|  |  |  |
| --- | --- | --- |
| 选题 | **2024年第十四届APMCM**  **亚太地区大学生数学建模竞赛（中文赛项）** | 参赛编号 |
| B | apmcm24102848 |

洪水灾害的数据分析与预测

**摘要**

洪水是由暴雨、急剧融冰化雪、风暴潮等自然因素引起的自然灾害，对经济、生态和人类生命构成严重威胁。随着人口增长和人类活动扩张，地表状态显著变化，洪灾程度和频率加剧，迫切需要利用先进技术进行防洪减灾。可以通过数据分析和数学建模的方法预测洪水发生的概率，通过对洪水数据的分析，识别与洪水发生密切相关的影响因素，建立洪水风险预警评价模型和洪水发生概率的预测模型，为制定防洪措施和应急预案提供科学依据，实现防洪减灾工作的智能化管理。

对于问题一，首先分析洪水数据，使用马氏距离法预处理数据，采用相关性分析和主成分分析等方法识别关键影响因素，并通过可视化手段展示结果，发现基础设施恶化、地形排水、季风强度等指标与洪水发生概率高度相关，而海岸线脆弱性、侵蚀、城市化等指标相关性较弱。

对于问题二，利用K-means聚类算法将洪水事件聚类成高、中、低风险类别，分析指标特征，选取合适指标建立基于梯度提升决策树（GBDT）的洪水预警评价模型，并进行灵敏度分析。

对于问题三，基于随机森林算法构建FloodRiskPredictor预测模型，选取20个指标中的关键因素进行洪水发生概率预测，并验证模型准确性。调整模型参数，使用交叉验证方法评估模型性能，随机森林模型在预测洪水发生概率方面表现出较高的准确性和稳定性，使用五个关键指标时模型性能依然良好。

对于问题四，应用FloodRiskPredictor模型预测test.csv中洪水发生概率，将结果填入submit.csv，并绘制直方图和折线图分析预测结果的分布特性，并通过QQ图、SW和KS检验方法进行检验，以验证数据的正态性，结果表明预测的洪水概率趋近于正态分布，不严格符合正态分布。

总的来说，本文通过数据预处理、相关性分析、K-means聚类算法、随机森林回归模型、决策树回归模型对洪水数据进行了深入分析，建立了有效的洪水风险预警和预测模型，为防洪减灾工作提供了有力的技术支持和决策依据，模型也可推广应用于其他自然灾害预测、城市管理和农业生产等领域，实现更广泛的社会效益。

**关键词：洪水灾害；梯度提升决策树；随机森林；K-means算法；KS检验**

目录

[一、 问题重述 3](#_Toc10954)

[1.1 问题背景 3](#_Toc30879)

[1.2 问题重述 3](#_Toc6146)

[二、 问题分析 3](#_Toc4649)

[2.1 问题一分析 3](#_Toc499)

[2.2 问题二分析 4](#_Toc8775)

[2.3 问题三分析 4](#_Toc4841)

[2.4 问题四分析 4](#_Toc10773)

[三、 模型假设 5](#_Toc20359)

[四、符号说明 5](#_Toc27587)

[五、模型的建立与求解 5](#_Toc18694)

[5.1 问题一的模型建立与求解 5](#_Toc21569)

[5.1.1数据预处理 5](#_Toc24246)

[5.1.2相关性分析 6](#_Toc15191)

[5.1.3结果分析 8](#_Toc28710)

[5.2 问题二的模型建立与求解 9](#_Toc30102)

[5.2.1数据标准化处理 9](#_Toc11738)

[5.2.2基于无监督聚类算法的风险聚类 9](#_Toc20168)

[5.2.3基于梯度提升决策树（GBDT）的洪水预警评价模型 16](#_Toc25729)

[5.3 问题三的模型建立与求解 21](#_Toc30908)

[5.3.1 全部指标 21](#_Toc16133)

[5.3.2 五个指标 25](#_Toc6570)

[5.4 问题四的模型建立与求解 27](#_Toc7660)

[六、模型评价 31](#_Toc16003)

[6.1 模型的优点 31](#_Toc22236)

[6.2 模型的不足 31](#_Toc22754)

[七、模型的改进与推广 31](#_Toc1359)

[7.1 模型的推广 31](#_Toc28034)

[7.2 模型的改进 32](#_Toc7593)

[八、参考文献 32](#_Toc32611)

[附录 33](#_Toc15960)

# 问题重述

* 1. **问题背景**

洪水是暴雨、急剧融冰化雪、风暴潮等自然因素引起的江河湖泊水量迅速增加，或者水位迅猛上涨的一种自然现象。洪水不仅会造成巨大的经济损失，还会对生态环境和人类生命安全构成严重威胁。历史上，无论是中国、欧洲还是西亚、非洲，都有大量关于洪水灾害的记载，显示了洪水灾害的普遍性和危害性。随着人口的增长和人类活动的扩张，如扩大耕地、围湖造田、乱砍滥伐等，地表状态发生了显著变化，汇流条件恶化，进一步加剧了洪灾的程度和频率。特别是森林砍伐导致的水土流失，已成为许多地区洪水频发的重要原因。自然生态系统的破坏，如湿地损失、河流湖泊淤积等，降低了自然环境的自我调节能力，使得洪水这种自然灾害的应对变得更为困难。

面对洪水灾害的严峻挑战，需要利用先进的技术手段进行数据分析和建模，以揭示洪水发生的规律和影响因素。这为数学建模和数据分析在防洪减灾中的应用提供了广阔的舞台。随着大数据和人工智能技术的快速发展，人们有能力处理和分析海量的数据，从中提取有价值的信息和规律，因此通过数据分析和数学建模的方法预测洪水发生的概率成为一种可行的手段。

本文通过对洪水数据的分析，分别找出相关性高的影响因素与相关性低的影响因素，基于这些相关性指标建立发生洪水不同风险的预警评价模型，并建立洪水发生概率的预测模型，这对于制定有效的防洪措施和应急预案具有重要意义，同时可以为决策者提供科学的依据和支持，实现防洪减灾工作的智能化管理。通过实时监测、预警和评估，可以更加有效地应对洪水灾害的挑战。最后本文将对预测模型进行验证与改进，以提高预测的的准确性与可靠性。

* 1. **问题重述**

问题 1.请分析附件train.csv中的数据，分析并可视化上述20个指标中，哪些指标与洪水的发生有着密切的关联？哪些指标与洪水发生的相关性不大？并分析可能的原因，然后针对洪水的提前预防，提出你们合理的建议和措施。

问题 2.将附件train.csv中洪水发生的概率聚类成不同类别，分析具有高、中、低风险的洪水事件的指标特征。然后，选取合适的指标，计算不同指标的权重，建立发生洪水不同风险的预警评价模型，最后进行模型的灵敏度分析。

问题 3.基于问题1中指标分析的结果，请建立洪水发生概率的预测模型，从20个指标中选取合适指标，预测洪水发生的概率，并验证你们预测模型的准确性。如果仅用5个关键指标，如何调整改进你们的洪水发生概率的预测模型？

问题 4.基于问题3中建立的洪水发生概率的预测模型，预测附件test.csv中所有事件发生洪水的概率，并将预测结果填入附件submit.csv中。然后绘制这74多万件发生洪水的概率的直方图和折线图，分析此结果的分布是否服从正态分布。

# 问题分析

**2.1 问题一分析**

问题一要求分析train.csv中的数据，分析并可视化哪些指标与洪水发生密切关联。

首先，我们需要对train.csv数据进行预处理，包括缺失值和异常值检验与处理，由于数据涉及到多个维度的特征，因此我们使用可视化箱形图作为数据预处理的方式；接着采用相关性分析和主成分分析等方法，计算并验证各个指标与洪水发生概率之间的相关系数，通过正态性检验决定使用皮尔逊系数；最后，我们通过可视化手段展示分析结果，识别出与洪水发生密切相关的关键因素。

