**审定成绩：**

|  |  |  |  |
| --- | --- | --- | --- |
| **重庆邮电大学**  **毕业设计（论文）** | | | |
|  | | | |
|  | **中文题目** | **关联分类规则挖掘软件设计与实现** |  |
|  |  |
| **英文题目** | **Design and Implementation of Software** |
|  | **for Mining Association Classification Rule** |
| **学院名称** | **计算机科学与技术学院/人工智能学院** |
| **学生姓名** | **文锐锴** |
| **专 业** | **智能科学与技术** |
| **班 级** | **04052001** |
| **学 号** | **2020211608** |
| **指导教师** | **周应华 副教授** |
| **答 辩 组**  **负 责 人** | **杨富平 副教授** |
|  | |

**2024 年 6 月**

**重庆邮电大学教务处制**

**计算机科学与技术/人工智能**学院本科毕业设计(论文)诚信承诺书

本人郑重承诺：

我向学院呈交的论文《 关联分类规则挖掘软件设计与实现 》，是本人在指导教师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不含任何其他个人或集体已经发表或撰写过的作品成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明并致谢。本人完全意识到本声明的法律结果由本人承担。

年级 2020

专业 智能科学与技术

班级 04052001

承诺人签名

年 月 日

学位论文版权使用授权书

本人完全了解重庆邮电大学有权保留、使用学位论文纸质版和电子版的规定，即学校有权向国家有关部门或机构送交论文，允许论文被查阅和借阅等。本人授权重庆邮电大学可以公布本学位论文的全部或部分内容，可编入有关数据库或信息系统进行检索、分析或评价，可以采用影印、缩印、扫描或拷贝等复制手段保存、汇编本学位论文。

（注：保密的学位论文在解密后适用本授权书。）

|  |  |
| --- | --- |
| 学生签名： | 指导老师签名： |
| 日期： 年 月 日 | 日期： 年 月 日 |

摘要

传统的关联规则挖掘算法意在从海量数据中挖掘出有意义的规则，以描述数据之间存在的相互联系。而分类问题则是利用分类器，根据样本的属性对样本类别进行预测。对于前者，所能发现的规则是无法事先知晓的；对于后者，目标类别的最终判定结果只能有一个。本文结合关联规则挖掘与分类技术，设计并实现了一个挖掘关联分类规则的软件。

本文首先选取并介绍了CBA-M1算法、CBA-M2算法、APR算法和CMAR算法的基本原理，然后通过需求分析、确定软件基本功能并对软件进行设计，最后基于Vue3和Flask框架的前后端分离架构进行软件开发，算法主体使用Python语言实现。在系统设计方面，本文采用软件工程原理，构建了一个模块化的系统，包括数据预处理、类关联规则提取和分类器构造等关键模块。此外，实现的软件简便易用，能够利用多种算法学习得到分类规则，然后使用这些规则对数据进行分类以及对比分析不同算法的性能。

实验部分，使用UCI公开数据集对算法进行了测试，并对不同算法的挖掘结果和效率进行了对比分析。结果表明，不同算法在准确率、时间开销、规则数目上各有优劣。其中，实验发现APR算法整体性能更好，在测试样本中分类准确率较高的同时，拥有较低的时间开销和较少的类关联规则；CMAR算法由于采用了高效的FP-growth算法和CR-tree结构，使得其在处理大规模数据集时，仍能表现出低时间开销。

**关键词**：数据挖掘，关联规则，分类，软件实现

**Abstract**

Traditional association rule mining algorithms aim to discover meaningful rules from vast amounts of data to describe the relationships between data. In contrast, classification problems use classifiers to predict the category of samples based on their attributes. For the former, the rules discovered are not known in advance; for the latter, the final determination of the target category can only be one. This thesis combines association rule mining and classification techniques to design and implement software for mining association classification rules.

First, this thesis selects and introduces the basic principles of the CBA-M1 algorithm, CBA-M2 algorithm, APR algorithm, and CMAR algorithm. Then, through requirement analysis, it determines the basic functions of the software and proceeds with software design. Finally, based on the front-end and back-end separation architecture of Vue3 and Flask frameworks, the software is developed with the main algorithms implemented in Python. In terms of system design, this thesis adopts software engineering principles to construct a modular system, including key modules such as data preprocessing, class association rule extraction, and classifier construction. Additionally, the developed software is user-friendly, capable of learning classification rules using various algorithms, and then using these rules for data classification and comparative analysis of different algorithms' performance.

In the experimental section, the algorithms were tested using publicly available UCI datasets, and the mining results and efficiency of different algorithms were compared and analyzed. The results indicate that different algorithms have their own advantages and disadvantages in terms of accuracy, time cost, and the number of rules. Among them, the experiment found that the APR algorithm performs better overall, with higher classification accuracy on the test samples, lower time cost, and fewer class association rules. The CMAR algorithm, due to its use of the efficient FP-growth algorithm and CR-tree structure, shows low time cost even when processing large-scale datasets.

**Keywords:** data mining, association rules, classification, software implementation

目录

[第1章 引言 1](#_Toc168828993)

[1.1 研究背景和意义 1](#_Toc168828994)

[1.2 国内外研究现状 2](#_Toc168828995)

[1.2.1 国内研究现状 2](#_Toc168828996)

[1.2.2 国外研究现状 3](#_Toc168828997)

[1.3 主要内容和工作安排 4](#_Toc168828998)

[第2章 关联分类规则挖掘算法 7](#_Toc168828999)

[2.1 CBA算法 7](#_Toc168829000)

[2.1.1 规则生成 7](#_Toc168829001)

[2.1.2 CBA-M1算法 9](#_Toc168829002)

[2.1.3 CBA-M2算法 9](#_Toc168829003)

[2.2 APR算法 10](#_Toc168829004)

[2.3 CMAR算法 11](#_Toc168829005)

[2.4 本章小结 12](#_Toc168829006)

[第3章 软件设计与实现 13](#_Toc168829007)

[3.1 需求分析 13](#_Toc168829008)

[3.2 总体设计 14](#_Toc168829009)

[3.3 详细设计 16](#_Toc168829010)

[3.3.1 功能设计 16](#_Toc168829011)

[3.3.2 页面设计 21](#_Toc168829012)

[3.4 软件实现 22](#_Toc168829013)

[3.4 本章小结 23](#_Toc168829014)

[第4章 软件展示与测试 25](#_Toc168829015)

[4.1 软件展示 25](#_Toc168829016)

[4.2 软件测试 30](#_Toc168829017)

[4.3 本章小结 31](#_Toc168829018)

[第5章 算法分析实验 33](#_Toc168829019)

[5.1 分类准确率 34](#_Toc168829020)

[5.2 模型构建及预测时间 36](#_Toc168829021)

[5.3 类关联规则数目 37](#_Toc168829022)

[5.4 本章小结 38](#_Toc168829023)

[第6章 总结与展望 39](#_Toc168829024)

[6.1 总结 39](#_Toc168829025)

[6.2 后续工作展望 40](#_Toc168829026)

[参考文献 43](#_Toc168829027)

[致谢 45](#_Toc168829028)

# 

# 第1章 引言

## 1.1 研究背景和意义

随着全球数据总量的持续攀升，数据分析与挖掘技术在各个领域的应用变得愈发重要。据国家统计局数据显示，截至2023年底，我国数据总量已达到数十ZB级别，并且这一数字预计将在未来几年继续增长。这一数据膨胀趋势使得如何有效地从海量数据中挖掘出有价值的信息，成为了一个亟待解决的问题。利用数据挖掘技术提升决策质量和效率，对于国家经济发展和社会进步具有重要意义。

在具体行业中，零售、医疗、金融等领域对数据挖掘技术的需求尤为强烈。根据前瞻产业研究院发布的《中国大数据行业市场前瞻与投资战略规划分析报告》，未来市场规模将突破七万亿元。各行业纷纷加大对数据挖掘技术的投入，以期通过数据分析提升业务洞察力和竞争优势。特别是在金融行业，通过数据挖掘技术进行风险和欺诈检测，已成为行业的常态化需求。因此，对于挖掘出海量数据中存在的潜在模式以及对数据进行分类有着很大的需求。

关联分类规则挖掘结合了关联规则挖掘和分类两种方法的优点，不仅能够发现隐藏在数据中的关联模式，还能通过这些模式对样本进行准确分类。作为数据挖掘领域的一个重要问题，人们对其进行了大量的研究，并衍生出多种不同的关联分类规则挖掘算法，其中较为经典的有CBA算法[1]、CMAR算法[2]。CBA算法包含两个部分，第一部分生成一组类关联规则而非传统的关联规则，这一部分实现的关键在于如何筛选出候选项集，以及如何对类关联规则进行规则剪枝。算法的第二部分就是训练分类器，CBA的分类器构造包含M1和M2两种算法。2019年，研究人员在CBA-M1算法的基础上进行改进，提出了基于主动剪枝规则实现的APR算法[3]。CMAR算法不像CBA算法依赖单一的类关联规则，而是利用多条类关联规则的组合进行分类。本论文选取了CBA-M1算法、CBA-M2算法、APR、CMAR算法进行实现，并整合在一个软件中，可以有效处理各类数据集。通过实验验证多种算法在不同数据集上的性能，发现该软件在不同的数据集中均有良好的分类效果。关联分类规则挖掘软件的设计与实现，为行业提供了一个强有力的分析工具，有助于提升相关行业的数据处理能力和决策水平。

## 1.2 国内外研究现状

### 1.2.1 国内研究现状

基于关联规则的分类算法在国内的数据挖掘和机器学习领域得到了广泛的研究和应用。国内的研究团队在关联分类规则挖掘算法的基础上进行了多种创新，特别是针对特定的行业需求和数据特性进行了深入的定制和优化，这些优化措施有效地提高了算法在处理大规模数据集时的性能，同时也增强了算法在复杂数据环境下的稳定性和准确性。

关联分类算法在数据挖掘领域中得到了广泛应用和创新改进。黎文娟等人提出的基于余弦度量的关联分类算法（IACD）通过使用余弦度量来评估项集与类别之间的相关性，以及使用新的规则强度对规则进行排序和剪枝，有效减少了冗余规则数量，并在多个UCI数据集上显示出较高的分类准确率[4]。秦晨普等人针对传统关联分类算法的不足，如资源消耗大和规则剪枝难等问题，提出了一种基于分类修剪的改进方案ACCP，通过分块挖掘和改进频繁项集及规则修剪过程，显著提高了分类准确率和运行效率[5]。

国内研究学者在改进现有算法以提高分类准确率和效率方面，提出了多种改进算法。李家辉等人针对支持度置信度框架难以产生大量高质量规则的问题，提出了改进的关联分类算法IAMC[6]。刘莉萍等人针对中药“性-效”数据的高关联度和属性稀疏问题，提出了一种改进的关联分类算法（ECBA），该算法采用垂直数据格式生成类关联规则，避免了传统关联分类算法产生大量无意义规则的不足[7]。郭明环等人通过关联挖掘算法对图书馆读者的行为特征进行了分析，有效地划分了读者类型，该方法不仅提高了分析效率和直观性，还减少了计算成本[8]。翟悦等人提出了一种带约束的改进关联分类规则挖掘方法，通过扩展概念格结构和引入差集概念，加速了支持度和置信度的计算，并在时间和空间性能上都表现出色，满足了重用性要求[9]。马晓剑等人针对信息高度冲突情况下，基于信度转移模型的关联二分类算法会出现分类精确率低、计算量大的问题，提出了一种基于概率转换函数与邓熵的关联分类融合算法。该算法通过构造基本概率指派函数、重构邓熵，并结合PPT散度优化分类规则，有效解决了概率转换函数的高冲突问题，显著提高了分类准确性，并且计算量大幅降低[10]。

面对分布不平衡的数据集，刘影等人提出了一种基于模糊关联的不平衡数据分类算法，通过结合爬虫技术、AdaBoost.M1W集成学习算法和模糊关联分类器，对不平衡数据集进行高精度分类。实验结果表明，该算法在自然不平衡数据集和人工不平衡数据集上的分类性能较优，分类耗时短，形成的关联规则数据量低且分类质量好[11]。崔巍等人提出来一种不平衡关联分类算法ACI，使用不平衡规则裁剪方法进行规则剪枝，发现在不平衡数据集上比基准算法的分类效果更好[12]。王俊红等人提出了一种基于欠采样和代价敏感的不平衡数据分类算法USCBooST，根据样本权重选取与少数类样本数量相当的多数类样本，之后将采样后的多数类样本权重归一化并与少数类样本组成临时训练集训练基分类器[13]。

这些研究表明关联分类算法的持续创新不仅解决了现有技术的局限性，而且在多个应用场景中展示了其强大的性能和实用性。

### 1.2.2 国外研究现状

目前国外研究团队在传统关联分类规则挖掘算法的基础上也有着多种创新，针对不同领域的具体应用对算法进行了优化，可以有效地提升分类任务的准确性、效率和可靠性。

H. Hasanpour等人针对由于挖掘的规则集过于庞大而导致的高处理开销问题，提出了一种将Apriori算法与基于分类的关联规则算法相结合的新方法，构建了一个高效的基于规则的分类器​​。研究结果表明，这种集成方法在处理大型数据集时，不仅能显著减少计算资源的消耗，还能提升分类性能[14]。R. Alazaidah等人在2023年提出了一种基于关联的多标签分类方法，通过挖掘标签之间的关系来提升分类结果准确率。结果显示，该方法在准确性和效率上都有显著提升，展示了其在应用科学中的广泛应用前景[15]。

