**人工智能系统中面向模型风险的测试系统设计与实现**

摘 要

在当今的科技时代，人工智能技术的应用已经渗透到生活的方方面面，从金融预测到医疗诊断，人工智能为各行业带来了革命性的变化。然而，随着人工智能模型的广泛应用，模型的可靠性、公平性和安全性问题逐渐显现，如何有效评估和控制模型风险成为了亟待解决的问题。本研究设计并实现了一个面向模型风险的测试系统，旨在从准确性、鲁棒性和公平性三个维度全面评估人工智能系统中的模型性能。

系统采用了前后端分离的架构，前端使用Vue框架实现，后端使用SpringBoot框架，并通过MyBatis进行数据库操作。模型评估部分则采用FastAPI与PyTorch实现，提供高效的模型评估服务。系统支持用户上传数据集和模型，能够对多种图像分类和检索模型进行全面评估，并生成详细的评估报告。这些功能帮助用户发现和改进模型在实际应用中的潜在问题，提升模型的整体性能和可靠性。

**关键词：** 人工智能，模型风险，准确性，鲁棒性，公平性

**Design and Implementation of Model Risk Oriented Testing System in Artificial Intelligence Systems**

**ABSTRACT**

In today's technological era, artificial intelligence (AI) has permeated various aspects of life, revolutionizing industries from financial forecasting to medical diagnosis. However, with the widespread application of AI models, issues of reliability, fairness, and security have emerged, making effective evaluation and control of model risks an urgent problem. This study designs and implements a model risk-oriented testing system to comprehensively evaluate the performance of AI models from the perspectives of accuracy, robustness, and fairness.

The system adopts a front-end and back-end separation architecture, with the front end implemented using the Vue framework and the back end using the SpringBoot framework, with MyBatis for database operations. The model evaluation part is implemented using FastAPI and PyTorch, providing efficient model evaluation services. The system supports users to upload datasets and models, comprehensively evaluate various image classification and retrieval models, and generate detailed evaluation reports. These functions help users identify and improve potential issues in model applications, enhancing the overall performance and reliability of models.

**Key words：**Artificial Intelligence, Model Risk, Accuracy, Robustness, Fairness

目 录

[1 引 言 1](#_Toc167889244)

[1.1 研究背景及意义 1](#_Toc167889245)

[1.2 主要研究内容 1](#_Toc167889246)

[1.3 论文组织结构 2](#_Toc167889247)

[2 相关理论与技术概述 3](#_Toc167889248)

[2.1 准确性的评估 3](#_Toc167889249)

[2.2 鲁棒性的评估 6](#_Toc167889250)

[2.2.1对抗性示例 6](#_Toc167889251)

[2.2.2 对抗性攻击 6](#_Toc167889252)

[2.2.3 鲁棒性评估指标 7](#_Toc167889253)

[2.3 公平性的评估 8](#_Toc167889254)

[2.4 相关技术概述 9](#_Toc167889255)

[3 系统需求分析 12](#_Toc167889256)

[3.1 需求分析概述 12](#_Toc167889257)

[3.2 功能性需求分析 12](#_Toc167889258)

[3.2.1 总体业务描述 12](#_Toc167889259)

[3.2.2 账号管理用例分析 14](#_Toc167889260)

[4.2.3 模型测试用例分析 16](#_Toc167889261)

[3.2.3 模型管理用例分析 19](#_Toc167889262)

[3.2.4 数据集管理用例分析 21](#_Toc167889263)

[4.3 系统数据建模 23](#_Toc167889264)

[4.4 非功能性需求分析 23](#_Toc167889265)

[4.4.1 性能需求 24](#_Toc167889266)

[4.4.2 可靠性需求 24](#_Toc167889267)

[4.4.3 安全性需求 24](#_Toc167889268)

[4.4.4 兼容性需求 24](#_Toc167889269)

[4 系统设计与实现 25](#_Toc167889270)

[4.1 系统架构设计 25](#_Toc167889271)

[4.2 系统类图设计 26](#_Toc167889272)

[4.2.1 账号管理类图 26](#_Toc167889273)

[4.2.2 模型测试类图 28](#_Toc167889274)

[4.2.3 模型管理类图 29](#_Toc167889275)

[4.2.3 数据集管理类图 30](#_Toc167889276)

[4.3 数据库设计 32](#_Toc167889277)

[5.3.1 用户表 32](#_Toc167889278)

[4.3.2 登陆凭证表 33](#_Toc167889279)

[4.3.3 记录表 33](#_Toc167889280)

[4.3.4 模型表 34](#_Toc167889281)

[4.3.5 数据集表 35](#_Toc167889282)

[4.4 核心功能详细设计与实现 36](#_Toc167889283)

[4.4.1 账号管理 36](#_Toc167889284)

[5.4.2 模型测试 38](#_Toc167889285)

[4.4.3 模型管理 44](#_Toc167889286)

[4.4.4 数据集管理 45](#_Toc167889287)

[5 系统测试 46](#_Toc167889288)

[5.1功能测试 46](#_Toc167889289)

[5.1.1 账号管理功能测试 46](#_Toc167889290)

[5.1.2 模型测试功能测试 49](#_Toc167889291)

[5.1.3 模型管理功能测试 52](#_Toc167889292)

[5.1.4 数据集管理功能测试 53](#_Toc167889293)

[5.2 非功能测试 54](#_Toc167889294)

[5.2.1 性能测试 54](#_Toc167889295)

[5.2.2 兼容性测试 55](#_Toc167889296)

[6 总结与展望 56](#_Toc167889297)

[6.1 总结 56](#_Toc167889298)

[7.2 展望 56](#_Toc167889299)

[参考文献 57](#_Toc167889300)

[谢 辞 59](#_Toc167889301)

# 1 引 言

## 1.1 研究背景及意义

在过去的几十年里，人工智能技术经历了飞速的发展。人工智能从简单的规则系统演变为今天的深度学习模型，不仅模仿了人类智能，甚至在某些领域超越了人类。神经网络作为人工智能的核心技术之一，借鉴了人脑神经元的结构和功能，赋予了计算机处理复杂任务的能力。得益于计算能力的增强和数据量的增加，深度神经网络在语音识别、图像处理和自然语言处理等领域取得了显著成就。

随着人工智能技术的持续进步，其应用已遍及金融、医疗、交通等多个行业。在金融领域，人工智能系统用于预测股票市场、管理风险和信用评分，极大提高了交易效率和风险控制。在医疗行业，人工智能在疾病诊断、基因研究和药物开发中发挥作用，帮助医生更快更准确地诊断并为患者提供治疗方案。在交通领域，人工智能用于交通管理、自动驾驶和流量优化，提升了安全性和效率。

在探讨人工智能技术的未来时，我们必须认识到，随着技术的不断进步，人工智能系统将变得更加智能和自主。这意味着人工智能将能够处理更复杂的任务，提供更高级的决策支持，并在更广泛的领域中发挥作用。例如，人工智能可以帮助解决全球性问题，如气候变化、资源分配和疾病控制。人工智能技术的进步也可能带来新的就业机会，促进经济增长。

尽管人工智能带来了巨大的便利，但它也面临着挑战。人工智能技术的发展也引发了关于伦理和社会影响的重要讨论。随着人工智能系统变得更加复杂，确保它们的行为符合人类价值观和道德标准变得更加困难。人工智能决策过程的透明度、数据隐私的保护、以及防止算法偏见和歧视，都是当前和未来需要解决的关键问题。人工智能模型在现实世界中的决策和预测要求极高的准确性。在实际应用中，人工智能模型可能面临误判、偏见和过拟合等风险，这些风险可能导致性能下降或严重错误，给用户和社会带来损失。因此，对人工智能模型进行全面的测试变得至关重要。传统测试方法通常只关注准确性，而忽视了模型的潜在风险。一个面向风险的测试系统需要考虑模型在不同情境下的鲁棒性、可解释性和公平性，以更全面地评估模型质量。通过测试，可以及时发现并纠正模型的偏差和错误，确保其在实际应用中的可靠性和安全性。因此，如何测试人工智能系统中的模型并提供相应工具，成为了一个迫切需要解决的问题。

## 1.2 主要研究内容

本论文主要围绕人工智能系统中模型风险的测试与评估展开，重点在于通过设计和实现一个综合性的测试系统，评估模型在图像分类与检索任务中的表现。具体研究内容如下：

（1）准确性评估

设计并实现了对模型准确性的评估模块，采用精确率、召回率、F1分数等指标，对模型在不同数据集上的分类性能进行全面评估。

（2）鲁棒性评估

通过生成对抗样本和噪声测试，评估模型在面对恶意攻击和数据扰动时的稳定性。抗鲁棒性。

（3）公平性评估

评估模型在不同群体中的表现差异，确保模型对各个群体的公平性。采用Demographic Parity和Equalized Odds等公平性指标，量化模型对不同敏感群体的影响。

## 1.3 论文组织结构

本文分为六个章节进行论述。组织结构如下：

第一章为引言，介绍研究背景和意义，阐述人工智能技术的发展现状及其应用场景，提出本文的研究目标。

第二章为相关理论与技术概述，介绍准确性、鲁棒性和公平性评估的相关理论和方法，综述实现系统所需的相关技术，包括前端开发技术、后端开发技术、数据库技术。

第三章为系统需求分析，分析系统的功能需求和非功能需求，确定系统应具备的主要功能和性能指标。通过需求分析，明确系统的目标和约束条件，提供系统设计和实现的基础。

第四章为系统设计与实现，介绍系统的总体架构和各个模块的详细设计与实现。包括前端模块、后端模块、数据库模块和模型评估模块的设计与实现。

第五章为系统测试，通过具体的实验和测试，评估本系统的性能和功能。

第六章为总结与展望，总结本研究的主要成果和贡献，提出未来研究的方向和需要进一步改进的方面。

# 

# 2 相关理论与技术概述

## 2.1 准确性的评估

评估模型的准确性是确定模型在预测任务中的表现的重要步骤。准确性评估可以帮助我们了解模型的预测质量以及模型在现实应用中的可用性。以下是评估模型准确性的一些常见指标：

（1） 准确率（Accuracy）

准确率（Accuracy）是机器学习中最常用的评估指标之一，通常用于衡量分类模型在预测任务中的表现。准确率表示模型在测试集中正确预测的样本数量与总样本数量的比率。它是衡量模型整体预测质量的一种简单而直接的度量方法。

对于一个分类模型，准确率的计算公式如下：

（2.1）

准确率是一个简单、直观的度量指标，适用于二元分类和多类分类任务。但不均衡数据集中，准确率可能会掩盖模型在少数类上的表现。模型可能通过简单地预测多数类来获得高准确率，但对少数类的预测效果不佳。为了全面评估模型的表现，通常将准确率与其他评估指标（如精确度、召回率、F1分数等）结合使用。

（2） 精确度（Precision）

精确度（Precision）是机器学习中的一个评估指标，用于衡量分类模型在预测任务中的表现。精确度也被称为“精准率”或“查准率”，表示模型在预测为正类的样本中正确预测的比例。换句话说，精确度衡量的是模型预测为正类的样本中实际为正类的样本所占的比例。

精确度的计算公式如下：

（2.2）

其中True Positives为模型正确预测为正类的样本数量，False Positives为模型错误预测为正类的样本数量。

精确度衡量模型在预测为正类时的准确性，有助于评估模型在正类预测上的表现。

（3）召回率（Recall）

召回率（Recall）是机器学习中的一个评估指标，用于衡量分类模型在预测任务中的表现。召回率表示模型在实际为正类的样本中正确预测的比例，即模型能够正确识别出实际正类样本的能力。

召回率的计算公式如下：

（2.3）

其中True Positives为模型正确预测为正类的样本数量。False Negatives为模型错误预测为负类的样本数量。召回率衡量模型在实际正类样本中正确识别的能力，通常将召回率与精确度（Precision）结合使用，以评估模型在正类预测中的综合表现。

（4） F1 分数（F1 Score）

F1 分数（F1 Score）是机器学习中的一个评估指标，用于衡量分类模型在预测任务中的综合表现。它是精确度（Precision）和召回率（Recall）的调和平均数，旨在平衡这两个指标之间的权衡。F1 分数在正负样本不均衡的数据集中尤其有用，因为它能够同时考虑模型在预测正类时的准确性和全面性。

F1 分数的计算公式如下：

（2.4）

（5） P-R 曲线

P-R 曲线，也称为精确度-召回率曲线（Precision-Recall Curve），是机器学习中的一种评估工具，用于衡量分类模型在预测任务中的表现。P-R 曲线绘制了模型在不同阈值下的精确度（Precision）和召回率（Recall）之间的权衡关系。通过观察曲线的形状，可以了解模型在不同阈值下在正类预测中的综合表现。

P-R 曲线下面的面积（AUC）是一个综合评估模型表现的指标，数值越大表示模型在正类预测中的综合表现越好。通过比较不同模型的 P-R 曲线，可以评估它们在正类预测中的综合表现，并选择最优模型。P-R 曲线可以帮助选择模型的最佳阈值，以实现模型在精确度和召回率之间的最佳权衡。

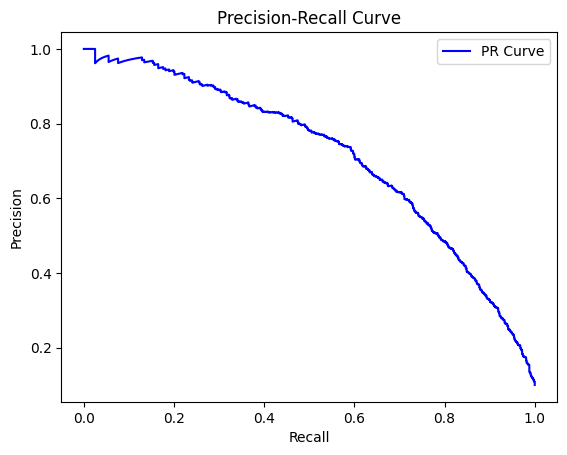


图2.1 P-R曲线

（6） ROC 曲线

ROC 曲线（Receiver Operating Characteristic Curve，接收者工作特性曲线）是机器学习中的一种评估工具，用于衡量分类模型在预测任务中的表现。ROC曲线描绘了模型在不同阈值下的真正率（True Positive Rate）和假正率（False Positive Rate）的权衡关系。

ROC 曲线下面的面积（AUC）是一个综合评估模型表现的指标，AUC值越大表示模型在分类任务中的综合表现越好。通过比较不同模型的ROC曲线，可以评估它们在分类任务中的综合表现，并选择最优模型。ROC曲线可以帮助选择模型的最佳阈值，以实现模型在真正率和假正率之间的最佳权衡。

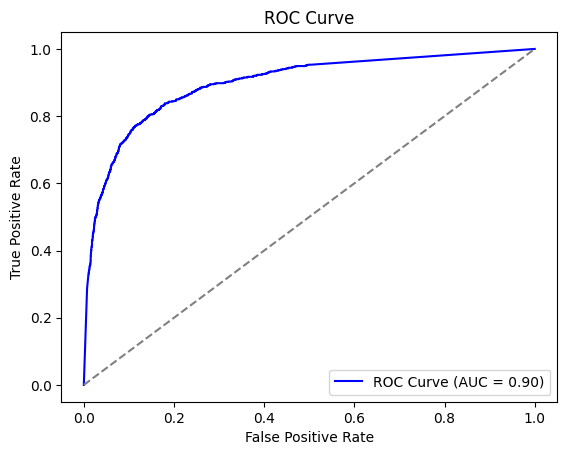


图2.2 ROC曲线

（7） 平均精度均值（MAP，Mean Average Precision）

平均精度均值（MAP）是多个类别的平均精度（Average Precision, AP）的均值。AP是在不同的召回率水平下，精确率的平均值，可以用P-R曲线下的面积来表示。对于多标签分类或多类别的情况，计算所有类别的AP，然后取均值作为MAP：

