建议先在本地使用样本数据集进行测试,测试无误后再使用全量数据集在集群提交任务。

## 主要包括以下四项任务(详细定义见后续内容):

- 图书数据预处理;
- 图书相似度计算,通过协同过滤挖掘潜在关联图书;
- 构建用户画像,挖掘用户偏好的图书特征,对用户进行聚类;
- 个性化推荐服务实现,结合用户画像与图书相似度生成用户个性化的推荐图书列表。

使用数据集: 开源图书评分数据

网址: Goodbooks-10k: a new dataset for book recommendations - FastML

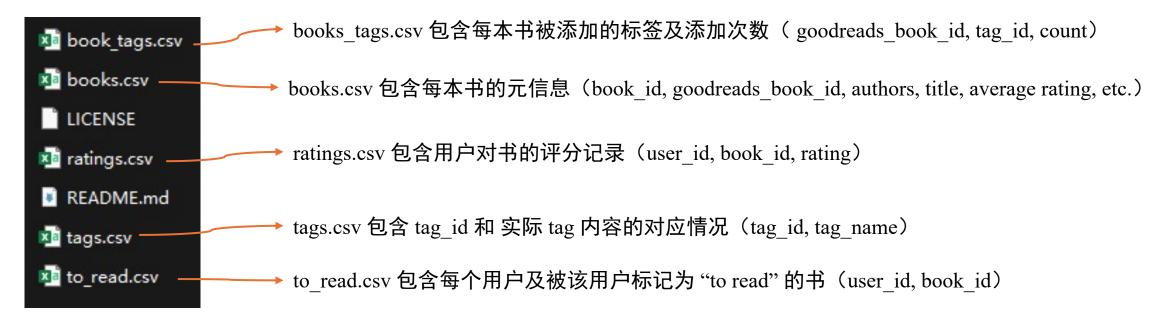
数据集在平台的存储目录: /user/root/FinalExp/FinalExp1 (注: 平台上的数据集与网址上的数据集略有差异,请使用平台中的数据集)







## 4. 数据集描述



**注意**:每本书可能有多个**版本**,数据集中的 goodreads\_book\_id 和 best\_book\_id 都代表一本书最受欢迎的版本,work\_id 则不区分版本地指代一本书(以《哈利·波特与魔法石》为例,无论哪个版本(英文原版、中文译本、20周年纪念版等), 它们的 work id 都相同,因为本质上是同一部作品。),book id 则是每本图书(区分版本)的序号。 ratings.csv 和 to read.csv 中的 book id 关联到 work id 而不是 goodreads book id, 这意味着用户对一本书的不同版本的评 分或 "to\_read" 标记会合并到同一个 work\_id 下。







## 4. 数据集描述

数据表	字段(部分)	描述
books.csv	book_id	一本图书(区分版本)的序号
	goodreads_book_id	一本书最受欢迎的版本号
	best_book_id	同 "goodreads_book_id"
	work_id	不区分版本地指代一本书
	authors	书的作者
	title	书名
ratings.csv	user_id	用户编号
	book_id	同books.csv中的book_id
	rating	评分
book_tags.csv	goodreads_book_id	同books.csv中的goodreads_book_id
	tag_id	标签编号
	count	被打该标签的次数

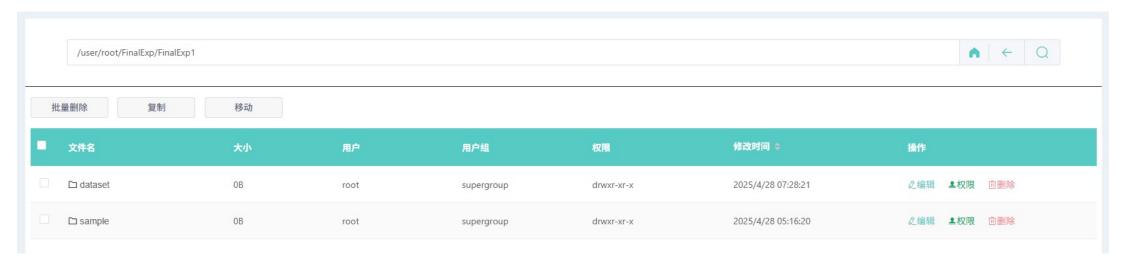






## 4. 数据集描述

数据表	字段(部分)	描述
tags.csv	tag_id	标签编号
	tag_name	标签内容
to_read.csv	user_id	用户编号
	book_id	一本图书(区分版本)的序号