国外研究学者还将剪枝和规则生成作为算法的改进方向，C. Thanajiranthorn等人摒弃传统的剪枝过程，提出了一种旨在提高分类精度的新型关联分类（AC）算法，可以直接发现紧凑且高效的分类规则​​。该方法通过严格筛选规则，确保每条规则都具有较高的分类效用，从而减少规则集的冗余，提高分类器的精度和效率。实验结果显示，这种直接发现高效规则的方法不仅能够保证分类准确率，还能显著减少规则数量，使得分类模型更加简洁高效[16]。R. Dhanalakshmi等人提出了一种基于信息质量值的模糊影响规则的关联规则生成和分类算法，旨在提高分类效率并降低错误率。该算法通过引入信息质量值评估规则的有效性，并结合模糊逻辑处理数据中的不确定性，特别适用于处理敏感数据。实验结果表明，这种方法在多个数据集上的分类效果优于传统方法，尤其是在处理带有噪声和不确定性的数据时，能够显著降低错误率，提高分类效率[17]。

在应用方面，应用范围也在不断扩展。关联分类算法已经被应用于多个领域，如文本数据集、图像识别等，关联方法在处理类别不平衡、重叠和典型模式时的性能，显示出在特定问题上关联方法的优势[18]，A. Tran等人使用关联分类算法对心力衰竭的严重情况进行分类，对患者的健康状况和心脏功能等级之间的相关性进行了分析[19]。在事故原因的分析方面，关联规则挖掘技术也得到了广泛应用。A. Verma等人利用关联规则挖掘分析职业事故调查报告，分析钢铁厂工作活动与事故​​之间的关系，揭示了其中的潜在关联​​，通过使用文本挖掘结合ARM技术，分析识别事故原因及其关系[20]。L. Cagliero等人通过从过去的学习者相关数据中训练分类模型来预测学生的学习成绩，提出了一种Explainable Learning Analytics解决方案 。该方法旨在在课程结束前预测考试成功率，并为教育工作者提供可解释的预测结果，以便进行针对性的教学干预。研究显示，这种方法不仅提高了学生成绩预测的准确性，还增强了模型的可解释性和实用性，为教育领域提供了一种新的分析工具[21]。

这些研究不仅说明关联分类规则挖掘技术在不同领域有着广泛的适用性，同时也表明了关联分类规则挖掘算法的有效性深受研究人员的欢迎。这些研究成果共同展示了关联分类规则挖掘技术在数据挖掘和分类方面的巨大潜力，推动了其在多个领域的应用和发展。

## 1.3 主要内容和工作安排

全文共分为6章，内容结构安排如下：

第1章是引言，引入课题的研究背景及意义。

第2章是关联分类规则挖掘算法，介绍四种算法的实现原理。

第3章对软件进行需求分析、总体设计和详细设计。

第4章是关联分类规则挖掘软件实现与测试，介绍软件的实现与测试过程。

第5章是算法分析实验，使用多个数据集对算法性能进行分析。

第6章是总结与展望，对本文及实验内容进行总结，并提出未来的发展方向。

# 第2章 关联分类规则挖掘算法

关联分类规则挖掘算法是一种将关联规则挖掘与分类任务相结合的技术。通过在数据集中挖掘频繁项集，得到一种将属性值与特定类标签相关联的特殊规则，称为关联分类规则或者类关联规则。在分类任务中，这些关联分类规则用于预测未知实例的类别，该算法的核心理念就是通过从训练集中找到强相关的规则、构建性能良好的分类器来提高分类的准确性。与其他传统的分类方法相比，关联分类规则挖掘算法具有较强的可解释性，能够帮助揭示数据中的潜在模式。

## 2.1 CBA算法

CBA(Classification Based on Association，基于关联规则的分类)算法[1]，结合了关联规则挖掘和分类器构建的思想，能够从数据中发现类关联规则并进行分类预测。该算法包含两个部分，第一部分是规则生成，这一步是在Apriori算法的基础上进行修改实现的，其核心思想与Apriori算法相似，不过生成的是针对特定类的关联规则而非传统的关联规则。在这个过程中，首先识别出频繁项集并生成类关联规则，然后对这些规则进行剪枝，从而减少规则的数量并提高规则的质量。第二部分是训练分类器，分类器构造部分包含M1和M2两种算法。其中M1算法基于贪心策略，按序逐步抽取类关联规则，与数据集中的样本一一进行匹配，最后构建一个有序规则集合作为最终的分类器。虽然M1算法形式相对简单，易于编程实现，但是当面对大型数据样本时，其效率并不高，因此又提出了M2算法。M2算法通过更复杂的规则选择和管理机制来优化分类过程，特别适合处理大规模和复杂的数据集。

### 2.1.1 规则生成

CBA算法的规则生成部分就是去寻找所有满足最小支持度和最小置信度要求的类关联规则，一个类关联规则就是一个由项集（X）和类别标签（Y）形成的组合（XY）。项集是一组特定的属性，而类别标签则是对应的分类结果。

类关联规则的支持度，反映了同时满足条件集和特定类标签的数据实例在整个数据集中的占比。而置信度度量了在满足条件集的数据实例中，同时也符合规则定义的类标签的占比。对于一个类关联规则XY，支持度计算公式为：

， (2.1)

式中，——数据集中同时包含X和Y的数据项数目

*N* ——数据项总数目

这条类关联规则的置信度计算公式为：

， (2.2)

式中，——数据集中同时包含X和Y的数据项数目

­­ ——数据集中包含X的数据项数目

支持度是一种重要的度量，用于评估规则的普遍性，置信度用于评估规则的可靠性，支持度和置信度越高，说明规则更加普遍且可靠。如果一个规则满足最小支持度约束并且满足最小置信度约束，那么它就是一个强关联规则，类关联规则应当都是强关联规则。生成类关联规则的关键步骤如图2.1所示。

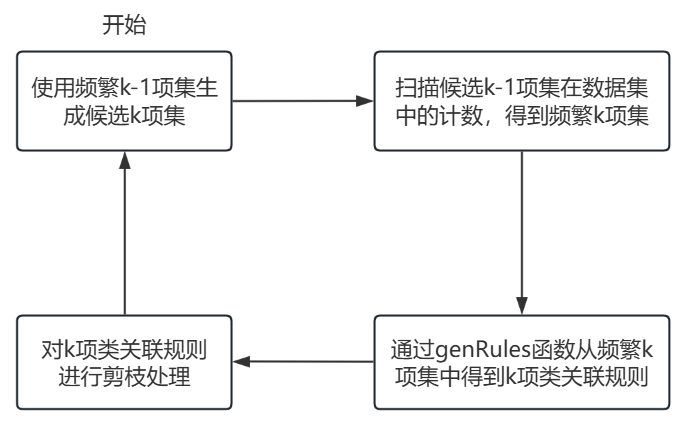


图2.1 生成类关联规则的关键步骤

### 2.1.2 CBA-M1算法

CBA-M1算法的主要核心是基于贪心策略，尽可能早的选出高优先级的规则。当存在多种规则时，需要确定某个样本优先匹配其中的哪一个规则，故需要对生成的类关联规则进行优先级排序。

给定两个规则ri和rj，规则ri优先级高于规则rj的条件是：

（1）ri的置信度大于rj的置信度。

（2）ri和rj置信度相同，但是ri的支持度大于rj的支持度。

（3）ri和rj的支持度和置信度均相同，但是ri比rj产生的时间早。

CBA-M1算法的基本思想就是根据生成的类关联规则，构建出一组高优先级有序规则序列< r1, r2, ··· , rn, default\_class >作为最终的分类器。序列中的规则按照优先级降序排列，其中default\_class是默认类，当前面的规则都不能覆盖这条记录时，则将其划分入默认类中。

M1算法首先根据规则序列依次选择规则，并逐一与样本进行匹配。如果样本满足当前规则的条件，则说明匹配成功，在每一轮中匹配中将移除所有与该规则匹配成功的样本，然后使用下一条规则继续匹配剩余的数据集。每轮中未匹配成功的样本将被分类为默认类，即剩余数据集中出现次数最多的类标签。在分类器构建过程中，会测试类关联规则集合中的每一条规则，并记录添加每条规则后的累计分类错误数。匹配结束后，筛选出分类错误总数最少的规则作为临界点，丢弃所有优先级低于临界点的规则，只保留该规则以及优先级高于临界点的规则集合。这些被保留的规则构成最终的分类器，用于对测试集进行分类。

### 2.1.3 CBA-M2算法

CBA-M2算法的中心思想和M1算法基本一致，仍然是先按照上述规则对类关联规则进行优先级排序，找出临界点并丢弃临界点之后的规则，不同的是分类器的构建方法。整个分类器构建过程包括三个阶段，首先通过缩小候选规则的方法，减少不必要的规则判定，提高训练效率。CBA-M2算法在阶段一中首先对数据集中的样本进行遍历，找出正确分类该样本的最高优先级规则crule，以及错误分类该样本的最高优先级规则wrule。若crule的优先级高于wrule，则该样本被正确分类，并将这条规则添加至指定集合U中。若wrule的优先级高于crule，则需要决定该样本由crule覆盖还是由wrule覆盖，记录后在阶段二处理。阶段二中，遍历集合U，如果某个样本的wrule是其他样本的crule，那么就对wrule的分类错误总数加一。否则使用wrule遍历整个数据集，记录其分类错误总数。阶段三中，对前两个阶段得到的规则集进行筛选，丢弃那些不会提高分类器准确率的规则。和M1算法类似，以分类错误数最少的规则为临界点，在此规则之后的所有规则都将被丢弃，因为它们只会产生更多的错误。

## 2.2 APR算法

APR（Active Pruning Rules，主动剪枝规则）算法在2019年提出，是在CBA-M1算法基础上引入了一种动态剪枝机制[3]。传统的CBA-M1算法按照固定的规则序列依次匹配样本，并在完成匹配后移除所有满足条件的样本。然而，这种方法会影响优先级较低的规则的有效性，进而降低分类器的性能。APR算法通过在每次规则匹配后更新剩余规则在未匹配数据集中的支持度和置信度，对不满足最小支持度和置信度的规则进行剔除，从而减少冗余，确保仅保留最有效的规则。这种动态调整机制提高了分类器的效率和准确性。

APR算法采用垂直结构存储项目的出现列表，通过交集生成频繁项集。在分类器构建阶段，CBA-M1算法通过对规则进行排序，每次选择最高优先级的规则对数据进行匹配，并在规则匹配后移除相应的数据项，最终选择错误率最小的规则作为分类器的临界点，保留错误总数最少及优先级高于临界点的规则作为最终的分类器。而APR算法每轮匹配后重新计算规则的支持度和置信度，更新规则的优先级，剔除不满足支持度和置信度阈值要求的规则。确保分类器中仅包含最有效的规则。理论分析而言，APR算法比CBA-M1算法的剪枝规则更加有效，生成的规则数量也会更少，构建的分类器更加简洁有效。

对数据进行分类预测时，APR算法针对每个数据样本，所有匹配成功的规则将根据标签分为不同的簇，对数据样本的类别进行投票，将包含最多规则的簇对应的类别标签作为分类结果。

APR算法通过动态规则剪枝和多规则匹配策略，在保证分类器中规则质量的同时，还避免了CBA-M1算法过度依赖单一高置信度规则的弊端。

## 2.3 CMAR算法

CMAR（Classification based on Multiple Association Rules，基于多重关联规则的分类）算法[2]在传统关联分类规则挖掘算法的基础上进行改进，旨在克服传统关联规则分类中的过度依赖单个高置信度规则的问题。CMAR算法与传统关联规则分类方法不同，它不依赖单个高置信度的规则，而是从关联规则中挑选出一组高置信度、相关的规则进行分类预测。同时，使用卡方分析方法进行加权，确保选择的多重关联规则集能够为新样本提供更全面的分类决策，减少预测中偏差从而提高分类的准确性。

为了高效地挖掘满足支持度和置信度要求的规则，CMAR使用改进的FP-growth算法挖掘类关联规则，通过构建关联FP-tree，能够快速地处理大型数据库中的频繁项集挖掘任务。同时，CMAR采用一种新的数据结构CR-tree用于存储和检索已挖掘的规则。该结构通过前缀树的方式来共享规则中的重复项，大大降低了规则的检索时间。

CMAR算法采用了多种剪枝策略来提高分类器的质量。首先是在规则生成阶段，通过设定最小支持度和置信度阈值，筛选出满足条件的规则。在挖掘新的类关联规则时，将通过计算卡方值来评估规则的有效性，剔除那些分类不显著的规则。同时使用基于数据覆盖率实现的剪枝算法，限制每个数据记录被规则覆盖的次数，确保每条记录不会被过多的规则覆盖；在CR-tree的构建过程中，如果新规则的置信度低于其父节点的置信度且两者分类类别相同，那么这条新规则不会被插入，从而减少冗余规则。

