****

**本 科 毕 业 设 计（论文）**

题 目：基于Android的电子书推荐系统的设计与实现

|  |  |
| --- | --- |
| 学生姓名： | 刘亚辉 |
| 学 号： | 1907010317 |
| 专业班级： | 计算1903 |
| 指导教师： | 张红霞 |

2023年6月10日

学位论文原创性声明

本人所提交的学位论文人工智能领域数据治理平台的设计与实现，是在导师的指导下，独立进行研究工作所取得的原创性成果。除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的研究成果。对本文的研究做出重要贡献的个人和集体，均已在文中标明。

本声明的法律后果由本人承担。

论文作者（签名）： 指导教师确认（签名）：

年 月 日 年 月 日

学位论文版权使用授权书

本学位论文作者完全了解中国石油大学（华东）有权保留并向国家有关部门或机构送交学位论文的复印件和磁盘，允许论文被查阅和借阅。本人授权中国石油大学（华东）可以将学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或其它复制手段保存、汇编学位论文。

保密的学位论文在\_\_\_\_\_\_\_年解密后适用本授权书。

论文作者（签名）： 指导教师（签名）：

年 月 日 年 月 日

摘 要

**Abstract**

目 录

[第1章 绪论 1](#_Toc135686272)

[1.1 选题目的及研究意义 1](#_Toc135686273)

[1.2 国内外研究现状 1](#_Toc135686274)

[1.2.1国内外的电子书推荐平台 1](#_Toc135686275)

[1.2.2 国内外的推荐系统的研究 2](#_Toc135686276)

[1.3 主要研究内容 4](#_Toc135686277)

[第2章 相关理论与技术 4](#_Toc135686278)

[2.1 Android 4](#_Toc135686279)

[2.2 Flask 5](#_Toc135686280)

[2.3 MySQL 5](#_Toc135686281)

[2.4 TensorFlow 6](#_Toc135686282)

[第3章 基于协同过滤的混合推荐算法 6](#_Toc135686283)

[3.1 推荐算法原理 6](#_Toc135686284)

[3.2 基于内存的协同过滤算法 7](#_Toc135686285)

[3.2.1 基于用户的协同过滤算法 7](#_Toc135686286)

[3.2.2 基于物品的协同过滤 9](#_Toc135686287)

[3.3 加权混合 10](#_Toc135686288)

[第4章 系统需求分析 11](#_Toc135686289)

[4.1 系统功能用例 12](#_Toc135686290)

[4.2 功能需求分析 12](#_Toc135686291)

[4.3非功能需求分析 12](#_Toc135686292)

[第5章 系统设计 14](#_Toc135686293)

[5.1 系统体系架构 14](#_Toc135686294)

[5.2 功能模块设计 16](#_Toc135686295)

[5.2.1 Android客户端功能模块设计 16](#_Toc135686296)

[5.2.2 系统后台管理各功能模块设计 17](#_Toc135686297)

[5.4 数据库设计 17](#_Toc135686298)

[5.5 总结 20](#_Toc135686299)

[第6章 系统实现 21](#_Toc135686300)

[6.1 系统开发环境 21](#_Toc135686301)

[6.2 系统核心功能模块实现 21](#_Toc135686302)

[6.2.1 登录注册模块 21](#_Toc135686303)

[6.2.2 电子书推荐模块 21](#_Toc135686304)

[6.2.3 热门图书模块 23](#_Toc135686305)

[6.2.4 搜索模块 23](#_Toc135686306)

[6.2.5 个人中心模块 24](#_Toc135686307)

[6.2.6 购物车模块 24](#_Toc135686308)

[6.2.7 管理模块 25](#_Toc135686309)

[第7章 系统测试 27](#_Toc135686310)

[7.1 登录注册模块 27](#_Toc135686311)

[7.2 电子书推荐模块 28](#_Toc135686312)

[7.3 搜索模块 28](#_Toc135686313)

[7.4 购物车模块 28](#_Toc135686314)

[7.5 个人中心模块 29](#_Toc135686315)

[7.6 电子书详情模块 29](#_Toc135686316)

[7.7 管理模块 29](#_Toc135686317)

[第8章 总结与展望 30](#_Toc135686318)

[8.1 总结 30](#_Toc135686319)

[8.2 展望 31](#_Toc135686320)

[致谢 31](#_Toc135686321)

[参考文献 32](#_Toc135686322)

# 绪论

## 选题目的及研究意义

随着移动互联网的快速发展，数字阅读已经成为人们生活中不可或缺的一部分，电子书市场也随之快速发展。但是，用户在海量电子书中寻找自己喜欢的作品仍然是一件很麻烦的事情。因此，如何为用户提供更加便捷、智能的电子书推荐服务是当前的研究热点之一。

本文将基于Android平台，研究并设计一款电子书推荐系统，旨在为用户提供便捷、智能的电子书推荐服务。具体研究内容包括：用户需求分析、系统架构设计、推荐算法研究、系统实现与测试等方面，对于电子书推荐系统的研究和实现具有一定的参考价值和应用意义。

本文旨在实现基于Android平台的电子书推荐系统，通过协同过滤算法，为用户提供更加个性化、精准的电子书推荐服务，优化用户体验。同时，本文还将研究电子书推荐系统的设计与开发方法，对于电子书推荐系统的研究和实现具有重要的参考价值

## 国内外研究现状

电子书市场快速发展，国内外研究和应用了很多电子书推荐系统，主要依赖于推荐算法的研究和技术的支持。基于协同过滤算法的推荐系统应用广泛，但存在推荐精度不高、用户个性化需求不能得到满足等问题。因此，如何提高电子书推荐系统的精度和个性化服务水平是当前研究的热点之一。

### 1.2.1国内外的电子书推荐平台

Amazon Kindle是世界上最早的电子书平台之一，采用了基于协同过滤算法的推荐系统。通过记录用户的阅读历史、评分和书评等信息，为用户推荐个性化的电子书和杂志。Google Play Books使用基于机器学习的推荐系统，通过分析用户的阅读偏好、搜索历史等信息，为用户推荐个性化的电子书和杂志。iBooks使用基于协同过滤算法的推荐系统，通过记录用户的阅读历史、评分和书评等信息，为用户推荐个性化的电子书和杂志。京东阅读使用基于深度学习的推荐系统，通过分析用户的阅读行为、社交网络和兴趣爱好等信息，为用户推荐个性化的电子书和杂志。微信读书使用基于深度学习的推荐系统，通过分析用户的阅读行为、社交网络和兴趣爱好等信息，为用户推荐个性化的电子书和杂志。

总的来说，随着推荐技术的发展和进步，电子书推荐平台的个性化推荐能力不断提升，为用户提供更加精准、个性化的阅读体验。电子书推荐平台的发展历程为推荐系统的研究和应用提供了重要的经验和启示。

### 1.2.2 国内外的推荐系统的研究

（1）在90年代，推荐算法主要采用协同过滤算法[1]，其中包括基于用户（UserCF）和基于物品（ItermCF）的算法。这些算法的优点在于模型简单，计算速度快。然而，这些模型的缺点在于行为数据非常稀疏，导致模型欠拟合。

（2）自2006年起，基于矩阵分解的协同过滤算法成为了一种新的推荐算法。这种算法的优点在于，它类似于embedding技术，可以改善模型数据的稀疏性，从而为没有行为数据的物品和用户做出推荐。然而，这种算法的缺点是它只能利用行为数据这一个特征，没有充分利用其他信息。Koren等[2]提出了一种名为SVD++的新型矩阵分解算法，该算法能够有效地解决数据稀疏性和数据冷启动等问题。Salakhutdinov等[3]则提出了一种概率矩阵分解算法，能够更好地处理非线性变换和噪声的影响。王婵在传统的协同过滤推荐中加入了权重,因子，使用权重组合来进行算法混合;王睿等混合了项目的横向和纵向相似度，将不同层次的相似度进行配对结合形成混合推荐,改善了推荐精度。

（3）2017年，阿里巴巴提出了DIN[5]模型，其场景出发点是认为将用户的embedding向量固定为一个值很难发现用户和目标商品之间的相关性。如果用户的embedding向量能随着待推荐商品的变化而变化，就能更精确地把握用户的兴趣向量，从而提高推荐准确度。DIN的baseModel将用户embedding向量、历史物品的平均池化embedding向量、待推荐目标物品embedding向量和场景信息embedding向量连接成一个长向量，并送入MLP进行训练。DIN的创新点在于引入了激活单元，即历史物品和待推荐物品之间的相关性权重，这是因为用户的历史物品和目标物品之间的关联程度不同，其权重应该是不同的。因此，使用平均池化会损失一些信息，因此我们先计算目标商品和历史商品之间的权重，然后再使用权重进行池化，从而有效地提取出用户的兴趣。