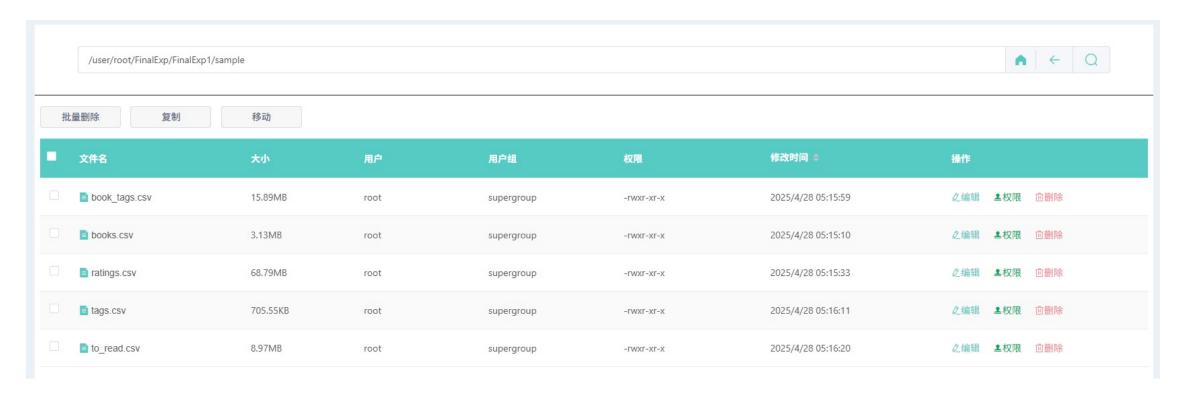
实验数据集路径: /user/root/FinalExp/FinalExp1







## 4. 数据集描述



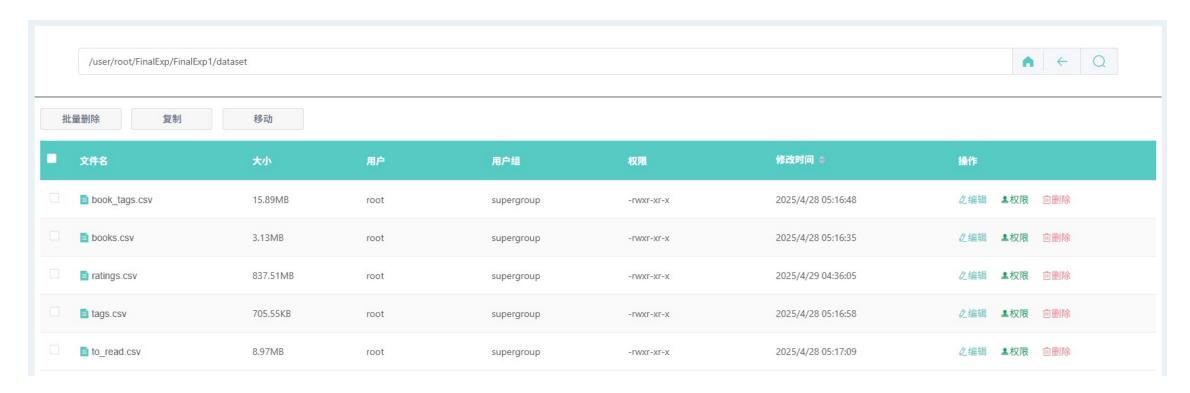
样本数据集路径: /user/root/FinalExp/FinalExp1/sample







## 4. 数据集描述



全量数据集路径: /user/root/FinalExp/FinalExp1/dataset







## 任务 1 - 数据预处理

本任务的主要工作是对原始数据集进行数据预处理,完成以下操作:

- 1. 修正 books.csv, 将该文件中的 original\_publication\_year 列的空值均改为 1900;
- 2. 修改 books.csv 中的 original publication year 字段,将其改为 YYYYs 格式(例如 2008 则改为 2000s);
- 3. 对修正后的 books.csv 进行提取操作,输出新文件 books simplified,每行格式为

(book\_id, goodreads\_book\_id, best\_book\_id, work\_id, authors, original\_publication\_decade, title)

4. 对 books\_simplified、book\_tags.csv 做联合操作,输出新文件 book\_tags\_flattened,每行格式为

(book id \t {tag id1}: {num1}, {tag id2}: {num2},...) (\t代表制表符,后续一样)

其中{tag\_id1}:{num}中 {tag\_id1} 表示 book\_id 所指代图书第一个标签的编号, {num1}表示该图书被打标签 {tag\_id1} 的次数, 后续标签依此类推。