**2.2 问题二分析**

问题二分为两小问。第一小问首先是对train.csv中的洪水发生概率聚类成不同类别（高、中、低风险），可以采用K-means聚类方法或者系统聚类等无监督的聚类方法进行聚类，聚类成三种类别，之后通过对聚类后的指标数据进行可视化，建立频率分布直方图，分析这三种类别的指标特征，对比不同风险类别的指标特征，识别关键影响因素。

第二小问选取合适的指标，根据指标重要性计算各指标的权重，之后基于权重和指标特征使用梯度提升决策树模型建立洪水风险预警评价模型，通过交叉验证法调整模型参数以及分割数据集进行模型训练和测试，评估模型性能，分析模型对变化的敏感程度。

**2.3 问题三分析**

在解决问题三的第一部分中，我们采用了随机森林算法来构建FloodRiskPredictor模型，涵盖了全部的20个指标。首先，对数据进行了必要的预处理，包括利用插值和中位数来处理缺失值，确保数据的完整性和一致性。我们将整个数据集按照7:3的比例分割，其中70%的数据被用作训练集，而剩余的30%则用作测试集。

为了优化模型的性能，我们运用遗传算法来调整随机森林的参数设置，以寻求最佳的训练效果。训练完成后，使用测试集对模型进行验证，通过一系列评价指标如精确度、召回率和F1分数等来评估模型的性能。此外，我们还对模型进行了灵敏性分析，以评估模型对于输入变量变化的敏感度，从而验证模型的稳健性。

基于第一小问中随机森林模型的输出，特别是各指标的重要性评分，我们选择了影响力最大的前五个指标来重新构建模型。这次，数据集的划分比例调整为8:2，即80%的数据用作训练集，20%作为测试集，这有助于我们在更大的测试集上验证模型的泛化能力。

在模型训练过程中，除了关注模型的预测精度，我们还特别关注均方误差（MSE）和均方根误差（RMSE）这两个评价指标，以准确地衡量预测值与实际值之间的偏差。通过交叉验证的方法，我们不仅能够评估模型在不同子集上的表现，还能进一步地检验模型的稳定性。基于这些分析，我们对模型进行了必要的调整和优化，以确保其在实际应用中的高效性和准确性。

**2.4 问题四分析**

在问题四中，我们采用了从问题三开发的洪水发生概率预测模型来分析 test.csv 中的数据。我们首先确保数据格式正确并进行了必要的预处理。接着，应用模型对测试数据进行了洪水概率预测，并将结果填充到 submit.csv 文件中。为了深入分析预测结果的分布特性，我们绘制了直方图和折线图，并通过QQ图和概率累积密度函数进一步评估了数据的正态性。虽然数据趋近于正态分布，但正态性检验如Kolmogorov-Smirnov和Anderson-Darling显示存在统计上显著的偏差。这些分析帮助我们验证了模型的有效性，并提供了对洪水概率分布深入理解的依据，展示了模型预测结果与理论正态分布之间的细微差异

# 模型假设

1. 假设不同的环境和人为因素（如地形排水、河流管理、季风强度）对洪水发生概率有显著影响。
2. 假设模型的参数能够通过优化算法找到，使得模型能够达到最佳的预测性能。
3. 假设使用训练集和测试集分割数据能够有效防止过拟合，确保回归模型在未知数据上表现稳定。

四、符号说明

|  |  |
| --- | --- |
| **符号** | **说明** |
|  | 斯皮尔曼相关系数 |
|  | 决策树的数目 |
|  | 学习率 |
|  | 簇类中心个数 |
|  | 树的最大深度 |
|  | 基尼不纯度 |
|  | 初始化均方损失函数 |
|  | 损失函数关于的梯度 |

五、模型的建立与求解

**5.1 问题一的模型建立与求解**

5.1.1数据预处理

对数据进行适当的预处理是建立有效预测模型的关键步骤之一。我们采取了多种数据清洗和特征工程技术以确保数据的质量和适用性，具体步骤如下：

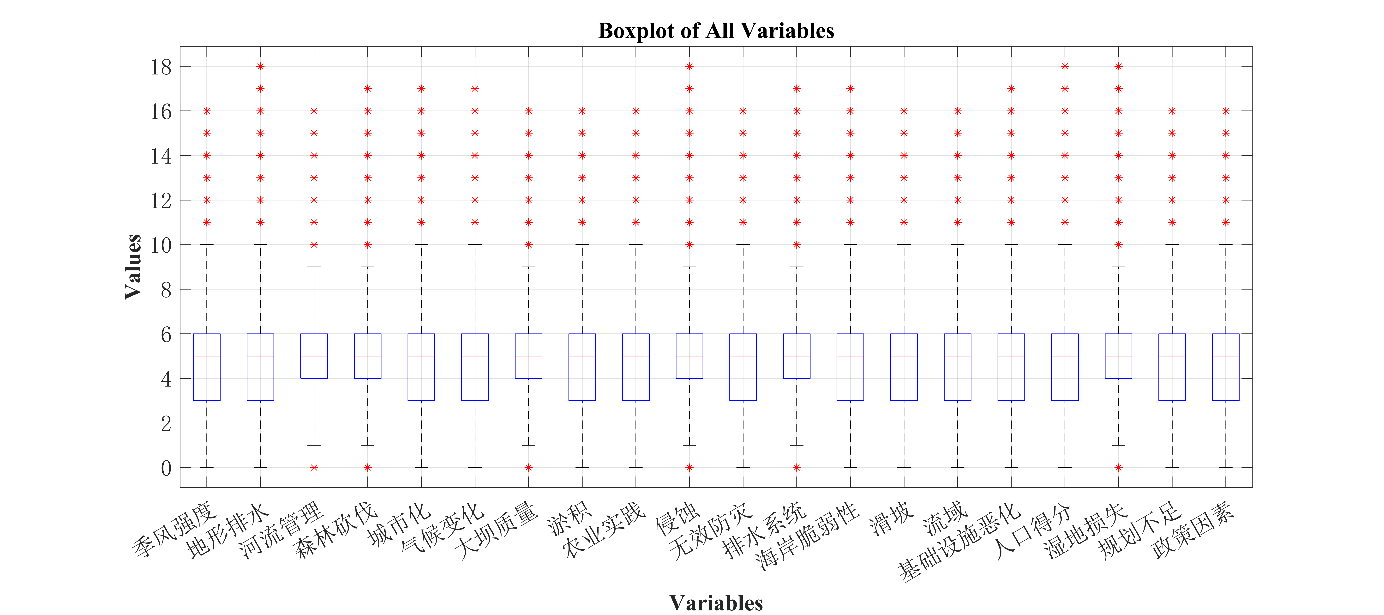
数据清洗：首先，我们从train.csv中删除了所有重复的记录，以确保数据的唯一性。接着，我们对缺失值进行了详细处理，对于缺失较少的特征，我们使用了该特征的

图 1：可视化箱型图

均值（对于连续变量）或众数（对于类别变量）进行填充。对于缺失严重的特征，我们考虑了其对模型的影响后选择予以删除。此外，我们通过可视化箱型图和统计方法IQR识别并处理了异常值，对极端值进行了适当的修剪或替换，以减少它们对整体模型性能的潜在负面影响。