对新样本进行分类预测时，CMAR首先从CR-tree中检索出与新样本特征相匹配的规则集合，这些规则按照类别标签进行分组，之后根据每个规则的卡方统计值和加权分析方法计算该分组的综合权重，从而评估分组的整体效果。通过对比不同类别分组的组合效果分数，CMAR选择综合效果最强的组标签作为最终预测的分类结果，从而对新样本进行准确的分类。如果匹配的规则不一致或存在冲突，则通过综合所有规则的效果进行最终决策，以确保分类预测的稳定性和准确性。这种基于多重规则的分类策略有效避免了单一规则引起的偏差与过拟合问题，提高了CMAR模型的预测性能。

## 2.4 本章小结

本章主要介绍了四种关联分类规则挖掘算法的基本原理，其中CBA-M1、CBA-M2算法都是依赖单一高置信度的规则进行分类，而CMAR算法克服了传统分类算法过度依赖单一规则的缺陷，通过多个类关联规则和卡方分析进行判别。APR算法引入了动态剪枝和多规则匹配策略，能够有效减少冗余规则，提升规则的质量。这些关联分类算法各有其特点，提供了一种相较于传统分类方法更具解释性的方式。目前已经深刻了解了四种算法的基本原理，这一步十分关键，对于下一步软件设计具有重要意义。

# 第3章 软件设计与实现

关联分类规则挖掘软件需要实现多种算法对指定数据的关联规则挖掘与分类预测，软件应具有良好的交互性设计以及数据的可视化展示，帮助用户更好地对比和分析算法结果。

## 3.1 需求分析

关联分类规则挖掘软件的核心功能是使用关联分类规则挖掘算法构建分类器实现对数据的分类，并根据最终的分类结果测试不同算法的性能。通过分析，该软件需要具备以下功能。

首先，用户需要能够自定义选择数据集，因此软件必须支持用户上传本地数据集文件，文件格式应为CSV等常见数据格式。软件需要识别并解析文件中的数据，并在上传完成后立即在展示区域进行数据预览，方便用户确认上传的数据是否正确。此外，软件需要提供数据预处理功能。数据预处理模块应负责处理缺失值、数据清洗和数据离散化等操作，确保输入数据的质量和一致性，提升分类器的构建效果。软件还需要提供灵活的参数设置功能，允许用户调整最小支持度和最小置信度参数。参数设置部分应设计为滑动条或输入框，支持用户快速调整参数值，并且参数设置应精确到小数点后两位，以满足不同算法和数据集的需求。此外，软件应支持算法选择功能，允许用户从多种关联分类算法中进行选择，例如CBA-M1、CBA-M2、APR和CMAR四种算法。用户可以根据自身需求和数据特性选择适合的算法进行数据分析。

为了展示算法的运行结果，软件应具备分类器生成后的类关联规则展示、分类准确率、运行时间等评价指标的显示功能。结果展示部分应设计为易于理解的图表形式，如柱形图和折线图等，帮助用户直观地了解算法的分类效果和性能差异。软件还应支持多算法比较功能，用户可以一次性选择多种算法进行测试，软件将分别展示每种算法的分类器、准确率、时间开销和规则数量等结果，并以图表形式进行对比，帮助用户清晰地看到不同算法的优缺点。

模型构建完毕后，为了以后使用已经构建好的模型对数据进行分类，软件应能够对模型进行保存和读取并调用。而且，为了方便地管理模型，软件应能够提供对模型进行管理的界面，同时做到模型的重命名、删除等操作。

## 3.2 总体设计

为了提高软件的可维护性和扩展性，计划采用前后端分离的软件架构。前端部分负责用户界面展示和交互，后端负责数据处理和算法运算，两者之间通过网络请求与响应实现高效交互。这种模式不仅能提高开发的灵活性，还便于后续的系统维护和升级。

前端设计包含操作区和展示区两部分。操作区需要实现文件的上传、参数设置和算法选择等功能，用户可以通过操作区上传本地数据文件，支持CSV等常见数据格式。文件上传按钮应方便易用，并在文件成功上传后显示确认信息。参数设置部分提供调整最小支持度和最小置信度的滑动条或输入框，用户可以通过鼠标或键盘精确调整参数值，并实时看到参数变化的效果。操作区还包括多个算法选择按钮，用户可以选择不同的关联分类算法（如CBA-M1、CBA-M2、APR和CMAR）进行分析。此外，操作区还应包含“测试”按钮，用户点击后可以启动多种算法运行，并将算法运行结果使用柱形图或者条形图在前端页面进行展示，实现对多种算法的性能进行分析。

展示区用于显示文件数据展示、算法运行结果以及不同算法的对比分析结果。文件数据预览功能使用户可以在上传文件后立即查看数据内容，确保数据上传正确。算法运行结果应包含分类器中的类关联规则、分类准确率、时间开销等指标，对于类关联规则，应采用表格进行展示，包含这条规则的属性、类别、实际支持度和置信度。不同算法的对比分析结果应以图表形式展示，如柱形图、折线图等，帮助用户直观地理解和分析数据挖掘结果。展示区提供对比功能，用户可以同时查看多种算法的结果，图表对比不同算法的准确率、时间开销和规则数量等，方便用户进行综合评估。

后端设计包括数据预处理模块、算法实现模块和服务器搭建。数据预处理模块接收前端上传的数据文件后，进行数据预处理，包括处理缺失值、数据清洗和数据离散化等操作，确保数据质量和一致性。数据清洗包括检测并处理数据中的空值、重复值和异常值，通过插值、删除或填补等方式提高数据的一致性。数据离散化将连续型数据转换为离散型数据，便于关联规则挖掘算法处理，采用基于区间划分或聚类的方法，将连续数据分段，并将每段映射为一个离散值。数据格式化对上传的数据进行格式检查和转换，确保数据结构符合算法要求，将用户上传的各种数据格式统一转换为标准的输入格式。

算法实现模块是软件的核心部分，负责实现多种关联分类规则挖掘算法，如CBA-M1、CBA-M2、APR和CMAR算法，每种算法应封装为高内聚低耦合的算法模块。每个算法模块接收参数并对数据进行处理，返回类关联规则、分类准确率、运行时间等结果，确保独立性和灵活性。

服务器部分作为软件的中枢，负责接收前端发送的网络请求，解析请求内容并调用相应的算法模块进行数据处理，算法运行完毕后将结果打包成JSON格式返回前端。为保障多用户使用时的高并发处理能力，服务器应具备高并发处理能力，确保多用户同时使用时的响应速度。设计服务器架构时考虑未来扩展需求，预留接口和模块，便于后续增加新算法或功能。前后端通过HTTP请求进行通信，前端发送POST请求将用户上传的数据和设定的参数传递给后端，后端处理完毕后将结果以JSON格式返回前端，前端接收返回结果后，将数据解析并渲染到展示区域。交互过程需考虑数据传输的安全性和效率，采用加密传输和数据压缩技术，确保用户数据的隐私和完整性。前端界面设计应注重用户体验，提供友好的交互方式和清晰的操作指引。页面布局应简洁大方，各功能模块分布合理，确保用户能够方便快捷地完成操作。前端应提供实时反馈机制，例如在算法运行时显示加载动画，在数据上传成功后显示提示信息等，提升用户使用体验。此外，软件应具有响应式设计，适应不同设备和屏幕尺寸，确保在笔记本和平板电脑上都有良好的使用效果。

通过以上设计，关联分类规则挖掘软件将具备高效的数据处理能力、灵活的参数调节功能、算法选择和直观的结果展示，能够帮助用户更好地进行数据挖掘和分析，提升分类器的构建效果和实用性。软件的高扩展性和用户友好性将使其在实际应用中发挥更大的作用，为各领域的数据分析和决策提供强有力的支持。软件整体的层次图如图3.1所示：

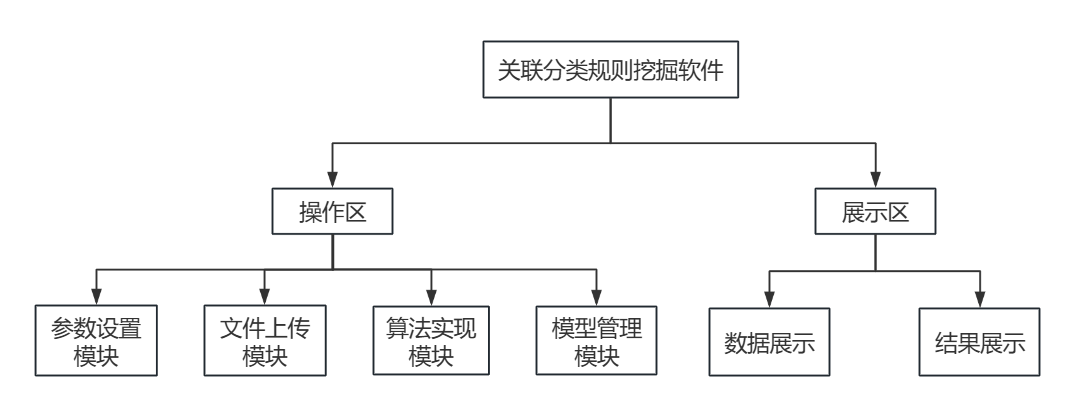


图3.1 关联分类规则挖掘软件层次图

## 3.3 详细设计

### 3.3.1 功能设计

软件主要包含文件上传与展示、参数设置、算法实现、网络请求管理、模型管理五个模块。其中网络请求管理模块为其他模块提供信息传递服务。

1）文件上传与展示模块

文件上传部分需要满足用户能够自定义上传本地CSV文件，该部分负责接收用户上传的数据，并借助POST网络请求将数据传送给服务端。同时，将用户上传的数据在前端页面进行展示，让用户可以在上传后立即查看数据的内容，确保上传的数据符合用户的预期。

2）参数设置模块

参数设置部分需要满足用户使用鼠标或键盘自由编辑最小支持度和最小置信度，并且阈值大小应精确到小数点后两位数，以满足用户对于这些参数的精细化调整需求。同时，功能实现中需要提供用户友好的界面，方便用户响应式地输入和调整最小支持度和最小置信度的数值，在用户进行修改后，页面需要即时反馈参数设置的变化。

3）算法实现模块

关联分类规则挖掘算法要求数据类型必须为离散型，因此需要对连续型数据进行离散化处理，同时为了减小分类器构建过程中的复杂度，需要对离散化数据值进行编码处理。而从实际中采集到的数据往往存在着格式不统一、属性值缺失、噪声点等一系列问题。尽管测试时所使用的数据是从UCI机器学习资源库中下载的，这些数据已经得到了相关人员的一些预处理，但是仍然存在数据缺失以及需要对连续型数据进行离散化处理两个问题。下面就这两个方面进行介绍。

一般来说，对于缺失值，需要分两种情况区考虑。对于缺失值较多的属性，应当毫无保留的删去，因为这样属性的存在将会对最后分类器的性能带来极大影响。对于缺失值较少的属性，则应当考虑保留。算法的预处理过程中，设定的缺失率阈值为0.5，当缺失率小于阈值时，利用缺失值所属特征的众数进行填充；对于缺失率高于阈值时，则丢弃这一列。

预处理过程中有一个问题是对连续型属性值离散化处理。所谓连续型属性值，就是该属性的取值范围并非有限集，与之相对的则是离散型特征，其取值只可能落在一个有限集合中。连续型属性值较为常见，但是在关联分类规则挖掘算法中，只能处理离散型的特征。因此需要对连续型属性进行离散化处理，将指定区间内的数值映射为同一类型。实现过程中，采用了基于信息增益的离散化方法，该算法根据数据的混乱程度找出最佳的离散化临界点。

关联分类算法的实现原理在第二章节已经进行过介绍，其中CBA算法包含CBA-RG（规则生成）和CBA-CB（分类器构建）两部分，为了更加准确地描述算法的设计过程，本处给出相关算法伪代码。CBA-M1算法伪代码如下所示：

|  |
| --- |
| 算法1 CBA-M1算法 |
| **输入**：数据集D，规则集R  **输出**：分类器C   1. *R* = sort(*R*) 2. *temp =* 3. **for** each rule *r* *R* in sequence **do** 4. **for** each case *d* *D* **do** 5. **if** d satisfies the conditions of *r* **then** 6. store *d*.id in *temp* and mark *r* if it correctly classifies *d* 7. **end if** 8. **end** **for** 9. **if** *r* is marked **then** 10. insert *r* at the end of *C* 11. delete all the cases with the ids the in *temp* from *D* 12. selecting a default class for the current *C* 13. compute the total number of errors of *C* 14. **end** **if** 15. **end** **for** |
| 1. 找出错误总数最少的第一条规则p，并且删除C中p之后的所有规则 2. 将p关联的默认类添加到C的末尾 3. **return** C |

CBA-M2算法首先对数据集样本遍历，找出正确分类该样本的最高优先级规则和错误分类该样本的最高优先级规则，并通过三个阶段对规则进行筛选，构建出最终的分类器。CBA-M2算法部分伪代码如下所示：

|  |
| --- |
| 算法2 CBA-M2算法 |
| **输入：**数据集D，规则集R  **输出：**分类器C   1. **for** each rule *r* **∈** *R* **do** 2. **if** *wRule* is marked **then** 3. *wRule*.classCasesCovered[*y*]++ 4. **else** 5. *wSet* = allCoverRules(*U*, *dID*.case, *cRule*) 6. **for** each rule *w wSet* **do** 7. *w*.replace = *w*.replace {<*cRule*, *dID*, *y*>} 8. *w*.classCasesCovered[*y*]++ 9. **end** **for** 10. **end** **if** 11. **end** **for** |