## 任务 2 - 图书相似度计算

本任务的目标是基于用户评分记录,构建图书相似度记录,用于后续向用户个性化推荐图书。

输入数据: books\_simplified、ratings.csv、book\_tags\_flattened

#### 输出格式:

- Part A:找出所有在任一用户的评分历史中共现的图书对,统计该共现图书对在所有用户的各自评分历史中的总共现次数,输出格式为  $((book\_id_i,book\_id_j),$  总共现次数),同一对图书只输出一次;
- Part B: 对于 Part A 中找到的每个图书对 (book\_id<sub>i</sub>, book\_id<sub>i</sub>), 计算以下维度的相似度
  - tags 相似度: 计算 Jaccard 系数 ( |tags 集合交集| / |tags 集合并集| );
  - authors 相似度: 计算 Jaccard 系数 ( |authors 集合交集 | / |authors 集合并集 | ); (注意一本图书可能有多个作者)
  - 出版年代相似度: 1-(绝对年份差/数据集中图书的最大年份差)(例如1980s 和 2010s:  $1-\frac{|1980-2010|}{Max\ Year\ Gap}$ );
- Part C:将相似度进行融合计算,输出文件 books\_similarity,其每行格式为((book\_id\_i, book\_id\_j), similarity(book\_id\_i, book\_id\_j)),其中

$$similarity(book\_id_i,book\_id_j) = \frac{1}{4}*\frac{book_i \pi book_j$$
总共现次数  $+\frac{1}{4}*tags$ 相似度  $+\frac{1}{4}*authors$ 相似度  $+\frac{1}{4}*$ 出版年代相似度。







任务 3 - 构建用户画像, 形成用户聚类

任务 3.1 - 构建用户画像

本任务的目标是为每位用户构建兴趣画像,并保存到本地。

输入数据: books\_simplified、ratings.csv、to\_read.csv、 book\_tags\_flattened 提示: 每次只统计一种阅读偏好(标签/作者/出版年代)。

输出格式:根据用户评过分的图书和 "to\_read" 的图书,统计每个用户的阅读偏好,输出文件 user\_preference

- 标签及标签在用户评过分的图书和 "to read" 的图书的标签中出现的次数;
- 作者及作者在用户评过分的图书和 "to\_read" 的图书的作者中出现的次数(注意一本图书可能有多个作者);
- 出版年代及出版年代在用户评过分的图书和 "to read" 的图书的出版年代中出现的次数;

 $user\_id \ \ tags=\{tag\_id1\}: \{t\_num1\} | \{tag\_id2\}: \{t\_num2\} | \dots; \\ a\_num1\} | \{author2\}: \{a\_num2\} | \dots; \\ y\_num1\} | \{y\_num1\} | \{y\_num1\} | \{y\_num2\} | \dots; \\ y\_num2\} | \dots; \\ y\_num3\} | \dots; \\ y\_num4$  |  $y\_num4$  |

#### 输出样例:

tags=21:3|32:2;authors=John Doe:4|Smith:1;years=2000s:3|2010s:2







## 任务 3 - 构建用户画像, 形成用户聚类

#### 任务 3.2 - 形成用户聚类

本任务的目标是使用 K-means 聚类方法对用户进行聚类, 此处 K 设定为 100。

输入数据: user preference、tags.csv

#### 特征编码:

• 标签:通过哈希映射为 512 维的向量 t; (哈希取模 Math.abs(str.hashCode() % length) 映射到向量索引,下同)

• 作者:通过哈希映射为 256 维的向量 a;

• 出版年代:针对所有出现的年代(如 2000s、2010s 等),建立 One-Hot 编码 y。

将三段向量(按顺序 t,a,y)拼接成一个新的向量 v,即用户i的特征向量  $v_i$ 。

使用用户各自的特征向量 v 进行 K-means 聚类操作(使用欧氏距离),输出新文件 user\_cluster。

输出格式: user id \t cluster id

输出样例: 123 \t 1 (说明 user\_id 为 123 的用户被聚类到类簇 1)

提示:初始点选取可以先随机采样一部分数据,执行一次聚类操作找出初始中心,再在全量数据上执行聚类;若聚类过程太慢,可以采用  $PCA^{[*]}$  将向量降维到 $100 \sim 200$ 维再聚类。







## 任务 3 - 构建用户画像,形成用户聚类

#### 任务 3.2 - 形成用户聚类

根据用户偏好信息、用户聚类信息,生成用户类簇的偏好信息

输入数据: user preference、user cluster

统计属于同一类簇中的用户的偏好标签、偏好作者、偏好出版年代,生成这一类簇的整体偏好信息。

- 统计每个类簇中出现的偏好标签及其总次数; (即所有属于该类簇的用户的偏好标签,并统计总次数)
- 统计每个类簇中出现的偏好作者及其总次数;
- 统计每个类簇中出现的偏好出版年代及其总次数。

为每一个类簇筛选出总次数最多的64个偏好标签、总次数最多的32个偏好作者、总次数最多的8个偏好出版年代,并按次数降序排列,作为整个类簇的偏好信息。若不足数量则全部输出。输出文件 cluster\_preference。

 $cluster\_id \ \ tags=\{tag\_id1\}: \{t\_num1\} | \{tag\_id2\}: \{t\_num2\}| ...; \\ a\_num1\} | \{author2\}: \{a\_num2\}| ...; \\ y\_ears=\{y\_ear1\}: \{y\_num1\} | \{y\_ear2\}: \{y\_num2\}| ...; \\ y\_ears=\{y\_ear1\}: \{y\_num2\}: \{y\_num2\}:$ 

## 输出样例:

58 tags=21:120|45:110...;authors=J.K. Rowling:23|Smith:19...;years=1980s:32|2010s:31...