朱晓东等[6]则进一步优化了DIN模型，提出了Deep Interest Evolution Network（DIEN）模型，能够更好地捕捉用户的兴趣演化过程。

DIN模型的优点在于引入了用户向量随商品变动的机制，用户向量更加灵活，兴趣更能提取。然而，该模型的缺点在于没有完全捕捉用户历史物品的时间序列特性，并且模型存在冷启动问题。

（4）YouTubeNet[7]是2016年Google提出的一种经典的模型，用于预测用户对视频播放时长，已被知乎、山烛视频等知名互联网公司采用。该模型的创新点在于其离线得到用户embedding向量和物品embedding向量的设计非常巧妙。具体来说，该模型使用用户的历史序列embedding向量做avg-pooling，将其与用户属性信息结合，送入3层神经网络，最后使用softmax预测目标视频。该模型使用召回层来快速召回物品，使用Faiss库进行实现。在线上模型部分，该模型使用加权逻辑回归作为最终输出，预测时使用指数进行输出，指数输出刚好是目标视频的观看时长。在上市公司信息推荐应用中，该模型的目标是预测用户对公司信息的阅读时长，从而让用户深入阅读公司信息以降低投资风险，并让用户接触到更多感兴趣的上市公司。

（5）DSIN模型[8]是阿里巴巴公司对DIEN模型的优化，引入了session的概念和attention机制。这些改进使得DSIN能够更好地挖掘用户兴趣随时间变迁的规律。DSIN模型将用户历史行为数据按照一定时间段(如30分钟)拆分成多个session，然后提取每个session的兴趣向量。接着，使用attention机制获得每个兴趣向量在全局中的embedding向量，以此来准确抓住兴趣随时间变化的规律。此外，DSIN模型还使用了BI-LSTM模型，以处理序列间的内在相关性。

然而，DSIN模型的复杂性较高，而且BI-LSTM是串行模型，无法高效并行计算，因此时效性较差。针对这些问题，可以进行以下改进。首先，可以将位置信息叠加到attention后的embedding向量中，一起进行训练，以便模型能直接学习到位置信息，且能够并行计算，这种方法可以参考ViT模型在图片分类中的处理技巧。其次，对于session的划分，可以采用更加智能的方式，而非固定的时间窗口，以更好地抓住用户兴趣变化的点。最后，可以进行降噪处理，比如根据近期物品和历史物品的相关性计算，过滤掉用户无意间的噪声点击，或将其作为第二兴趣特征进行平行训练，具体处理方式需要根据业务特点进行确定。

可以看出，个性化推荐技术虽然发展已久，但对推荐算法存在问题的研究一直在继续。没有一个算法是十全十美的，都有自己的适用场景，因此在对个性化推荐算法进行改进时需要因势利

## 主要研究内容

在上述背景下，本文将研究一个基于安卓的电子书推荐系统，该系统旨在根据用户的兴趣和行为推荐合适的电子书，提高用户的阅读体验。本文将研究内容分为四个层级，即数据采集层、数据处理层、推荐算法层和应用服务层，针对不同的层级，采取不同的技术选型和手段进行研究。

数据采集层负责收集用户阅读行为和电子书元数据信息，数据处理层对收集到的数据进行清洗、预处理和特征提取等操作，为推荐算法提供符合要求的数据。

推荐算法层使用基于物品和用户的协同过滤算法，根据用户画像和电子书元数据信息，对用户进行电子书推荐。

应用服务层通过友好的安卓界面向用户展示推荐结果，并提供用户使用。该部分为系统最上层结构，通过应用服务层最终实现电子书推荐的价值。系统采用Flask框架，设计基于MVC模式的前后端分离系统，保证推荐实时性和准确性。

# 相关理论与技术

本章将介绍该系统中使用到的关键技术，包括用于客户端的Android，用于服务器的Flask，用于数据存储的MySQL，用于深度学习的框架TensorFlow和基于物品和用户的协同过滤算法。下面将针对上述技术讲述其主要架构或原理以及优势。

## 2.1 Android

Android是由Google公司开发的一种基于Linux操作系统的移动设备操作系统,有着开放性、免费、多样化、易于开发的特点。Android系统提供了丰富的应用程序框架，包括Activity、Service、BroadcastReceiver、ContentProvider等四大组件，它们可以协同工作，完成各种应用程序功能。并采用了资源管理机制，可以根据不同的设备类型、屏幕分辨率、语言等来管理和加载相关资源文件，保证应用程序在不同的设备上具有较好的兼容性和适配性。

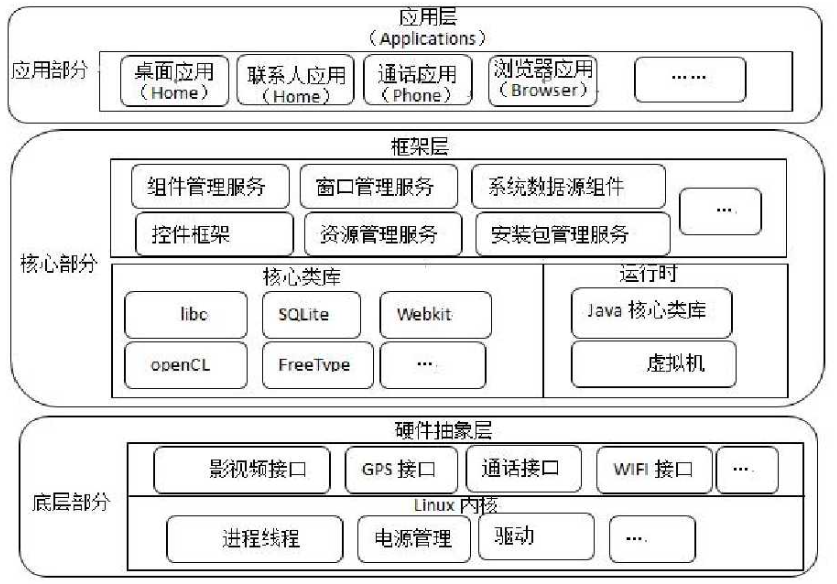


图2-1 安卓系统架构

## 2.2 Flask

Flask是一种轻量级的Python Web框架，具有简单、灵活、易于扩展的特点。其主要架构是基于Werkzeug和Jinja2两个库来实现的。Flask提供了丰富的扩展机制，可以轻松集成其他Python库或框架，同时还支持自定义扩展，满足不同的开发需求。

## 2.3 MySQL

MySQL是一种开放源代码的数据库管理系统，具有可移植性、高效率、多种编码规则支持、多种连接方式、多种存储引擎支持等特点。MySQL的源代码一般采用C语言编写，可以在多种操作系统中使用，支持库嵌入和独立执行，提供多种工具进行数据库的维护、优化、压缩等操作。MySQL支持大型数据库，能够高效地处理大量数据。同时，MySQL还提供多种存储引擎，可以满足不同应用场景下的需求。由于其代码开放、速度快、体积小、节约成本等优点，MySQL被很多开发者所青睐。

## 2.4 TensorFlow

TensorFlow是由Google Brain团队开发的一款开源机器学习框架，具有可移植性、高效率、多种编码规则支持、多种连接方式、多种存储引擎支持等特点。TensorFlow可以在多种操作系统中使用，支持库嵌入和独立执行，提供多种工具进行数据库的维护、优化、压缩等操作。TensorFlow支持大型数据库，能够高效地处理大量数据。同时，TensorFlow还提供了多种深度学习模型的实现，可以满足不同应用场景下的需求。由于其代码开放、速度快、体积小、节约成本等优点，TensorFlow被很多开发者所青睐。

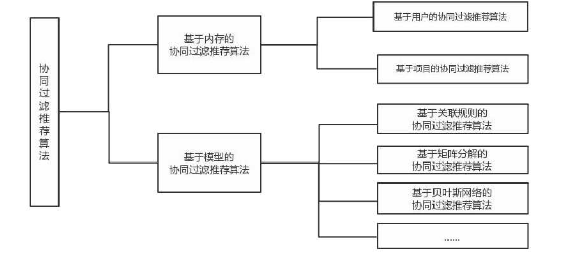
# 第3章 基于协同过滤的混合推荐算法

本文将探讨基于Android的电子书推荐系统的设计与实现。该系统旨在为用户提供最新、最感兴趣的电子书内容，满足用户的阅读需求。在实现过程中，我们需要考虑如何通过推荐算法层面的设计，收集、处理用户信息，建立和更新用户的兴趣模型，以及解决推荐系统的冷启动问题等挑战。