（2.5）

其中，是类别的总数，第个类别的平均精度。

（8） Top-K 准确率（Top-K Accuracy）

Top-K 准确率（Top-K Accuracy）是检索领域中一个重要的评价指标，衡量检索系统在返回的前 K 个结果中包含正确结果的能力。统计所有查询中，返回正确结果的查询比例，即为 Top-K 准确率：

（2.6）

## 2.2 鲁棒性的评估

### 2.2.1对抗性示例

包括深度神经网络在内的一些机器学习模型很容易受到对抗性示例的影响[1]。对抗性示例是指通过对输入数据进行微小的、经过计算的修改，以欺骗机器学习模型的输入，从而导致模型输出错误的示例。这些修改通常是在人类难以察觉的范围内进行的，但却足以使模型做出错误的分类或回归预测。

### 2.2.2 对抗性攻击

我们可以使用许多攻击算法来生成对抗性示例，以下是一些常用的攻击算法。

（1） 快速梯度符号法（Fast Gradient Sign Method）[1]

FGSM利用神经网络模型的梯度信息，针对输入数据进行微小的扰动，使得模型产生错误的分类结果。其基本思想是，在模型的损失函数关于输入的梯度的方向上，对输入数据进行一小步的移动，以使损失函数增加，从而导致模型输出错误的结果。

设扰动为，为模型的参数，为模型的输入，为对应的真实标签，为训练神经网络所使用的成本函数，为成本函数关于的梯度， 为一个小的扰动量，用于控制扰动的大小，函数返回其参数的符号。扰动用如下的公式来表示：

（2.7）

最终生成的对抗性示例可以用如下公式表示：

（2.8）

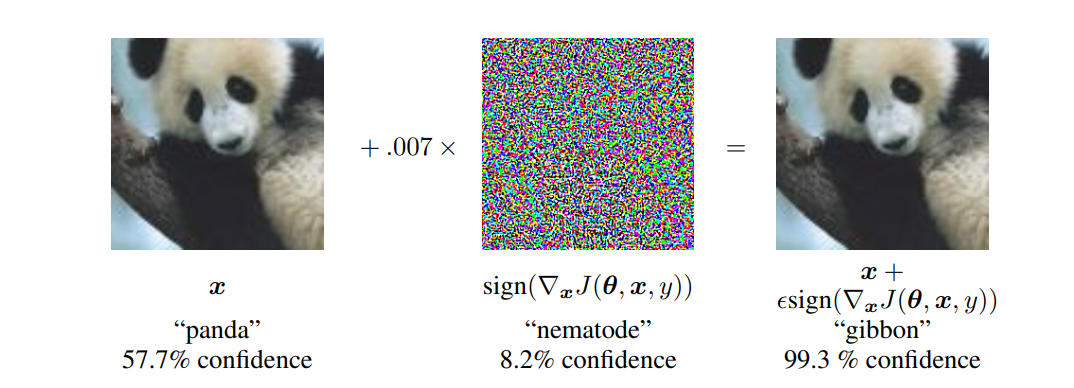


图2.3 快速梯度符号法[1]

（2）DeepFool[2]

DeepFool的基本思想是在数据点周围找到一个足够小的线性扰动，使得将数据点沿着决策边界移动到另一个类别的决策边界。这个扰动是通过迭代的线性化来计算的，直到数据点被错误分类为止。DeepFool能有效计算出欺骗深度神经网络的扰动，从而可靠地量化这些分类器的鲁棒性。

给定一个深度神经网络模型，设待攻击的输入样本为，其对应的真实标签为，扰动为，迭代次数为，阈值为。对于每一次迭代，计算输入样本相对于每个类别的梯度,然后找到相对于当前类别的对校最小梯度对应的类别，即：

（2.9）

然后计算样本关于当前类别和下一个类别的最小梯度方向的投影，并将扰动沿着这个方向进行更新，即：

（2.10）

直到或样本被错误分类时，停止迭代。

### 2.2.3 鲁棒性评估指标

评估模型的对抗鲁棒性时，可以使用多种指标来量化模型对抗性攻击的表现。以下是一些常见的评估指标：

（1）实证鲁棒性（Empirical Robustness）[2, 4]

实证鲁棒性指通过对模型在真实数据集上的实际表现进行观察和分析，以确定模型在面对对抗性攻击时的鲁棒性。它等同于成功发起攻击所需要的最小平均扰动。

给定一个分类器，一个非目标攻击，测试数据集。为所有攻击成功的样本的下表集合，即对于任意,。实证鲁棒性可以被定义为：

（2.11）

其中为用于生成对抗性示例的范数。

（2） 损失敏感性（Loss Sensitivity）[4, 5]

损失敏感度是一种基于梯度的局部性质度量，可以帮助评估模型对输入的微小变化的敏感程度，因此可以在不考虑具体攻击方法的情况下提供有关模型鲁棒性的信息。

利普希茨连续性常数衡量了函数在输入空间中的局部变化与输出空间中的变化之间的关系。对于一个函数，如果存在一个常数，使得对于输入空间中的任意两个点和，函数值之间的距离不超过输入之间的距离的倍，即：

（2.12）

那么常数被称为函数的利普希茨连续性常数。这个常数可以理解为函数的局部斜率上界，即函数在输入空间中的局部变化率的最大值。

我们可以通过估计模型的利普希茨连续性常数来衡量其损失敏感性。给定一个分类器和一个测试数据集，损失敏感性可以被定义为：

（2.13）

其中表示损失函数关于输入和对应标签的梯度。

那么常数被称为函数的利普希茨连续性常数。这个常数可以理解为函数的局部斜率上界，即函数在输入空间中的局部变化率的最大值。

（3）安全曲线（Security Curve）

安全曲线（Security Curves）是一种用来评估和比较机器学习模型对抗性鲁棒性的方法。它们在文献[7,8]中被广泛使用，以确定和比较模型在面对对抗性示例时的表现。

安全曲线通过展示模型在不同攻击扰动下的性能，揭示模型在面对逐渐增加的攻击强度时的表现。这通常是通过衡量模型在不同范数下的攻击扰动对模型的性能（例如准确性）的影响来实现的。

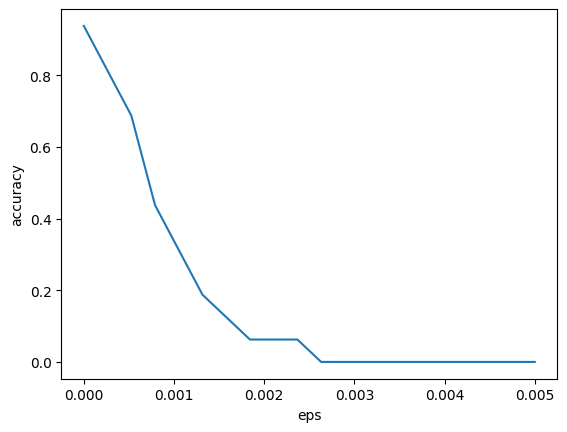


图2.4 使用无穷范数的DeepFool算法对预训练的ResNet18模型进行攻击后得到的安全曲线

## 2.3 公平性的评估

公平性是一个无法直接观察或测量的概念，需要通过各种测量模型对模型进行量化，以便理解和评估模型公平性。在评估人工智能系统的公平性时，比较不同群体之间量化的危害是非常重要的。这有助于了解模型在不同敏感特征群体中的表现差异，特别是在错误方面的差异。我们可以从指标值的差异、指标值的比率或是最差群体的表现中来评估不同敏感特征群体。我们使用代表用于预测的特征向量，使用来代表敏感特征，使用来代表真实标签。

以下是一些可能会使用到的指标。

（1） Demographic Parity

Demographic Parity是一个公平性度量指标，其目标是确保机器学习模型的预测与敏感群体的成员身份无关，即当某个预测的概率不依赖于敏感群体的成员身份时，就实现了Demographic Parity。

在二元分类的情况下，设分类器为，当分类器的预测结果与敏感特征之间统计独立时，分类器在分布下满足Demographic Parity，即[2]：

（2.14）

我们可以通过计算不同敏感特征群体间的选择率差异来衡量Demographic Parity。选择率表示模型预测为正例的样本数量与样本总数的比率。如果某一群体的选择率与最高选择率的比率小于0.8，则可能存在不公平。

在回归的情况下，设预测器为，当与敏感特征独立时，预测器满足在分布下的Demographic Parity，即[3]：

（2.15）

Demographic Parity作为公平性指标存在一定局限性[4]。Demographic Parity忽略了其他可能影响预测结果的因素，因此不一定能够准确反映模型的公平性。

（2） Equalized Odds

Equalized Odds是一种公平性度量指标，其目标是确保机器学习模型在不同群体中表现同样良好。与Demographic Parity相比，Equalized Odds更为严格，因为它不仅要求机器学习模型的预测与敏感群体独立，还要求不同群体具有相同的错误正率（False Positive Rate）和真正正率（True Positive Rate）。

设分类器为，设分类器为，当分类器的预测结果与敏感特征条件独立时，分类器在分布下满足Equalized Odds，即[2]：

（2.16）

## 2.4 相关技术概述

（1） MySQL数据库

MySQL是一种流行的开源关系数据库管理系统（RDBMS）。它以其高性能、高可靠性和易用性而闻名，是全世界使用最广泛的数据库之一。MySQL数据库具有优秀的查询性能，适合处理大型数据库和高并发请求，且可以运行在多种操作系统上，如Windows、Linux和macOS。MySQL支持主从复制、集群和高可用性架构，确保数据的可靠性和可用性，并提供了强大的安全机制，包括用户认证、权限管理和加密通信。MySQL支持多种存储引擎（如InnoDB和MyISAM），用户可以根据需求选择合适的存储引擎。

（2） Java

Java由Sun Microsystems公司于1995年推出，现在由Oracle公司维护和发展。Java以其跨平台特性、高安全性、良好的性能以及广泛的应用而著称，是当前世界上最流行的编程语言之一。Java是一种纯面向对象的编程语言，支持类、对象、继承、多态、封装等面向对象的编程概念，有助于代码的重用和系统的扩展。Java程序可以在任何支持Java虚拟机（JVM）的平台上运行，而无需重新编译。

（3） Spring Boot框架

Spring Boot是由Pivotal团队开发的一个开源框架，它基于Spring框架，旨在简化Spring应用程序的开发和部署。Spring Boot通过提供一系列开箱即用的配置和自动化工具，使开发者能够更快速地创建基于Spring的独立生产级应用。

Spring Boot提供了强大的自动配置功能，能够根据项目的依赖和配置自动配置Spring应用程序。这减少了繁琐的手动配置工作，让开发者专注于业务逻辑的实现，并内置了Tomcat、Jetty和Undertow等流行的Web服务器，开发者可以将应用程序打包为一个独立的JAR或WAR文件，直接运行，无需额外配置服务器环境。

（4） MyBatis框架

MyBatis是一种优秀的持久层框架，提供了支持自定义SQL、存储过程以及高级映射的持久化实现。MyBatis 避免了几乎所有的JDBC代码和手动设置参数以及获取结果集。MyBatis可以通过简单的XML或注解来配置和映射原生信息，将Java对象和SQL语句关联起来。

MyBatis使用XML和注解两种方式来配置和映射Java对象与SQL语句之间的关系。相比于其他对象关系映射框架，MyBatis更贴近原生SQL，开发者可以完全掌控SQL执行过程。MyBatis允许使用动态SQL，提供了丰富的标签和表达式语言来构建动态SQL，能够应对复杂查询需求。MyBatis支持高级映射功能，包括一对一、一对多和多对多的复杂关系映射。通过配置文件，可以方便地将数据库表与Java对象之间进行映射。

（5） Python

Python是一种由Guido van Rossum于1991年发布的高级编程语言。由于其设计哲学强调代码的可读性和简洁性，Python迅速成为了编程社区中的一种流行语言。Python是一种解释型、动态类型、面向对象的编程语言，广泛应用于Web开发、数据分析、人工智能、科学计算、自动化运维等领域。Python是一种跨平台语言，可以在Windows、macOS、Linux等多种操作系统上运行。只需编写一次代码，就可以在不同的平台上运行，极大地提高了开发效率。

Python是人工智能和机器学习领域的首选语言。Scikit-learn、TensorFlow、Keras和PyTorch等库提供了广泛的机器学习和深度学习算法，实现了从数据预处理、模型训练到模型部署的一站式解决方案。

（6） FastAPI

FastAPI基于Starlette和Pydantic，是一个现代、高性能的Web框架，由Sebastián Ramírez开发。FastAPI。FastAPI以其快速的性能、易用性和自动生成文档的能力而广受欢迎。

FastAPI使用Pydantic进行数据验证和序列化，能够自动校验请求数据和响应数据的格式，确保数据的准确性和安全性。FastAPI全面支持Python的异步编程特性，允许开发者使用async/await来编写高性能的异步代码。

（7） PyTorch

PyTorch是一个开源的深度学习框架，由Facebook的人工智能研究小组（FAIR）开发和维护。自2016年发布以来，PyTorch因其动态计算图、直观的设计和强大的功能，迅速成为学术界和工业界广泛使用的深度学习工具。

PyTorch使用动态计算图（Dynamic Computational Graph），也称为即时计算（Eager Execution）。这意味着计算图是在运行时构建的，允许开发者以更自然和直观的方式调试和修改代码。PyTorch内置了强大的自动微分库Autograd，能够自动计算张量操作的梯度。对于复杂的模型和优化算法，这大大简化了开发过程。PyTorch提供了灵活且高效的张量操作，支持GPU加速计算。其张量库与NumPy兼容，允许无缝地集成现有的科学计算代码。PyTorch的nn模块提供了丰富的神经网络层和损失函数，开发者可以方便地定义和训练深度学习模型。高层的torch.optim模块提供了常用的优化算法，使模型训练变得更加简单和高效。PyTorch原生支持多GPU训练，开发者可以轻松地将模型和数据分布到多个GPU上，提高训练速度和效率。

（8） Vue.js

Vue.js（简称Vue）是一个用于构建用户界面的渐进式JavaScript框架。由Evan You于2014年创建，Vue采用自底向上的增量开发设计，使其易于集成于其他项目中，同时对于单页应用（SPA）的开发也非常适合。Vue以其灵活性、易用性和高性能而受到广泛欢迎。

Vue使用声明式渲染，通过简洁的模板语法将数据绑定到DOM。开发者可以在模板中使用指令（如v-bind和v-for）来动态更新DOM，代码更加直观和易读。Vue支持组件化开发，开发者可以将应用分解为可复用的组件，每个组件包含其自己的HTML、CSS和JavaScript。这种方式提高了代码的可维护性和复用性。

# 3 系统需求分析

## 3.1 需求分析概述

本系统旨在对图片的分类或检索任务的模型进行全面评估，主要包括模型的准确性、鲁棒性和公平性评估。系统采用前后端分离的架构，前端使用Vue框架，后端使用SpringBoot框架，数据库采用MySQL，并使用MyBatis进行数据库操作。模型评估部分则采用FastAPI与PyTorch实现，提供高效的模型评估服务。

在功能需求方面，系统支持用户上传打包好的数据集和模型，能够对多种图像分类与检索模型进行评估，并提供详细的结果和报告生成功能。系统还支持用户注册、登录。

性能需求方面，系统设计旨在高效处理和预处理大规模数据，支持实时评估功能。系统还注重用户体验，提供简洁友好的界面和快速响应能力。

非功能性需求则包括系统的可靠性、安全性和可扩展性，确保系统能够长期稳定运行，保护用户数据和隐私，并能够方便地扩展新的功能和评估指标。

本需求分析旨在明确系统的各项需求，为后续的设计和实现提供清晰的指导。

## 3.2 功能性需求分析

### 3.2.1 总体业务描述

本系统旨在提供一个平台，用于对图片分类及检索任务的机器学习模型进行全面评估。系统通过评估模型的准确性、鲁棒性和公平性，帮助用户了解模型的性能并发现潜在的问题，进而优化模型。系统的整体用例分析如图3.1所示。该系统可以分为以下四个主要模块：