## 任务 4 - 个性化图书推荐服务

本任务的目标是为每位用户生成个性化推荐列表,综合用户兴趣与图书相似度,输出 Top - 10 推荐图书。

输入数据: books\_simplified、book\_tags\_flattened、books\_similarity、ratings.csv、to\_read.csv、user\_cluster、cluster\_preference

#### 基于兴趣匹配推荐

- 1. 从该用户所在类簇中提取类簇偏好信息:偏好标签、偏好作者、偏好年代段;
- 2. 获取该用户已评分的或已标记为 'to read' 的图书集合 R;
- 3. 从 books\_simplified 中找出与提取出的类簇偏好信息相似度最高的 10 + |R| 本的图书,作为候选图书集 C,不足10 + |R| 本则全部保留;(参考任务2相似度的计算方法,不考虑共现次数,仅考虑标签、作者、出版年代,系数均从  $\frac{1}{4}$  调整为  $\frac{1}{3}$  );
- 4. 从 C 中排除出现在 R 中的图书, 形成推荐集合  $W_1$ , 即  $W_1 = C R$ 。

#### • 基于协同过滤推荐

- 1. 从 books similarity 中获取与集合 R 中任一图书相似度最高的 10 + |R| 本图书,形成候选图书集 S,不足10 + |R| 本则全部保留;
- 2. 从 S 中排除出现在 R 中的图书,形成推荐集合  $W_2$ ,即  $W_2 = S R$ 。

#### • 生成加权推荐列表

分别以权重 [0.1,0.9;0.2,0.8;...;0.8,0.2;0.9,0.1] 从  $W_1$  和  $W_2$  中取 10 本图书形成向用户推荐的图书列表,输出文件 recommend。若对于权重[u,1-u], $|W_1|<10u$  但  $W_2$  满足  $|W_2| \ge 10-|W_1|$ ,则额外从  $W_2$  取出  $10u-|W_1|$  本图书,反之亦然。若  $|W_1|+|W_2|<10$  则全部输出。

输出格式: (user\_id \t {rec\_book\_id\_1}, {rec\_book\_id\_2},...), 其中 {rec\_book\_id1} 表示向用户推荐的第一本图书的 book\_id, 依此类推。







# 课程设计要求

#### 1. 提交材料

- 1. 程序源代码,包含完整目录结构的 src 包,并提供编译方法说明;
- 2. 可执行 jar 包以及 jar 包的执行方式说明;
- 3. 程序设计报告(pdf 格式),报告内容包括每个子任务涉及的程序的主要流程、程序运行结果截图、在实现过程中进行的优化工作、优化取得的效果说明(如有)。

以上材料打包为一个zip压缩包。

- 4. 课程设计报告(pdf格式),报告内容包括:
  - 1. 小组信息(姓名、学号、联系方式)
  - 2. 小组分工情况:明确各成员在课程设计中分工的内容,要求以 git 形式分工合作(小组组长使用 git.nju.edu.cn 创建 一个项目仓库,其余小组成员以开发者身份加入该项目,注意要在仓库中备注小组信息)。
  - 3. 详细设计说明,包括详细程序设计、程序框架、功能模块、参数的选定、主要类的设计说明,包括主要类、函数的输入输出参数、尤其是 map 和 reduce 函数的输入输出键值对详细数据格式和含义,主要功能和算法代码中加清晰的注释说明,如果有优化点或创新点,请明确说明。
  - 4. 总结:程序的特点总结,功能、性能、扩展性等方面存在的不足和可能的改进之处







# 课程设计要求

#### 2. 完成周期

课程设计完成周期为一个月,验收截止日期:2025年6月3日实验课结束

后续实验课均要求到教室

提交方式:上传至课程网站

#### 3. 提醒

- 1. 设计程序时不一定要完全按照文档中给出的方法,可以有自己的解决方案,鼓励探索新方案;
- 2. 提交材料前请务必检查好是否有遗漏、文件格式是否符合要求;
- 3. 切勿抄袭代码:
- 4. 先使用提供的部分数据集在本地进行验证,确认无误后再使用大数据平台运行完整数据,程序执行说明务 必详细。