在系统的核心部分，即个性化推荐算法和用户模型的设计方面，目前已有成熟的技术可供选择。例如，基于TensorFlow框架的协同过滤推荐项目能够很好地帮助我们实现最基本的推荐目的。通过分析用户历史行为和相似用户的行为数据，该算法能够准确地分析用户兴趣，从而推荐相关的电子书内容。

## 3.1 推荐算法原理

协同过滤推荐算法（Collaborative Filtering Recommendation），简称CF算法，是目前应用最为广泛的推荐算法之一。其基本原理是通过分析用户过去的兴趣爱好，来发现与用户品味或习惯相似的“邻居”群体，从而为用户推荐邻居群体都偏好的商品或内容。以电商平台为例，由于商品数量远远超过用户数量，如何增加推荐商品被用户喜爱的可能性就成为了一个关键问题。协同过滤算法通过分析用户的特征和行为信息，寻找用户的“邻居”群体，从而预测用户的兴趣，有的放矢地向用户推荐可能符合偏好的商品或内容。该算法操作简单、性能良好，因此备受好评。根据相似度计算的主体不同，协同过滤推荐算法可以被分为基于内存和基于模型两大类。基于内存的算法通过相似度计算来寻找邻居群体，然后预测用户的兴趣。基于模型的算法则通过建立模型来分析用户的行为和兴趣，从而预测用户的兴趣。具体来说，基于内存的算法包括用户-用户协同过滤（User-Based Collaborative Filtering）和物品-物品协同过滤（Item-Based Collaborative Filtering）两种类型；而基于模型的算法包括矩阵分解（Matrix Factorization）和基于深度学习的模型（Deep Learning-based Models）等。该算法的具体分类如图2.3所示。



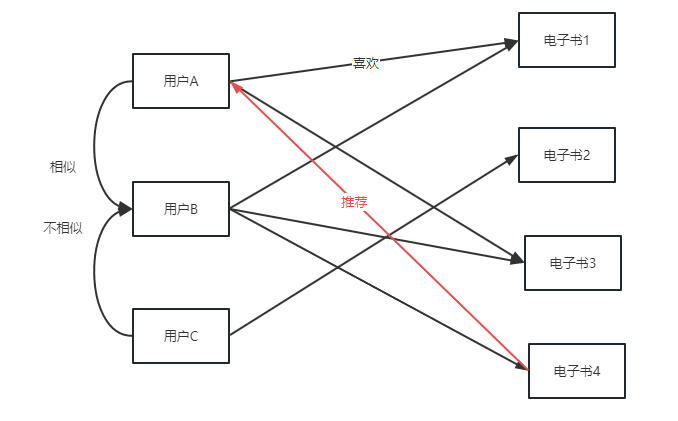
本文主要使用的是基于内存的协同过滤。

## 3.2 基于内存的协同过滤算法

### 3.2.1 基于用户的协同过滤算法

图2.4所示的是基于用户的协同过滤推荐算法的原理。该算法的核心思想是通过寻找用户的“邻居”群体，根据邻居群体的偏好来产生推荐。具体来说，该算法首先通过相似度计算来度量用户之间的相似性，然后找到与目标用户相似度最高的邻居群体，根据邻居群体对商品或内容的评分或者喜好程度来预测目标用户的兴趣，从而有的放矢地向目标用户推荐可能符合偏好的商品或内容。

以电子书为例，假设用户A喜欢电子书1和电子书3，用户B喜欢电子书1、电子书3和电子书4，用户C只喜欢电子书2。根据基于用户的协同过滤算法的原理，我们可以计算用户之间的相似度，发现用户A和B有两个共同喜欢的节目，可以认为他们是较为相似的用户；而用户C和用户A、用户B的喜好没有重合，相似度较低。当需要给用户A做推荐时，我们可以根据用户B的喜好，向用户A推荐用户B喜欢而用户A未接触过的节目，比如电子书4。



基于用户的协同过滤的基本操作步骤如下：

（1）根据用户的历史行为构建物品评分表。这个表是由用户历史行为的评分组成的，例如购买、收藏或评分。其目的是为了帮助我们了解用户对产品或的喜好程度。

（2）选择一个目标用户并计算该用户与其他用户之间的相似度。这里我们使用了修正的余弦相似度计算方法，因为传统的余弦相似度计算方法忽略了维度的差异。在修正的方法中，我们首先去除了评分的均值，然后计算修正后的余弦相似度。这样做可以更好地衡量用户之间的相似程度。

其中，表示用户u和用户v都评分过的项目集合，而用户u和用户v评分的均值则分别用与表示。

（3）为用户产生推荐。当目标用户的最近邻选取完后，就可以根据基于协同过滤算法的核心思想，具有相似兴趣偏好的用户，其对信息的需求也存在一定的相似性，来对目标用户对未知项目的评分进行预测，并最终将推荐结果反馈用户。[8]加权平均值公式如下:

其中，分别为用户u与用户v项目的评分的平均值。

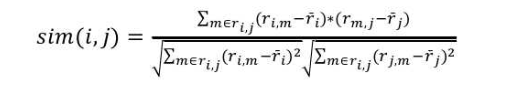
### 3.2.2 基于物品的协同过滤

基于物品的协同过滤推荐算法是根据物品之间的相似程度来为用户产生推荐。如果一个用户喜欢某个物品，那么与该物品相似的其他物品也可能会受到用户的喜欢。以电子书为例，如果一个用户喜欢看电子书1，那么与电子书1相似的另一个电子书3也可能会受到用户的喜欢。

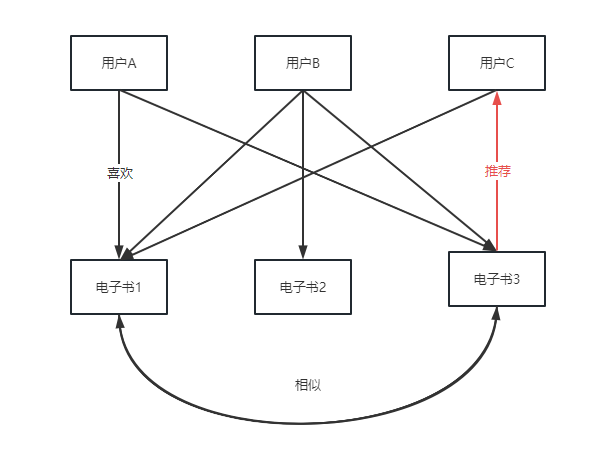
基于物品的协同过滤推荐算法的基本原理是通过计算物品之间的相似度来找到物品的“邻居”，进而为目标用户产生推荐。算法的具体步骤如下：

（1）构造用户-物品偏好矩阵，计算物品之间的相似度。通过用户对物品的评价，构建一个用户-物品偏好矩阵。

（2）将计算出的物品相似度进行排序，并选择与用户偏好物品相似度最高的前k个最近邻居物品。我们可以对计算出的物品相似度进行排序，然后选择与用户偏好物品相似度最高的前k个最近邻居物品。这些物品与用户偏好的物品相似度高，因此可能会被用户接受。



（3）从这k个邻居物品中为目标用户产生推荐集。最后，我们可以从这k个邻居物品中为目标用户产生推荐集。这些推荐物品与用户偏好的物品相似度高，并且可能会被用户接受。



## 3.3 加权混合

加权型混合顾名思义即是为两个及两个以上的传统算法赋予不同的权重。首先需要在推荐场景中,用两个或两个以上的传统推荐算法计算出推荐结果,接着按照实际需求，根据传统算法所被赋予的不同权重，用每个算法的结果乘以权重，最后相加，即可得到混合算法的结果。在这一混合方式中，每个算法的权重如何赋值是很关键的。一般来说，每个算法都有自己的适用场景，所以在进行算法混合时通常将更适应当前推荐场景或更能明确用户偏好的传统算法确定为主算法,赋予更高的权重。加权型混合计算如下所示:

其中作为最终评分，分别是算法1、2、...、n的预测评分，是所赋予的各个算法的权重，且满足下列条件：

基于用户和项目混合的协同过滤算法,其特征在于,包括以下步骤:

步骤1:对用户-项目评分数据集进行整理,建立用户-项目评分矩阵U;

步骤2:计算皮尔逊系数,由上文中提到的公式3-3可以计算物品之间的相似度,并将相似度按照从大到小进行排序。

步骤3:根据物品之间的相似度排序,产生物品的“最近邻居N”,利用最近邻居N预测用户T对其他物品的评分;

步骤4:采用步骤2得出的皮尔逊系数,计算目标用户T与其他用户之间的相似度﹐并将相似度按照从大到小的顺序进行排序;

步骤5:根据用户之间的相似度排序,产生用户的“最近邻居K”,利用最近邻居K预测用户T对其他物品的评分;