APR算法在CBA-M1算法的基础上，在每轮匹配过程中都对规则集合中余下的规则更新优先级。具体而言，对于其中的每条规则，重新计算该规则在剩余数据集中的支持度和置信度，并进行优先级排序，对于不满足最小支持度和最下置信度要求的规则进行剪枝处理，这种动态剪枝策略能够保证每次取出的规则都是高质量的。APR算法伪代码如下所示：

|  |
| --- |
| 算法3 APR算法 |
| **输入：**数据集D，规则集R  **输出：**分类器C   1. **for** each rule R in sorted rules **do** 2. **for** each dataset example d in **D** 3. **if** e is covered by R **then** 4. Add R to classifier C 5. Remove d from D 6. **continue** 7. **end** **if** 8. **end** **for** 9. **for** each remaining rule R' in sorted rules **do** 10. update support and confidence of R' based on remaining data in D 11. Re-rank rules 12. **end** **for** 13. **end** **for** 14. 找出错误总数最少的第一条规则p，并且删除C中p之后的所有规则 15. 将p关联的默认类添加到C的末尾 16. **return** C |

CMAR算法首先使用FP-growth改进算法挖掘出数据中的类关联规则，在此过程中，FP-tree需要在恰当的节点处添加该路径的类别。CMAR算法在对类关联规则进行处理时，主要采用数据覆盖率的剪枝算法。

基于数据覆盖率的剪枝方法与传统的关联分类规则挖掘算法有所不同，CBA算法每次匹配成功后会直接丢弃匹配到的数据，而CMAR算法通过设置覆盖阈值选择一部分高质量的规则进行分类。CMAR算法允许数据对象被多个规则覆盖，从而在分类时可以参考多个规则，提高分类的准确性。CMAR算法首先根据规则的优先级对规则进行排序，初始化记录覆盖次数的字典和未被充分覆盖的数据对象集合。然后，遍历排序后的规则，对于每个规则，检查其是否覆盖记录的项集，并检查记录的覆盖次数是否在预设范围内。如果规则覆盖的记录中有与规则标签匹配的记录，则保留该规则，并更新相应记录的覆盖次数。最后，对未被充分覆盖的记录统计其标签出现的次数，选择出现次数最多的标签作为默认标签。

在规则存储方面，CMAR采用了CR-tree结构来高效地存储和检索规则。CR-tree是一种前缀树结构，利用规则之间的共享性来实现紧凑存储，并且作为规则的索引结构，能够快速检索规则。在规则插入过程中，若发现存在更通用且置信度更高的规则，当前规则将被剪枝，以确保存储的规则具有最高的质量和效能。在分类过程中，CMAR算法不依赖于单一规则进行分类，而是选择一组高置信度且相关性强的规则，并分析这些规则之间的关联性。通过一种加权的卡方分析技术，CMAR能够综合考虑条件支持度和类别分布，得出更为可靠的分类结果。

4）网络请求管理模块

在Web应用开发中，前后端分离架构已经成为一种主流模式，其核心在于前端和后端通过HTTP网络通信进行数据交互。后端部分借助Flask框架搭建服务端，主要有两个方面的考虑，一是考虑到是一个基于Python的轻量级Web框架，非常适合快速构建和原型设计；二是Flask的易于扩展性和庞大的生态系统使其在开发数据密集型Web应用时显得尤为便利。通过Flask框架，开发者可以轻松地处理请求、管理路由以及集成各种第三方库，提升开发效率。

前端部分则使用Axios库进行网络请求的封装。Axios是一个基于Promise的HTTP库，支持客户端和Node.js环境，因其简单易用的API设计和丰富的功能而备受欢迎。在前后端分离架构中，前端通过Axios发送POST请求，将本地的参数或数据传递给后端的指定路由，Flask接收到请求后进行相应的处理，执行算法或业务逻辑，并将结果返回给前端，前端再通过Axios接收响应的数据，并将结果渲染到页面上，完成整个交互流程。这种架构不仅提升了开发效率，还使得应用更具可维护性和可扩展性，适应不断变化的业务需求。

5）模型管理模块

模型管理模块在关联分类规则挖掘软件中扮演着至关重要的角色。需要能够高效地保存和加载训练好的模型，从而在需要时方便地重用这些模型。例如，模型训练完成后，可以调用模型对新数据进行预测。为实现这一目标，可以利用Python中的pickle模块，将模型对象序列化后保存为文件。pickle模块不仅可以将模型对象序列化并保存到文件中，还可以在需要时将其反序列化加载到内存中，以便程序再次调用该模型。同时，模型管理模块还需要与服务器后端中存储的pickle文件进行交互，以便读取指定目录下的pickle文件并在前端页面进行展示。用户在模型管理页面选中指定模型后，后端将加载指定模型对当前数据进行分类预测，并将分类结果展示在前端页面中。

除了保存和加载模型之外，模型管理模块还需要提供一系列操作，例如重模型命名、重命名以及删除模型等功能。这些操作能够满足系统用户在模型管理方面的各种需求，使模型管理更加灵活和便捷。例如，当某个模型经过优化后需要更新名称以区分不同版本，或者当某个模型不再需要使用时，可以将其删除以节省存储空间。此外，为了与其他模块和外部服务进行交互,模型管理模块还应该提供相应的交互接口。这些接口可以为服务器提供相关的服务，并允许其他系统组件轻松地与模型管理模块进行通信和协作。

总之，模型管理模块不仅需要确保模型的有效保存和加载，还需要支持对模型的全面管理和维护，包括重命名、删除以及信息的持久化存储。同时，通过提供友好的交互接口，确保模型管理模块能够无缝地与其他系统组件和外部服务进行协作，从而为整个系统提供可靠和高效的模型管理服务。

### 3.3.2 页面设计

页面设计的主要目标是提供一个简洁易用的界面，方便用户能够上传文件并选择不同的算法进行处理。页面设计遵循简洁明了、响应式设计和一致性的原则，确保用户能够轻松理解并使用各项功能。页面布局包括左侧的操作区和右侧的展示区。操作区包括文件上传按钮、多个算法选择按钮、模型的保存和管理以及参数设置滑动栏，而展示区用于显示数据和算法运行结果。页面设计注重界面的美观和易用性，同时还注重用户体验的连贯性和流畅度，采用简洁明了的设计风格和实时反馈机制，使用户能够轻松地完成各项操作，并对系统的运行状态有清晰的了解，从而提高用户满意度和系统使用效率。

通过线框图和原型设计，确定各个组件的位置和功能，确保用户可以方便地使用软件对挖掘出数据中的类关联规则并实现对数据的分类。软件首先需要实现文件上传按钮，用于实现上传本地的数据集。其次软件中不同的算法应单独对应一个按钮，按钮中注明当前软件绑定的算法，用于实现不同算法挖掘当前数据集的类关联规则以及实现分类。同时，还应实现模型管理模块，该模块由“保存模型”和“模型管理”两个按钮组成，分别负责实现保存当前模型以及管理之前已经存储的模型，从而实现对模型的复用。最后，软件需要实现测试功能，可以展示不同算法对当前数据集的分类结果，并以图表等形式对算法结果进行展示。

用户界面设计强调易用性和一致性，按钮和滑动条使用统一的颜色和风格，用户可以直观地看到当前设置的“最小支持度”和“最小置信度”参数。在用户体验设计方面，通过实时反馈机制，当用户上传文件或调整参数时，展示区会立即更新，提升用户交互的顺畅度和满意度。

## 3.4 软件实现

关联分类规则挖掘软件采用前后端分离架构，前端基于Vue3框架开发，负责实现数据的上传与展示，用户操作以及算法结果的可视化功能，后端采用Flask框架搭建，负责实现算法的主体功能和网络通信。

软件的页面主要基于Vue 3框架进行开发，它以其响应式和组件化的特性而闻名，使开发人员能够更轻松地构建动态且高效的用户界面。Vue3 引入了一系列创新功能，包括 Composition API、Teleport、Suspense 等，大大提升了开发者的工作效率和应用性能。其更快的渲染速度、更小的依赖包以及更强大的 TypeScript 支持，使得 Vue3 成为构建现代 Web 应用的理想选择。因此，基于以上的优势，软件选取Vue3实现前端页面，以确保软件的稳定性、高效性和美观性。

软件界面整体分为操作区和展示区两大部分，这两部分密切配合以实现软件的所有功能。利用Vue3的响应式编程和组件化的特性，可以实现功能的模块化开发，并且使用Pinia库实现状态管理，用于实现不同组件之间的数据通信。该框架的组合式API使状态管理更加直观和灵活，极大适应了复杂交互页面的需求。前端部分实现了数据上传按钮、参数设置选项、算法选择按钮、多算法对比以及模型管理的在内的多个界面组件。在页面开发中，结合了Element-Plus组件库和ECharts可视化库来实现数据结果的动态可视化展示，确保用户能够上传数据，选择合适的参数和算法，并且能够清晰地看到算法运行结果。

前端页面的操作栏部分主要负责用户操作和参数设置，包含多个交互组件。用户可以通过上传文件按钮选择本地文件进行上传。点击该按钮后，会弹出一个文件选择面板，用户可以在其中选择要上传的文件。确认文件后，文件将上传到服务器。同时，页面右侧的数据展示部分将显示上传文件的预览内容。左侧操作栏包含两个滑动栏，可以使用鼠标最小支持度和最下置信度参数。此外，操作栏还包括选择不同算法的按钮。点击按钮后，将会触发监听事件，向服务器发送POST网络请求以及相关参数，服务器会对已上传的数据进行一系列处理，包括数据清洗、数据离散化和数字编码等一系列数据预处理操作。预处理后的数据将被传入封装好的算法模块，算法处理后的结果将通过网络响应返回给客户端，客户端接收后将结果渲染到数据展示区，方便用户查看算法运行结果。用户触发事件后当响应结果未返回时，将展示加载页面提醒用户。

软件的后端部分由Flask框架搭建，Flask框架以简洁、灵活著称，提供了构建Web应用所需的基本功能，同时保持了良好的扩展性。Flask不会强加任何特定的项目结构或依赖关系，开发者可以根据自己的喜好和项目需求来组织代码。基于以上的优势，后端选取Flask框架实现前后端的HTTP通信。

对于模型的管理模块，使用Python的pickle模块将分类器的参数保存为pickle文件，并存储在指定目录下。当用户点击“模型管理”按钮后，后端将读取指定目录下的pickle文件并进行展示，用户选择指定模型后，将读取对应pickle文件中的分类器参数对数据集进行分类预测，并将结果返回到前端进行展示。

## 3.4 本章小结

本章主要描述了关联分类规则挖掘软件的设计和实现过程。在需求分析部分，确定了软件需具备文件上传、参数设置、算法选择、模型管理和结果展示等核心功能。总体设计部分采用前后端分离架构，用于提高系统的灵活性和可维护性。详细设计部分深入探讨了五个功能模块的实现方法，并对软件页面进行了详细设计。在软件实现部分，详细介绍了软件实现所需的技术栈以及软件的具体实现功能，对软件实现过程进行了详细介绍。

# 第4章 软件展示与测试

关联分类规则挖掘软件采用前后端分离架构，前端基于Vue3框架开发，负责实现数据的上传与展示，用户操作以及算法结果的可视化功能，后端采用Flask框架搭建，负责实现算法的主体功能和网络通信。

## 4.1 软件展示

软件整体分为操作区和展示区两部分，操作区负责实现用户的算法操作和模型保存等功能，并提供两个滑动栏允许用户调节最小支持度和最小置信度参数，调整后的参数数值应在页面中实时显示。调整支持度和置信度的滑动栏时，用户可以使用鼠标进行修改，当鼠标无法满足精细化需求时，也可以使用键盘中的上下箭头按键对参数进行微调。

数据展示部分则负责展示上传文件的原始数据和算法处理结果。用户上传文件后，原始数据将显示在此区域。点击算法按钮后，展示区域的上半部分为算法执行过程中预处理后的数据，包括对连续型数据进行离散划分、填补缺失值等处理。展示区域的下半部分为算法的执行结果，包括分类器中的类关联规则和算法运行结果的评价指标。软件的初始页面如图4.1所示。