我们使用可视化箱型图来识别异常值，如图1所示，这张箱线图揭示了数据集中各变量的分布特征和潜在的数据质量问题。从图中可见，大部分变量的数据分布相对对称，中位数通常位于箱体中心，显示出数据的均匀性。然而，每个变量都表现出大量的异常值，特别是在高值区域，这些异常值以红色“+”符号标识。异常值的广泛存在可能表明数据收集过程中的偏差或测量误差，建议进一步调查这些异常值的来源，并根据其性质决定是否进行修正或删除，在本研究的数据预处理阶段，我们对原始数据集 train.csv 进行了彻底清理。该数据集最初包含 1,048,575 条记录。通过识别和移除包含缺失值、异常值的数据条目后，共清理了 381,636 条数据。这一处理大幅减少了数据集的规模，确保了剩余数据的质量，最后将清洗后的数据储存在cleaned\_data.csv中，为进一步的分析和模型训练提供了更准确和可靠的基础。

数据集划分：我们将数据集分为训练集和测试集，通常采用80/20的划分比例，以确保模型在未见数据上的表现能够被公正地评估。训练集数据大小为533,551，测试集大小为133,388。

**5.1.2相关性分析**

在对数据进行预处理后，我们需要对数据进行相关性分析，通过计算各个指标与洪水发生概率之间的相关系数，判断这些指标与洪水发生的关联强度。

为选择合适的相关性分析模型，我们对所有指标数据进行KS检验计算各个指标的P值来定量判断它们是否符合正态分布的规律。KS检验通过比较样本分布与标准正态分布的差异，来检验数据的正态性。具体而言，我们对每个指标进行KS检验，得到相应的P值。如果P值小于显著性水平（通常设为0.05），则拒绝原假设，认为该指标不服从正态分布；否则接受原假设。

KS检验的结果显示，20个指标的P值都为0，小于显著性水平0.05，即这20个指标的数据分布都不符合正态分布，因此我们使用斯皮尔曼相关系数（Spearman's rank correlation coefficient）进行相关性分析。

斯皮尔曼相关系数基于排名而不是原始数据，适用于数据不符合正态分布时，并且不要求标量之间的关系是线性的，它的核心思想是将原始数据转换为排名，然后计算排名之间的相关性。其计算过程如下：

1.数据排名：将原始数据按照大小转换为排名数据。如果有重复值，则将这些值分配至其排名的平均值。

2.计算排名差异：计算任意两个变量之间的排名差异，这里我们可以只计算每个特征与洪水发生概率的排名差异。

3.计算相关系数：使用相应公式计算斯皮尔曼相关系数。

具体公式如下所示：

式中是斯皮尔曼相关系数，是第i对数据的排名差异，即两个变量对应排名的差值，n是数据对的数量。

在进行斯皮尔曼相关系数计算后，我们将结果可视化，绘制斯皮尔曼相关系数热力图，其中相关性分析的值矩阵存储在spearman.csv中，在分析各特征与洪水概率的相关性时，我们根据所得到的斯皮尔曼相关系数的特点将0.176作为阈值，得到表1所示结果,并将结果绘制成如图2所示的斯皮尔曼相关系数热力：

表 1：指标相关系数示意图

|  |  |  |  |
| --- | --- | --- | --- |
| **强相关指标** | **相关系数** | **弱相关指标** | **相关系数** |
| **基础设施恶化** | **0.1816** | **湿地损失** | **0.1756** |
| **地形排水** | **0.1807** | **农业实践** | **0.1756** |
| **季风强度** | **0.1805** | **流域** | **0.1749** |
| **大坝质量** | **0.1797** | **政策因素** | **0.1742** |
| **河流管理** | **0.1792** | **规划不足** | **0.1734** |
| **淤积** | **0.1791** | **城市化** | **0.1729** |
| **人口得分** | **0.1783** | **侵蚀** | **0.1718** |
| **气候变化**  **滑坡**  **森林砍伐**  **无效防灾** | **0.1775**  **0.1772**  **0.1772**  **0.1764** | **排水系统**  **海岸脆弱性** | **0.1705**  **0.1703** |

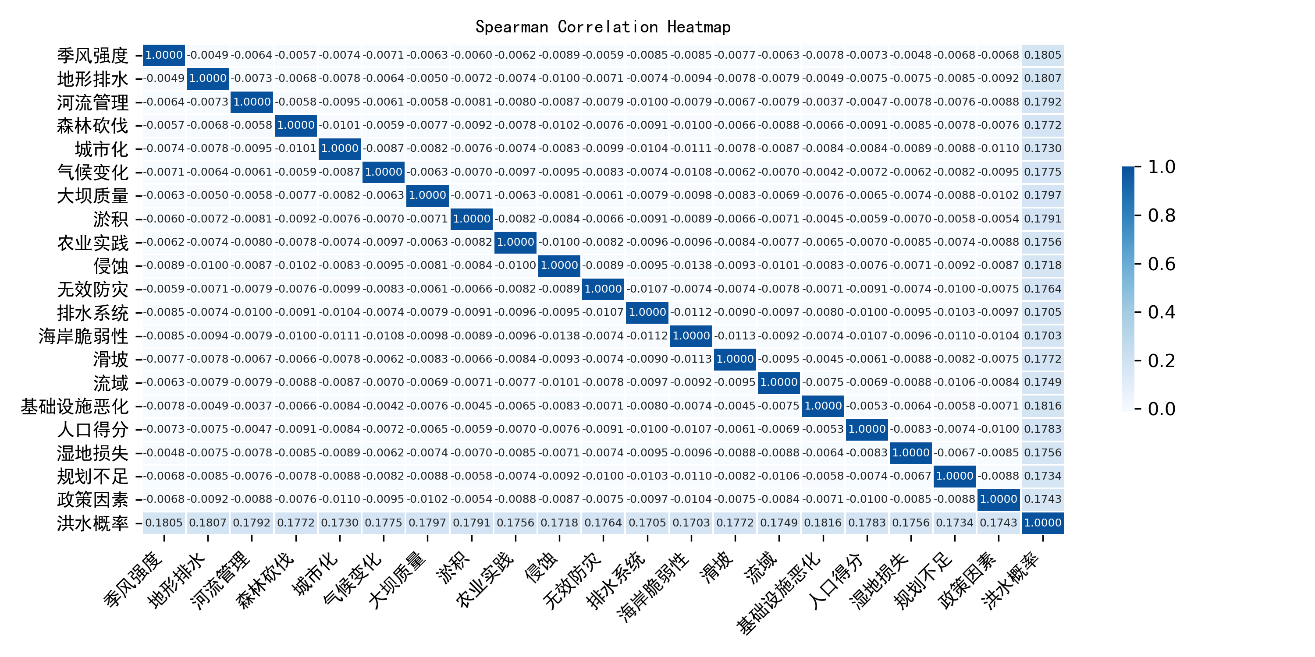


图 2：斯皮尔曼相关系数热力图

**5.1.3结果分析**

从表格1中可知，基础设施恶化、地形排水、季风强度等与洪水发生概率相关性较强，这是符合实际情况的，当基础设施过度老化时会直接影响洪湖水的排水和防护能力，导致城市内涝。地形排水直接决定了雨水的汇集和排出速度，在一个低洼地带，由于地势较低，雨水容易汇集而不易排出，导致洪水风险增加。而在山地地区，尽管雨水能够快速汇集，但如果降雨强度过大，仍然可能引发山洪。季风强度决定了降雨量和降雨强度。在季风季节，强季风带来的持续性暴雨会迅速增加河流水位，超过河道和排水系统的承载能力，导致洪水发生，尤其是当强季风与老化的基础设施和不良的地形排水条件结合时，洪水风险会显著增加。

而海岸线脆弱性、侵蚀、城市化等指标与洪水发生概率相关性较弱，海岸线脆弱性主要影响沿海地区，而洪水发生不仅局限于这些地区。内陆地区的洪水与海岸线的状况关系不大。海岸线脆弱性主要影响的是海洋洪水或风暴潮，而不是因降雨引起的河流洪水或内涝。因此，在总体洪水发生概率分析中，这一因素的影响相对较小。土壤侵蚀更多影响的是土壤质量和农业，而不是直接导致洪水发生。侵蚀会导致土地退化，但与降雨引起的洪水关联性较弱。城市化过程包括多种因素的综合影响，如人口密度增加、土地使用变化、基础设施建设等。城市化本身不是一个单一的因素，因此其与洪水发生的直接关联较弱。