步骤6:综合利用物品相似度和利用用户相似度得到的评分,如果目标用户对物品的评分预测既在“最近邻居N”中,又在“最近邻居K”中,则对预测评分进行加权计算。

确定或者优化权重系数可以采用交叉验证的方法，通过在训练集和验证集上进行多次实验，选取使得预测效果最优的权重系数。

具体的步骤如下：

（1）将数据集分为训练集和验证集，通常采用 5 折交叉验证的方法，即将数据集分为 5 个子集，每次将其中一个子集作为验证集，其余 4 个子集作为训练集，依次进行实验，得到 5 个预测效果，取平均值作为最终结果。

（2）在训练集上使用基于用户的协同过滤算法和基于物品的协同过滤算法分别预测出评分。

（3）在验证集上计算加权混合算法的预测评分，根据预测效果选择最优的权重系数。可以采用均方根误差（RMSE）等指标来评估预测效果，公式如下：

其中是实际评分，是预测评分，N是样本数。

（4）重复2和3步骤，尝试不同的权重系数，选择使得预测效果最优的权重系数。

得到最优的权重的系数后通过公式3-5，即可得到加权混合后的预测评分。

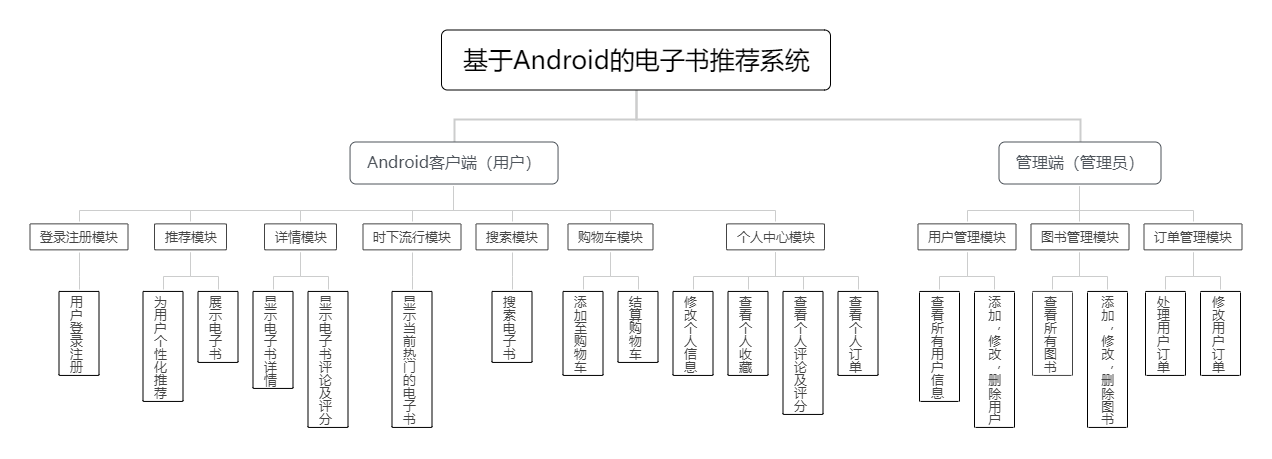
步骤7:据最近邻居X的排序,按照预测评分从大到小的排序输出top-N推荐项目集。

# 第4章 系统需求分析

系统需求分析是系统开发的首要阶段，系统设计、系统编码、系统测试与系统维护阶段的工作都会以系统需求分析的结果为基础。本章节从功能和非功能两个角度对系统需求进行了分析。

## 4.1 系统功能用例

## 4.2 功能需求分析



（1）电子书推荐。系统需要根据用户的历史浏览记录、搜索记录、购买记录等，推荐相关的电子书。推荐结果应该能够满足用户的需求，提高用户的购买率和满意度。

（2）电子书搜索。用户可以通过关键词搜索电子书，系统提供快速、准确的搜索结果。系统支持模糊搜索、关键词联想等功能，提高用户的搜索效率。搜索结果包含电子书的基本信息，例如书名、作者、价格、评分等。

（3）电子书购买。用户可以将电子书加入购物车，方便批量购买。系统还需要提供订单管理功能，用户可以查看订单状态、退货、评价等。

（4）个人信息查询和管理。用户可以查看个人信息，例如个人资料，个人收藏，个人订单等。用户可以修改个人信息，例如昵称、头像、密码等。系统需要保障用户信息的安全性和隐私性，防止信息泄露。

（5）用户、图书、订单管理。系统需要提供用户管理功能，管理员可以查看、添加、修改、删除用户信息。

系统需要提供图书管理功能，管理员可以查看、添加、修改、删除图书信息。

系统需要提供订单管理功能，管理员可以查看、添加、修改、删除订单信息。

管理员需要设置权限，保证系统的安全性和稳定性。

## 4.3非功能需求分析

系统的非功能性需求是指为了满足用户的业务需求，除了功能需求之外，软件产品必须具备的特性。这些特性包括安全性、可靠性、健壮性等，因此非功能性需求的阐述应该从多方面展开，而不应该单单局限于系统的并发量、响应速度等方面。在系统工程及需求工程中，非功能性需求通常从性能、易用性、可靠性、可拓展性、可维护性和安全性六个方面来进行论述。

表3-2 系统非功能性需求及其说明

|  |  |
| --- | --- |
| 非功能性需求 | 需求说明 |
| 性能 | 系统必须快速响应用户的请求，不超过3秒钟，定位系统从点击到第一个界面显示出来所需要的时间不得超过300毫秒，在非高峰时间根据编号和名称特定条件进行搜索，可以在3秒内得到搜索结果等。 |

|  |  |
| --- | --- |
| 易用性 | 指系统从用户使用角度的友好程度，即系统界面简洁、操作简单、重点突出、排版工整、导航清晰 |
| 可靠性 | 系统必须对输入进行提示，并对数据进行检查，以避免数据异常。系统的健壮性应该强，能够处理系统运行过程中出现的各种异常情况，包括人为操作错误、输入非法数据、硬件设备失败等，并能够正确处理和回避这些异常情况。50个字概括 |
| 可拓展性 | 可以适应不断变化的需求和规模，实现高效的系统扩展和升级。实现可拓展性需要采取模块化设计、灵活的架构和可配置的组件等实践。 |
| 可维护性 | 可维护性是软件系统易于维护和修改的性质，通过采取良好的实践提高可维护性，如文档记录、模块化设计、清晰的代码结构、良好的命名和注释、适当的测试和调试等。 |
| 安全性 | 需要实现严格的权限访问控制来保护数据不被非法访问和篡改，确保数据的机密性和完整性。同时，保护系统数据不受干扰，确保系统的稳定性和可用性也是安全性的目标。 |

# 第5章 系统设计

本章节在上文需求分析的基础上，设计系统架构，并将其划分为不同的功能模块，对每个模块详细介绍其架构和原理，在设计时充分考虑系统对可拓展性、可理解性和可维护性的要求，保证系统结构清晰易懂，便于日后维护与二次拓展开发。以下是系统的详细设计。

## 5.1 系统体系架构

在电子书推荐系统实际运行中，我们将面对大量的电子书数据的处理和用户行为的响应，而后期可能还要对隐含的缺陷做出修复并保持系统的更新。结合系统的业务需要和性能需求，我们对基于Android的电子书推荐系统做出了三个层级的划分以及对系统业务逻辑处理、显示、数据存储进行解耦，三层结构如下图5.1所示：

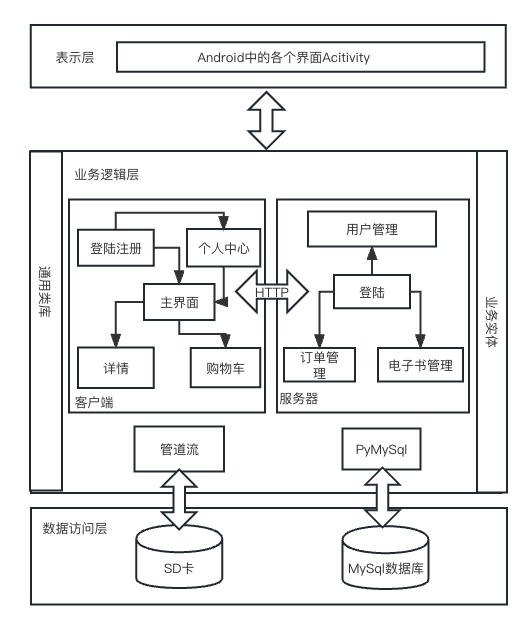


图4.1 系统的三层结构

(1) 表示层：本系统中的表示层与传统的三层架构中的表示层不同，不再由web页面负责。在本系统中，android客户端的操作交互由本层和业务逻辑层中的业务类共同负责，例如各个Button、View的监听响应。显示内容则是由各个Activity与服务器通信后接受的消息和数据。