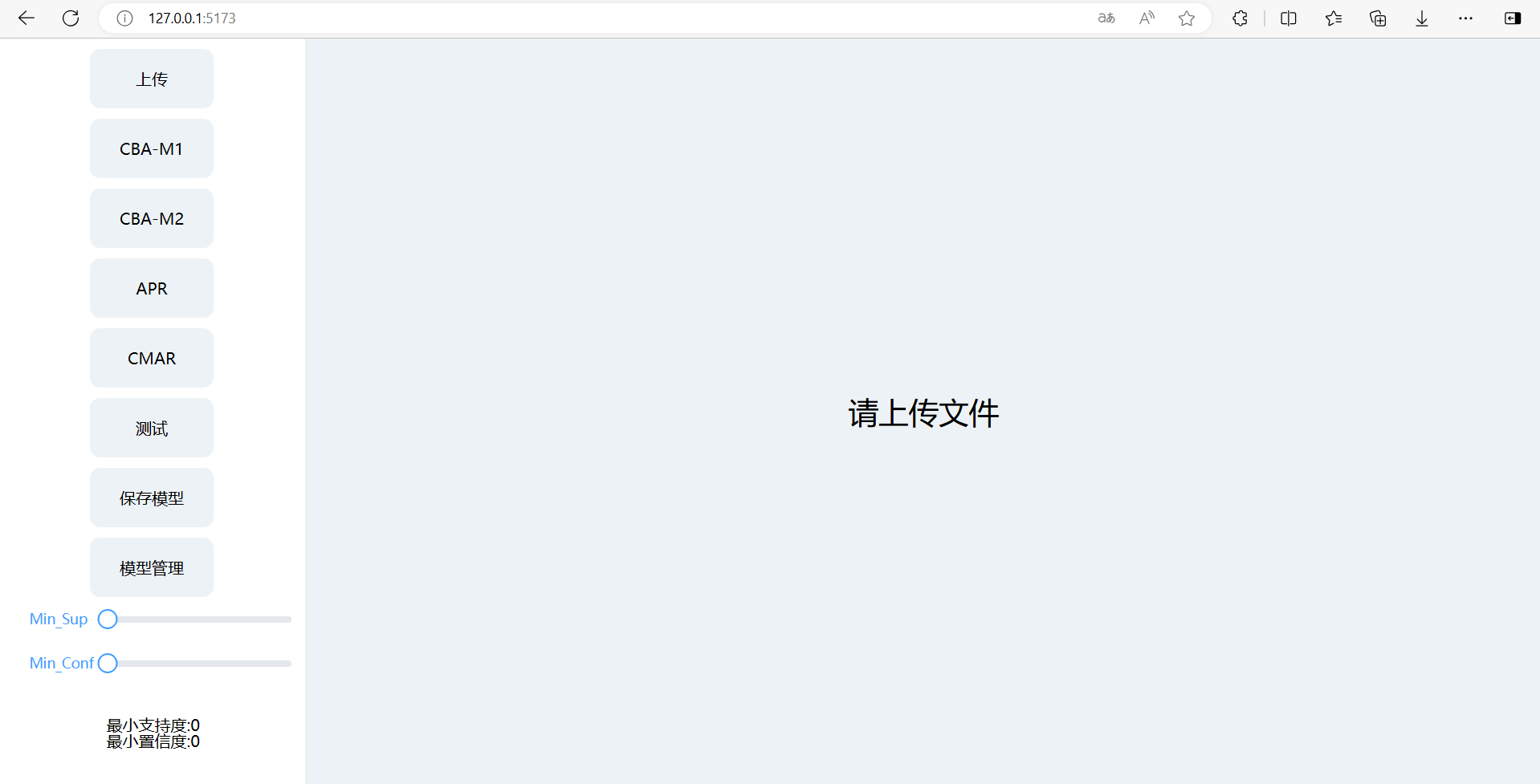


图4.1 软件初始界面

用户点击“上传”按钮后，可以选择本地csv文件进行上传，文件数据将在软件页面右侧的展示区进行展示，此处以UCI机器学习数据库中的鸢尾花（iris）数据集为例，软件页面如图4.2所示。

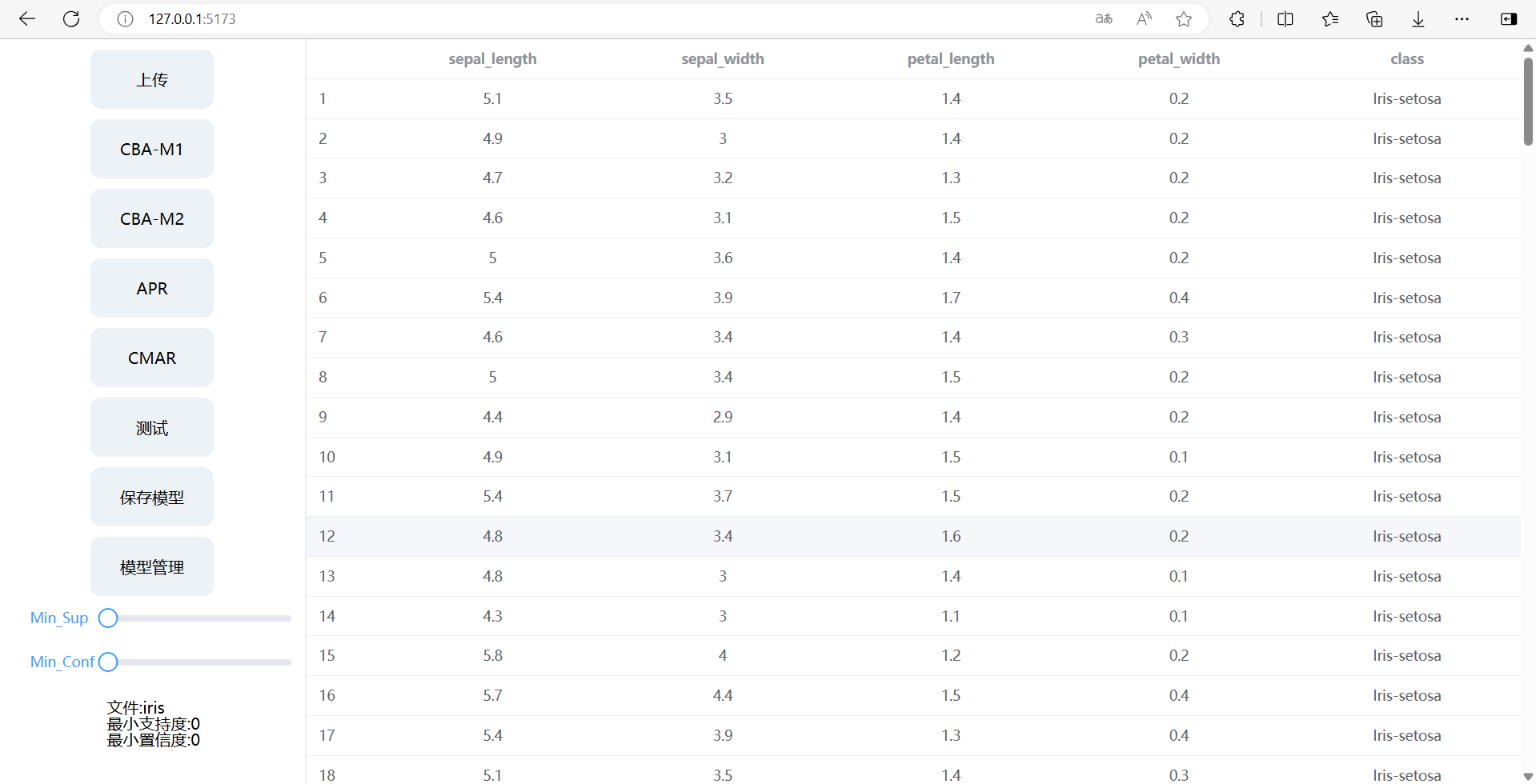


图4.2 iris数据集展示

使用鼠标操作最小支持度和最小置信度滑动栏，设置支持度为0.01，置信度为0.5，点击“CBA-M1”等对应算法的按钮，右侧将展示对应的算法运行结果以及算法运行结果的评价指标，如图4.3所示。

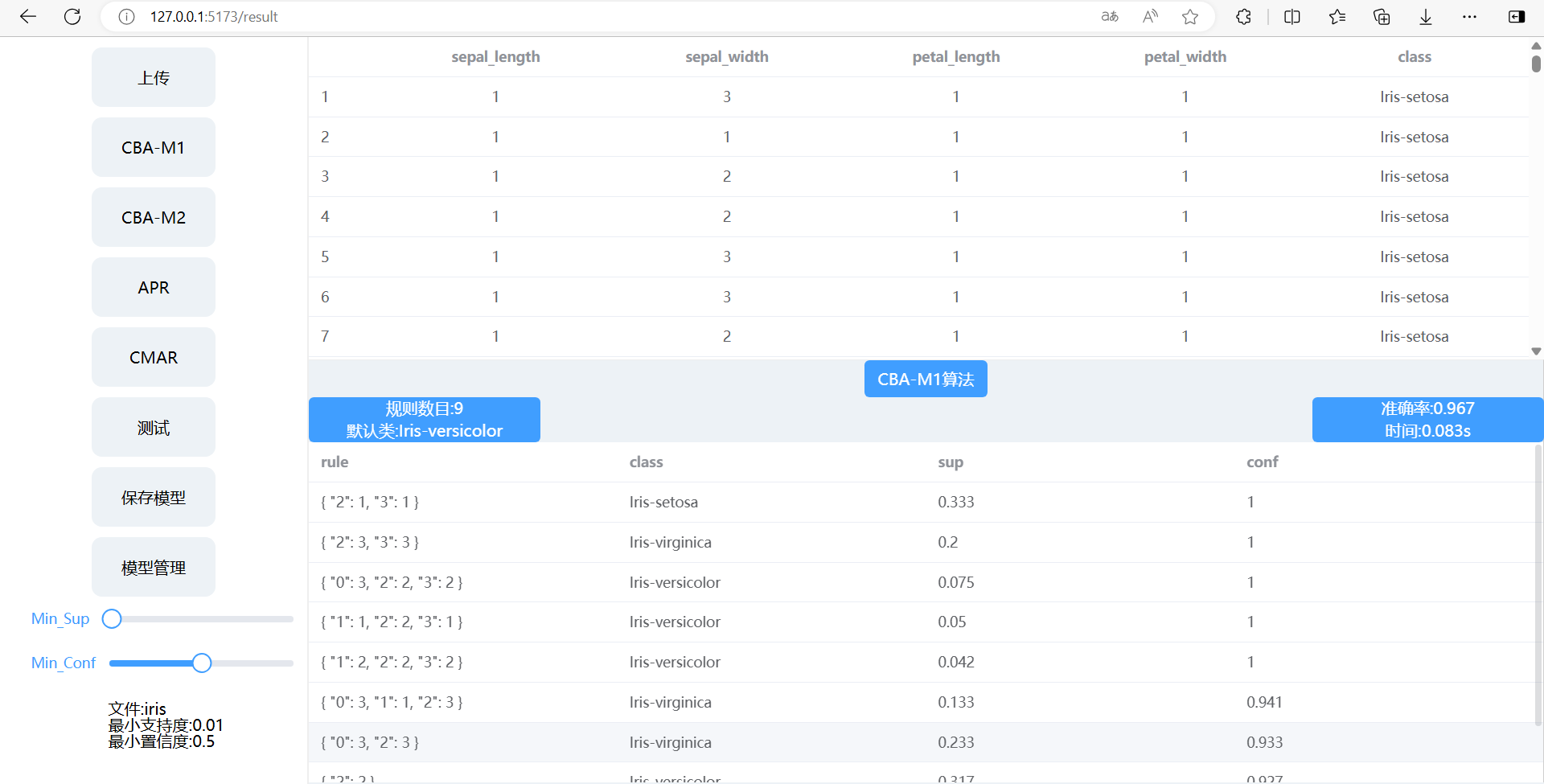


图4.3 CBA-M1算法运行结果

为了直观地比较不同算法，在页面中添加了测试选项，点击“测试”按钮后，后端算法模块将对数据集进行四种算法的十折交叉验证，最后取其平均值作为每种算法在当前数据集下的运行结果。前端部分接收到运算结果后，页面中将通过图表对结果进行可视化展示，包含准确率、规则数目、时间开销三种指标。结果展示包含多种展示方式，默认展示柱状折线图，如图4.4所示。