我们针对强相关的各项指标，分析关联性强的原因并建议采取相应措施：

1. **基础设施恶化：** 衰退的基础设施可能导致排水系统失效或不足，增加了城市和地区的洪水风险。采取的措施包括定期检查和维护城市排水系统、桥梁和道路，确保其畅通有效。
2. **地形排水：** 地形复杂或排水不良的地区容易积水，进而增加洪水发生的可能性。采取的措施包括在城市和乡村规划中，考虑地形的自然排水特征，避免建设在低洼易积水区域。
3. **季风强度：** 季风带来的大量降水可能导致河流水位上升，引发洪水。采取的措施包括建立高效的气象监测网络，实时跟踪降水情况，并提前预警可能的洪水风险。
4. **大坝质量：** 大坝如果维护不善或设计不当，可能会导致溃坝，从而引发严重洪水。采取的措施包括对所有大坝进行定期检查，修复裂缝和结构问题，确保其安全性和稳定性。
5. **河流管理：** 不当的河流管理可能导致河道淤积、漫滩等问题，增加了洪水发生的频率和强度。采取的措施包括保护河流的生态系统，维护河道的自然生态平衡，减少人为干预对水流和水质的影响。
6. **淤积：** 河道淤积会减少河道容量，加剧降水期间的洪水风险。采取的措施包括对河道进行定期清淤和疏浚，确保水流畅通，减少积水和泛滥的风险。
7. **人口密度：** 人口密集地区容易引发土地开发过度，减少自然水文系统的缓冲能力，导致洪水的频率和影响增加。采取的措施包括通过城市和乡村规划，合理布局人口密度，避免过度集中，减少自然灾害对人口的影响。
8. **气候变化：** 气候变化可能导致降水模式改变，增加了极端天气事件（如暴雨）的频率和强度，从而增加了洪水的风险。采取的措施包括开展适应性措施，包括改进防洪工程、提升水资源管理能力等，以减少洪水灾害的影响。
9. **滑坡：** 滑坡会阻塞河道、增加泥石流风险，从而导致洪水的发生和加剧。采取的措施包括通过植被恢复和土地保护措施，增强地表的稳定性，减少滑坡的发生。
10. **森林砍伐：** 大规模森林砍伐会减少植被覆盖，降低土壤保持能力，增加了水土流失和河流泛滥的风险。采取的措施包括采取可持续的森林资源管理措施，确保森林覆盖率和生态系统功能，减少水土流失和洪水风险。
11. **无效防灾措施：** 实施不当或维护不及时的防灾措施可能失效，无法有效减少洪水灾害的影响。采取的措施包括确保防洪堤、防护墙等防灾设施的建设质量和运行效能，定期检查和维护设施状态。

通过上述措施，我们可以有效降低洪水发生的概率。

**5.2 问题二的模型建立与求解**

5.2.1数据标准化处理

首先，我们需要将cleaned\_data.csv中所有的特征数据进行标准化处理，在进行K均值聚类之前，我们需要先对所有特征进行标准化处理，这是因为K均值聚类算法依赖于欧氏距离来计算样本之间的相似性。不同特征的量纲和取值范围可能差异很大，直接使用原始数据进行聚类会导致数值较大的特征对距离计算的影响过大，掩盖其他特征的重要性。通过标准化处理，我们将所有特征的均值归一化为0，标准差归一化为1，从而确保每个特征在聚类过程中的影响力相同，提高聚类结果的准确性和可靠性。

5.2.2**基于无监督聚类算法的风险聚类**

问题二分两小问，第一小问要求根据洪水发生的概率将洪水事件聚类成高、中、低风险三种类别，并分析这三种类别对应的洪水事件的指标特征。第二小问要求选取合适的指标，计算不同指标的权重，建立发生洪水不同风险的预警评价模型，最后进行模型的灵敏度分析。

针对第一小问，我们应用无监督聚类算法将洪水事件聚类为高、中、低三个风险级别，然后通过可视化分析不同类别的数据的特征分布。

无监督聚类算法用于将数据集中的样本划分为个簇，使得同一簇内的样本相似度较高，而不同簇之间的样本相似度较低。

算法基于“距离”来衡量样本之间的相似度，通常使用欧氏距离。算法的核心思想是迭代地将样本分配到K个簇中，使得每个样本到其所属簇的中心（即簇内所有样本的均值）的距离最小，算法的目标是使得所有样本到其簇中心的距离平方和最小，即最小化目标函数。该算法简单高效、可解释性强、适合大规模数据集。

算法的实现步骤图如图3：

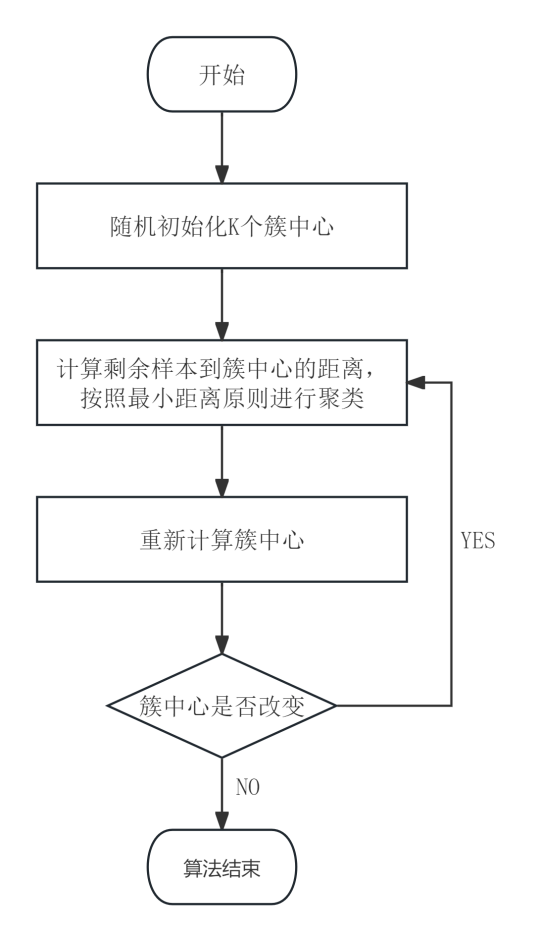


图 3：算法实现步骤图

其目标函数如下：

其中表示簇类中心的数目，为样本坐标，为簇类中心坐标。

我们在**Algorithm 1**中总结了算法的实现步骤。

|  |
| --- |
| Algorithm 1 K-means |
| Require: Dataset D, number of clusters k |
| 1: Initialize: Randomly select k data points as initial cluster centroids. |
| 2: Repeat |
| 3: For each data point x in D do |
| 4: Compute distances from x to each cluster centroid. |
| 5: Assign x to the nearest cluster centroid. |
| 6: End for |
| 7: For each cluster centroid do |
| 8: Compute the mean of all data points assigned to the cluster, update cluster centroid. |
| 9: End for |
| 10: Until Convergence |
| 11: Return the final cluster centroids and data point assignments. |