（1）用户管理

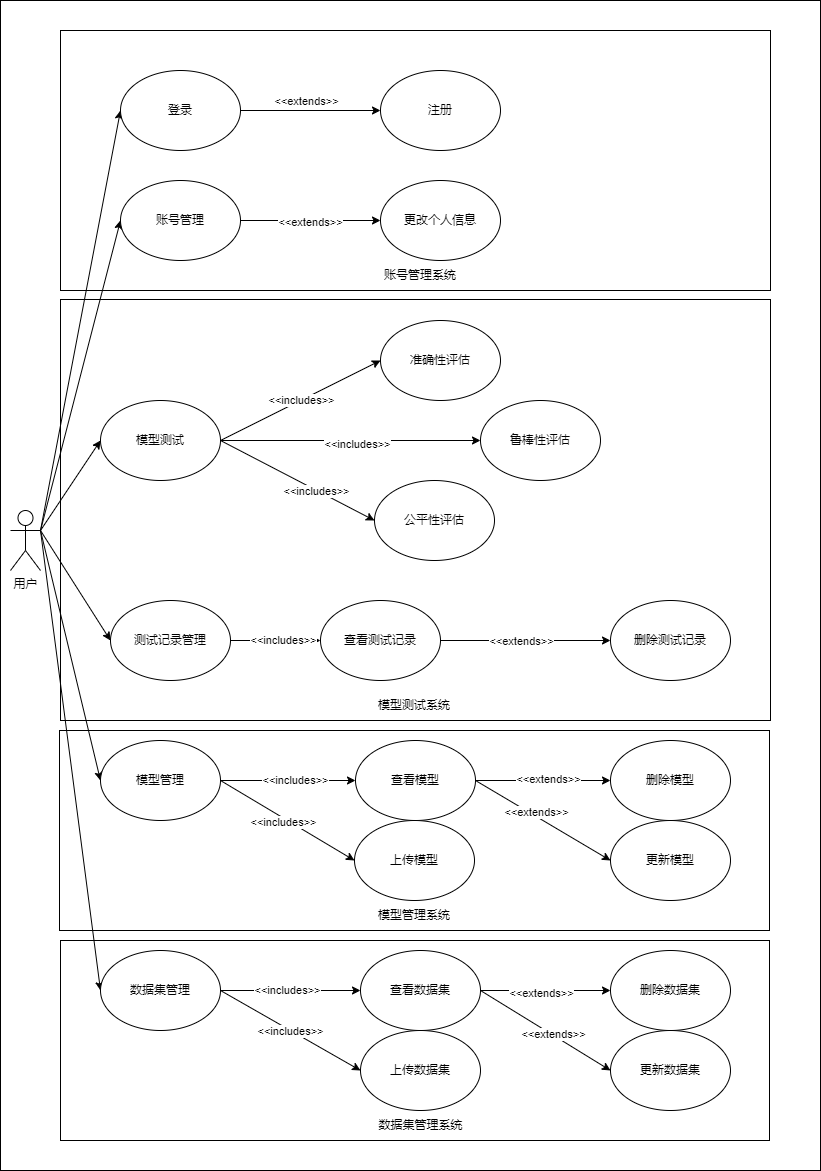
用户可以通过系统进行注册和登录。用户必须先注册才能登录。注册成功后，用户可以登录系统，进入个人账户进行后续操作。用户可以管理自己的账户信息，包括修改个人信息和查看账号管理系统的相关内容。

（2）模型测试

用户可以上传并测试各种图片分类或检索模型。系统将对模型的性能进行全面评估，包括准确性评估、鲁棒性评估和公平性评估。准确性评估包括精确率、召回率和F1分数等指标。鲁棒性评估通过生成对抗样本和噪声测试模型的鲁棒性，评估模型在面对恶意攻击和数据扰动时的稳定性。公平性评估通过评估模型在不同群体中的表现差异，确保模型对各个群体的公平性，常用的公平性指标包括差异影响和均等机会。

（3）测试记录管理

用户可以查看历史测试记录，包括各项测试结果的详细数据和评估报告。用户可以删除不再需要的测试记录，保持测试记录管理系统的整洁和高效。

图3.1 系统用例图

（4）模型管理

用户可以对模型进行管理，包括查看模型、上传新模型、更新模型和删除模型。用户可以查看已上传的模型信息和状态。用户可以上传新的模型类定义文件和模型权重文件，以便进行测试和评估。用户可以更新已有模型的权重文件或定义文件，确保模型保持最新状态。用户可以删除不再需要的模型，保持模型管理系统的简洁。

（5）数据集管理

用户可以管理用于模型测试的数据集，包括查看数据集、上传新数据集、更新数据集和删除数据集。用户可以查看已上传的数据集信息和状态。用户可以上传打包好的图片数据集，系统会自动解压和处理数据。用户可以更新已有的数据集，确保数据集保持最新状态。用户可以删除不再需要的数据集，保持数据集管理系统的简洁。是一些常用的攻击算法。

### 3.2.2 账号管理用例分析

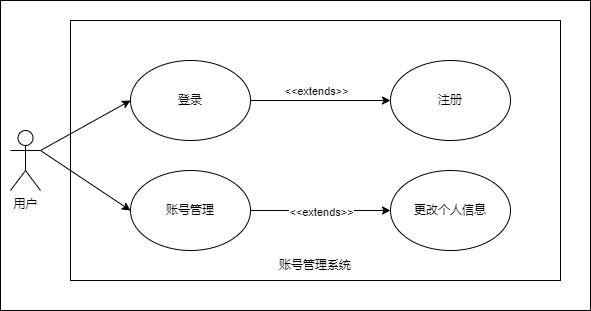
账号管理系统是本平台的重要组成部分，旨在提供用户注册、登录和个人信息管理等功能。通过账号管理系统，用户可以方便地管理自己的账户信息。

图3.2 账号管理系统用例图

登录用例的用例规约如表3.1所示。该用例描述了已注册并完成邮箱验证的用户如何通过输入用户名和密码，成功登录并访问系统。该过程确保只有经过验证的用户才能访问系统的账号管理和其他功能。

表3.1 登录用例规约

|  |  |
| --- | --- |
| 用例名称 | 登录 |
| 用例描述 | 已经注册并完成邮箱验证的用户通过输入用户名和密码，登录并访问系统。 |
| 参与者 | 用户 |
| 前置条件 | 用户已注册账号并完成邮箱验证。 |

续表3.1

|  |  |
| --- | --- |
| 后置条件 | 用户成功登录系统。 |
| 主要事件流 | 1. 用户访问登录页面。 2. 用户输入用户名和密码。 3. 用户提交登录信息。 4. 系统验证用户提供的用户名和密码。 5. 如果验证通过，系统检查用户是否已完成邮箱验证。 6. 如果用户已完成验证，系统允许用户登录。 7. 用户可以访问账号管理和其他系统功能。 |
| 备选事件流 | 5a. 用户未完成邮箱验证   1. 系统提示用户完成邮箱验证。 2. 用户通过“重新发送验证邮件”功能完成邮箱验证。 |

注册的用例规约如表3.x所示。该用例描述了用户通过填写注册表单，提供必要的个人信息，在系统中创建一个新账号的过程。系统会验证用户提供的信息，并在验证通过后完成注册。注册过程中会发送验证邮件，用户需要通过邮件中的链接完成验证，激活账号。

表3.2 注册用例规约

|  |  |
| --- | --- |
| 用例名称 | 注册 |
| 用例描述 | 用户通过填写注册表单，提供必要的个人信息，创建一个新的账号。系统会验证用户提供的信息（如邮箱是否唯一），并在验证通过后完成注册。注册过程中会发送验证邮件，用户需要通过邮件中的链接完成验证，激活账号。 |
| 参与者 | 用户 |
| 前置条件 | 用户未注册账号。 |
| 后置条件 | 新用户账号创建成功，用户已完成邮件验证。 |
| 主要事件流 | 1. 用户访问注册页面。 2. 用户填写注册表单，包括用户名、密码、邮箱等信息。 3. 用户提交注册表单。 4. 系统验证用户提供的信息。 5. 如果信息合法且邮箱唯一，系统发送验证邮件到用户提供的邮箱。 6. 用户点击邮件中的验证链接，激活账号。 7. 系统提示用户注册成功。 8. 用户可以使用注册信息登录系统。 |
| 备选事件流 | 5a. 用户邮箱已被使用   1. 系统提示用户邮箱已被使用，注册失败。 2. 用户重新输入一个未使用的邮箱地址。   6a. 用户未收到验证邮件   1. 用户在一段时间内未收到验证邮件。 2. 用户访问系统提供的“重新发送验证邮件”功能。 3. 系统重新发送验证邮件。 |

账号管理用例规约如表3.3所示。用户在登录系统后，可以查看和管理个人账号信息的过程。用户能够查看当前的个人信息，如用户名、邮箱。

表3.3 账号管理用例规约

|  |  |
| --- | --- |
| 用例名称 | 账号管理 |
| 用例描述 | 用户登录系统后，可以查看和管理个人账号信息。 |
| 参与者 | 用户 |
| 前置条件 | 用户已成功注册并登录系统。 |
| 后置条件 | 用户可以查看当前的个人信息。 |
| 主要事件流 | 1. 用户登录系统后，进入账号管理。 2. 系统显示当前用户的账号信息，包括用户名、邮箱、联系方式等。 |
| 备选事件流 | 无 |

更改个人信息用例规约如表3.4所示，用户可以在系统中更新其个人信息，并保存这些更改。

表3.4 更改个人信息用例规约

|  |  |
| --- | --- |
| 用例名称 | 更改个人信息 |
| 用例描述 | 用户可以修改个人信息（如用户名、邮箱、密码等）。 |
| 参与者 | 用户 |
| 前置条件 | 用户已成功注册并登录系统。 |
| 后置条件 | 账号信息修改成功，用户可以查看更新后的个人信息。 |
| 主要事件流 | 1. 用户在“账号管理”界面选择“修改个人信息”。 2. 系统显示可以修改的个人信息项。 3. 用户修改相关信息项后，点击“保存”。   系统验证用户输入的修改信息是否合法。   1. 如果验证通过，系统保存用户修改后的信息。 2. 系统返回账号管理界面，并提示“信息修改成功”。 3. 用户可以查看更新后的个人信息。 |
| 备选事件流 | 无 |

### 4.2.3 模型测试用例分析

图3.3 模型测试系统用例图

模型测试系统是本平台的核心功能之一，旨在提供一个平台，用于对图片分类及检索任务的机器学习模型进行全面评估。系统通过评估模型的准确性、鲁棒性和公平性，帮助用户了解模型的性能并发现潜在的问题，进而优化模型。

准确性评估用例规约如表3.5所示，用户选择需要测试的模型和数据集，并由系统对模型进行准确性评估。评估过程包括计算精确率、召回率和F1分数或其他指标，并生成详细的评估报告。

表3.5 准确性评估用例规约

|  |  |
| --- | --- |
| 用例名称 | 准确性评估 |
| 用例描述 | 用户选择需要测试的模型和数据集，系统对模型进行准确性评估。 |
| 参与者 | 用户 |
| 前置条件 | 用户已登录并上传了模型和数据集。 |
| 后置条件 | 系统生成评估报告，用户可以查看详细的评估结果。 |
| 主要事件流 | 1. 用户选择已上传的模型和数据集。 2. 系统加载模型和数据集。 3. 系统对模型进行准确性评估，计算精确率、召回率和F1分数或其他指标。 4. 系统显示评估结果，用户可以查看模型的准确性评估报告。 |
| 备选事件流 | 无 |

鲁棒性评估用例规约如表3.6所示，用户选择需要测试的模型和数据集，系统对模型进行鲁棒性评估。评估过程中，系统生成对抗样本或加入噪声，对模型的稳定性进行测试，并生成详细的评估报告。

表3.6 鲁棒性评估用例规约

|  |  |
| --- | --- |
| 用例名称 | 鲁棒性评估 |
| 用例描述 | 用户选择需要测试的模型和数据集，系统对模型进行鲁棒性评估。 |
| 参与者 | 用户 |
| 前置条件 | 用户已登录并上传了模型和数据集。 |
| 后置条件 | 系统生成评估报告，用户可以查看详细的评估结果。 |
| 主要事件流 | 1. 用户上传模型和数据集。 2. 系统加载模型和数据集。 3. 系统生成对抗样本或加入噪声，对模型进行鲁棒性测试。 4. 系统对模型进行鲁棒性评估，评估模型在面对恶意攻击和数据扰动时的稳定性。 5. 系统显示评估结果，用户可以查看模型的鲁棒性评估报告。 |
| 备选事件流 | 无 |

公平性评估用例规约如表3.7所示，用户选择需要测试的模型和数据集，并由系统对模型进行公平性评估。评估过程旨在评估模型在不同群体中的表现差异，确保模型对各个群体的公平性，并生成详细的评估报告。

表3.7 公平性评估用例规约

|  |  |
| --- | --- |
| 用例名称 | 公平性评估 |
| 用例描述 | 用户选择需要测试的模型和数据集，系统对模型进行公平性评估。 |
| 参与者 | 用户 |
| 前置条件 | 用户已登录并上传了模型和数据集。 |
| 后置条件 | 系统生成评估报告，用户可以查看详细的评估结果。 |
| 主要事件流 | 1. 用户上传模型和数据集。 2. 系统加载模型和数据集。 3. 系统对模型进行公平性评估，评估模型在不同群体中的表现差异，确保模型对各个群体的公平性。 4. 系统显示评估结果，用户可以查看模型的公平性评估报告。 |
| 备选事件流 | 无 |

测试记录管理用例规约如表3.8所示，用户查看和管理其模型测试记录，包括查看测试记录和删除不需要的记录。用户可以通过系统查看详细的测试报告，并管理这些记录。

表3.8 测试记录管理用例规约

|  |  |
| --- | --- |
| 用例名称 | 测试记录管理 |
| 用例描述 | 用户可以查看和管理模型测试记录，包括查看测试记录和删除不需要的记录。 |
| 参与者 | 用户 |
| 前置条件 | 用户已登录并进行过模型测试。 |
| 后置条件 | 用户可以查看详细的测试记录和报告。 |
| 主要事件流 | 1. 用户选择“测试记录管理”功能。 2. 系统显示当前用户的所有测试记录，包括模型、测试日期、测试结果等信息。 3. 用户可以选择查看详细的测试报告。 |
| 备选事件流 | 无 |

删除测试记录用例规约如表3.x所示，用户可以选择并删除特定的测试记录，系统会更新测试记录列表以反映这一更改。

表3.9 删除测试记录用例规约

|  |  |
| --- | --- |
| 用例名称 | 删除测试记录 |
| 用例描述 | 用户可以删除不再需要的测试记录，保持测试记录管理系统的整洁和高效。 |
| 参与者 | 用户 |
| 前置条件 | 用户已登录并进行过模型测试。 |
| 后置条件 | 选定的测试记录被删除，系统更新测试记录列表。 |
| 主要事件流 | 1. 用户选择“测试记录管理”功能。 2. 系统显示当前用户的所有测试记录。 3. 用户选择要删除的测试记录。 4. 系统提示用户确认删除操作。 5. 用户确认删除操作。 6. 系统删除选定的测试记录，并更新测试记录列表。 |
| 备选事件流 | 无 |

### 模型管理用例分析

图3.4 模型管理系统用例图

模型管理系统是本平台的重要组成部分，旨在提供用户对模型进行管理的功能。通过模型管理系统，用户可以方便地上传、查看、更新和删除模型。

模型管理用例规约如表3.10所示，用户在登录系统后管理自己的模型。用户可以查看已上传的模型、上传新模型、更新已有模型以及删除不需要的模型。

表3.10 模型管理用例规约

|  |  |
| --- | --- |
| 用例名称 | 模型管理 |
| 用例描述 | 用户登录系统后，可以管理自己的模型，包括查看模型、上传新模型、更新已有模型和删除不需要的模型。 |
| 参与者 | 用户 |
| 前置条件 | 用户已登录系统。 |
| 后置条件 | 用户可以查看已上传的模型及其详细信息。 |
| 主要事件流 | 1. 用户登录系统后，进入模型管理页面。 2. 系统显示用户已上传的所有模型列表，包括模型名称、模型类型等信息。 |
| 备选事件流 | 无 |

