# 软件项目开发计划书

第八组：梁爽 杨祥睿 黄英伦 杜家琛

目录

[软件项目开发计划书 1](#_Toc9115)

[1 引言 2](#_Toc6930)

[1.1 编写目的 2](#_Toc16710)

[1.2 背景 2](#_Toc32397)

[1.3 定义 3](#_Toc30209)

[1.4 参考资料 3](#_Toc1279)

[2 项目概述 3](#_Toc19393)

[2.1 项目目标 3](#_Toc4329)

[2.2 产品目标与范围 4](#_Toc27633)

[2.3 假设与约束 4](#_Toc5324)

[2.4 项目工作范围 4](#_Toc25)

[2.5 应交付成果 4](#_Toc4492)

[2.5.1 需完成的软件 4](#_Toc5203)

[2.5.2 需提交用户的文档 5](#_Toc4877)

[2.5.3 须提交内部的文档 5](#_Toc15456)

[2.6 项目开发环境 5](#_Toc24525)

[2.7 项目验收方式与依据 5](#_Toc10116)

[3 项目团队组织 6](#_Toc18904)

[3.1 组织结构 6](#_Toc2081)

[3.2 人员分工 6](#_Toc26724)

[3.3 协作与沟通 6](#_Toc5491)

[3.3.1 项目团队内部协作 6](#_Toc1845)

[3.3.2 项目接口人员 7](#_Toc24531)

[3.3.3 项目团队外部沟通与协作模式 7](#_Toc16955)

[4 实施计划 7](#_Toc13420)

[4.1 风险评估及对策 7](#_Toc14091)

[（1）工程／规模／进度上的风险 7](#_Toc29517)

[（2）技术上的风险 7](#_Toc17176)

[（3）其它：开发中可能出现新的需求等。 7](#_Toc4913)

[4.2 工作流程 8](#_Toc15589)

[4.3 总体进度计划 8](#_Toc10368)

[5 支持条件 8](#_Toc18301)

[5.1 内部支持 8](#_Toc12621)

[6 关键问题 9](#_Toc6901)

[参考文献 9](#_Toc13924)

## 1 引言

### 1.1 编写目的

为了保证项目团队按时保质地完成项目目标，便于项目团队成员更好地了解项目情况，使项目工作开展的各个过程合理有序，因此以文件化的形式， 把对于在项目生命周期内的工作任务范围、各项工作的任务分解、项目团队组织结构、各团队成员的工作责任、团队内外沟通协作方式、开发进度、经费预算、项目 内外环境条件、风险对策等内容做出的安排以书面的方式，作为项目团队成员以及项目干系人之间的共识与约定，项目生命周期内的所有项目活动的行动基础，项目 团队开展和检查项目工作的依据。

为了更好地完成项目，便于老师和组内同学了解项目。

### 1.2 背景

项目的名称：图书推荐系统

项目的委托单位：第八组

项目的任务提出者：共同商议确定

项目建设背景：随着互联网的快速发展，人们越来越倾向于在网上查询与阅读书籍，尤其是在大学领域，读者对书籍和文献的需求量极大。然而，图书馆的图书种类繁多，读者往往会迷惑于该寻找哪一本适合自己的书籍。因此，我们决定设计一个《图书推荐系统》。该系统会根据图书评分状况，向其推荐适合他阅读相关书籍。这样，面对不熟悉的学科领域，图书推荐系统就可以为读者推荐图书同时也可以引导读者读书，避免了读者在众多图书中盲目寻找的麻烦。而对于没有明确需求的读者，图书推荐系统也可以为他们提供个性化推荐，为读者提供他们感兴趣的图书，这样一来，不但可以节约读者找寻书籍时间，还能提高他们的阅读兴趣，增加读者的阅读量，拓宽阅读面。

### 1.3 定义

协同过滤推荐算法：算法通过对用户历史行为数据的挖掘发现用户的偏好，基于不同的偏好对用户进行群组划分并推荐品味相似的商品。协同过滤推荐算法分为两类，分别是基于用户的协同过滤算法(user-based collaborative filtering)，和基于物品的协同过滤算法(item-based collaborative filtering)。

欧几里得距离：欧几里得度量（也称欧氏距离）是一个通常采用的距离定义，指在m维空间中两个点之间的真实距离，或者向量的自然长度（即该点到原点的距离）。在二维和三维空间中的欧氏距离就是两点之间的实际距离。