(2) 业务逻辑层：本系统中的业务逻辑层分为Android的业务逻辑和电子书管理后台的业务逻辑两部分。Android的业务逻辑由Android提供的通用类库SDK接口给开发者完成具体的实现。电子书管理后台的业务逻辑除了处理管理员管理业务之外，更重要的是封装数据的格式成实体类，以便进行层间传输。此外，该层还处理来自Android应用和后台的业务逻辑。它向前承接了表示层，向后获取数据访问层的数据，处于整个系统的核心部分。

(3) 数据访问层：在基于Android的电子书推荐系统中，该层负责访问数据。该层使用pymysql库对数据库进行增删改查操作。为了方便操作，该层重新封装了与电子书、用户、用户行为相关的DAO接口和实现类。

在本系统中，体系架构决定了系统的总体组织、全局控制、通信协议、同步方式、数据存取等。经典的三层结构即从上到下的表示层、业务逻辑层、数据访问层在实际开发中得到广泛的运用。其优点在于有利于将各个层次分开，降低各层次之间的依赖性，提高系统的稳定性和可维护性。

## 5.2 功能模块设计

### 5.2.1 Android客户端功能模块设计

Android客户端模块包括：登陆注册模块、个性化推荐模块、时下流行模块、购物车模块以及个人中心模块等。

1. 登陆注册模块：用户初次使用或者注销重新使用时显示登录和注册界面。对于新用户来说点击注册进入到注册页面，已经注册的用户则直接进入系统。
2. 个性化推荐模块：个性化推荐页面是APP的主页面，界面以图文的形式展示推荐给用户的电子书。以列表的形式进行展示，用户点击后会跳转至详情页面。从上至下显示电子书封面，标题，作者，评分等，以及加购物车、评论、收藏等按钮，最下面以列表的形式展示所有用户对这本书等评论。
3. 购物车模块：购物车页面以列表的形式展示用户已经添加到购物车中的电子书信息，包括电子书封面、标题、作者、价格等信息。页面底部会显示购物车中所有电子书的总价，以便用户进行结算。用户可以对购物车中的电子书进行操作，如删除、修改数量等。在删除电子书时，需要进行二次确认，避免用户误操作。在修改数量时，需要进行价格计算，并及时更新购物车总价。在购物车页面底部通常会添加一个“去结算”按钮，用户点击后可以进入结算页面，查看购物车中的电子书信息并进行订单结算。结算页面需要显示电子书的价格、数量、总价等信息，并提供支付方式的选择。结算完成后，系统会生成订单并将订单信息保存到数据库。
4. 个人中心模块：在个人中心页面中，用户可以方便地查看个人信息、订单、收藏和评论等内容。用户可以通过点击修改个人信息入口，对个人信息进行修改，并进行校验。订单列表展示了订单编号、订单状态和订单总价等信息，并提供了查看订单详情和取消订单等操作入口。收藏列表展示了电子书的封面、标题和作者等信息，并提供了取消收藏等操作入口。评论列表显示了电子书的封面、标题、评论内容和评论时间等信息，并提供了修改和删除等操作入口，需要进行二次确认，以防误操作。
5. 热门电子书模块：该模块以六宫格的形式呈现，每个电子书需要展示封面、标题、作者、评分等信息，并提供查看详情、购买等操作入口。热门排行可按照电子书的热度排序。

### 5.2.2 系统后台管理各功能模块设计

管理端主要包括模块：用户管理、电子书管理和订单管理模块。

1. 用户管理模块：管理所有的用户信息，用户列表显示了所有用的UID等简略信息，管理员可以对用户进行信息查看或者修改、评论查看或修改、账号注销等操作。
2. 电子书管理模块：电子书管理页面以列表的形式展示电子书的信息，包括封面、标题、作者、出版社、价格和库存等，并提供编辑、删除等操作入口。管理员可以对电子书进行编辑操作，包括修改电子书的信息和上传、修改封面图片。

电子书管理模块需要提供搜索功能，支持模糊搜索和精确搜索。

（3）订单管理模块：管理员可以在该页面中查看订单列表，其中包括订单号、用户信息、电子书信息、订单金额和状态等信息。此外，该页面还应提供查看、编辑和删除订单的操作按钮

详细介绍系统的各个功能模块的设计，包括用户管理、电子书管理、推荐模块等。对于每个模块，说明其功能、输入、输出和调用方式等信息，并通过流程图或时序图等展示模块的交互过程

## 5.4 数据库设计

系统主要模块的功能设计完成后，根据数据库设计的基本原则[9]，包括原子性、原始性、演绎性和稳定性，接下来需要进行各个功能模块包含的数据关系的设计。本系统的数据库设计包括八个数据库表，它们分别是：用户信息表、用户收藏表、用户评论表、用户对书评分表、用户推荐信息表、电子书信息表、购物车信息表、订单信息表。

1. 用户信息表（user）：包含用户编号、用户名、昵称、密码、地址、手机号、性别、用户头像等字段。

表4.4.1用户信息表（user）

|  |  |  |  |
| --- | --- | --- | --- |
| 键名 | 类型 | 是否主键 | 说明 |
| uid | int | 是（自增） | 用户编号 |
| username | varchar（50） |  | 用户名 |
| nickname | varchar（50） |  | 昵称 |
| password | varchar(50) |  | 密码 |
| address | varchar(255) |  | 地址 |
| phone | varchar(100) |  | 手机号 |
| gender | varchar(50) |  | 性别 |
| avatar | varchar(255) |  | 用户头像 |

(2) 用户收藏表(user\_favorite): 包含用户名、电子书编号、标题、图书封面、作者、评分、收藏日期字段

表4.4.2 用户收藏表(user\_favorite)

|  |  |  |  |
| --- | --- | --- | --- |
| 键名 | 类型 | 是否主键 | 说明 |
| uid | Int | 否 | 用户编号 |
| bookid | varchar(30) |  | 电子书编号 |
| title | varchar(100) |  | 电子书标题 |
| author | varchar(100) |  | 电子书作者 |
| bookcover | varchar(255) |  | 电子书封面 |
| rating | varchar(30) |  | 电子书评分 |
| date | varchar(30) |  | 收藏日期 |

（3）用户评论表（user\_reviews）:包含用户编号、图书编号、评论内容等

字段。

表4.4.3 用户评论表（user\_reviews）

|  |  |  |  |
| --- | --- | --- | --- |
| 键名 | 类型 | 是否主键 | 说明 |
| uid | int | 否 | 用户编号 |
| bookid | varchar(30) |  | 电子书编号 |
| review | varchar(255) |  | 评论内容 |

（4）用户对书评分表（ratings）：包含用户编号、图书编号、评分字段。

表4.4.4 用户对书评分表（ratings）

|  |  |  |  |
| --- | --- | --- | --- |
| 键名 | 类型 | 是否主键 | 说明 |
| uid | int | 否 | 用户编号 |
| bookid | varchar(30) |  | 电子书编号 |
| rating | varchar(30) |  | 评分 |

（5）用户推荐信息表（user\_recom\_books）: 用户编号、图书编号字段

表4.4.5用户推荐信息表（user\_recom\_books）

|  |  |  |  |
| --- | --- | --- | --- |
| 键名 | 类型 | 是否主键 | 说明 |
| uid | int | 否 | 用户编号 |
| bookid | varchar(30) |  | 电子书编号 |

(6) 电子书信息表(books):包含电子书编号、标题、封面、作者、评分、价格、语言、出版时间、简介等字段

表4.4.6电子书信息表(books)

|  |  |  |  |
| --- | --- | --- | --- |
| 键名 | 类型 | 是否主键 | 说明 |
| bookid | varchar(30) | 是 | 电子书编号 |
| title | varchar(100) |  | 电子书标题 |
| author | varchar(100) |  | 电子书作者 |
| bookcover | varchar(255) |  | 电子书封面 |
| rating | varchar(30) |  | 电子书评分 |
| price | varchar(30) |  | 电子书价格 |
| language | varchar(30) |  | 电子书语言 |
| year | varchar(30) |  | 出版时间 |
| description | varchar(255) |  | 简介 |

（7）购物车信息表（cart）：包含电子书编号、标题、封面、作者、价格、用户编号、数量字段。

4.4.7 购物车信息表（cart）

|  |  |  |  |
| --- | --- | --- | --- |
| 键名 | 类型 | 是否主键 | 说明 |
| bookid | varchar(30) | 否 | 电子书编号 |
| title | varchar(100) |  | 电子书标题 |
| author | varchar(100) |  | 电子书作者 |
| bookcover | varchar(255) |  | 电子书封面 |
| count | varchar(255) |  | 购买数量 |
| price | varchar(30) |  | 电子书价格 |
| uid | int |  | 用户编号 |