使用K-means聚类算法根据洪水发生概率将洪水事件分类为高、中、低风险类别的结果如下散点图4所示：

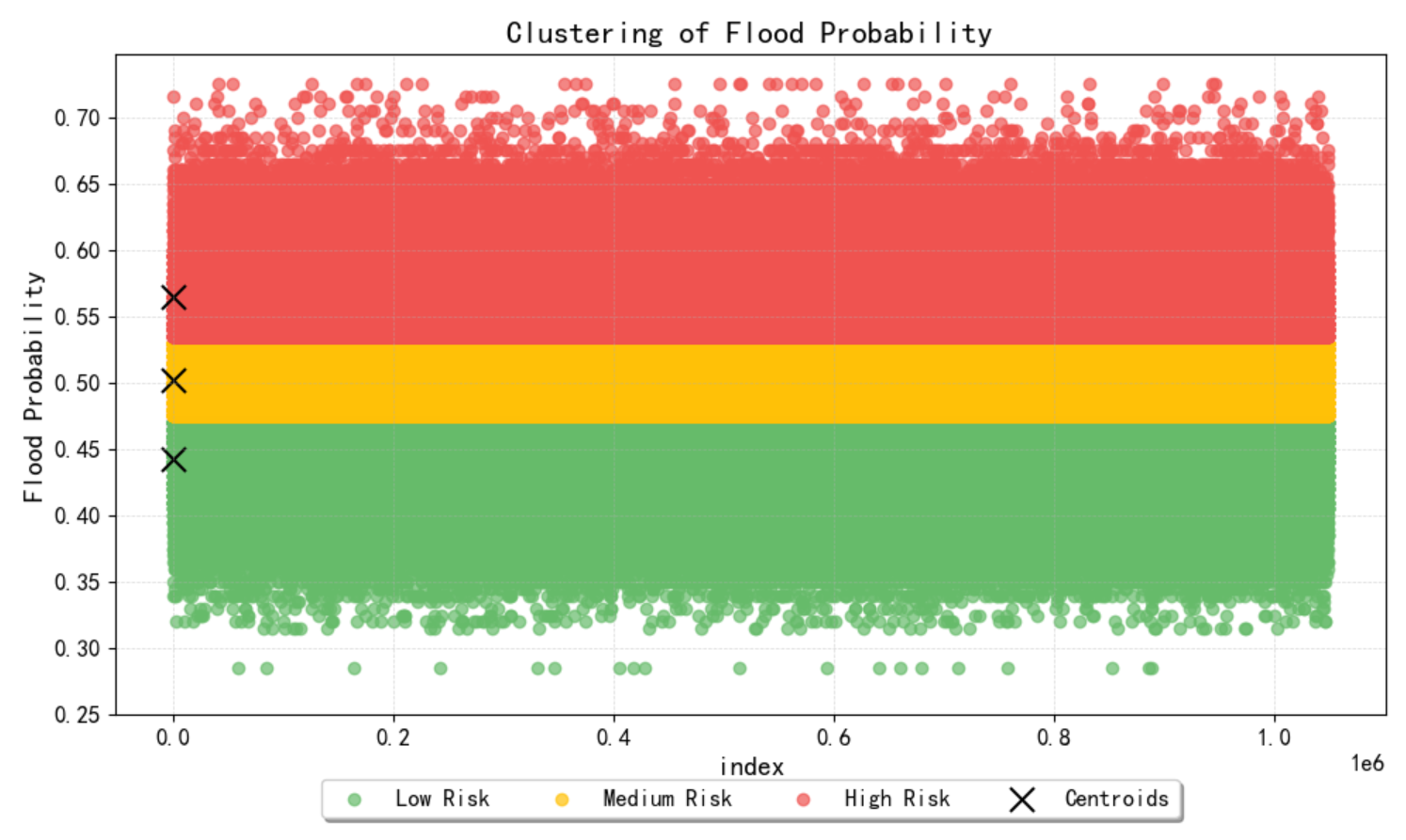


图 4:洪水发生概率聚类图

其中横坐标轴代表索引，纵坐标反应了洪水发生的概率，不同颜色的区域代表了不同的洪水风险类别，绿色区域代表低风险类别，黄色区域代表中风险类别，红色区域代表高风险类别，代表聚类后每一个类别的质心。可以发现高风险类别的洪水事件洪水发生的概率分布在0.53以上，中风险类别的洪水事件洪水发生的概率分布在0.47-0.53之间，低风险类别的洪水事件洪水发生的概率分布在0.47以下，即如式（3）所示：