上传模型用例规约如表3.11所示，用户可以通过系统界面上传新的模型文件。系统提供了上传表单，用户可以填写模型名称并选择文件进行上传。系统会验证上传的文件是否合法，确保文件格式和内容正确后，保存模型文件并更新模型列表

表3.11 上传模型用例规约

|  |  |
| --- | --- |
| 用例名称 | 上传模型 |
| 用例描述 | 用户可以上传新的模型，包括模型类定义文件和模型权重文件。 |
| 参与者 | 用户 |
| 前置条件 | 用户已登录系统。 |
| 后置条件 | 新模型上传成功，系统更新模型列表。 |

续表3.11

|  |  |
| --- | --- |
| 主要事件流 | 1. 用户在模型管理页面选择“上传模型”功能。 2. 系统显示上传模型的表单，用户填写模型名称和选择模型文件（包括类定义文件和权重文件）。 3. 用户提交上传表单。 4. 系统验证上传的文件格式和内容是否合法。 5. 如果验证通过，系统保存模型文件并更新模型列表。 6. 系统提示用户上传成功。 |
| 备选事件流 | 无 |

更新模型用例规约如表3.12所示，用户可以通过系统界面更新已有模型的权重文件。系统提供了更新表单，用户可以选择新的权重文件进行上传。系统会验证上传的文件是否合法，确保文件格式和内容正确后，保存新的权重文件并更新模型的信息。

表3.12 更新模型用例规约

|  |  |
| --- | --- |
| 用例名称 | 更新模型 |
| 用例描述 | 用户可以更新已有模型的权重文件，确保模型保持最新状态。 |
| 参与者 | 用户 |
| 前置条件 | 用户已登录系统，并已上传过模型。 |
| 后置条件 | 模型文件更新成功，系统保存最新的模型信息。 |
| 主要事件流 | 1. 用户在模型管理页面选择某个已上传的模型。 2. 用户选择“更新模型”功能。 3. 系统显示更新模型的表单，用户选择新的权重文件。 4. 用户提交更新表单。 5. 系统验证上传的文件格式和内容是否合法。 6. 如果验证通过，系统更新模型文件并保存更改。 7. 系统提示用户更新成功。 |
| 备选事件流 | 无 |

删除模型用例规约如表3.13所示，用户可以选择并确认删除特定模型。系统会在用户确认后删除模型文件，并更新模型列表，确保用户的模型管理界面保持最新状态。

表3.13 删除模型用例规约

|  |  |
| --- | --- |
| 用例名称 | 删除模型 |
| 用例描述 | 用户可以删除不再需要的模型。 |
| 参与者 | 用户 |
| 前置条件 | 用户已登录系统，并已上传过模型。 |
| 后置条件 | 选定的模型被删除，系统更新模型列表。 |
| 主要事件流 | 1. 用户在模型管理页面选择某个已上传的模型。 2. 用户选择“删除模型”功能。 3. 系统提示用户确认删除操作。 4. 用户确认删除操作。 5. 系统删除选定的模型文件，并更新模型列表。 6. 系统提示用户删除成功。 |
| 备选事件流 | 无 |

### 3.2.4 数据集管理用例分析

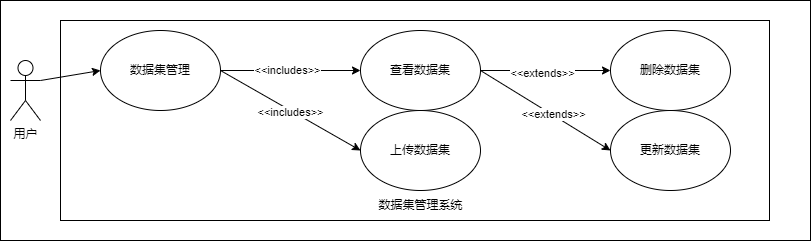
数据集管理系统是本平台的重要组成部分，旨在提供用户对数据集进行管理的功能。通过数据集管理系统，用户可以方便地上传、查看、更新和删除数据集。

图3.5 模型管理系统用例图

数据集管理用例规约如表3.14所示，用户可以通过系统界面方便地管理其数据集。系统提供查看、上传、更新和删除数据集的功能，确保用户能够灵活管理其数据集。

表3.14 数据集管理用例规约

|  |  |
| --- | --- |
| 用例名称 | 数据集管理 |
| 用例描述 | 用户登录系统后，可以管理自己的数据集，包括查看数据集、上传新数据集、更新已有数据集和删除不需要的数据集。 |
| 参与者 | 用户 |
| 前置条件 | 用户已登录系统。 |
| 后置条件 | 用户可以查看已上传的数据集及其详细信息。 |
| 主要事件流 | 1. 用户登录系统后，进入数据集管理页面。 2. 系统显示用户已上传的所有数据集列表，包括数据集名称、数据集类型等信息。 |
| 备选事件流 | 无 |

上传数据集用例规约如表3.15所示，用户可以通过系统界面上传新的数据集文件。系统提供了上传表单，用户可以填写数据集名称和描述，并选择文件进行上传。系统会验证上传的文件是否合法，确保文件格式和内容正确后，保存数据集文件并更新数据集列表。

表3.15 上传数据集用例规约

|  |  |
| --- | --- |
| 用例名称 | 上传数据集 |
| 用例描述 | 用户可以上传新的数据集文件。 |
| 参与者 | 用户 |
| 前置条件 | 用户已登录系统。 |
| 后置条件 | 新数据集上传成功，系统更新数据集列表。 |

续表3.15

|  |  |
| --- | --- |
| 主要事件流 | 1. 用户在数据集管理页面选择“上传数据集”功能。 2. 系统显示上传数据集的表单，用户填写数据集名称、描述并选择数据集文件。 3. 用户提交上传表单。 4. 系统验证上传的文件格式和内容是否合法。 5. 如果验证通过，系统保存数据集文件并更新数据集列表。 6. 系统提示用户上传成功。 |
| 备选事件流 | 无 |

更新数据集用例规约如表3.16所示，用户可以通过系统界面更新已有的数据集文件或描述信息。系统提供了更新表单，用户可以选择新的数据集文件或修改描述信息进行更新。系统会验证上传的文件是否合法，确保文件格式和内容正确后，保存新的数据集文件或描述信息，并更新数据集的信息。

表3.16 更新数据集用例规约

|  |  |
| --- | --- |
| 用例名称 | 更新数据集 |
| 用例描述 | 用户可以更新已有的数据集文件或描述信息，确保数据集保持最新状态。 |
| 参与者 | 用户 |
| 前置条件 | 用户已登录系统，并已上传过数据集。 |
| 后置条件 | 数据集文件更新成功，系统保存最新的数据集信息。 |
| 主要事件流 | 1. 用户在数据集管理页面选择某个已上传的数据集。 2. 用户选择更新数据集功能。 3. 系统显示更新数据集的表单，用户选择新的数据集文件或修改描述信息。 4. 用户提交更新表单。 5. 系统验证上传的文件格式和内容是否合法。 6. 如果验证通过，系统更新数据集文件并保存更改。 7. 系统提示用户更新成功。 |
| 备选事件流 | 无 |

删除数据集用例规约如表3.17所示，用户可以通过系统界面删除不再需要的数据集。系统提供了删除功能，用户可以选择并确认删除特定数据集。系统会在用户确认后删除数据集文件，并更新数据集列表，确保用户的数据集管理界面保持最新状态。

表3.17 删除数据集用例规约

|  |  |
| --- | --- |
| 用例名称 | 删除数据集 |
| 用例描述 | 用户可以删除不再需要的数据集 |
| 参与者 | 用户 |
| 前置条件 | 用户已登录系统，并已上传过数据集。 |
| 后置条件 | 选定的数据集被删除，系统更新数据集列表。 |

续表3.17

|  |  |
| --- | --- |
| 主要事件流 | 1. 用户在数据集管理页面选择某个已上传的数据集。 2. 用户选择删除数据集功能。 3. 系统提示用户确认删除操作。 4. 用户确认删除操作。 5. 系统删除选定的数据集文件，并更新数据集列表。 6. 系统提示用户删除成功。 |
| 备选事件流 | 无 |

## 4.3 系统数据建模

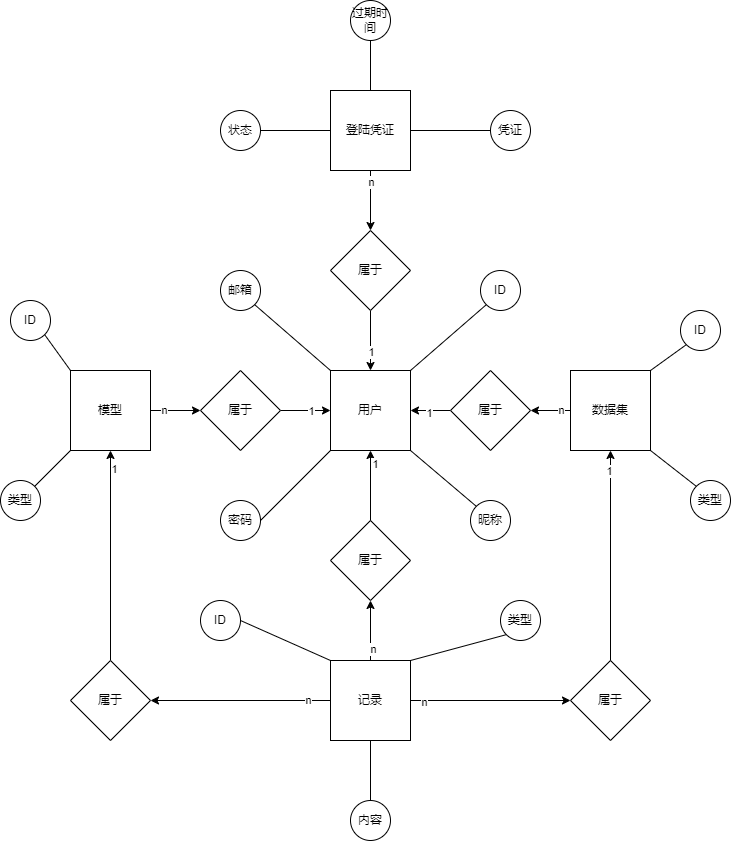
如图4.x所示，本系统主要的实体包括：用户、登录凭证、模型、数据集、记录。一个用户可以拥有一个或多个登录凭证，登录凭证包含过期时间、状态和凭证内容。一个用户可以拥有多个记录，每个记录有一个唯一的ID和内容。一个用户可以使用多个模型，每个模型有一个唯一的ID和类型。一个用户可以关联多个数据集，每个数据集有一个唯一的ID和类型。一个模型可以关联多个记录，一个数据集可以关联多个记录。

图4.1 系统E-R图

## 4.4 非功能性需求分析

非功能性需求是系统在功能之外必须满足的一系列标准，确保系统的性能、可靠性、安全性和用户体验等方面达到预期。

### 4.4.1 性能需求

（1）响应时间：80%的操作请求应在2秒内完成，所有请求应在5秒内完成。

（2）吞吐量：系统应能够支持至少100个并发用户同时上传、查看、更新和删除数据集。

（3）资源利用率：在高负载情况下，CPU和内存利用率不超过80%。

### 4.4.2 可靠性需求

系统的全年可用性应达到99.9%，即全年不可用时间不超过8.76小时。

### 4.4.3 安全性需求

（1）数据安全：所有敏感数据在传输和存储时都应进行加密。

（2）身份验证：用户在登录时必须进行身份验证。

### 4.4.4 兼容性需求

（1）浏览器兼容性：系统应在主流浏览器上正常运行，包括Chrome、Firefox、Safari和Edge。

（2）平台兼容性：系统应能够在不同操作系统和硬件环境下正常运行，包括Windows、macOS和Linux。

# 4 系统设计与实现

## 4.1 系统架构设计

图4.1 系统体系结构图

系统总体架构由前端模块、后端模块、数据库模块、模型评估模块构成。前端模块使用Vue框架构建，提供用户交互界面，负责数据展示和用户操作。Axios用于发送HTTP请求，与后端SpringBoot服务进行数据交互。Vue Router用于管理应用中的路由，实现单页面应用。Vuex用于状态管理，管理应用中不同组件之间的共享状态。Element Plus用于构建用户界面。

后端模块使用SpringBoot框架构建，负责业务逻辑处理、数据管理和API接口。Controller层负责处理Vue前端发来的HTTP请求，Service层提供具体的业务逻辑，DAO层负责相关的数据库操作。

模型评估模块使用Pytorch和FastAPI框架等实现，负责模型的准确性、鲁棒性和公平性评估。ART用于生成对抗性示例，Fairlearn用于公平性评估，Scikit-learn用于数据分析。

数据库模块使用MySQL数据库存储用户信息、数据集、模型信息和评估结果，并使用MyBatis与数据库进行交互。

## 4.2 系统类图设计

### 4.2.1 账号管理类图

图4.2 账号管理类图

User类表示系统中的用户实体，包含用户的基本信息和属性，如用户名、密码、电子邮件、账号状态等。LoginTicket类表示用户登录凭证，用于管理用户的登录状态，包括凭证字符串、关联用户ID、凭证状态和过期时间。

UserMapper 接口用于定义用户相关的数据库操作方法。selectByEmail():通过电子邮件查询用户。selectByName()通过用户名查询用户。selectById()通过用户ID查询用户。updateStatus()更新用户状态。updatePassword()更新用户密码。insertUser()将新用户信息插入数据库。LoginTicketMapper接口用于定义登录凭证相关的数据库操作方法。selectByTicket()通过票据字符串查询登录凭证。updateLoginTicket()更新登录票据状态。insertLoginTicket()将新的登录凭证信息插入数据库。

UserService类包含用户相关的业务逻辑，实现用户注册、登录、注销、激活等功能。register()验证用户信息合法性，插入新用户信息，返回注册结果。activation()验证激活码，更新用户状态为已激活。logout()使登录凭证失效，清除用户会话信息。findUserById()根据用户ID查找用户，返回用户信息。login()验证用户名和密码，生成登录票据，返回登录结果和票据信息。findLoginTicket()根据凭证字符串查找登录凭证。

UserController类处理与用户相关的HTTP请求，调用UserService执行具体的业务逻辑。updateUser()接收更新用户信息的请求，调用UserService执行更新操作，返回操作结果。getUserById()接收获取用户信息的请求，调用UserService获取用户信息，返回用户对象。LoginController类处理与用户登录、注册、注销、激活等相关的HTTP请求。login()接收登录请求，调用UserService执行登录逻辑，返回登录结果。getKaptcha()生成并返回验证码图像。register(接收注册请求，调用UserService执行注册逻辑，返回注册结果。logout()接收注销请求，调用UserService执行注销逻辑，返回注销结果。activation()接收激活请求，调用UserService执行激活逻辑，返回激活结果。

### 模型测试类图

图4.3 模型测试类图

Record类表示系统中的评估记录实体，包含评估任务的基本信息和属性。RecordMapper接口用于定义评估记录相关的数据库操作方法。insertRecord()将新的记录信息插入数据库。updateRecord()更新记录信息。deleteRecord()删除指定ID的记录。selectRecordById()根据记录ID查询记录。selectRecord()查询用户所有记录。searchModel()根据关键词搜索记录。

EvaluationService类包含与评估相关的业务逻辑，实现评估报告生成、记录管理等功能。getAccuracyReport()生成准确性评估报告。getFairnessReport()生成公平性评估报告。getRobustnessReport()生成鲁棒性评估报告。addRecord()添加新的评估记录。findRecordById()根据记录ID查找评估记录。findRecord()查找所有评估记录。modifyRecord()修改评估记录。searchModel()根据关键词搜索记录。