归一化处理：归一化方法有两种形式，一种是把数变为（0，1）之间的小数，一种是把有量纲表达式变为无量纲表达式。主要是为了数据处理方便。

### 1.4 参考资料

CSDN：协同过滤算法（推荐算法）、归一化处理

道客巴巴：图书推荐系统的设计与实现

## 2 项目概述

### 2.1 项目目标

第一目标：建立图书分类系统

第二目标：通过评分、简介等记录进行图书协同过滤

第三目标：加入个性化图书推荐功能

### 2.2 产品目标与范围

该系统致力于为收集图书信息来为用户推荐最相关的书籍。对于用户，通过获取数据库用户信息、、图书信息等原始数据，快速分析出用户兴趣图书的集合，完成个性化推荐功能。对于管理者，该系统能完成读者查询个人信息、图书评分检索等功能。

### 2.3 假设与约束

本次项目的截止时间约为五月中旬，整体开发时间约为八个周，项目小组成员为四人，使用设备为笔记本电脑。

### 2.4 项目工作范围

前端与后端开发，数据库开发，数据库与系统的连接，调试与成品的测试使用。

### 2.5 应交付成果

2.5.1 需完成的软件

完成程序名称：图书推荐系统

程序内容：源程序、数据库对象创建语句、可执行程序、支撑系统的数据库数据、配置文件、第三方模块、界面文件、界面原稿文件、声音文件、安装软件、安装软件源程序文件等等。

所用编程语言：Python

软件开发方式：

1）系统总体设计

2）系统详细设计

3）编码

4）测试

2.5.2 需提交用户的文档

需求规格说明书，概要设计说明书，详细设计说明书等。

2.5.3 须提交内部的文档

软件项目开发计划书，设计文档，项目报告等。

### 2.6 项目开发环境

操作系统：windows10

开发工具：PyCharm

数据库系统：mySQL

### 2.7 项目验收方式与依据

项目完成后首先由开发小组内部进行测试和验收，确定无误后进行演示，随后结束项目开发验收的整个流程，项目验收依据为项目文档，如项目开发计划书、需求文档、设计文档和详细设计说明书等。

## 3 项目团队组织

### 3.1 组织结构

组长：负责项目开发进度的跟踪，开发任务的分配，进行项目总结，解决人员问题

前端开发：负责项目的前端开发

后端开发：负责项目的后端开发

数据库人员：负责数据库部分的具体实现

测试人员：负责对已经完成的代码部分进行测试反馈

### 

### 3.2 人员分工

|  |  |
| --- | --- |
| 姓名 | 分工 |
| 梁爽 | 项目团队管理，后端开发 |
| 黄英伦 | 分析新功能、系统测试 |
| 杜家琛 | 需求分析、数据库设计 |
| 杨祥睿 | 前端设计编写、模块规划 |

### 3.3 协作与沟通

3.3.1 项目团队内部协作

协作模式：团队线上开发

沟通方式：线下和线上相结合，使用诸如github和QQ等软件工具

沟通频次：每周不少于2次

沟通成果记录办法：电子文档或纸质文档

3.3.2 项目接口人员

梁爽：负责内部人员组织交流

3.3.3 项目团队外部沟通与协作模式

本项目主要为项目内部项目

## 4 实施计划

### 4.1 风险评估及对策

（1）工程／规模／进度上的风险

对于小团队来说规模较大，规模估算可能存在误差；就目前确定的规模而言，开发时间和进度较为紧张；尽量提前一段时间完成阶段开发，避免测试时代码错误，若未能按时完成，则暂时推迟一到两天并加大工作量完成未完成部分。

（2）技术上的风险

数据集庞大，需要强大的计算能力支持。

（3）其它：开发中可能出现新的需求等。

### 4.2 工作流程

使用瀑布法工作流程。

### 4.3 总体进度计划

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 起止时间点 | 所需资源 | 完成工作 | 应提交成果 | 检查点 |
| 2-3周 | 相关的需求和开发数据 | 项目计划和需求分析 | 项目需求报告和项目开发计划书 | 完成报告和计划书编写 |
| 3-4周 | 相关开发经验和工具 | 项目开发前期准备 | 开发环境配置和相关工具和数据的准备 | 完成前期准备 |
| 4-6周 | 开发人员和相关技术支持 | 大致完成项目整体的编写 | 已经完成的项目初期版本 | 项目大体编写完成 |
| 7周 | 测试人员和相关测试经验支持 | 完成后期收尾和测试等工作 | 项目的最终成果 | 开发结束，完成项目测试 |
| 8周 | 开发流程数据汇总 | 项目汇报 | 项目成品 | 完成项目汇报 |