这个散点图直观、清晰地展示出了洪水风险的分布规律，通过它我们可以快速了解不同洪水事件的洪水风险等级，为制定防洪减灾措施提供科学依据。

接下来进行分析具有高、中、低风险的洪水事件的指标特征，我们首先绘制了每个类别所有样本的所有指标特征雷达图，可以清晰地观察到每个风险等级的所有指标的影响程度。

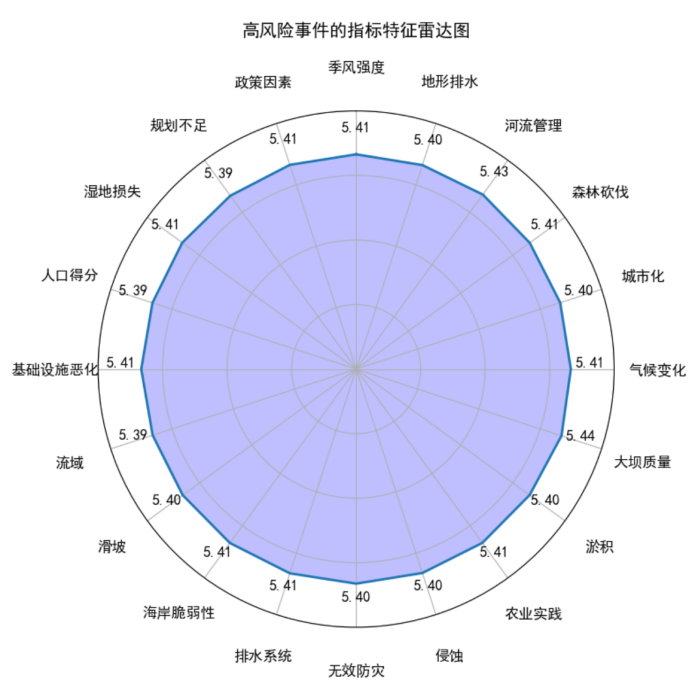


图 5:高风险事件的指标特征雷达图

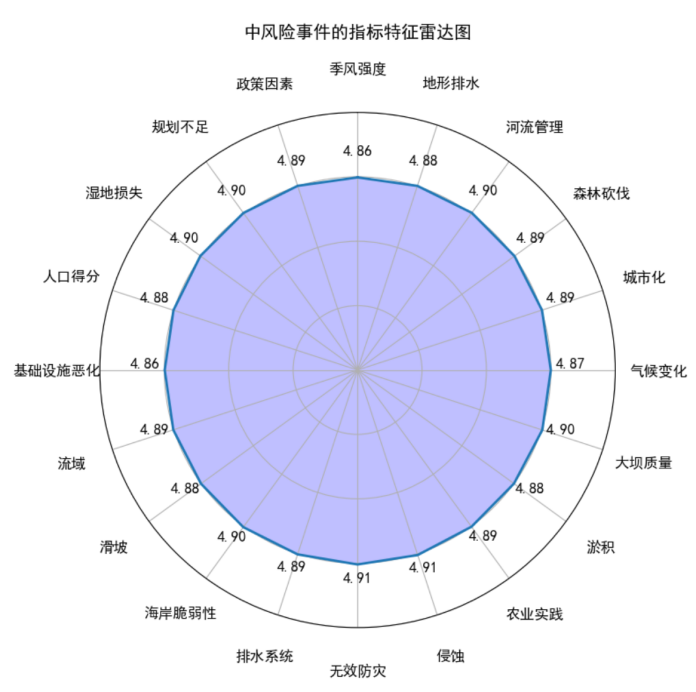


图 6：中风险事件的指标特征雷达图

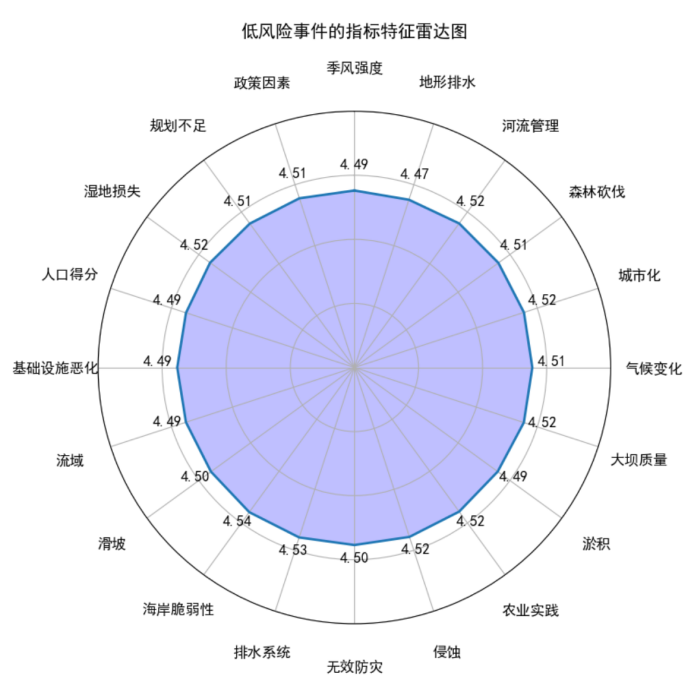


图 7：低风险事件的指标特征雷达图

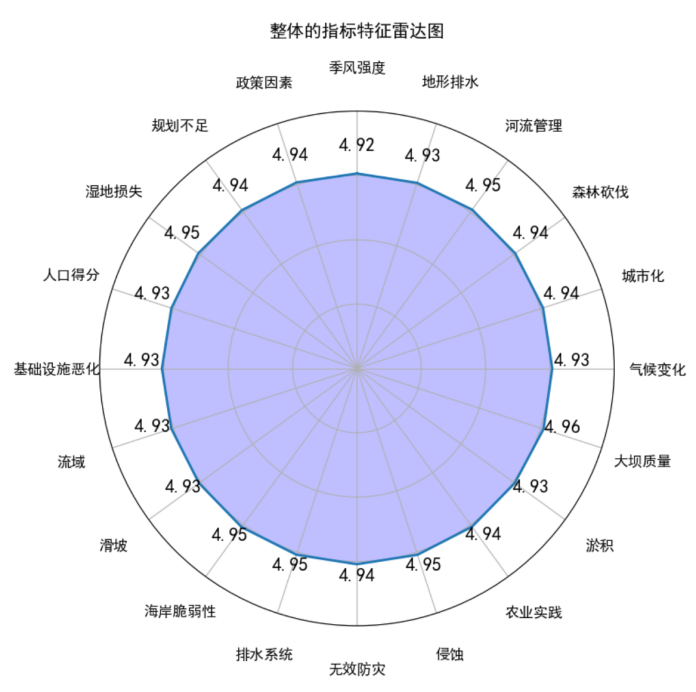


图 8：整体数据指标特征雷达图

特征雷达图可以直观地反应出每个特征的平均特征值，通过对比整体的指标特征雷达图和三个风险等级的指标特征雷达图，每一个风险等级的每个指标的平均特征值与整体的每个指标的平均特征值差距越大，说明该指标对于洪水的发生概率的影响程度就越大。