EvaluationController类处理与评估相关的HTTP请求，调用EvaluationService执行具体的业务逻辑。evalRobustness()处理鲁棒性评估请求。evalFairness()处理公平性评估请求。evalAccuracy()处理准确性评估请求。RecordController类处理与评估记录相关的HTTP请求，调用EvaluationService执行具体的业务逻辑。searchRecord()根据关键词搜索评估记录。deleteRecord()删除指定ID的评估记录。getRecordById()根据记录ID获取评估记录。searchModel()处理搜索记录的请求，返回搜索结果。

### 4.2.3 模型管理类图

图4.4 模型管理类图

Model类表示系统中的模型实体，包含模型的基本信息和属性。ModelMapper接口用于定义模型相关的数据库操作方法。insertModel()将新的模型信息插入数据库。updateModel()更新模型信息。deleteModel()删除指定ID的模型。selectModel()查询所有模型。selectModelById()根据模型ID查询模型。searchModel()根据关键词搜索模型。

ModelService类包含与模型管理相关的业务逻辑，实现模型的增删改查等功能。updateModel()更新模型信息。findModel()查询所有模型。insertModel()添加新的模型信息。findModelById()根据模型ID查找模型。deleteModel()删除指定ID的模型。searchModel()根据关键词搜索模型。

ModelController类处理与模型相关的HTTP请求，调用ModelService执行具体的业务逻辑。deleteModel()处理删除模型的请求，调用ModelService执行删除操作，返回操作结果。searchModel()处理搜索模型的请求，调用ModelService执行搜索操作，返回搜索结果。addModel()处理添加模型的请求，调用 ModelService执行添加操作，返回操作结果。updateModel()处理更新模型的请求，调用ModelService执行更新操作，返回操作结果。

### 数据集管理类图

图4.5 数据集管理类图

Dataset 类表示系统中的数据集实体，包含数据集的基本信息和统计属性。DatasetMapper接口用于定义数据集相关的数据库操作方法。insertDataset()将新的数据集信息插入数据库。updateDataset()更新数据集信息。deleteDataset()删除指定ID的数据集。selectDataset()查询所有数据集。selectDatasetById()根据数据集ID查询数据集。searchDataset()根据关键词搜索数据集。

DatasetService 类包含与数据集管理相关的业务逻辑，实现数据集的增删改查等功能。updateDataset()更新数据集信息。findDataset()查询所有数据集。insertDataset()添加新的数据集信息。findDatasetById()根据数据集ID查找数据集。deleteDataset()删除指定ID的数据集。searchDataset()根据关键词搜索数据集。

DatasetController类处理与数据集相关的HTTP请求，调用DatasetService执行具体的业务逻辑。deleteDataset()处理删除数据集的请求，调用 DatasetService 执行删除操作，返回操作结果。searchDataset()处理搜索数据集的请求，调用 DatasetService执行搜索操作，返回搜索结果。addDataset()处理添加数据集的请求，调用DatasetService执行添加操作，返回操作结果。updateDataset()处理更新数据集的请求，调用DatasetService执行更新操作，返回操作结果。

## 4.3 数据库设计

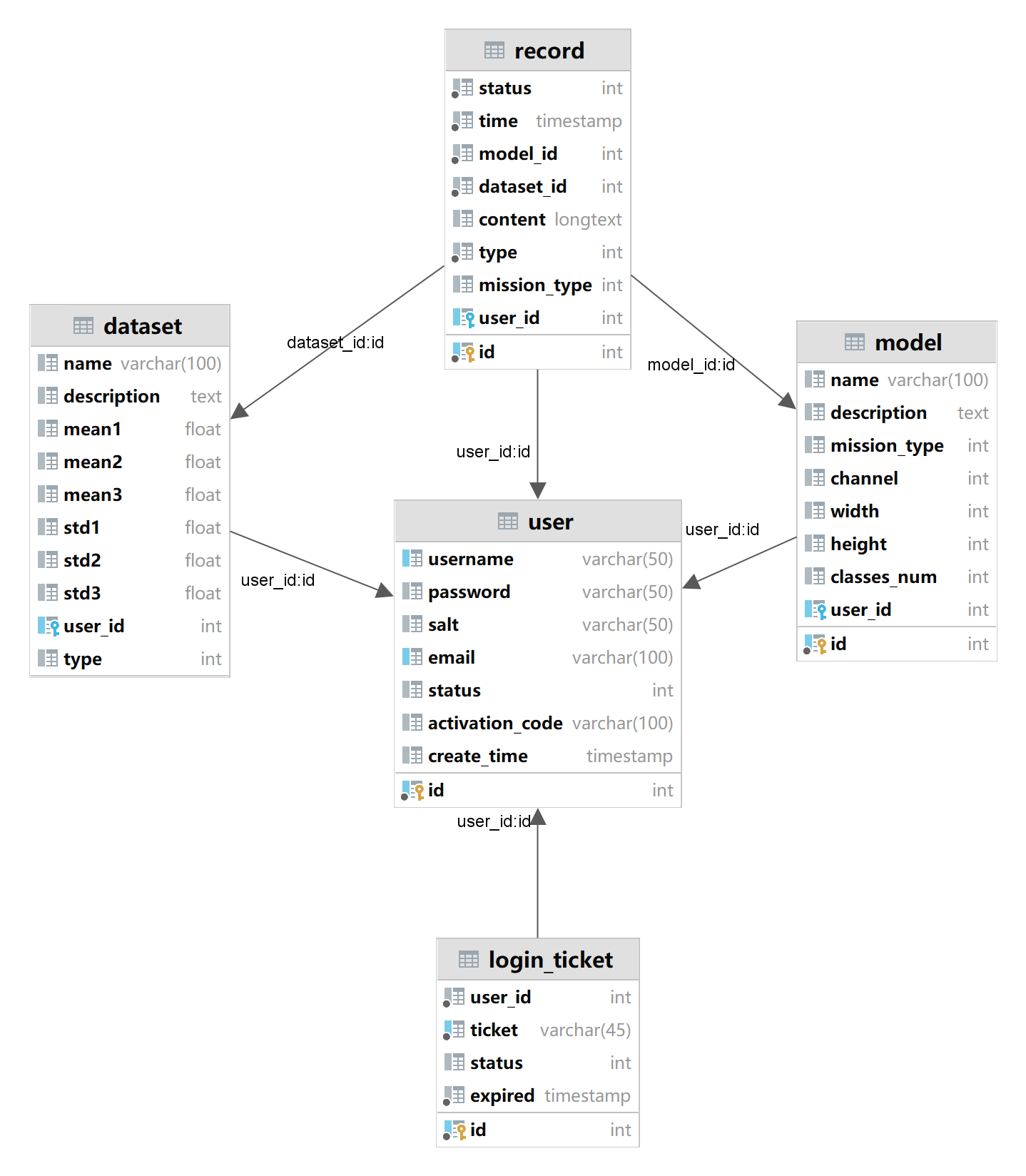
基于数据库E-R图设计，我们进一步设计数据库的信息表的具体内容。数据库信息表设计如图4.6所示

图4.6 数据库信息表设计

### 5.3.1 用户表

用户表存储系统中所有用户的基本信息，用于管理和验证用户身份。

表4.1 用户表设计

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 主键 | 非空 | 说明 |
| id | int | 是 | 是 | 用户ID |
| username | varchar(50) | 否 | 是 | 用户名 |
| password | varchar(50) | 否 | 是 | 密码 |

续表4.1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 主键 | 非空 | 说明 |
| salt | varchar(50) | 否 | 是 | 盐，用于密码加密 |
| email | varchar(100) | 否 | 是 | 电子邮件地址 |
| status | int | 否 | 是 | 用户状态 |
| activation\_code | varchar(100) | 否 | 否 | 激活码 |
| create\_time | timestamp | 否 | 是 | 创建时间 |

### 4.3.2 登陆凭证表

登录凭证表用于存储用户的登录凭证信息，管理用户的会话和登录状态。

表4.2 登陆凭证表设计

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 主键 | 非空 | 说明 |
| id | int | 是 | 是 | 凭证ID |
| user\_id | int | 否 | 是 | 用户ID（外键） |
| ticket | varchar(45) | 否 | 是 | 登录凭证 |
| status | int | 否 | 是 | 凭证状态 |
| expired | timestamp | 否 | 是 | 过期时间 |

### 4.3.3 记录表

记录表存储与用户、模型和数据集相关的操作记录或日志。

表4.3 记录表设计

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 主键 | 非空 | 说明 |
| id | int | 是 | 是 | 记录ID，这是记录表的主键，用于唯一标识每条记录。 |
| status | int | 否 | 是 | 记录状态，表示记录的当前状态。 |
| time | timestamp | 否 | 是 | 时间，记录操作发生的时间。 |
| model\_id | int | 否 | 是 | 模型ID（外键），关联到模型表的id，表示使用的模型。 |

续表4.3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 主键 | 非空 | 说明 |
| dataset\_id | int | 否 | 是 | 数据集ID（外键），关联到数据集表的id，表示使用的数据集。 |
| content | longtext | 否 | 否 | 内容，记录的详细信息。 |
| type | int | 否 | 是 | 类型，表示记录所评估的指标。 |
| mission\_type | int | 否 | 是 | 任务类型，表示记录涉及的任务类别。 |
| user\_id | int | 否 | 是 | 用户ID（外键），关联到用户表的id，表示记录的操作人。 |

### 4.3.4 模型表

模型表存储系统中使用的各种模型的信息，包括模型的配置和属性。

表4.4 模型表设计

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 主键 | 非空 | 说明 |
| id | int | 是 | 是 | 模型ID，这是模型表的主键，用于唯一标识每个模型。 |
| name | varchar(100) | 否 | 是 | 模型名称，模型的名称。 |
| description | text | 否 | 否 | 描述，模型的详细描述。 |
| mission\_type | int | 否 | 是 | 任务类型，表示模型的用途或任务类型。 |
| channel | int | 否 | 是 | 通道数，表示模型输入的通道数量。 |
| width | int | 否 | 是 | 宽度，表示模型输入的宽度。 |
| height | int | 否 | 是 | 高度，表示模型输入的高度。 |
| classes\_num | int | 否 | 是 | 类别数量，表示模型可以分类的类别数量。 |
| user\_id | int | 否 | 是 | 用户ID（外键），关联到用户表的id，表示模型的拥有者。 |

### 4.3.5 数据集表

数据集表存储系统中使用的各种数据集的信息，包括数据集的统计信息和属性。

表4.5 数据集表设计

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 主键 | 非空 | 说明 |
| id | int | 是 | 是 | 数据集ID，这是数据集表的主键，用于唯一标识每个数据集。 |
| name | varchar(100) | 否 | 是 | 数据集名称，数据集的名称。 |
| description | text | 否 | 否 | 描述，数据集的详细描述。 |
| mean1 | float | 否 | 否 | 平均值1，数据集的统计信息。 |
| mean2 | float | 否 | 否 | 平均值2，数据集的统计信息。 |
| mean3 | float | 否 | 否 | 平均值3，数据集的统计信息。 |
| std1 | float | 否 | 否 | 标准差1，数据集的统计信息。 |
| std2 | float | 否 | 否 | 标准差2，数据集的统计信息。 |
| std3 | float | 否 | 否 | 标准差3，数据集的统计信息。 |
| user\_id | int | 否 | 是 | 用户ID（外键），关联到用户表的id，表示数据集的拥有者。 |
| type | int | 否 | 是 | 类型，表示数据集的类别或类型。 |

## 4.4 核心功能详细设计与实现

### 4.4.1 账号管理

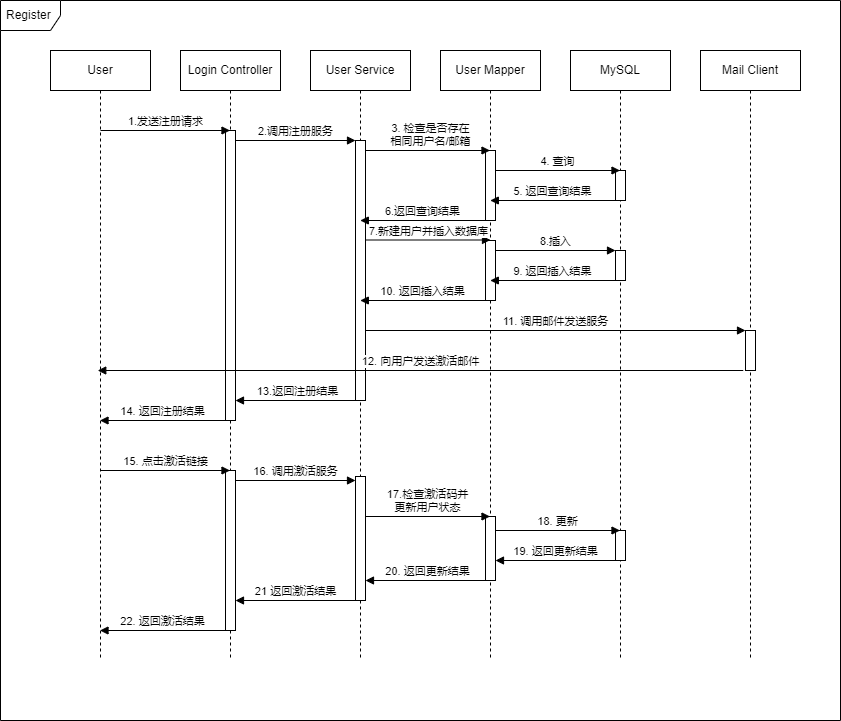


图4.7 注册与激活时序图

图5.x所示为用户注册与激活所对应的时序图。用户通过前端界面发送注册请求到 Login Controller。Login Controller 接收到请求后，调用 User Service 提供的注册服务。User Service 调用 User Mapper，检查数据库中是否存在相同的用户名或邮箱。User Mapper执行查询操作。User Mapper 将查询结果返回给 User Service。

如果查询结果表明用户名或邮箱不存在，User Service调用 User Mappe将新用户数据插入数据库。User Mapper 将插入结果返回给 User Service。User Service 调用 Mail Client 发送激活邮件。Mail Client 将激活邮件发送给用户。User Service 将注册结果返回给 Login Controller。Login Controller 将注册结果返回给用户。

用户通过邮箱中的激活链接访问激活服务。Login Controller 接收到激活请求后，调用 User Service 提供的激活服务。User Service 检查激活码的有效性，并调用 User Mapper更新用户状态为已激活。User Mapper 将更新结果返回给 User Service。User Service 将激活结果返回给 Login Controller。Login Controller 将激活结果返回给用户。

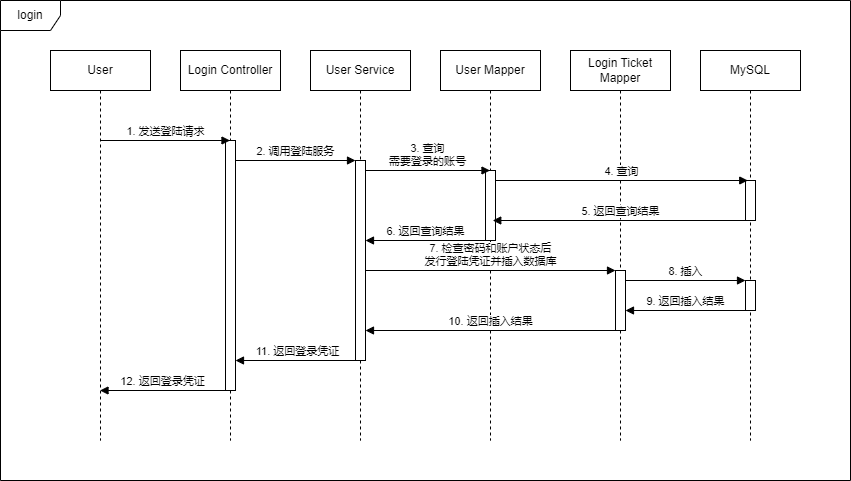


图4.8 登录时序图

图5.x所示为用户登录所对应的时序图。用户通过前端界面发送登录请求到 Login Controller。Login Controller 接收到请求后，调用 User Service 提供的登录服务。User Service 调用 User Mapper，根据用户名查询数据库中对应的账号信息。User Mapper 将查询结果返回给 User Service。User Service 验证用户输入的密码是否正确，并检查账户状态（如是否已激活）。如果验证通过，生成登录凭证。User Service 调用 Login Ticket Mapper，将生成的登录票据插入数据库。Login Ticket Mapper 将插入结果返回给 User Service。User Service 将登录票据返回给 Login Controller，然后 Login Controller 将登录凭证返回给用户。