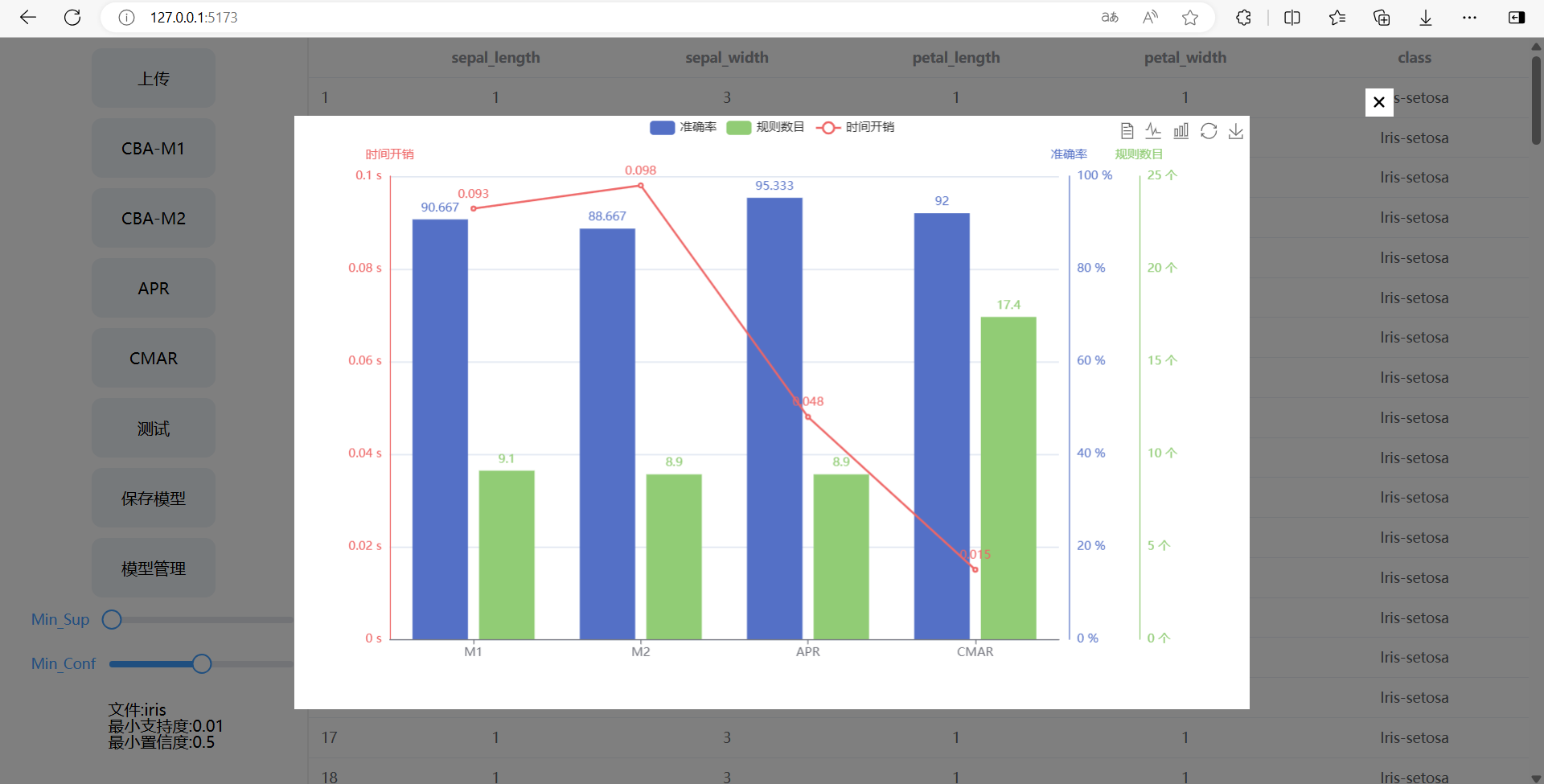


图4.4 四种算法测试结果的柱状折线图

同时，软件提供了多种展示方式，包括柱状图、折线图和表格，点击右上方的下载按钮可以下载图片来保存结果，此处以表格统计为例，结果如图4.5所示。

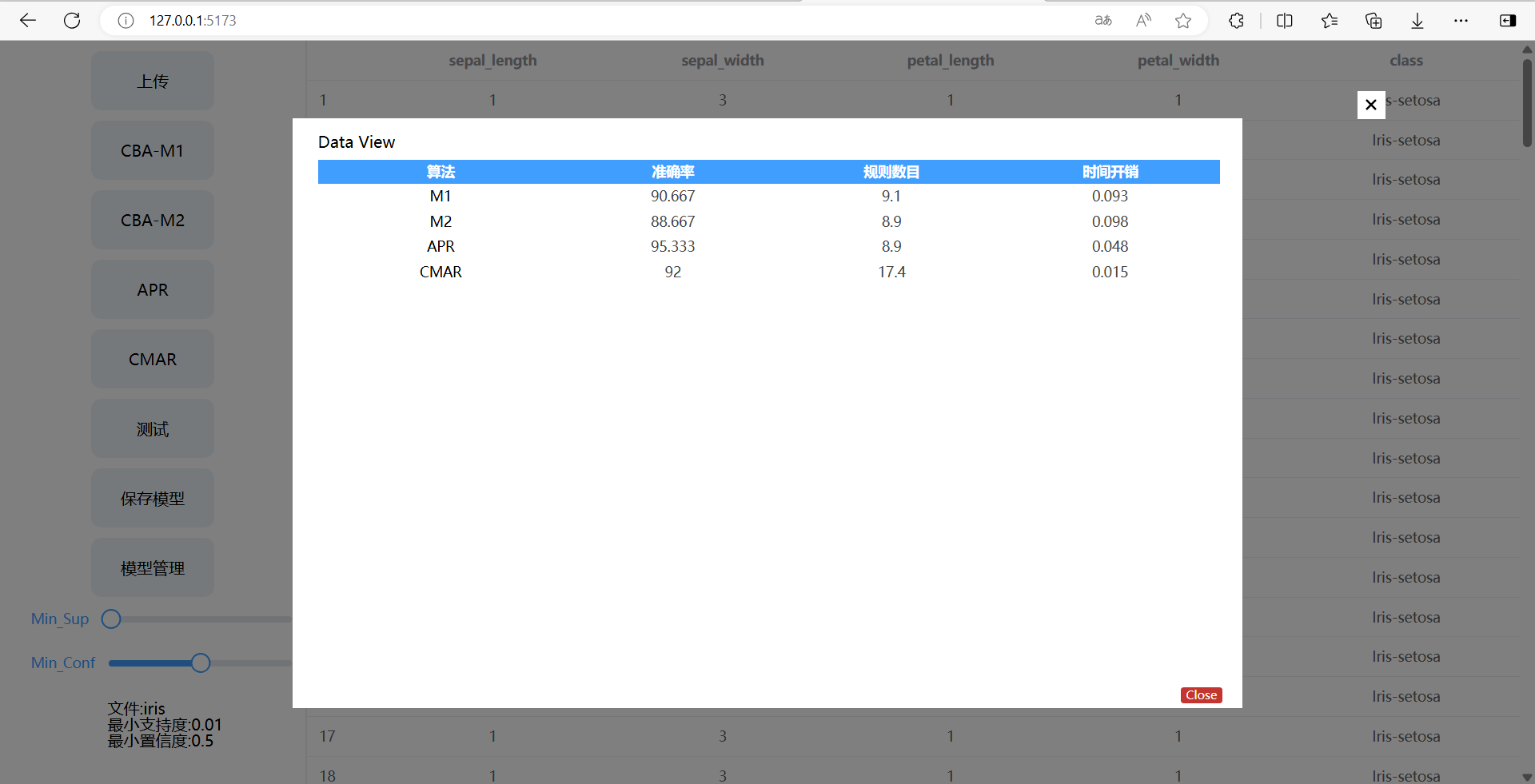


图4.5 四种算法测试结果的统计图

当用户需要保存当前模型时，可以点击左侧的“保存模型”按钮来保存，模型默认名为算法名称和数据集名称拼接而成，用户也可以通过输入框对保存的模型自定义命名，如图4.6所示。

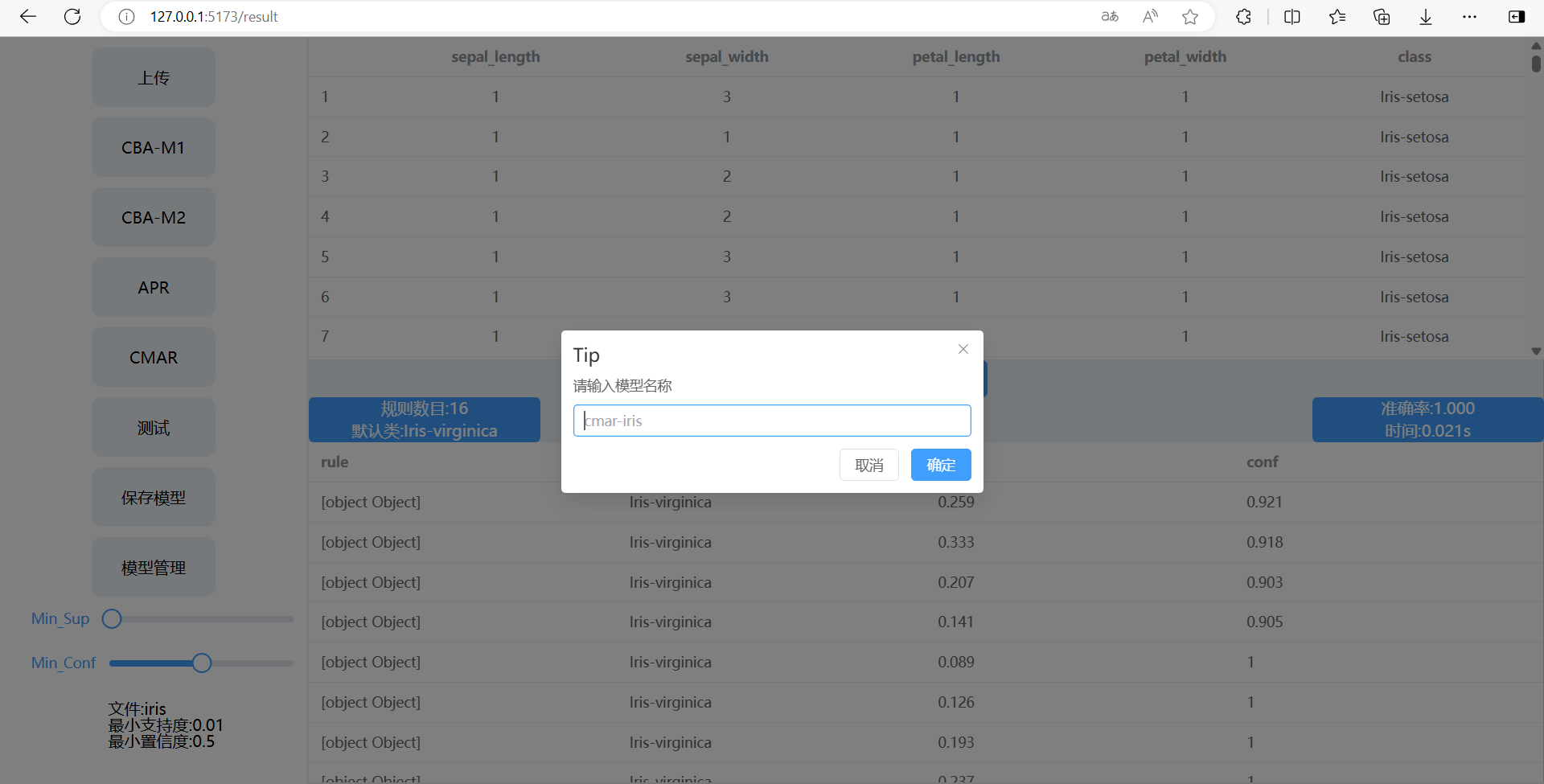


图4.6 保存模型

用户也可以选择之前保存的模型对当前数据进行处理，点击“模型管理”按钮后，将展示保存的已有模型。这些模型以pickle文件的方式存储模型的参数，用户可以自定义选择其中的一个模型对参数进行读取，从而对数据进行分类预测，如图4.7所示。

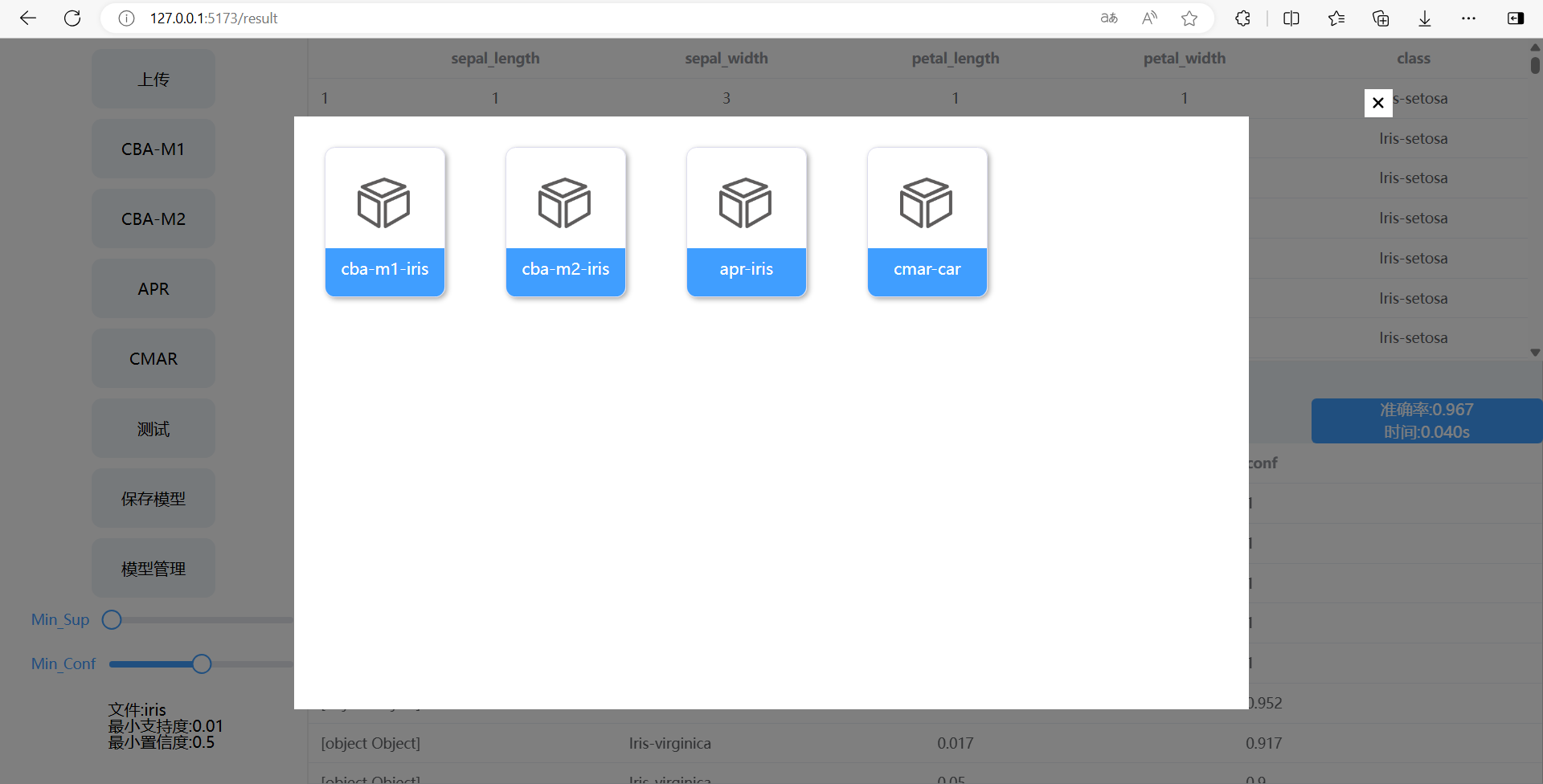


图4.7 模型管理界面

同时，软件不仅注重软件的整体功能，还关注软件使用中的细节问题。当数据量过大时，可能会出现发送网络请求后结果长时间未返回的情况，因此，软件中需要对用户的操作做出及时的响应，以此来优化用户体验。