（8）订单信息表（orders）：包含订单号、电子书编号、标题、封面、作者、价格、用户编号、数量、购买者姓名、收货地址、手机号、支付状态等字段

4.4.8订单信息表（orders）

|  |  |  |  |
| --- | --- | --- | --- |
| 键名 | 类型 | 是否主键 | 说明 |
| orderid | int | 是（自增） | 订单号 |
| bookid | varchar(30) |  | 电子书编号 |
| title | varchar(100) |  | 电子书标题 |
| author | varchar(100) |  | 电子书作者 |
| bookcover | varchar(255) |  | 电子书封面 |
| count | varchar(255) |  | 购买数量 |
| price | varchar(30) |  | 电子书价格 |
| uid | int |  | 购买者用户编号 |
| buyername | varchar(100) |  | 购买者姓名 |
| address | varchar(255) |  | 收货地址 |
| phone | varchar(100) |  | 手机号 |
| 支付状态 | int |  | 是否支付 |

## 5.5 总结

总结系统设计的关键点，强调系统的优势和特点

# 第6章 系统实现

## 6.1 系统开发环境

系统开发环境共包含以下几个方面：

1. 硬件设施：AMD Ryzen 7 3750H 2.30 GHz处理器，16 GB 2666 MHz DDR4内存,GTX1660Ti图形处理器；
2. 操作系统：Windows 10 21H2；
3. 集成开发环境: PyCharm 2023.1 (Professional Edition)，Python 3.8，Android Studio 2022.12，Java 15.0.1；
4. MySQL：5.6；
5. Flask: 2.2.3；
6. TensorFlow：2.12.0；

## 6.2 系统核心功能模块实现

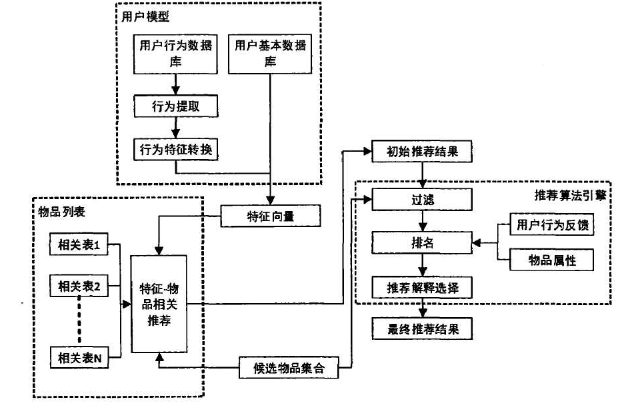
### 6.2.1 登录注册模块

（1）注册：用户在系统首页点击“注册”按钮后跳转至注册页面。用户在注册页面输入帐号、密码等信息，页面进行表单验证，验证用户输入是否合法。用户输入完成后，点击注册按钮，前端控制器使用MyUser对象接收表单数据，并依次调用RegisterService类的checkData()方法和User类的isRegistered()方法来判断该帐号是否已被注册。如果已被注册，系统会提示帐号已被注册，注册失败；如果未被注册，则调用RegisterService类的register()方法进行注册

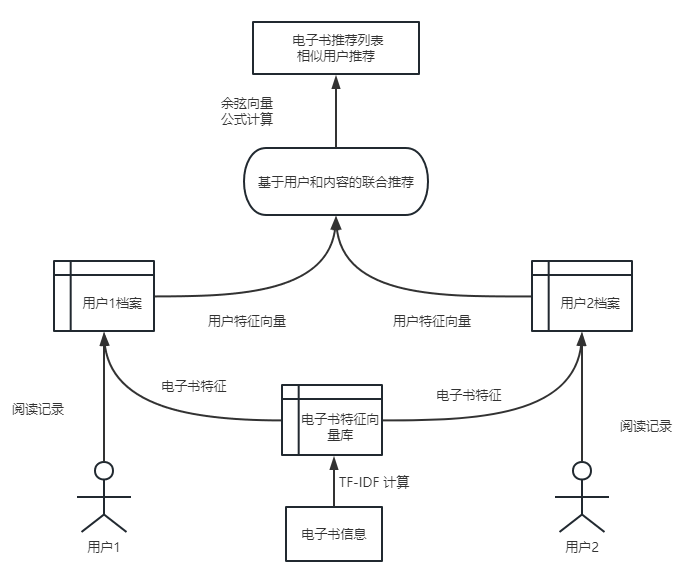
（2）登录：用户注册成功后，系统会自动将用户登录，页面会自动跳转至主页面。而在用户登录时，用户需要输入正确的帐号和密码，然后点击“登录”按钮。系统将前端提交的表单数据作为登录方法的形参，调用LoginService类的userLogin()方法来验证用户名和密码是否正确。如果用户名或密码不正确，系统会提示用户输入的用户名或密码错误；如果正确，则用户将能够进入主页面。

### 6.2.2 电子书推荐模块

电子书推荐模块是本系统的重要模块，用户登录后，用户的操作会触发一系列请求和响应操作，从中挖掘统计出推荐结果并展示给用户。用户可以在推荐结果中选择感兴趣的电子书进行查看，请求会发送至服务器端，服务器会根据开发者制定的智能推荐算法组合技术，将用户个性化的要求和电子书相关的具体信息一一展示给用户，包括电子书定价、出版社、内容简介、作者等。



本项目采用基于用户和物品的联合推荐方法，分别对电子书的两种类型进行推荐。基于用户和物品的联合推荐方法为用户提供了同类型的书籍推荐、不同类型的书籍推荐、相似用户推荐三种不同的服务。在集合亚马逊的书籍数据和大量用户数据后，最终形成了用户数据集合。通过找出兴趣点相同的用户，基于用户和内容的联合推荐方法如图4-12所示，实现了智能推荐服务。



该算法使用TensorFlow深度学习框架作为基础，使用基于用户和物品的协同过滤算法进行推荐。

1. 准备数据：从 CSV 文件中读取用户-物品评分矩阵，评分范围为 1 到 5，0 表示用户没有评分（未读）。并将数据放入数据库中。删除重复的记录，并将评分数据转换为矩阵。
2. 使用矩阵分解，将一个大矩阵分解成多个小矩阵的技术，这样可以使得矩阵的计算和存储更加高效。在协同过滤中，我们通常使用矩阵分解来将用户-物品评分矩阵分解成两个低维矩阵，分别表示用户和物品的隐含特征，这些隐含特征可以用来预测用户对未评分物品的评分。tf.keras.layers.Embedding构造隐向量，将输入的整数序列嵌入到一个可训练的低维空间中，同时可以将整数序列作为其输入并将其嵌入矩阵作为输出。嵌入矩阵的每一行都代表了一个整数序列中对应整数的嵌入向量。在推荐系统中，该函数通常用于将用户 ID 和物品 ID 建立嵌入，并将其转换为具有相同维数的低维向量空间中的向量，这些向量之间的相似度可以用于计算推荐的物品。
3. 使用TensorFlow GradientTape 对模型进行前向传播和反向传播，计算损失函数的梯度，并使用优化器对模型的参数进行梯度下降。使用测试数据计算均方误差（MSE），用于评估模型的性能。
4. 在验证集上计算加权混合算法的预测评分，根据预测效果选择最优的权重系数。用最优的权重系数进行加权混合
5. 使用模型进行推荐：为指定的用户推荐指定数量的未读过物品。

### 6.2.3 热门图书模块

当用户进入热门图书页面后，系统调用getTopBooks()方法获取当前热门图书信息，若有数据则以GridView组件的形式展示。点击“换一批”按钮后，系统将重新向后端服务器请求数据，刷新当前页面。当用户点击某本热门图书时，系统进入详情页面。

### 6.2.4 搜索模块

用户在APP首页点击“搜索图书”按钮后跳转至搜索页面。用户在搜索页面输入关键词，页面进行表单验证，验证用户输入是否合法。用户输入完成后，点击搜索按钮，前端控制器使用Book对象接收表单数据，并调用Book类的searchBook()方法来查询相关图书信息。如果查询结果为空，系统会提示“未找到相关图书”；如果查询结果不为空，则展示相关图书信息。

用户可以点击某一本图书查看详细信息，在详细信息页面，用户可以选择购买、评论该图书或者返回上一级页面。如果用户选择购买该图书，则调用BuyService类的addToCart()方法进行添加购物车操作。如果用户选择收藏该图书，则调用FavoriteService类的addToFavorites()方法进行添加收藏操作。如果用户选择评论该图书，则页面跳转至评论页面，用户可以在评论页面输入评论内容，并点击提交按钮，前端控制器使用Comment对象接收表单数据，并调用CommentService类的addComment()方法进行添加评论操作。如果添加评论成功，则提示用户评论成功；否则提示用户评论失败。