### 模型测试

图4.9 模型测试时序图

模型测试是本系统的核心功能。图5.x所示为用户执行模型测试并查看测试记录所对应的时序图。用户通过前端界面选择需要评估的模型、数据集、评估类型（准确性、鲁棒性、公平性）与所需的参数，发送模型测试请求。 EvaluationController接收到测试请求后根据评估类型，调用相应的服务方法。EvaluationService接收到请求后，执行相应的测试逻辑。

在执行测试逻辑之前，EvaluationService通过RecordMapper将新纪录记录插入数据库，然后调用FastAPI接口执行具体的模型评估逻辑。测试完成后，EvaluationService更新记录状态，通过RecordMapper将更新后的记录保存到数据库。

用户通过前端页面选择需要查看的记录，发送记录查询请求。RecordController接收到请求后，调用相应的服务方法。EvaluationService通过RecordMapper查询记录，并将记录返回给RecordController。RecordController将查询到的记录返回给用户。

对于不同的评估任务，我们采用了不同的评估逻辑。以图像分类任务为例，具体评估逻辑如下：

（1） 数据预处理

transform\_list = []  
if enable\_resize:  
 transform\_list.append(transforms.Resize((resize1, resize2))) transform\_list.append(transforms.ToTensor())  
if enable\_normalize:  
 transform\_list.append(transforms.Normalize((mean1, mean2, mean3), (std1, std2, std3))) transform = transforms.Compose(transform\_list)  
testloader = DataLoader(testset, batch\_size=batch\_size, shuffle=False)

模型测试过程中根据用户所需要的参数灵活处理图像数据，确保输入数据格式统一且经过适当预处理。enable\_resize检查是否启用图像大小调整。如果启用，则添加Resize变换，将图像调整为指定的尺寸(resize1, resize2)。enable\_normalize检查是否启用图像归一化。如果启用，则添加 Normalize变换，使用指定的均值(mean1, mean2, mean3)和标准差(std1, std2, std3)对图像进行归一化。

（2） 预测

进行前向传播、获取预测结果并收集实际标签和预测标签，以便后续进行评估。

（3） 准确性评估

图像分类任务的准确性评估使用classification\_report函数，该函数是 Scikit-learn中用于生成分类报告的工具。该函数用于计算分类指标，例如精确度（Precision）、召回率（Recall）、F1 分数（F1-Score）以及每个类的支持数（Support）。这些指标帮助评估分类模型的性能。

report = classification\_report(y\_test, y\_pred, output\_dict=True)

（4） 鲁棒性评估

图像分类的鲁棒性评估使用ART进行。我们使用PyTorchClassifier 封装了PyTorch模型，使其能够在ART框架中进行各种机器学习任务。

classifier = PyTorchClassifier(

model=net,

loss=torch.nn.CrossEntropyLoss(),

optimizer=None,

input\_shape=(channels, width, height),

nb\_classes=classes\_num,

device\_type='gpu'

)

然后根据用户要求的扰动大小和扰动步长计算实证鲁棒性与敏感损失性。计算实证鲁棒性时使用快速梯度符号法。

sensitivity = loss\_sensitivity(classifier, test\_images\_preprocessed, test\_labels\_one\_hot.numpy())

epsilons = np.arange(0, eps, eps\_step)

params = {'eps\_step': eps\_step, 'eps': eps}

emp\_robustness = empirical\_robustness(classifier, test\_images\_preprocessed, 'fgsm', params)

我们使用DeepFool算法对模型进行攻击，观察模型在不同攻击强度下准确率的变化情况，然后根据结果绘制安全曲线。

accuracies = []

for epsilon in epsilons:

attack = DeepFool(classifier, epsilon, batch\_size=4)

x\_test\_adv = attack.generate(x=test\_images\_preprocessed, transform=transform)

adv\_dataset = MyDataset(testset.targets, x\_test\_adv)

testloader = DataLoader(adv\_dataset, batch\_size=16, shuffle=False)

correct = 0

total = 0

with torch.no\_grad():

for inputs, labels in testloader:

inputs = inputs.to(device)

labels = labels.to(device)

outputs = net(inputs)

\_, predicted = torch.max(outputs, 1)

correct += (predicted == labels).sum().item()

total += labels.size(0)

accuracy = correct / total

accuracies.append(accuracy)

plt.plot(epsilons, accuracies)

plt.xlabel('Epsilon')

plt.ylabel('Accuracy')

plt.title('Security Curve')

plt.savefig('result.png')

（5） 公平性评估

当评估模型的公平性时，很多公平性指标是基于二分类的。为了简化分析，我们根据用户的需求将实际标签与预测标签转换为二进制标签。

report["demographic parity difference"] = demographic\_parity\_difference(y\_true=y\_test, y\_pred=y\_pred, sensitive\_features=sensitive\_feature)  
report["demographic parity ratio"] = demographic\_parity\_ratio(y\_true=y\_test, y\_pred=y\_pred, sensitive\_features=sensitive\_feature)  
report["equalized odds difference"] = equalized\_odds\_difference(y\_true=y\_test, y\_pred=y\_pred,  
sensitive\_features=sensitive\_feature)  
report["equalized odds ratio"] = equalized\_odds\_ratio(y\_true=y\_test, y\_pred=y\_pred, sensitive\_features=sensitive\_feature)

for i in range(len(y\_test)):  
 if y\_test[i] == pos\_label:  
 y\_test[i] = 1  
 else:  
 y\_test[i] = 0  
 if y\_pred[i] == pos\_label:  
 y\_pred[i] = 1  
 else:  
 y\_pred[i] = 0

使用Fairlearn中的MetricFrame对不同敏感特征组进行性能指标计算。

metrics = {  
 "accuracy": accuracy\_score,  
 "precision": precision\_score,  
 "false positive rate": false\_positive\_rate,  
 "false negative rate": false\_negative\_rate,  
 "selection rate": selection\_rate,  
 "count": count,  
}  
metric\_frame = MetricFrame(  
 metrics=metrics, y\_true=y\_test, y\_pred=y\_pred, sensitive\_features=sensitive\_feature,  
)  
report = metric\_frame.by\_group.to\_dict()

进一步计算公平性指标，如人口平等差异和比例、平等机会差异和比例。

对于图像检索任务，评估的具体逻辑如下：

（1） 数据预处理

同图像分类任务。

（2） 特征提取

提取模型在测试集上的特征输出，并将这些特征和对应的标签收集到列表中，最终转换为NumPy数组。这些提取的特征可以用于后续的分析、可视化。

（3） 准确性评估

计算平均精度均值（MAP）、Top-K 准确率、ROC AUC面积，并绘制精确度-召回率曲线。

map\_score = evaluate\_map(features, labels)  
top\_k\_accuracy = compute\_top\_k\_accuracy(features, labels, k=k)  
roc\_auc = compute\_roc\_auc(features, labels)  
plot\_precision\_recall\_curve(features, labels)

（4） 鲁棒性评估

为样本加上噪音、旋转样本、压缩样本，然后观察模型在不同程度的扰动下MAP和Top-K准确率的下降程度。

def plot\_robustness\_curve(transform\_fn, k, perturbation\_levels, perturbation\_name):  
 map\_scores = []  
 top\_k\_accuracies = []  
 for level in perturbation\_levels:  
 map\_score, top\_k\_accuracy = evaluate\_robustness(transform\_fn, k, level)  
 map\_scores.append(map\_score)  
 top\_k\_accuracies.append(top\_k\_accuracy)  
 plt.figure()  
 plt.plot(perturbation\_levels, map\_scores, marker='o', label='mAP')  
 plt.plot(perturbation\_levels, top\_k\_accuracies, marker='x', label='Top-K Accuracy')  
 plt.xlabel(perturbation\_name)  
 plt.ylabel('Score')  
 plt.title(f'Robustness Curve for {perturbation\_name}')  
 plt.legend()  
 plt.savefig(perturbation\_name + '.png')  
plot\_robustness\_curve(add\_noise, k, noise\_levels, 'noise\_level')  
plot\_robustness\_curve(rotate\_image, k, rotation\_levels, 'rotation')  
plot\_robustness\_curve(compress\_image, k, compression\_levels, 'quality')

（5）公平性评估

评估模型在不同群体上的公平性，计算不同群体的MAP和 Top-K 准确率，并绘制相应的图表。

def evaluate\_fairness(features, labels, sensitive\_features):  
 unique\_groups = np.unique(sensitive\_features)  
 group\_map\_scores = {}  
 for group in unique\_groups:  
 group\_indices = np.where(sensitive\_features == group)[0].astype(int)  
 print(group\_indices)  
 group\_features = features[group\_indices]  
 group\_labels = np.array(labels)[group\_indices]group\_map\_score = evaluate\_map(group\_features, group\_labels)  
 group\_map\_scores[group] = float(group\_map\_score)  
 print(f"Group {group} mAP: {group\_map\_score}")max\_map\_score = max(group\_map\_scores.values())  
 min\_map\_score = min(group\_map\_scores.values())  
 fairness\_gap = abs(max\_map\_score - min\_map\_score)  
 print(f"Fairness Gap: {fairness\_gap}")  
 return group\_map\_scores, fairness\_gap

def evaluate\_top\_k\_accuracy(similarity\_matrix, sensitive\_features, labels, k=5):  
 unique\_groups = np.unique(sensitive\_features)  
 top\_k\_accuracies = {}  
 for group in unique\_groups:  
 group\_indices = np.where(sensitive\_features == group)[0].astype(int)  
 group\_similarities = similarity\_matrix[group\_indices][:, group\_indices]  
 group\_labels = np.array(labels)[group\_indices]  
 accuracies = []  
 for i in range(len(group\_similarities)):  
 true\_label = group\_labels[i]  
 predicted\_scores = group\_similarities[i]top\_k\_indices = np.argsort(predicted\_scores)[-k:][::-1]  
 top\_k\_labels = group\_labels[top\_k\_indices]accuracies.append(1 if true\_label in top\_k\_labels else 0)  
 top\_k\_accuracies[group] = float(np.mean(accuracies))  
 print(f"Group {group} Top-{k} Accuracy: {top\_k\_accuracies[group]}")  
 return top\_k\_accuracies

### 模型管理

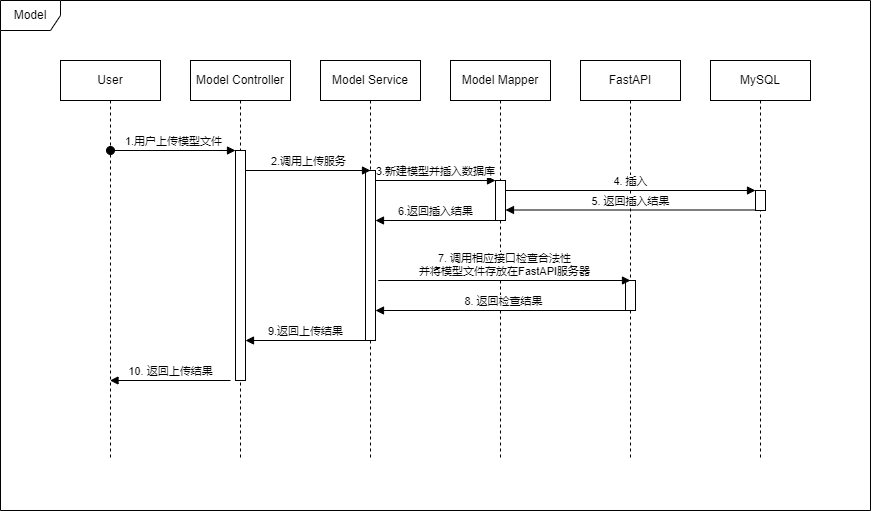


图4.10 模型上传时序图

图5.x所示为模型上传所对应的时序图。用户通过前端界面上传模型文件到 Model Controller。Model Controller 接收到请求后，调用 Model Service 提供的上传服务。Model Service 调用 Model Mapper，将新建模型信息插入数据库。Model Mapper 将插入结果返回给 Model Service。Model Service 调用 FastAPI 服务，将模型文件存储在 FastAPI 服务器，并进行合法性检查。FastAPI 返回检查结果给 Model Service。Model Service 将上传结果返回给 Model Controller。Model Controller 将上传结果返回给用户。

### 数据集管理

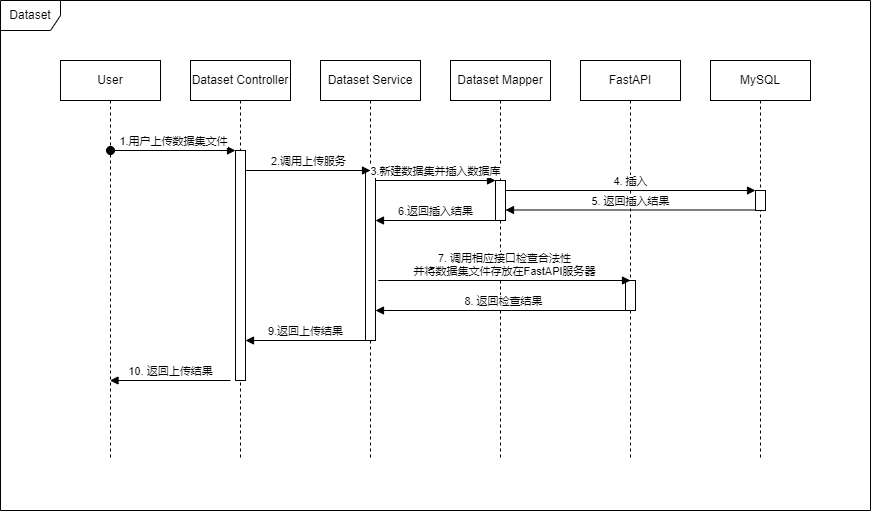


图4.11 模型上传时序图

图5.x所示为数据集上传所对应的时序图。用户通过前端界面上传数据集文件到 Dataset Controller。Dataset Controller 接收到请求后，调用 Dataset Service 提供的上传服务。Dataset Service 调用 Dataset Mapper，将新建数据集信息插入数据库。Dataset Mapper 将插入结果返回给 Dataset Service。Dataset Service 调用 FastAPI 服务，将数据集文件存储在 FastAPI 服务器，并进行合法性检查。FastAPI 返回检查结果给 Dataset Service。Dataset Service 将上传结果返回给 Dataset Controller。Dataset Controller 将上传结果返回给用户。

# 

# 

# 5 系统测试

## 5.1功能测试

### 5.1.1 账号管理功能测试



图5.1 账号管理界面

账号管理功能测试主要包括用户注册、登录、个人信息修改、账号激活等功能的测试，确保用户能够正常注册、登录和管理个人信息。

注册测试用例如表5.1所示，这些测试用例涵盖了用户注册过程中常见的输入情况，包括合法输入和几种非法输入。通过执行这些测试用例，系统能够验证其对不同类型输入的处理能力，确保用户注册功能的稳定性和可靠性。

表5.1 注册测试用例

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 用例编号 | 测试项目 | 执行步骤 | 输入数据 | 预期结果 | 实际结果 |
| TC-001 | 用户注册 | 合法输入 | 用户名："testuser"  密码："Test@123" 邮箱："[testuser@example.com](mailto:testuser@example.com)" | 系统发送验证邮件，用户点击激活链接后账号被激活，能够成功登录 | 一致 |