接着我们绘制了每个类别所有样本的所有指标的频率分布直方图，可以直观地观察到各个指标特征在不同的风险等级中的分布情况。

下面图中天蓝色柱状图代表各个指标特征值的频率分布，红色曲线表示其相应的概率密度函数。

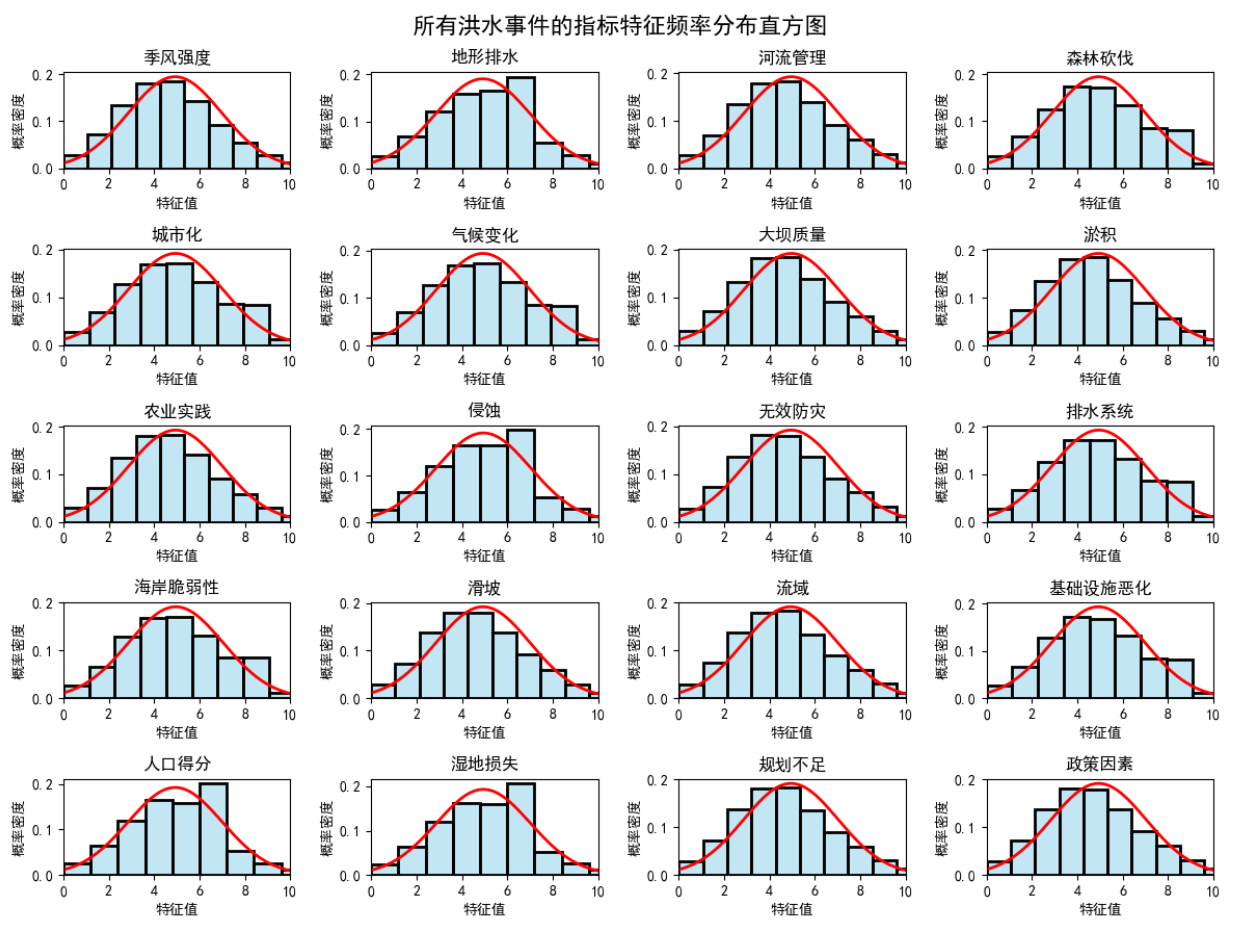


图 9：所有洪水事件指标特征频率分布直方图

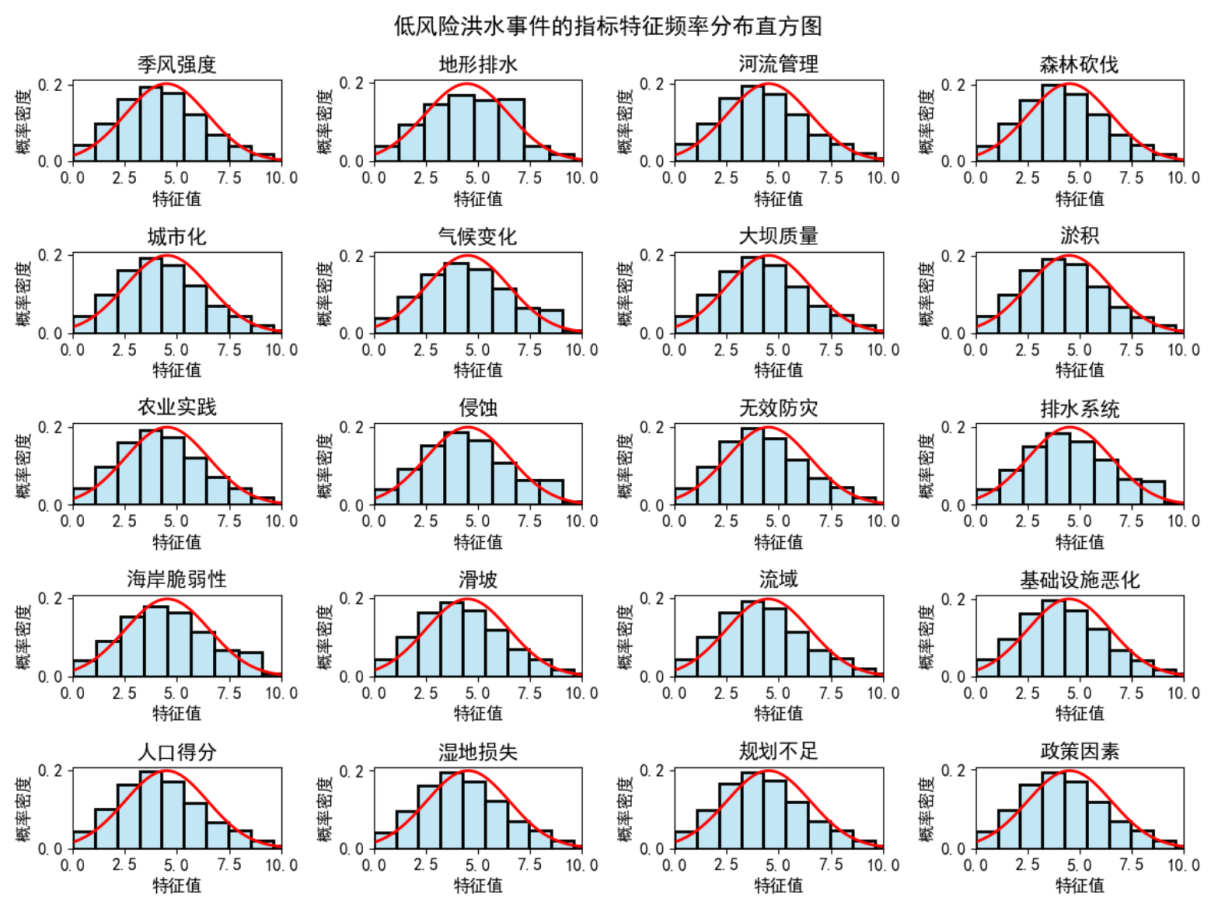


图 10：低风险事件指标特征频率分布直方图

处于低风险等级的洪水事件的特征分布通常较为接近整体样本的中位数，这些特征的波动较小，表明其对洪水发生的概率影响较弱。



图 11：中风险事件指标特征频率分布直方图

处于中风险等级的洪水事件的特征分布较为均衡，但仍有一些特征指标集中于某些区间，这些特征有助于识别出介于高风险等级和低风险等级之间的洪水事件。

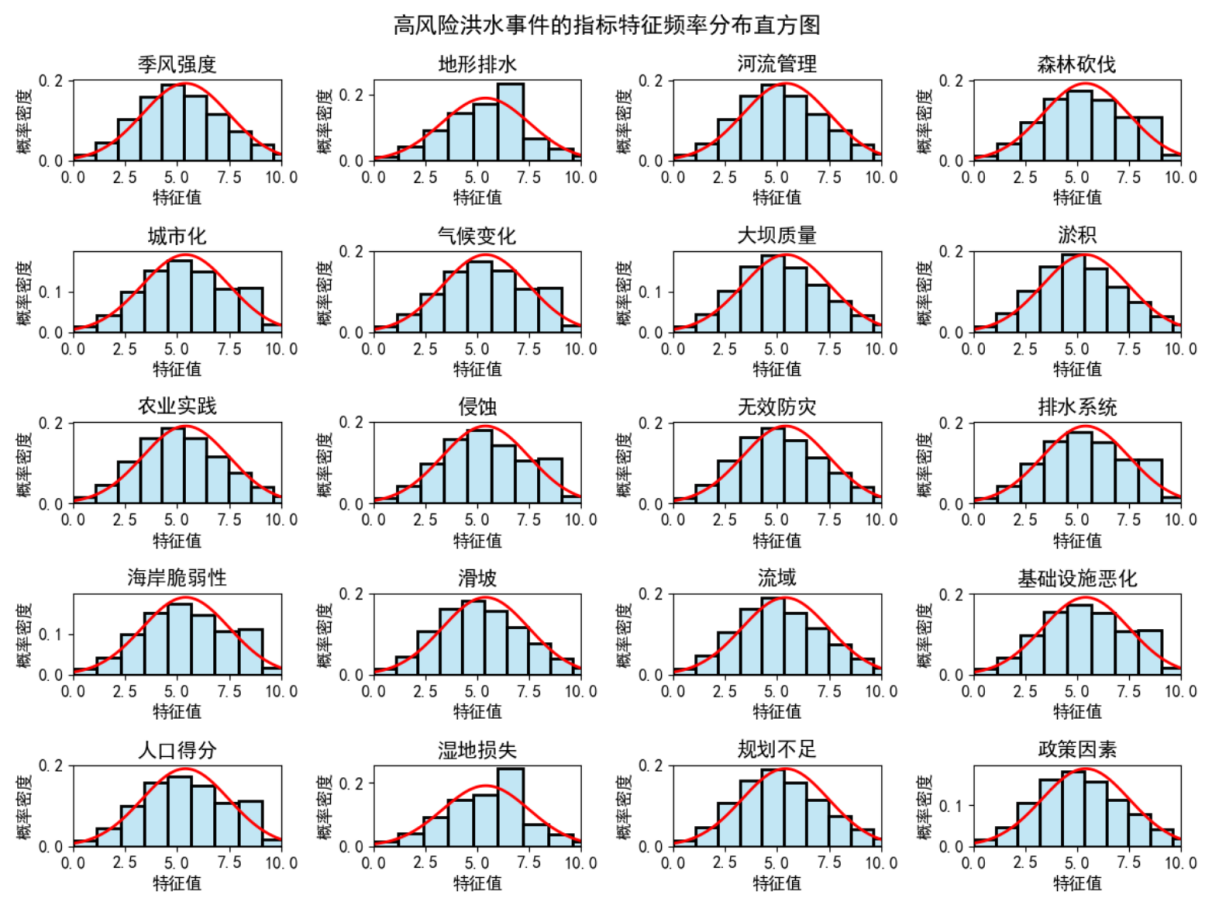


图 12：高风险事件指标特征频率分布直方图

对于处于高风险等级的洪水事件关注其指标特征分布是否有比较明显的峰值和极值，在高风险等级中表现出峰值偏移较多的指标特征的值可能在高风险洪水事件中占据主导作用。

通过观察指标特征雷达图和指标特征频率分布直方图，将三种类别的特征雷达图和指标频率分布直方图与整体的进行对比，可以发现海岸脆弱性、排水系统、城市化、农业实践等指标的变化幅度小，说明这几个指标特征对于洪水事件的发生影响较小，同时也可以发现大坝质量、地形排水、基础设施恶化、人口得分等指标的变化幅度大、说明这几个指标对于洪水事件的发生影响较大。