### 6.2.5 个人中心模块

个人中心模块共有四个部分，个人信息、我的订单、我的收藏、我的评论。

1. 个人信息部分，用户可以编辑自己的昵称、密码、手机号、常用地址等信息。当用户点击保存后系统调用modifyUserInfo()方法通过Volley库将http请求加入RequestQueue，以异步请求的方式将用户个人信息上传至后端服务器中。
2. 我的订单部分，用户可以查看自己的订单信息。
3. 我的收藏部分，用户进入此页面后，系统调用netUnit类中getFavoriteInfo()方法获取收藏信息，并用RecyclerView以列表的形式展示，包括电子书的封面、标题、作者、收藏时间等。用户长按某个电子书时可以删除收藏，系统调用deleteFavorite()方法向服务器同步数据。我的评论部分类似于我的收藏，不再赘述。

### 6.2.6 购物车模块

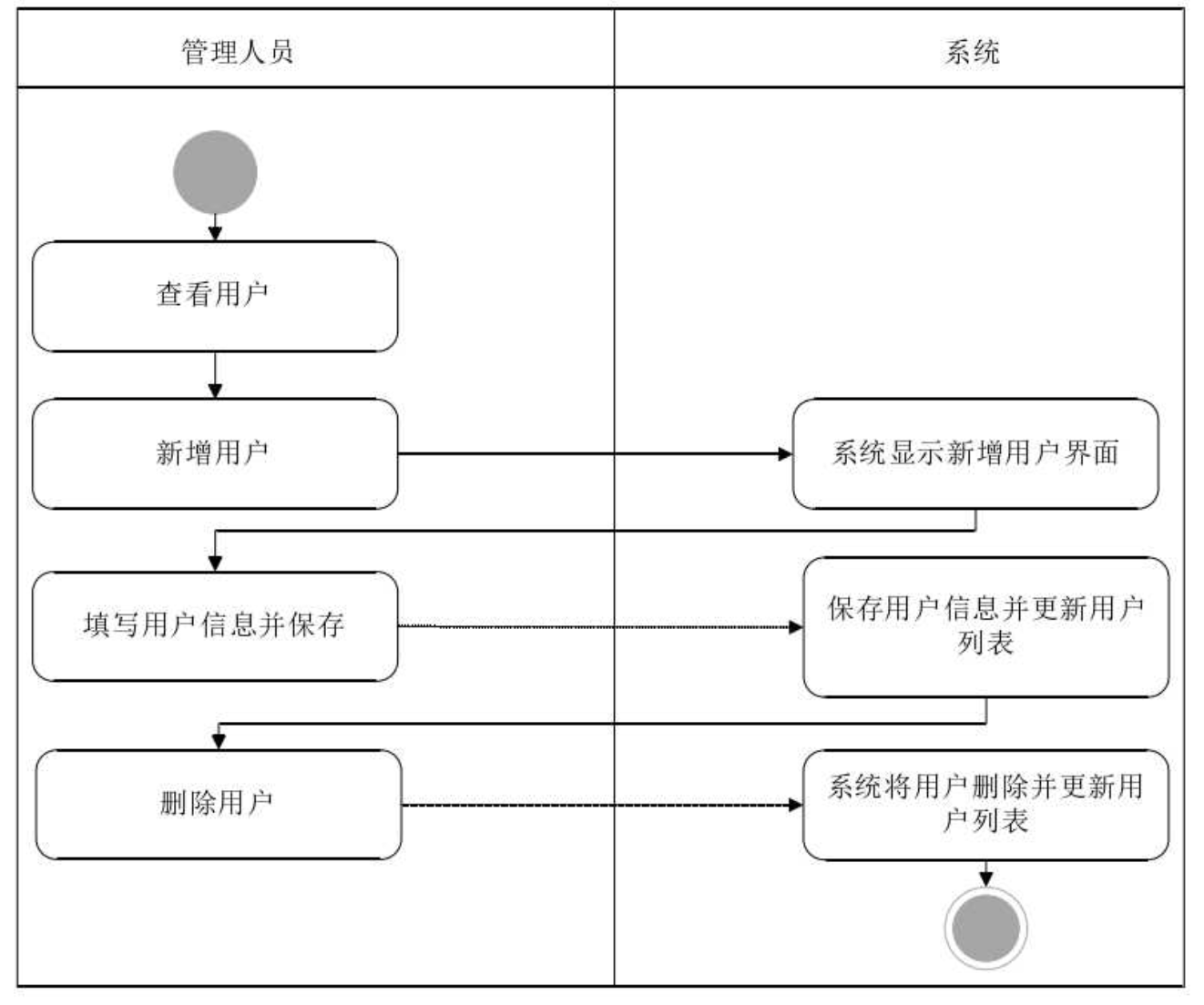
当进入购物车页面后，系统调用myCartInfo()方法查询当前用户的购物车信息，若有数据则以RecyclerView组件的形式展示。当用户添加商品至购物车时，服务端会获取用户添加的商品的信息，根据商品的ID和数量，调用addToCart()方法创建相应的购物车商品明细对象proObj，并从 Product表中获取该商品的商家信息。然后判断购物车对象列表cartList中有没有当前用户的购物车对象cart，若没有，将创建新的购物车对象cart，并将proObj添加进 cartList;若有，proObj的总价和数量。以上步骤完成后，系统通过Volley库将http请求加入RequestQueue，以异步请求的方式将购物车信息上传至后端服务器中。

当用户在购物车页面选择结算时，系统调用calculateTotal()方法计算购物车中商品的总价，并展示在页面上。用户可以选择修改商品数量或者删除商品，这会触发相应的事件，前端控制器会调用CartService类的相应方法进行修改或删除操作。如果修改或删除成功，则刷新购物车页面展示最新的购物车信息。用户确认结算后，进入订单结算页面，系统调用OrderService类的createOrder()方法创建订单，并将订单信息上传至服务器。如果创建订单成功，则提示用户订单创建成功，并跳转至订单详情页面。否则提示用户订单创建失败。

### 6.2.7 管理模块

管理模块包含三个部分：用户管理、电子书管理和订单管理。

1. 用户管理模块主要用于对系统中的用户进行管理。以RecyclerView的形式展示当前系统中全部用户，管理员可以在此模块中添加、删除和修改用户信息。当管理员需要添加新用户时，系统会调用addUser()方法，通过Volley库向后端服务器发送http请求，以异步方式将用户信息上传至服务器中。当管理员需要修改或删除用户信息时，进入用户信息编辑页面，可以对用户的头像，用户名，密码，地址手机号进行修改，点击保持后系统会调用modifyUser()方法向服务器发送数据，点击删除按钮时系统则调用deleteUser()方法，同样通过Volley库向服务器发送http请求，实现同步更新用户信息。



1. 电子书管理模块主要用于对系统中的电子书进行管理。管理员可以在此模块中添加、删除和修改电子书信息。当管理员需要添加新电子书时，点击加号浮动按钮，系统进入添加电子书页面，点击封面，安卓则调用changeAvatarClicked()方法，开启新的活动，用户可以选择从图库或者相机中获取图片作为图书的封面。用户在EditView中输入电子书的标题，作者等信息后，点击保存按钮，系统会调用addBook()方法，通过Volley库向后端服务器发送http请求，以异步方式将电子书信息上传至服务器中。同样，当管理员需要修改电子书信息时，系统会调用modifyBook()或deleteBook()方法，同样通过Volley库向服务器发送http请求，实现同步更新电子书信息。当服务器收到添加或者删除电子书的请求后，会判断信息是否符合规范，若符合规范则写入到数据库中。并在子线程中调用train方法对模型进行更新。
2. 订单管理模块主要用于对系统中的订单进行管理。管理员可以在此模块中查看、取消和修改订单信息。当管理员需要取消订单时，系统会调用cancelOrder()方法，当管理员需要修改订单信息时，系统会调用modifyOrder()方法，通过Volley库向服务器发送http请求，实现同步更新订单信息。管理员还可以在此模块中查看系统中所有的订单信息，以便更好地进行管理和调度。

# 第7章 系统测试

系统测试是将已经完成开发的系统的各个模块集成之后作为一个整体，在实际运行环境下对计算机系统的每个功能场景进行的测试和确认。测试方法通常分为白盒测试和黑盒测试两种，白盒测试一般是伴随开发过程进行，通过分支覆盖、谓词覆盖等方式测试程序中的所有可能路径，黑盒测试是将系统作为一个黑盒，不关注系统内部结构，从功能角度对系统进行测试。