续表5.1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 用例编号 | 测试项目 | 执行步骤 | 输入数据 | 预期结果 | 实际结果 |
| TC-002 | 用户注册（非法输入：用户名为空） | 1. 访问注册页面  2. 填写注册表单：  用户名：""  密码："Test@123"  邮箱："[testuser@example.com](mailto:testuser@example.com)"  3. 提交注册表单 | 用户名："" 密码："Test@123" 邮箱："[testuser@example.com](mailto:testuser@example.com)" | 系统提示用户名不能为空，用户无法完成注册 | 一致 |
| TC-003 | 用户注册（非法输入：密码少于6位） | 1. 访问注册页面  2. 填写注册表单：  用户名："testuser"  密码："Test" 邮箱："[testuser@example.com](mailto:testuser@example.com)"  3. 提交注册表单 | 用户名："testuser"  密码："Test"  邮箱："[testuser@example.com](mailto:testuser@example.com)" | 系统提示密码长度不合法，用户无法完成注册 | 一致 |
| TC-004 | 用户注册（非法输入：邮箱格式不正确） | 1. 访问注册页面  2. 填写注册表单：  用户名："testuser"  密码："Test@123"  邮箱："testuser"  3. 提交注册表单 | 用户名："testuser"  密码："Test@123" 邮箱："testuser" | 系统提示邮箱格式不正确，用户无法完成注册 | 一致 |

登录测试用例如表5.2所示，这些测试用例涵盖了用户登录过程中常见的输入情况，包括正确的用户名和密码、错误的用户名以及错误的密码。通过执行这些测试用例，系统能够验证其对不同类型输入的处理能力，确保用户登录功能的稳定性和可靠性。

表5.2 用户登录测试用例

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 用例编号 | 测试项目 | 测试步骤 | 输入数据 | 预期结果 | 实际结果 |
| TC-005 | 用户登录（正确的用户名和密码） | 1. 访问登录页面  2. 输入用户名："testuser"  密码："Test@123"  3. 提交登录信息 | 用户名："testuser"  密码："Test@123" | 用户成功登录系统，可以访问账号管理和其他系统功能 | 一致 |
| TC-006 | 用户登录（错误的用户名） | 1. 访问登录页面  2. 输入用户名："wronguser"  密码："Test@123"  3. 提交登录信息 | 用户名："wronguser"  密码："Test@123" | 系统提示用户名或密码错误，用户无法登录 | 一致 |
| TC-007 | 用户登录（错误的密码） | 1. 访问登录页面  2. 输入用户名："testuser"  密码："Wrong@123"  3. 提交登录信息 | 用户名："testuser"  密码："Wrong@123" | 系统提示用户名或密码错误，用户无法登录 | 一致 |

修改个人信息测试用例如表5.3所示，这些测试用例涵盖了用户修改个人信息过程中常见的输入情况，包括合法输入和几种非法输入。通过执行这些测试用例，系统能够验证其对不同类型输入的处理能力，确保用户个人信息修改功能的稳定性和可靠性。

表5.3 修改个人信息测试用例

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 用例编号 | 测试项目 | 测试步骤 | 输入数据 | 预期结果 | 实际结果 |
| TC-008 | 修改个人信息（合法输入） | 1. 登录系统  2. 进入账号管理页面  3. 选择“修改个人信息”  4. 修改用户名："newtestuser"  5. 提交修改信息 | 新用户名："newtestuser" | 系统提示信息修改成功，用户可以查看更新后的个人信息 | 一致 |
| TC-009 | 修改个人信息（非法输入：用户名为空） | 1. 登录系统  2. 进入账号管理页面  3. 选择“修改个人信息”  4. 修改用户名：""  5. 提交修改信息 | 新用户名："" | 系统提示用户名不能为空，用户无法完成信息修改 | 一致 |
| TC-010 | 修改个人信息（非法输入：邮箱格式不正确） | 1. 登录系统  2. 进入账号管理页面  3. 选择“修改个人信息”  4. 修改邮箱："newtestuser"  5. 提交修改信息 | 新邮箱："newtestuser" | 系统提示邮箱格式不正确，用户无法完成信息修改 | 一致 |

账号激活测试用例如表5.4所示，这些测试用例涵盖了用户账号激活过程中常见的输入情况，包括有效激活码和无效激活码。通过执行这些测试用例，系统能够验证其对不同类型激活码的处理能力，确保用户账号激活功能的稳定性和可靠性。

表5.4 账号激活测试用例

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 用例编号 | 测试项目 | 测试步骤 | 输入数据 | 预期结果 | 实际结果 |
| TC-011 | 账号激活（有效激活码） | 1. 完成注册  2. 收到验证邮件  3. 点击激活链接 | 有效激活码 | 激活成功，可登录系统 | 一致 |
| TC-012 | 账号激活（无效激活码） | 1. 完成注册  2. 收到验证邮件  3. 点击激活链接 | 无效激活码 | 激活失败 | 一致 |

### 5.1.2 模型测试功能测试

模型测试功能测试主要包括模型的准确性评估、鲁棒性评估、公平性评估以及测试记录管理的测试。

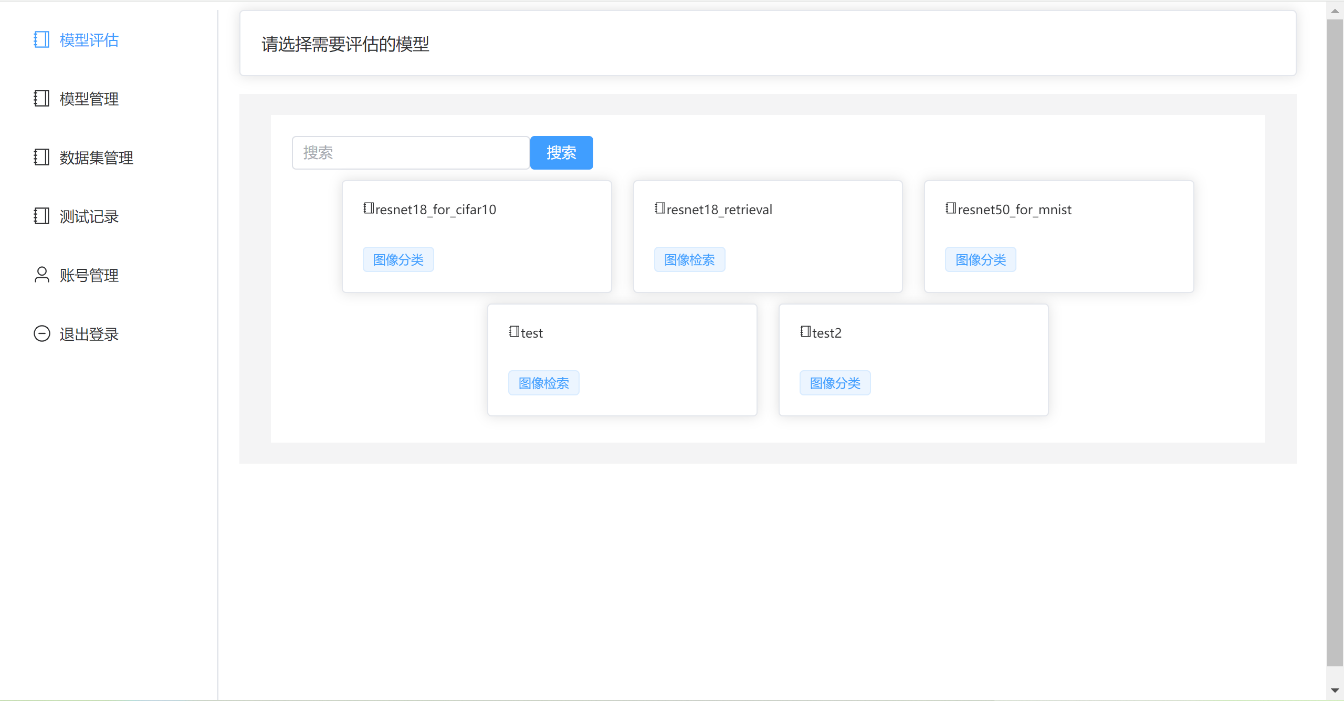
****

图5.2 模型测试界面

模型测试测试用例如表5.5所示，这些测试用例涵盖了模型评估过程中常见的输入情况，包括合法评估参数和不合法评估参数。通过执行这些测试用例，系统能够验证其对不同类型输入的处理能力，确保模型评估功能的稳定性和可靠性。

表5.5 模型测试测试用例

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 用例编号 | 测试项目 | 测试步骤 | 输入数据 | 预期结果 | 实际结果 |
| TC-101 | 准确性评估 | 1. 登录系统  2. 选择已上传的模型和数据集  3. 输入评估参数  4. 选择准确性评估  5. 提交评估请求 | 模型文件  数据集文件  评估参数 | 系统完成评估，生成准确性报告 | 一致 |
| TC-102 | 准确性评估（不合法评估参数） | 1. 登录系统  2. 选择已上传的模型和数据集  3. 输入不合法评估参数  4. 选择准确性评估  5. 提交评估请求 | 模型文件  数据集文件  不合法评估参数 | 系统提示评估参数无效 | 一致 |

续表5.5

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 用例编号 | 测试项目 | 测试步骤 | 输入数据 | 预期结果 | 实际结果 |
| TC-103 | 鲁棒性评估 | 1. 登录系统  2. 选择已上传的模型和数据集  3. 输入评估参数  4. 选择鲁棒性评估  5. 提交评估请求 | 模型文件  数据集文件  评估参数 | 系统完成评估，生成鲁棒性报告 | 一致 |
| TC-104 | 鲁棒性评估（不合法评估参数） | 1. 登录系统  2. 选择已上传的模型和数据集  3. 输入不合法评估参数  4. 选择鲁棒性评估  5. 提交评估请求 | 模型文件  数据集文件  不合法评估参数 | 系统提示评估参数无效 | 一致 |
| TC-105 | 公平性评估 | 1. 登录系统  2. 选择已上传的模型和数据集  3. 输入评估参数  4. 选择公平性评估  5. 提交评估请求 | 模型文件  数据集文件  评估参数 | 系统完成评估，生成公平性报告 | 一致 |
| TC-106 | 公平性评估（不合法评估参数） | 1. 登录系统  2. 选择已上传的模型和数据集  3. 输入不合法评估参数  4. 选择公平性评估  5. 提交评估请求 | 模型文件  数据集文件  不合法评估参数 | 系统提示评估参数无效 | 一致 |

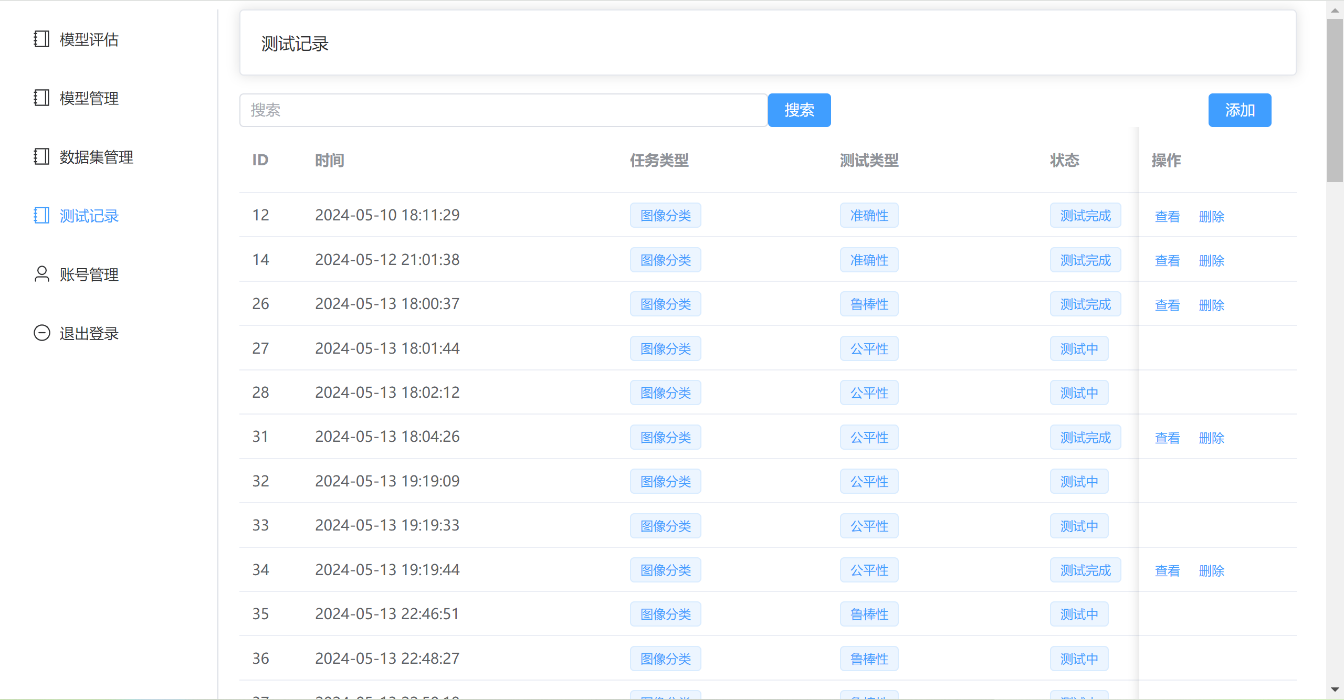
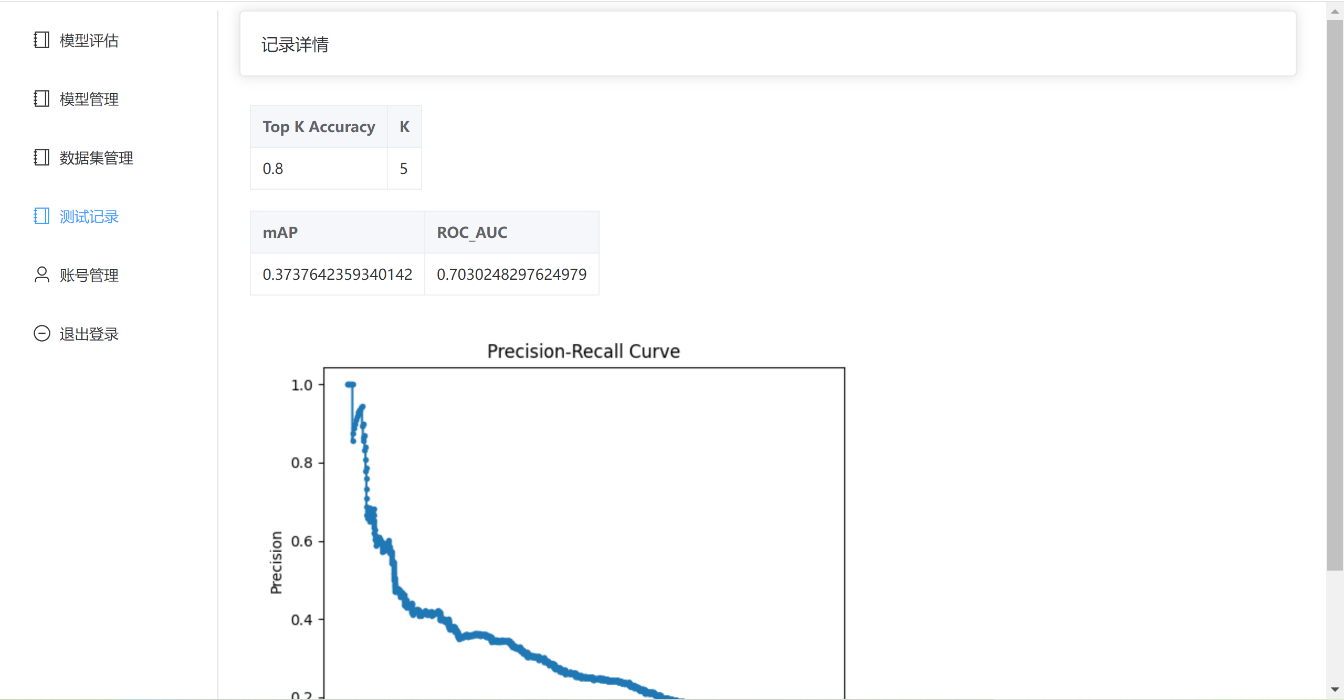