## 5 支持条件

所需要的编译环境：PyCharm，MySQL，Django

所需要的设施：电脑（windows10系统）

### 5.1 内部支持

5.1.1 立项阶段：小组所有成员完成项目立项。

5.1.2 需求分析阶段：小组所有成员根据所确立的项目进行需求分析，完成需求分析报告与软件项目开发计划书。

5.1.3 项目开发阶段：

梁爽完成系统需求，项目计划，项目团队管理，检查进度。

杜家琛进行模块分配、数据库设计。使用MySQL。

黄英伦分析新功能、软件框架扩展、代码。使用PyCharm。

杨祥睿进行前端设计编写，使用PyCharm。

5.1.4 项目测试阶段：由全体组员分别使用白盒测试和黑盒测试完成。

## 6 关键问题

协同过滤推荐算法是推荐系统中最基本的算法。它分为基于用户的协同过滤算法和基于物品的协同过滤算法。

基于物品的协同过滤算法主要分为两个步骤。

1）计算物品之间的相似性。

2）根据物品相似度与用户历史行为的，向用户提供推荐列表。

第一步骤中的关键点是计算项之间的相似度。除了使用基于内容的相似性，它是计算有多少类似的物品，而是看喜欢i的用户中，有多少人喜欢j的，因此计算是基于用户。该兴趣一般都比较确定和不容易改变。当一个用户都喜欢的物品，我们通常可以认为，这两个物品可能属于同一类别。令N(i)表示购买物品i的用户数，则物品i和物品j的相似度可以用公式来计算。

第一步的时间复杂度的改进方法：以UserCF类似，我们可以创建一个用户，项目查找表，通过计算，认为用户有beenhave这些项目之间的相似性时，它可以保证计算的相似性。这样能够保证相似度是有用的，而不用对那些零（可靠地稀疏矩阵）花费大量的计算量。

第一步相似的改进方法1：如果按上述公式计算的相似性，可以发现，这个物品我和受欢迎的物品j之间的相似性是非常高的，因为流行的读数偏高，所以基本上每个人都会买它。具有较高的知名度的商品不太区分的，所以我们需要惩罚流行物品j的权重[10]。

第一步相似性改进方法2：需要惩罚用户的活动。如果用户不活跃，只有购买的图书数量有限，那么这些书很可能在计算项目中感兴趣的一个或两个区域的相似性是有益的，但如果一个书店卖家提供折扣，如果你买90%打折的书籍，然后赚取差价，那么用户的行为对计算物品的相似性不会有任何作用，因为90％的书肯定会涵盖了很多的范围，所以你应该惩罚用户的活动，可以采取第一个方法。

第一步相似性改进方法3：物品的相似性的归属。规范化不仅提高了建议的准确性，还增加了建议的覆盖范围和多样性。例如，在京东上，用户的爱好种类繁多，有相机爱好者、耳机爱好者、电脑爱好者等。很少有人说爱好集中在一个类别中。假设有两种类型的A和B.A类之间的相似性是0.5，B类之间的相似性是0.8，A和B之间的相似性是0.2。当用户购买A类的5本书和B类的5本书后，我们必须向用户提供推荐。如果我们按照前面的方法并按相似性排序，那么推荐的方法应该是B类项目。即使B类别较低，它仍然优于A类。为了高相似性，所以相似性的相似性应该基于类别，因此A的相似性为1，B的相似性也为1 ，以便排序后推荐的A和B产品具有更高的准确性，覆盖范围和多样性。

第二步则比较简单，计算物品与用户已买物品的相似度（权重和），然后根据相似度排序选出topN。

ItemCF在实际系统中运用的比较多，主要有两个优点：

1）item-item表相比如user-user表要小的多，处理起来比较容易。

2）itemcf很容易为推荐提供理由，比如提前进行数据挖掘，可提高可靠性，改善用户与推荐系统的相互作用，并进一步加强定制推荐前推荐数据挖掘等。

基于物品的协同过滤算法与用户的协同过滤算法相比，基于用户的协同过滤算法有两大缺点。

1）随着网站用户数量的增加，计算用户数量的相似性就更加困难了。计算的时间复杂度和空间复杂度与用户的增长基本成平方关系。

2）基于用户的合作过滤算法很难对推荐的结果作出解释和建议。

基于物品的协作过滤算法是“目标用户”，用于查找与其喜欢的项目类似的项目。从实际情况角度出发，在对于本课题中的需求——图书推荐来说，每个用户对于个性化推荐书籍的需求都比较强烈，此时采用基于物品的协同过滤算法就可以更为充分地挖掘用户的兴趣领域，并且很容易根据用户的历史数据来对推荐结果做出解释，从而使用户更加信任系统做出的推荐结果。

## 参考文献

《C++ primer》[美]Stanley B.Lippman 著，王刚 杨巨峰 李忠伟改编

《计算机算法设计与分析》（第五版） 王晓东著

《数据库系统概念》(美国)希尔伯沙茨著

《Android应用开发揭秘》 杨丰盛著

《Android studio应用开发实战详解》 王翠萍著

《MySQL技术内幕：SQL编程》 姜承尧著