在发出网络请求但响应结果未返回时，系统会显示一个加载页面，提醒用户已正确发送数据，后端部分仍在处理中。此时，页面中会显示网络加载提示，通过简单的动画和“Loading”字样，让用户知道已经正确发送数据，系统正在处理请求，如图4.8所示。

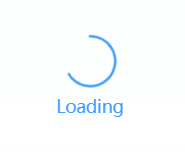


图4.8 加载页面

当用户未连接网络，或者网络发生故障导致网络请求无法发送时，系统会出现一个提示页面，及时为用户做出反馈。通过这种即时的错误提示机制，不仅能够显著提升用户体验，还能帮助用户迅速定位问题并采取相应措施，减少操作过程中的失误，提示页面如图4.9所示。



图4.9 提示页面

用户保存当前模型时，若模型已成功保存，页面中也将出现提示框进行通知。通过合理的界面设计和即时的反馈机制，能够使用户在使用关联规则分类软件时获得更好的体验，帮助用户更好地理解和操作软件，从而减少困惑和操作失误，使用户与软件之间的交互更加顺畅和高效。

## 4.2 软件测试

软件测试的目的是找出软件中潜在的各种错误和缺陷，通过修正各种错误和缺缺陷提高软件质量。软件实现过程中，对软件进行了白盒测试和黑盒测试，白盒测试则侧重于内部逻辑的正确性，黑盒测试主要关注软件的功能性。在算法实现模块，对每种算法进行了白盒测试，确保所有的代码路径都能被测试到，尤其算法中逻辑较为复杂的部分。在软件页面中进行了黑盒测试，测试过程中设计了一系列测试案例，涵盖所有用户可能的输入和操作路径，以确保软件在所有预定的功能上都能按预期工作。经过测试后，软件执行过程中发现的错误均已改正。

软件初步构建时，测试了软件的边界条件处理能力，比如上传错误类型的文件，以确保软件能妥善处理异常输入；当软件被手动修改跳转到其他路由时，软件能够自动检测异常情况并进行处理。

每种算法实现过程中，编写了测试案例来检测算法运行结果，包括算法执行前的预处理模块，单独采用含缺失值的测试样例，确保每一过程的实现符合预期。在前后端交互过程中，测试了参数信息能否被正确接收等，避免数据传输过程中发生错误。对软件进行结构化测试之后，对对软件的各项功能进行了全面的测试。例如在实现界面中，上传了标准的iris数据集，并使用四种不同的算法模型进行了处理。测试中包括检查算法是否能正确识别和分类iris数据集中的不同种类，同时确保算法的结果展示符合预期。

经过多轮测试，主要发现并修正了以下几个问题。上传大型数据集时，软件响应较慢，造成卡顿。之后在软件中使用懒加载技术，仅在需要时加载数据，减少内存使用；实现CMAR算法时，由于使用卡方计算时的逻辑错误，导致分类结果异常。之后对代码进行审查，查阅文档，重新编写该部分代码；使用滑动栏需修改参数时，页面中显示的参数与实际参数不一致。发现是页面修改的参数没有进行响应式处理，页面参数的改变并没有引起具体值得改变，最终使用ref函数进行响应式处理。点击“测试”按钮后，前端可以接收到后端返回的结果，但是展示部分出现了数据混乱局面，之后通过查看ECharts官网中的教程文档，重新梳理并修改参数，问题得以解决。

通过上述多次测试，不断发现并解决了多种潜在问题，逐步提升了软件的质量和用户体验，最终软件的各项功能都能按照预期运行。

## 4.3 本章小结

本章介绍了关联分类规则挖掘软件的实现和测试。软件页面基于Vue 3框架实现，前端分为操作区和数据展示两部分，使用Element-Plus和ECharts实现界面和数据可视化。操作栏支持数据上传、参数设置和算法选择，数据展示部分显示上传的数据和算法结果。软件封装了CBA-M1、CBA-M2、APR、CMAR四种算法，确保处理不同数据集并输出分类结果。软件添加了测试和模型管理功能，允许用户对四种算法进行十折交叉验证继而对算法性能进行分析，以及保存模型并对已有的模型进行管理。测试方面进行了白盒和黑盒测试，确保算法逻辑正确和软件功能正常，发现并修正了软件测试过程中存在的多个问题。

经过多轮测试和优化，软件达到了预期功能目标，有效执行多种算法并直观展示结果，提升了用户的操作便利性和数据分析能力。

# 第5章 算法分析实验

本文实现的软件中算法部分使用Python编写，从UCI机器学习数据库中选取10个数据集对程序进行测试，设置的支持度阈值为0.01，置信度阈值为0.5[3]。所有指标均采用10折交叉检验，取10次检验结果的平均值来评估其总体性能指标。

选取的十个数据集详情如表5.1所示。

表5.1 数据集详情

|  |  |  |  |
| --- | --- | --- | --- |
| 数据集 | 数据量/（条） | 属性数目/（个） | 类别数目/（个） |
| car | 1728 | 7 | 4 |
| diagnosis | 120 | 7 | 4 |
| iris | 150 | 5 | 3 |
| monks | 432 | 7 | 2 |
| vertebra | 310 | 7 | 2 |
| tic-tac-toe | 958 | 9 | 2 |
| seeds | 210 | 8 | 3 |
| mammographic | 961 | 6 | 2 |
| qualitative-bankruptcy | 250 | 7 | 2 |
| user-knowledge-modeling | 258 | 6 | 4 |

这些数据集均来自UCI机器学习数据库，是各类研究和应用中的经典数据集，涵盖了生物信息学、医疗诊断、农业和汽车工业等多个领域。详细介绍如下：

1. car

该数据集包含有关汽车评估的信息，基于多种属性进行分类，包括购买、维护成本、车门数量、乘客数量、后备箱大小和安全性等属性，每条记录被分类为 unacc、acc、good或vgood。

1. diagnosis

该数据集包含详细的诊断信息，涵盖多种医疗属性，包括患者的体温、症状发生率、是否有腰痛、排尿推动感、排尿疼痛感、皮肤瘙痒等属性，以及一个表示诊断结果的类属性，帮助医生和研究人员分析疾病特征和诊断结果的关系。

1. iris

该数据集描述了鸢尾花的测量数据，如萼片长度、萼片宽度、花瓣长度和花瓣宽度，每朵花被分类为三个物种之一：Iris-setosa、Iris-versicolor 或 Iris-virginica。

1. monks

该数据集用于测试和验证机器学习算法的性能，包含 a1 到 a6 等属性，以及一个表示结果的类属性，每条记录被分为两个类别。

1. vertebra

该数据集描述了各种椎骨特征，包括骨盆倾角、骨盆倾斜、腰椎前凸角、骶骨斜度、骨盆半径和脊柱滑脱等级等属性，每条记录被分类为 AB、NO 等类别。

1. tic-tac-toe

该数据集包含井字棋游戏的结果，每条记录描述棋盘的状态，包括top-left-square、bottom-right-square等属性，每条记录被分类为positive或者negative，用于研究模式识别和游戏策略分析。

1. seeds

该数据集包括小麦种子的各种测量数据，如面积、周长、紧凑性、长度、宽度、不对称系数和内核槽长度，每条记录被分类为 1、2 或 3 类，用于农业研究和植物分类的算法测试。

1. mammographic

该数据集提供了有关乳房肿块的信息，包括 BI-RADS 评估、年龄、形状、边缘和密度等属性，每条记录被分类为 0（良性）或 1（恶性）。

1. qualitative-bankruptcy

该数据集包含有关公司的定性信息，具有工业风险（IR）、管理风险（MR）、财务灵活性（FF）、信誉（CR）、竞争力（CO）和经营风险（OR）等属性，每条记录被分类为 B（破产）或 NB（非破产）。

1. user-knowledge-modeling

该数据集关注用户的知识水平，具有 STG、SCG、STR、LPR 和 PEG 等属性，每条记录被分类为 very\_low、low、middle和 high 等类别，用于教育研究和个性化学习系统的开发。

## 5.1 分类准确率

在分类任务中，准确率是衡量算法性能的一个关键指标，针对每个数据样本，正确分类的准确率为分类器正确分类以及默认类别正确分类的准确率之和。本文使用关联分类规则挖掘软件对UCI机器学习数据库中的十个数据集进行了测试，测试结果如表5.2所示。

表5.2 分类结果准确率

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 数据集 | 准确率/(%) | | | |
| CBA-M1 | CBA-M2 | APR | CMAR |
| car | 75.584 | 71.757 | 80.344 | 76.43 |
| diagnosis | 80.833 | 77.5 | 80.833 | 95 |
| iris | 94.667 | 90.667 | 95.333 | 92.667 |
| monks | 100 | 94.238 | 100 | 50.419 |
| vertebra | 68.71 | 82.581 | 84.194 | 75.484 |
| tic-tac-toe | 79.807 | 77.183 | 82.14 | 80.26 |
| seeds | 88.571 | 88.571 | 90 | 92.381 |
| mammographic | 82.516 | 76.797 | 82.097 | 74.294 |
| qualitative-bankruptcy | 100 | 97.6 | 100 | 98.8 |
| user-knowledge-modeling | 81.964 | 78.679 | 84.376 | 83.879 |

如表5.2所示，APR算法在大多数数据集上表现出较高的准确率，特别是在car、iris、user-knowledge-modeling和vertebra数据集上表现尤为突出。而CMAR算法在diagnosis和seeds数据集上表现最佳，CBA-M1和CBA-M2算法的表现则相对较弱。具体而言，APR算法在整体测试中都显现出较高的准确率，这表明其所采用的动态剪枝策略具有显著的提升效果。

APR算法通过动态剪枝，能够实时筛选和调整类关联规则的优先级，从而确保分类器只保留高质量的规则。这种机制不仅提高了分类器的准确率，还增强了其在不同数据集上的适应性和稳定性。CMAR算法基于卡方检验的加权机制，该机制在分类过程中考虑了规则的重要性和相关性，能够在分类过程中考虑规则的重要性和相关性，特别适合分类数据分布不均衡的数据集，这种机制使得CMAR算法在一些类别分布和特征相关性较强的数据集上表现出色，能够充分发挥其优势，确保分类结果的准确性和可靠性。相比之下，CBA-M1算法和CBA-M2算法在规则生成和筛选过程中相对简单，对复杂模式的捕捉能力有限。这也导致它们在处理复杂数据模式时准确率较低。

从准确率方面来看，APR算法和CMAR算法通常表现较好，适用于对分类精度要求较高的场景。APR通过全面的模式捕捉和严格筛选，CMAR通过加权机制有效处理复杂类别分布。

## 5.2 模型构建及预测时间

统计的时间开销包含模型构建以及分类预测的时间之和，模型构建及预测时间开销如表5.3所示。

表5.3 模型构建及预测时间

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 数据集 | 时间开销/（s） | | | |
| CBA-M1 | CBA-M2 | APR | CMAR |
| car | 7.201 | 10.199 | 2.866 | 0.839 |
| diagnosis | 0.094 | 0.19 | 0.083 | 0.147 |
| iris | 0.088 | 0.092 | 0.046 | 0.013 |
| monks | 1.986 | 2.208 | 1.725 | 0.446 |
| vertebra | 4.947 | 4.859 | 3.672 | 0.157 |
| tic-tac-toe | 1.274 | 1.287 | 1.357 | 1.268 |
| seeds | 8.869 | 8.587 | 1.289 | 1.432 |
| mammographic | 2.181 | 2.258 | 2.118 | 0.089 |
| qualitative-bankruptcy | 4.504 | 4.825 | 1.119 | 0.577 |
| user-knowledge-modeling | 0.004 | 0.005 | 0.004 | 0.073 |