图5.3 测试记录管理界面

测试记录管理测试用例如表5.6所示，这些测试用例涵盖了测试记录管理过程中常见的操作，包括查看所有测试记录、搜索特定测试记录和删除单条测试记录。通过执行这些测试用例，系统能够验证其对不同类型操作的处理能力，确保测试记录管理功能的稳定性和可靠性。

表5.6 测试记录管理测试用例

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 用例编号 | 测试项目 | 测试步骤 | 输入数据 | 预期结果 | 实际结果 |
| TC-107 | 查看测试记录 | 1. 登录系统  2. 进入测试记录管理页面  3. 查看已完成的测试记录 | 无 | 系统显示所有测试记录 | 一致 |
| TC-108 | 查看特定测试记录 | 1. 登录系统  2. 进入测试记录管理页面  3. 搜索并查看特定测试记录 | 特定测试记录相关信息 | 系统显示匹配的测试记录 | 一致 |
| TC-109 | 删除单条测试记录 | 1. 登录系统  2. 进入测试记录管理页面  3. 选择要删除的测试记录  4. 点击删除按钮 | 无 | 系统提示删除成功，记录被删除 | 一致 |

图5.4 测试记录详情界面

### 5.1.3 模型管理功能测试

图5.5 模型管理界面

模型管理功能测试主要包括上传模型、更新模型和删除模型等功能的测试。

模型管理测试用例如表5.x所示，这些测试用例涵盖了模型管理过程中常见的操作，包括上传、更新和删除模型。通过执行这些测试用例，系统能够验证其对不同类型输入的处理能力，确保模型管理功能的稳定性和可靠性。

表5.7 模型管理测试用例

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 用例编号 | 测试项目 | 测试步骤 | 输入数据 | 预期结果 | 实际结果 |
| TC-201 | 上传模型（合法输入） | 1. 登录系统  2. 进入模型管理页面  3. 选择上传模型  4. 提交模型文件 | 模型文件 | 系统提示上传成功，模型出现在模型列表中 | 一致 |
| TC-202 | 上传模型（非法输入） | 1. 登录系统  2. 进入模型管理页面  3. 选择上传模型  4. 提交非法模型文件 | 非法模型文件 | 系统提示上传失败，模型未出现在模型列表中 | 一致 |
| TC-203 | 更新模型（合法输入） | 1. 登录系统  2. 进入模型管理页面  3. 选择要更新的模型  4. 提交新模型文件 | 新模型文件 | 系统提示更新成功，模型列表中显示更新后的模型 | 一致 |

续表5.7

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 用例编号 | 测试项目 | 测试步骤 | 输入数据 | 预期结果 | 实际结果 |
| TC-204 | 更新模型（非法输入） | 1. 登录系统  2. 进入模型管理页面  3. 选择要更新的模型  4. 提交非法模型文件 | 非法模型文件 | 系统提示更新失败，模型列表中未显示更新后的模型 | 一致 |
| TC-205 | 删除模型 | 1. 登录系统  2. 进入模型管理页面  3. 选择要删除的模型  4. 点击删除按钮 | 无 | 系统提示删除成功，模型从模型列表中消失 | 一致 |

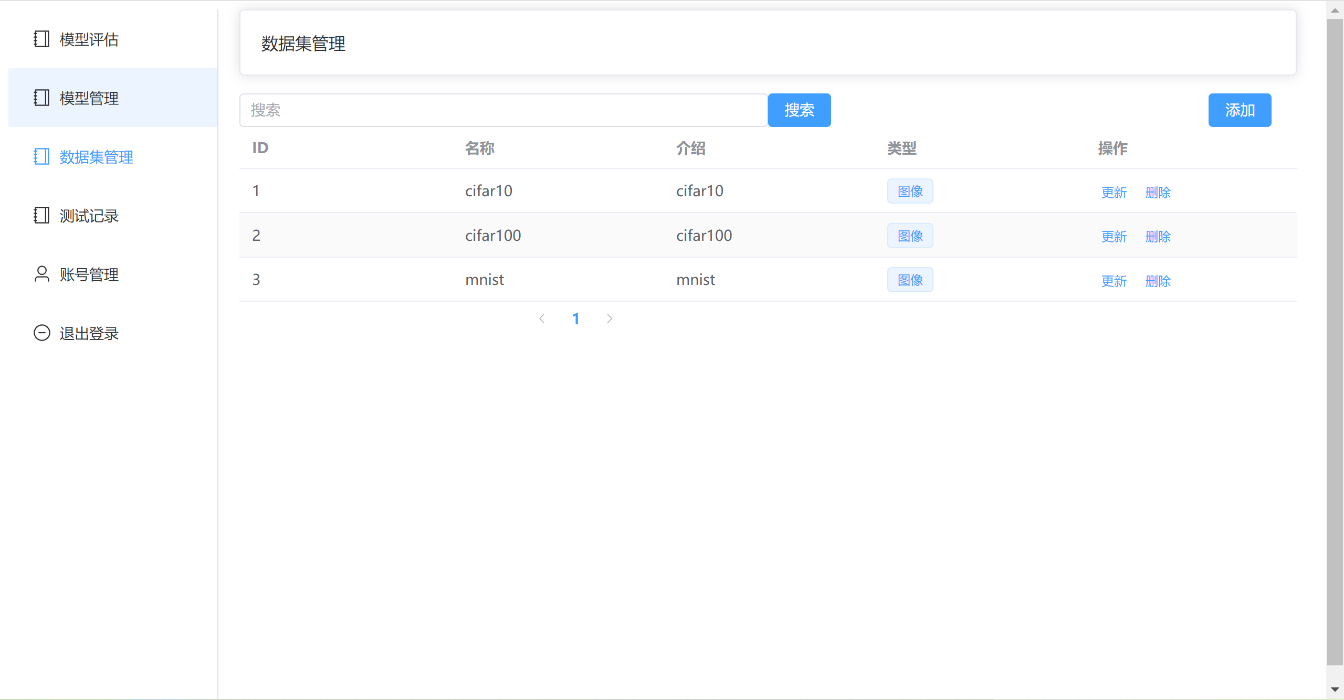


图5.6 数据集管理界面

### 5.1.4 数据集管理功能测试

数据集管理功能测试主要包括上传数据集、更新数据集和删除数据集等功能的测试。

数据集管理测试用例如表5.8所示，这些测试用例涵盖了数据集管理过程中常见的操作，包括上传、更新和删除数据集。通过执行这些测试用例，系统能够验证其对不同类型输入的处理能力，确保数据集管理功能的稳定性和可靠性。

表5.8 数据集管理测试用例

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 用例编号 | 测试项目 | 测试步骤 | 输入数据 | 预期结果 | 实际结果 |
| TC-301 | 上传数据集（合法输入） | 1. 登录系统  2. 进入数据集管理页面  3. 选择上传数据集  4. 提交数据集文件 | 数据集文件 | 系统提示上传成功，数据集出现在数据集列表中 | 一致 |
| TC-302 | 上传数据集（非法输入） | 1. 登录系统  2. 进入数据集管理页面  3. 选择上传数据集  4. 提交非法数据集文件 | 非法数据集文件 | 系统提示上传失败，数据集未出现在数据集列表中 | 一致 |
| TC-303 | 更新数据集（合法输入） | 1. 登录系统  2. 进入数据集管理页面  3. 选择要更新的数据集  4. 提交新数据集文件 | 新数据集文件 | 系统提示更新成功，数据集列表中显示更新后的数据集 | 一致 |
| TC-304 | 更新数据集（非法输入） | 1. 登录系统  2. 进入数据集管理页面  3. 选择要更新的数据集  4. 提交非法数据集文件 | 非法数据集文件 | 系统提示更新失败，数据集列表中未显示更新后的数据集 | 一致 |
| TC-305 | 删除数据集 | 1. 登录系统  2. 进入数据集管理页面  3. 选择要删除的数据集  4. 点击删除按钮 | 无 | 系统提示删除成功，数据集从数据集列表中消失 | 一致 |

## 5.2 非功能测试

### 5.2.1 性能测试

性能测试是确保系统在不同负载条件下能够稳定、高效运行的重要环节。通过对系统进行响应时间、吞吐量和资源利用率等方面的测试，可以全面评估系统的性能表现。我们采用ApiFox进行系统的性能测试。系统在不同并发数的性能情况如表5.9所示。

表5.9 性能测试

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 并发数 | 平均响应时间/ms | 最小响应实际/ms | 最大响应实际/ms | 吞吐量 | 失败率 |
| 10 | 16 | 7 | 281 | 7.51/s | 0.00% |
| 50 | 20 | 6 | 284 | 29.23/s | 0.00% |
| 100 | 37 | 7 | 666 | 7.44/s | 0.00% |

### 5.2.2 兼容性测试

兼容性测试是为了确保系统在不同环境下都能正常运行，包括不同的操作系统、浏览器等。通过兼容性测试，可以发现并解决系统在不同环境中可能存在的兼容性问题，保证用户在各种使用环境下都能获得一致的体验。本系统的兼容性测试结果如下：

系统在 Windows 10、macOS 和 Ubuntu 上均能正常运行。系统在 Google Chrome、Mozilla Firefox、Microsoft Edge 和 Safari 浏览器中均能正常显示和运行。

# 6 总结与展望

## 6.1 总结

本研究设计并实现了一个面向模型风险的测试系统，旨在全面评估人工智能系统中使用的图像分类和检索模型的性能。系统主要从准确性、鲁棒性和公平性三个维度对模型进行评估。具体而言，通过准确性评估可以量化模型在不同任务中的预测能力；通过鲁棒性评估可以了解模型在面对对抗性攻击和数据扰动时的稳定性；通过公平性评估可以确保模型对不同群体的公正性。

系统的架构采用前后端分离的设计，前端使用Vue框架实现，后端则采用SpringBoot框架，并通过MyBatis进行数据库操作。同时，模型评估部分使用FastAPI与PyTorch实现，提供高效的模型评估服务。系统支持用户上传数据集和模型，能够对多种图像分类和检索模型进行全面评估，并生成详细的结果报告。这些功能帮助用户发现和改进模型在实际应用中的潜在问题，提升模型的整体性能和可靠性。

## 7.2 展望

尽管本研究所设计的测试系统在模型评估方面取得了一定的成果，但仍有许多方面需要进一步优化和扩展。

（1） 评估指标

未来可以引入更多的评估指标，以提供更全面的评估结果。

（2） 应用领域

当前系统主要针对图像分类和检索任务，未来可以扩展到其他领域，如自然语言处理、语音识别等。

（3） 体验优化

未来可以通过改进界面设计、增加帮助文档和教程等方式，帮助用户更好地理解和操作系统。

# 参考文献

[1] GOODFELLOW I J, SHLENS J, SZEZEDY C. Explaining and Harnessing Adversarial Examples[EB/OL]. (2014-12-20)[2024-05-29]. <https://arxiv.org/abs/1412.6572>.

[2] MOOSAVI-DEZFOOLI S M, FAWZI A, FROSSARD P. DeepFool: A Simple and Accurate Method to Fool Deep Neural Networks[C]//2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2016: 2574-2582.

[3] CARLINI N, WAGNER D. Towards Evaluating the Robustness of Neural Networks[C]//2017 IEEE Symposium on Security and Privacy (SP). IEEE, 2017: 39-57. [2024-05-29]. <http://dx.doi.org/10.1109/SP.2017.49>.

[4] NICOLAE M I, SINN M, TRAN M N, et al. Adversarial Robustness Toolbox v1.0.0[EB/OL]. (2018-07-03)[2024-05-29]. <https://arxiv.org/abs/1807.01069>.

[5] ARPIT D, JASTRZĘBSKI S, BALLAS N, et al. A Closer Look at Memorization in Deep Networks[EB/OL]. (2017-06-16)[2024-05-29]. <https://arxiv.org/abs/1706.05394>.

[6] WENG T W, ZHANG H, CHEN P Y, et al. Evaluating the Robustness of Neural Networks: An Extreme Value Theory Approach[EB/OL]. (2018-01-31)[2024-05-29]. <https://arxiv.org/abs/1801.10578>.

[7] MADRY A, MAKELOV A, SCHMIDT L, et al. Towards Deep Learning Models Resistant to Adversarial Attacks[EB/OL]. (2017-06-19)[2024-05-29]. https://arxiv.org/abs/1706.06083.

[8] DONG Y, FU Q A, YANG X, et al. Benchmarking Adversarial Robustness[EB/OL]. (2019-12-26)[2024-05-29]. <https://arxiv.org/abs/1912.11852.2.3>

[9] WEERTS H, DUDÍK M, EDGAR R, et al. Fairlearn: Assessing and Improving Fairness of AI Systems[J]. Journal of Machine Learning Research, 2023, 24. [2024-05-29]. http://jmlr.org/papers/v24/23-0389.html.

[10] AGARWAL A, BEYGELZIMER A, DUDIK M, et al. A Reductions Approach to Fair Classification[C]//DY J, KRAUSE A, eds. Proceedings of the 35th International Conference on Machine Learning. Proceedings of Machine Learning Research, PMLR, 2018, 80: 60-69. [2024-05-29]. <https://proceedings.mlr.press/v80/agarwal18a.html>.

[11] AGARWAL A, DUDIK M, WU Z S. Fair Regression: Quantitative Definitions and Reduction-Based Algorithms[C]//CHAUDHURI K, SALAKHUTDINOV R, eds. Proceedings of the 36th International Conference on Machine Learning. Proceedings of Machine Learning Research, PMLR, 2019, 97: 120-129. [2024-05-29]. <https://proceedings.mlr.press/v97/agarwal19d.html>.

[12] BAROCAS S, HARDT M, NARAYANAN A. Fairness and Machine Learning: Limitations and Opportunities[M]. MIT Press, 2023.

[13] HARDT M, PRICE E, SREBRO N. Equality of Opportunity in Supervised Learning[C]//LEE D, SUGIYAMA M, LUXBURG U, GUYON I, GARNETT R, eds. Advances in Neural Information Processing Systems. Curran Associates, Inc., 2016, 29. [2024-05-29]. https://proceedings.neurips.cc/paper\_files/paper/2016/file/9d2682367c3935defcb1f9e247a97c0d-Paper.pdf.

[14] MySQL官方文档[EB/OL]. [2024-05-29]. <https://dev.mysql.com/doc/>.

[15] DUBOIS P. MySQL[M]. Addison-Wesley, 2013.

[16] ZAITSEV P, TKACHENKO V. High Performance MySQL: Optimization, Backups, and Replication[M]. O'Reilly Media, 2012. [2024-05-29]. <https://books.google.co.jp/books?id=D0b_Xg3UeXEC>.

[17] ORACLE. The Java Tutorials[EB/OL]. (2023)[2024-05-29]. <https://docs.oracle.com/javase/tutorial/>.

[18] GOSLING J J, JOY B, STEELE G L, et al. The Java Language Specification, Java SE 8 Edition[M]. Pearson Education, 2014. [2024-05-29]. <https://books.google.co.jp/books?id=1DaDAwAAQBAJ>.

[19] Spring官方文档[EB/OL]. [2024-05-29]. <https://spring.io/projects/spring-boot>.

[20] WALLS C. Spring in Action, Sixth Edition[M]. Manning, 2022. [2024-05-29]. <https://books.google.co.jp/books?id=9IBhzgEACAAJ>.

[21] MyBatis官方文档[EB/OL]. [2024-05-29]. <https://mybatis.org/mybatis-3/>.

[22] Python官方文档[EB/OL]. [2024-05-29]. https://docs.python.org/3/.

[23] FastAPI官方文档[EB/OL]. [2024-05-29]. https://fastapi.tiangolo.com/.

[24] PyTorch官方文档[EB/OL]. [2024-05-29]. https://pytorch.org/docs/.

[25] PASZKE A, GROSS S, MASSA F, et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library[EB/OL]. (2019-12-03)[2024-05-29]. <https://arxiv.org/abs/1912.01703>.

[26] Vue.js官方文档[EB/OL]. [2024-05-29]. https://vuejs.org/.

# 谢 辞

正文内容