## 7.1 登录注册模块

进行用户的注册和登录测试。在进行用户注册和登录测试时，需要验证用户能否成功注册、未能成功注册和使用已存在账号进行注册。测试结果显示系统可以完成以上目标。对于用户登录测试，主要关注成功登录和登录失败两种情况。结果显示本系统能过通过测试。

表6.1 用户注册功能测试用例表

|  |  |  |  |
| --- | --- | --- | --- |
| 功能描述 | 注册 | | |
| 编号 | 输入/动作 | 期望的输出/相应 | 测试结果 |
| 1 | 未输入用户名 | 请输入用户名提示 | 符合 |
| 2 | 输入用户名或密码不符合规范 | 请按要求输入的提示 | 符合 |
| 3 | 输入用户吗存在 | 显示用户名已存在 | 符合 |

表6.2 用户登录功能测试用例表

|  |  |  |  |
| --- | --- | --- | --- |
| 功能描述 | 登录 | | |
| 编号 | 输入/动作 | 期望的输出/相应 | 测试结果 |
| 1 | 输入正确的用户名和密码 | 成功登录并进入首页 | 符合预期 |
| 2 | 输入错误的用户名 | 提示该用户不存在 | 符合预期 |
| 3 | 输入正确的用户和错误的密码 | 提示密码错误 | 符合预期 |

## 7.2 电子书推荐模块

表6.3 电子书推荐模块测试

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 测试功能 | 流程 | 预期结果 | 测试结果 |
| 1 | 冷启动推荐 | 新用户登录后进入推荐页面 | 显示推荐的电子书 | 符合 |
| 2 | 个性化推荐 | 老用户登录后进入推荐页面 | 显示推荐的电子书 | 符合 |
| 3 | 更新推荐 | 用户对图书做出评分 | 更新用户推荐列表 | 符合 |
|  |  |  |  |  |

## 7.3 搜索模块

表6.4 搜索模块测试

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 测试功能 | 流程 | 预期结果 | 测试结果 |
| 1 | 模糊搜索 | 用户输入电子书标题 | 显示电子书 | 符合预期 |
| 2 | 模糊搜索 | 管理员输入用户信息 | 显示用户 | 符合预期 |

## 7.4 购物车模块

表6.5 购物车模块测试

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 测试功能 | 流程 | 预期结果 | 测试结果 |
| 1 | 加入购物车 | 在详情页点击加入购物车按钮 | 购物车内新添加此物品 | 符合预期 |
| 2 | 查看购物车 | 点击我的购物车 | 显示已添加的商品 | 符合预期 |
| 3 | 修改商品数量 | 在购物车页面点击“+”或“-”按钮 | 增加或减少物品 | 符合预期 |
| 4 | 删除商品 | 勾选要删除的商品，点击“删除”按钮 | 商品从购物车中删除 | 符合预期 |
| 5 | 结算购物车 | 点击结算按钮 | 跳转至结算页面 | 符合预期 |

## 7.5 个人中心模块

表6.6 个人中心模块测试

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 测试功能 | 流程 | 预期结果 | 测试结果 |
| 1 | 个人信息编辑 | 用户修改个人信息 | 修改成功 | 符合预期 |
| 2 | 查看收藏 | 用户点击我的收藏按钮 | 显示用户收藏的书 | 符合预期 |
| 3 | 查看订单 | 用户点击我的订单按钮 | 显示用户订单 | 符合预期 |
| 4 | 查看评论 | 用户点击我的评论按钮 | 显示我的评论 | 符合预期 |
| 5 | 查看评分 | 用户点击我的评分 | 显示我的评分 | 符合预期 |
| 6 | 删除收藏 | 用户点击取消收藏按钮 | 取消收藏 | 符合预期 |
| 7 | 删除评论 | 用户点击删除评论按钮 | 删除评论 | 符合预期 |

## 7.6 电子书详情模块

表6.7 电子书详情模块测试

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 测试功能 | 流程 | 预期结果 | 测试结果 |
| 1 | 电子书浏览 | 点击图书封面或者表题等 | 进入详情页面 | 符合预期 |
| 2 | 添加购物车 | 点击加入购物车按钮 | 购物车内新添加此物品 | 符合预期 |
| 3 | 收藏电子书 | 用户点击加入收藏按钮 | 收藏成功 | 符合预期 |
| 4 | 发表评论 | 用户输入评论并提交 | 显示并保存添加的评论 | 符合预期 |
| 5 | 对书评分 | 用户点击评分栏并提交评分 | 显示并保存用户评分 | 符合预期 |

## 7.7 管理模块

表6.8 管理端模块测试

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 测试功能 | 流程 | 预期结果 | 测试结果 |
| 1 | 添加用户 | 点击添加按钮并填写信息 | 保存添加的用户 | 符合预期 |
| 2 | 修改用户信息 | 点击用户并填写用户新的信息 | 更改用户信息成功 | 符合预期 |
| 3 | 删除用户 | 点击删除用户按钮 | 删除用户成功 | 符合预期 |
| 4 | 添加图书 | 点击添加按钮并填写信息 | 保存添加的图书 | 符合预期 |
| 5 | 修改图书信息 | 点击图书并填写图书新的信息 | 显示并保存电子书评分 | 符合预期 |
| 6 | 删除电子书 | 点击删除电子书按钮 | 删除电子书成功 | 符合预期 |
| 7 | 修改订单 | 点击订单并填写新的信息 | 修改订单成功 | 符合预期 |
| 8 | 删除订单 | 点击删除订单按钮 | 删除订单成功 | 符合预期 |

# 第8章 总结与展望

## 8.1 总结

本文首先介绍了选题的研究背景及其意义，阐述了推荐系统在国内的发展状况，并分析了基于Android的电子书推荐系统的现状。接着，对协同过滤算法进行了详细介绍。

其次，本文明确了系统开发的技术路线，采用了MVC架构，前端使用Android，后端使用Python编程语言，Flask框架以及MySQL数据库等。

通过前期充分的调查与考究，结合现有状况，采用自顶向下的方法，对系统进行了全面的可行性分析、功能性分析以及非功能性分析，确定了系统所要实现的目标。然后，根据系统要实现的目标，确定了本系统的整个体系结构，得出系统的总体设计方案。本系统一共划分为登录注册、推荐功能、搜索功能、购物车功能、个人中心功能以及后台管理六个重要成分。根据划分的功能模块对每个模块进行详细的设计，描绘出其流程图，并通过E-R图以及信息表完成系统的数据库的概念结构设计和物理结构设计。

最后，本文对系统功能模块进行开发，实现各个功能模块的需求，并对每个功能进行测试操作。测试结果表明，该系统能成功实现每个功能。该基于Android的电子书推荐系统能够方便用户购买自己喜欢的电子书，在海量电子书中寻找自己喜欢的作品，提供更加便捷、智能的电子书推荐服务，。

## 8.2 展望

由于时间和知识的受限，目前系统还存在很多不完善的地方，在未来的工作中，我将继续完善该系统，并探索一些新的研究方向。

1. 我计划加入更多的推荐算法和技术，以提高推荐的准确性和个性化程度。
2. 优化系统界面设计，拓展更多的功能。
3. 增加系统的安全性以及鲁棒性。

# 致谢

11111

# 参考文献

1. Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, John Riedl, GroupLens: an open architecture for collaborative filtering of netnews, Computer Supported Cooperative Work, pp175-186, Chapel Hill, North Carolina, 1994.
2. Koren, Yehuda, Robert Bell, and Chris Volinsky. "Matrix factorization techniques for recommender systems." Computer 8 (2009): 30-37.
3. Salakhutdinov, Ruslan, and Andriy Mnih. "Probabilistic matrix factorization." Advances in neural information processing systems. 2008.
4. 王睿.基于图网络结构的推荐方法研究[D].哈尔滨理工大学,2019.
5. 周广宇，朱晓东，宋承宪，庄宇轩，严宇超，肖鑫，姜向阳. 基于深度兴趣网络的点击率预测[J]. 中国科学: 信息科学, 2018, 48(10): 1294-1312.
6. 朱晓东，朱雪，戴晓晨，何彦，马俊. 基于深度兴趣演化网络的点击率预测[J]. 计算机学报, 2019, 42(8): 1546-1557.
7. Covington, P., Adams, J., & Sargin, E. (2016). Deep neural networks for YouTube recommendations. In Proceedings of the 10th ACM Conference on Recommender Systems (pp. 191-198).
8. 李旭，陈毅，冯杰，周国栋. 深度会话兴趣网络用于点击率预测[J]. 计算机研究与发展, 2020, 57(9): 1947-1961.
9. 刘建军.基于JDBC 的 Web 数据库接口技术研究[J].科技创新导报，2008,(10):24-24.