从表5.3可以看出，CMAR算法在大多数数据集上表现出较快的计算速度，APR算法次之，CBA-M1和CBA-M2算法的时间开销则相对较大。

CMAR算法的时间效率与其使用的FP-growth算法和CR-tree结构密切相关。FP-growth算法通过构建频繁模式树（FP-tree），能够高效地发现频繁项集，这种结构显著减少了数据扫描次数，并通过压缩数据表示来提升计算效率。同时，CMAR算法利用CR-tree（Classification Rule-tree）来组织和存储生成的分类规则，这种树结构不仅有助于快速查找和匹配规则，还能有效地剪枝，避免不必要的计算。正是这种高效的数据结构和算法设计，使得CMAR在时间开销上表现优异。

相比之下，APR算法虽然在准确率上表现出色，但其规则生成过程中需要遍历大量的候选项集，计算频繁度和置信度。因此，在处理大规模数据时，APR算法比CMAR算法的时间开销高。不过，APR算法通过剪枝和优化策略，依然在多数情况下表现出较好的时间效率。CBA-M1和CBA-M2算法在时间开销上表现较差，主要原因在于其规则生成和筛选策略相对简单直接，缺乏高效的剪枝机制。这使得在处理大规模数据时，生成和筛选规则的时间显著增加。此外，CBA-M2由于其基本计算步骤较多，导致总体时间开销较高。

数据的规模和复杂性也对时间开销产生了显著影响。在较小且简单的数据集（如iris和user）上，各算法的时间差异不大；而在较大且复杂的数据集上，CMAR算法的时间优势更为明显。

从时间开销来看，CMAR算法在多数情况下速度更快，适合处理大规模数据且对速度有要求的任务。其高效的规则筛选机制显著减少了计算量。

## 5.3 类关联规则数目

表5.4 类关联规则数目

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 数据集 | 规则数目/(个) | | | |
| CBA-M1 | CBA-M2 | APR | CMAR |
| car | 24.6 | 8.2 | 12.9 | 29.7 |
| diagnosis | 4 | 2.9 | 4 | 62.3 |
| iris | 9.3 | 9.2 | 9 | 15.6 |
| monks | 22.3 | 21.2 | 25.8 | 21.1 |
| vertebra | 15.3 | 9.6 | 11.8 | 74.4 |
| tic-tac-toe | 57 | 18.9 | 30.9 | 169 |
| seeds | 23 | 18.4 | 16.1 | 140.9 |
| mammographic | 63.1 | 15.9 | 31.7 | 12.5 |
| qualitative-bankruptcy | 10 | 8.2 | 5.8 | 72.4 |
| user-knowledge-modeling | 4.9 | 3.2 | 5.9 | 24.6 |

如表5.4所示，CBA-M2算法整体上生成的规则数目最少，APR次之，CBA-M1稍多，CMAR生成的规则数目最多，分类器中的类关联规则数目直接影响了模型的复杂性。

CBA-M2算法在规则生成过程中采用了更严格的筛选标准，虽然可能牺牲一部分准确率，但确保了生成规则的简洁性。严格的筛选机制减少了冗余规则，有助于提升模型的可解释性和应用效率。CBA-M1算法在规则生成上相对宽松，导致生成的规则数目较多。宽松的规则生成策略虽然提高了覆盖范围，但也增加了冗余规则，影响了模型的简洁性。APR算法在CBA-M1算法的基础上引入的动态剪枝策略，使得APR在规则数量少于CBA-M1。CMAR算法由于采用了基于卡方检验的加权机制，生成了大量规则。这种机制在提高分类准确率的同时，也增加了规则数目，导致模型复杂度较高。虽然这种方法能够在复杂数据集中取得较高的准确率，但规则数目的增加也带来了模型的解释难度和应用负担。

从规则数目来看，CBA-M2算法生成的规则数目最少，适用于对模型简洁性要求较高的应用场景。

## 5.4 本章小结

本章使用关联分类规则挖掘软件对四种主要算法（CBA-M1、CBA-M2、APR、CMAR）的性能进行了全面分析，重点关注了准确率、时间开销和规则数目三个维度。通过在十个不同数据集上的测试结果，发现各算法在不同数据集上的表现各有优劣。APR算法在大多数数据集上表现出较高的准确率， CMAR算法在部分数据集上表现较佳，但其生成的规则数目较多，导致模型复杂度增加。在时间开销方面，CMAR算法在大多数数据集上表现出较快的计算速度，其时间效率得益于其使用的FP-growth算法和CR-tree结构，这种高效的数据结构和算法设计显著减少了计算量，使得CMAR在时间开销上表现优异。在规则数目方面，CBA-M2算法整体上生成的规则数目最少，APR次之，CBA-M1稍多，CMAR生成的规则数目最多。在实际应用中，用户应根据具体需求综合考虑准确率、时间开销和模型简洁性等因素，选择最适合的算法。经过本章的测试，对于不同的算法的表现也有了更加客观的认识。

# 第6章 总结与展望

## 6.1 总结

本文围绕关联分类规则挖掘软件的设计与实现，进行了系统性研究和分析，取得了一些显著的成果。选择并实现了CBA-M1、CBA-M2、APR和CMAR四种关联分类算法。这些算法在关联规则挖掘和分类任务中具有代表性，能够有效地处理不同类型的数据集。特别是在APR算法的实现过程中引入动态剪枝策略，提高了规则筛选的效率和分类准确性。这种优化不仅减少了冗余规则，还显著提升了算法的运行速度和准确性。

软件采用前后端分离的架构进行开发，前端使用Vue3框架，后端使用Flask框架。这种架构设计提高了软件的扩展性和维护性，前端实现了数据的上传与展示、用户操作和算法结果的可视化功能，后端则负责数据处理和算法实现。通过这种设计和模块化开发，软件能够灵活应对用户的不同需求，提供高效的数据处理和结果展示。

软件在界面展示中注重数据可视化和用户体验的优化。通过使用ECharts可视化库，软件能够直观地展示不同算法的运行结果，包括分类准确率、运行时间和规则数目等算法评价指标。用户界面设计简洁明了，操作便捷，提升了用户体验。软件提供了参数调节功能，允许用户根据具体需求调整最小支持度和最小置信度参数，从而灵活控制算法的执行。最后，为了实现对模型的复用，软件还实现了模型管理功能，可以选择当前的模型进行保存以及选择已保存的模型对数据集进行分类预测。

通过对多个UCI公开数据集的实验测试，本文详细分析了四种算法在准确率、时间开销和规则数目等方面的性能表现。实验结果显示，APR算法在分类准确率高的同时，也拥有较少的时间开销和规则数目，综合性能优异;CMAR算法在时间效率上具有明显优势;而CBA-M2算法在生成规则数目上占优，构造的分类器最为简洁。通过这些分析，为实际应用中的算法选择提供了客观依据和指导。

## 6.2 后续工作展望

尽管本论文在关联分类规则挖掘软件实现取得了一定的进展，但仍有许多值得进一步探索和改进的方面。未来的研究工作可以从以下几个方面展开：

1. 分布式处理：

现有算法在处理大规模数据集时仍存在效率问题。未来的研究可以进一步优化这些算法的计算效率和分类效果，探索更高效的数据结构和优化策略，例如采用Hadoop、MapReduce等技术对类关联规则进行并行计算和分布式处理，以提升算法的可扩展性和实时性。

1. 算法创新：

可以在现有的关联分类规则挖掘算法基础上进行创新。通过改进分类器的构造方法，挖掘出更加高质量的规则，从而提高分类的准确率并减少规则的数量。同时，可以结合人工智能技术，特别是深度学习的特征提取能力和关联规则的可解释性，构建出更加高效和准确的分类器。这种结合可以充分利用两者的优势，形成更强大的数据挖掘工具。

1. 用户界面和交互设计优化：

通过改进软件的用户界面和交互设计，可以显著提升用户的使用体验。例如增加更多的参数调节选项，使用户可以根据具体需求灵活调整最小支持度和最小置信度参数，进而控制算法的执行。增强数据可视化功能，使用户能够更加直观地理解和分析分类结果，通过丰富的图表和交互手段，帮助用户更好地进行数据分析和决策支持。这不仅提升了软件的功能性，也使其更加易于使用和理解。

1. 扩展软件的实用性和应用场景：

进一步扩展关联分类规则挖掘软件的应用场景，针对具体行业（如金融、医疗、零售等）进行定制化优化，以满足不同领域的特定需求。同时，研究如何有效整合跨领域的数据集，通过综合分析提升决策的准确性和效能。通过这些方面的研究和改进，可以进一步提升关联分类规则挖掘软件的性能和实用性，为更多领域的数据挖掘和分类任务提供强有力的技术支持，最终推动数据挖掘技术在各个行业中的广泛应用和发展。

通过这些方面的研究和改进，可以进一步提升关联分类规则挖掘软件的性能和实用性，为更多领域的数据挖掘和分类任务提供强有力的技术支持，最终推动数据挖掘技术在各个行业中的广泛应用和发展。持续的创新和优化将使得关联分类规则挖掘在面对不断变化的数据环境和业务需求时，依然能够提供高效、准确的解决方案，为科学研究和商业应用带来更大的价值。

# 参考文献

1. Liu B, Hsu W, Ma Y. Integrating classification and association rule mining[C]. Proceedings of The Fourth International Conference on Knowledge Discovery and Data Mining, New York, USA, 1998: 80-86.
2. Li W, Han J, Pei J. CMAR: Accurate and efficient classification based on multiple class-association rules[C]. Proceedings 2001 IEEE International Conference on Data Mining, New York, USA, 2001: 369-376.
3. Rajab K D. New associative classification method based on rule pruning for classification of datasets[J]. IEEE Access, 2019, 7: 157783-157795.
4. 黎文娟, 周忠眉. 基于大数据与人工智能的关联分类改进算法[J]. 广东通信技术, 2022, 42(01): 63-66.
5. 秦晨普, 张云华. 基于分类修剪的关联分类算法改进[J]. 计算机系统应用, 2019, 28(4): 194-198.
6. 李家辉, 周忠眉. 基于多次学习和关联度的关联分类改进算法[J]. 南京大学学报 (自然科学版), 2019, 55(4): 564-572.
7. 刘莉萍, 李欢, 何正宏. 关联分类算法改进及其在中药“性-效”分析中的应用[J]. 现代信息科技, 2023, 7(18): 150-153.
8. 郭明环, 代素香, 吕亚娟. 关联挖掘在图书馆读者分类中的应用探索[J]. 内蒙古科技与经济, 2021, (17): 120-123.
9. 翟悦, 李楠, 于文武. 基于扩展概念格的带约束关联分类规则挖掘方法[J]. 大连交通大学学报, 2021, 42(4): 88-93.
10. 马晓剑, 张家绪, 王奥, 等. 基于概率转换函数与邓熵的关联分类融合算法[J]. 计算机工程与设计, 2023, 44(05): 1392-1399.
11. 刘影, 徐辉. 基于模糊关联的不平衡数据分类算法研究[J]. 齐齐哈尔大学学报(自然科学版), 2023, 39(04): 21-27.
12. 崔巍, 贾晓琳, 樊帅帅, 等. 一种新的不均衡关联分类算法[J]. 计算机科学, 2020, 47(S1): 488-493.
13. 王俊红, 闫家荣. 基于欠采样和代价敏感的不平衡数据分类算法[J]. 计算机应用, 2021, 41(1): 48-52.
14. Hasanpour H, Meibodi R G, Navi K. Improving rule-based classification using Harmony Search[J]. PeerJ Computer Science, 2019, 5: 188-189.
15. Alazaidah R, Samara G, Almatarneh S, et al. Multi-Label Classification Based on Associations[J]. Applied Sciences, 2023, 13(8): 5081-5083.
16. Thanajiranthorn C, Songram P. Efficient rule generation for associative classification[J]. Algorithms, 2020, 13(11): 299-302.
17. Dhanalakshmi R, Anitha K, Rukmani Devi D, et al. Association rule generation and classification with fuzzy influence rule based on information mass value[J]. Journal of Ambient Intelligence and Humanized Computing, 2021, 12(6): 6613-6620.
18. Mahgoub H, Rösner D, Ismail N, et al. A text mining technique using association rules extraction[J]. International Journal of Computer and Information Engineering, 2008, 2(6): 2044-2051.
19. Tran A, Chan P, Jones P, et al. Comparison of patient self-reported health status with clinician-assigned New York heart association classification[J]. JAMA Network Open, 2020, 3(8): 2014319-2014321.
20. Verma A, Dhalmahapatra K, Maiti J. Forecasting occupational safety performance and mining text-based association rules for incident occurrences[J]. Safety Science, 2023, 159: 106014-10616.
21. Cagliero L, Canale L, Farinetti L, et al. Predicting student academic performance by means of associative classification[J]. Applied Sciences, 2021, 11(4): 1420-1422.

# 致谢

行文至此，毕业设计已经进入尾声。从查阅相关资料、编写算法代码，到完成论文的编写，一路的酸甜苦辣已回荡在脑海之中。在这个过程中收获十分丰富，学习到了许多知识，各方面的能力也得到了锻炼。在此，首先感谢我的指导老师周应华老师，他在我毕业设计完成的过程中提供了很多宝贵建议，对于毕业论文的撰写进行了耐心的指导。

同时，我还要感谢在毕业设计期间一起交流的同学。他们的指导给予我很大的帮助，正是通过交流和探讨，我才能发现自己在某些方面的不足，从而不断完善毕业设计。

最后，再次对给予我帮助的导师和同学表示由衷的感谢。