通过分析不同指标的影响力，这对于制定针对性的风险管理策略、优化资源配置以及提高整体风险防控能力具有重要意义。

5.2.3基于梯度提升决策树（GBDT）的洪水预警评价模型

为选取合适的指标并计算不同指标的权重，建立发生洪水不同风险的预警评价模型，我们选择使用梯度提升决策树**[2]**（GBDT）模型进行拟合。GBDT模型是一种集成学习方法，通过组合多个弱学习器（通常是决策树）来提升模型的预测性能,在回归和分类任务中表现出色,能够自动处理非线性关系和特征之间的复杂交互作用，并且不需要对数据进行过多的预处理。通过训练GBDT模型**[3]**，我们可以得到各个特征的重要性评分，这些评分反映了每个特征再模型预测发生概率中的贡献度，从而可以作为个指标的权重。此外GBDT可以帮助我们更清晰地理解不同指标对洪水风险地影响机制。

为构建模型并验证其性能，我们使用“cvparttition”方法按7：3比例分割清洗后的数据，将数据集划分为训练集和测试集，使用70%的数据来训练决策树模型，从而拟合出各个特征的重要性权重，并保留30%的数据作为测试集来评估模型的泛化能力和预测精度，防止模型过拟合，确保其在未知数据集上的表现。

GBDT通过逐步构建决策树，每棵新树都是在之前所有树的基础上，拟合当前的残差来减少误差。其核心思想是利用梯度下降法来优化损失函数。具体实施过程如图13所示

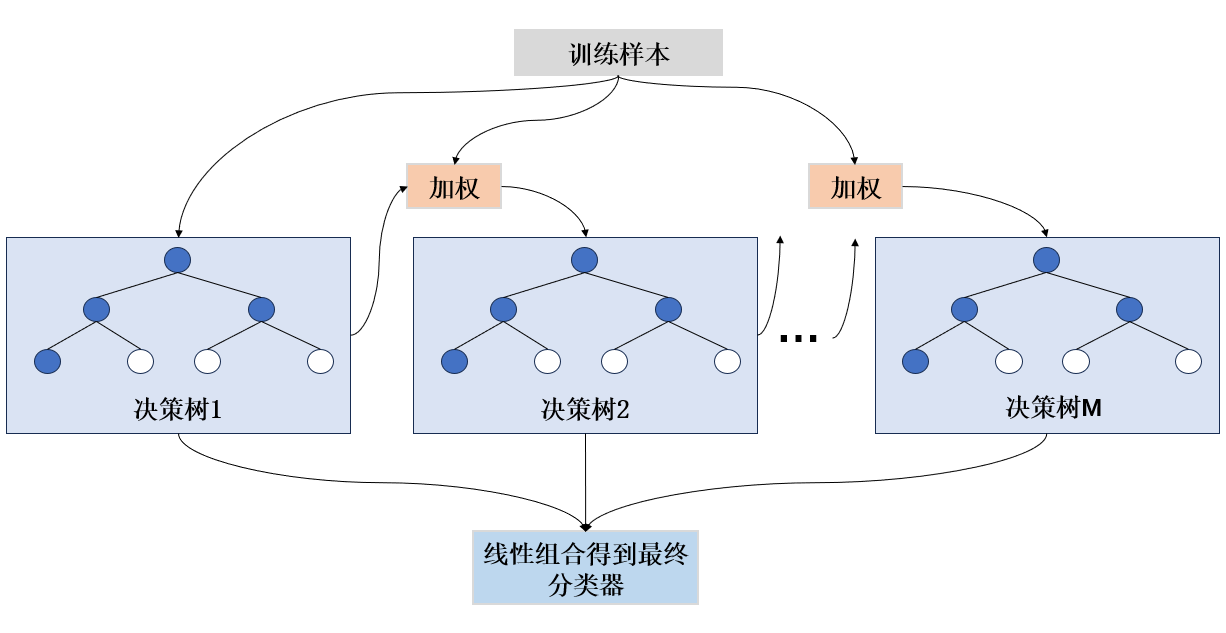


图 13:梯度提升决策树模型框架图

首先我们对模型进行初始化，常用初始预测值为目标变量的均值：

对于均方误差损失函数初始化为：

对于每一棵树,计算当前模型的残差（负梯度），并计算均方误差损失函数：

之后用一棵新的决策树来拟合这些残差并通过式（）更新模型，学习率：

经过次迭代后，最终模型为：

在上述模型的构建过程中，有 棵树，每棵树 中有 个节点。特征 在节点 的分裂带来的误差减少量记作 ，训练完成后，特征 的总重要性得分 可以表示为:

为了进行归一化，最终的特征重要性得分可以表示为:

式中 是所有特征重要性得分的总和。

我们将经过标准化处理的训练数据输入到梯度提升决策树（GBDT）模型，并通过交叉验证的方法，确定最优的参数配置，根据交叉验证的结果将基础决策树数量设置为150学习率设置为0.2，每颗决策树的最大深度限制为5，得到特征重要分数如下图所示

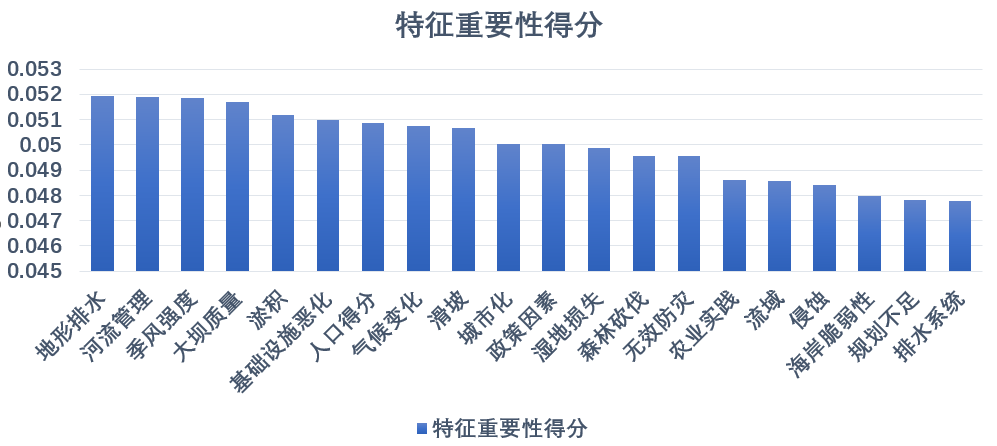


图 14：基于梯度提升决策树的特征重要性得分

根据每个特征的重要性和样本的特征值，我们可以计算风险评分,风险评分反映了每个样本的综合风险。根据前面式（3）所定义的风险类别分类标准，我们对测试集样本进行风险评估,以检验模型的性能，通过计算评价指标均方误差（MSE）及决定系数来验证模型的泛化能力。

MSE 是预测值与实际值之间误差的平方和的平均值。值越小表示预测值与实际值之间的差距越小。

衡量模型解释的方差比例。其值范围在0到1之间，值越接近1，表示模型解释了越多的方差。

MSE 和 RMSE的数据如表2：

表 2：方误差（MSE）及决定系数

|  |  |
| --- | --- |
|  | 0.0004498 |
|  | 0.826423 |

MSE为0.0004498，这表明模型在整体上具有较高的预测准确性。这个值尤其在洪水预测这样的任务中非常有意义，因为预测的准确性直接关系到防灾减灾的效果。

的值为0.826423，这表明模型能够解释82.6%的数据变化，模型性能较好。

同时我们使用聚类好的的数据将数据根据阈值划分为高、中、低三类风险，使用我们训练好的预测评价模型对测试集进行测试，测试结果如图15所示，测试精度达到了78%，并得到如图16所示的混淆矩阵热力图。

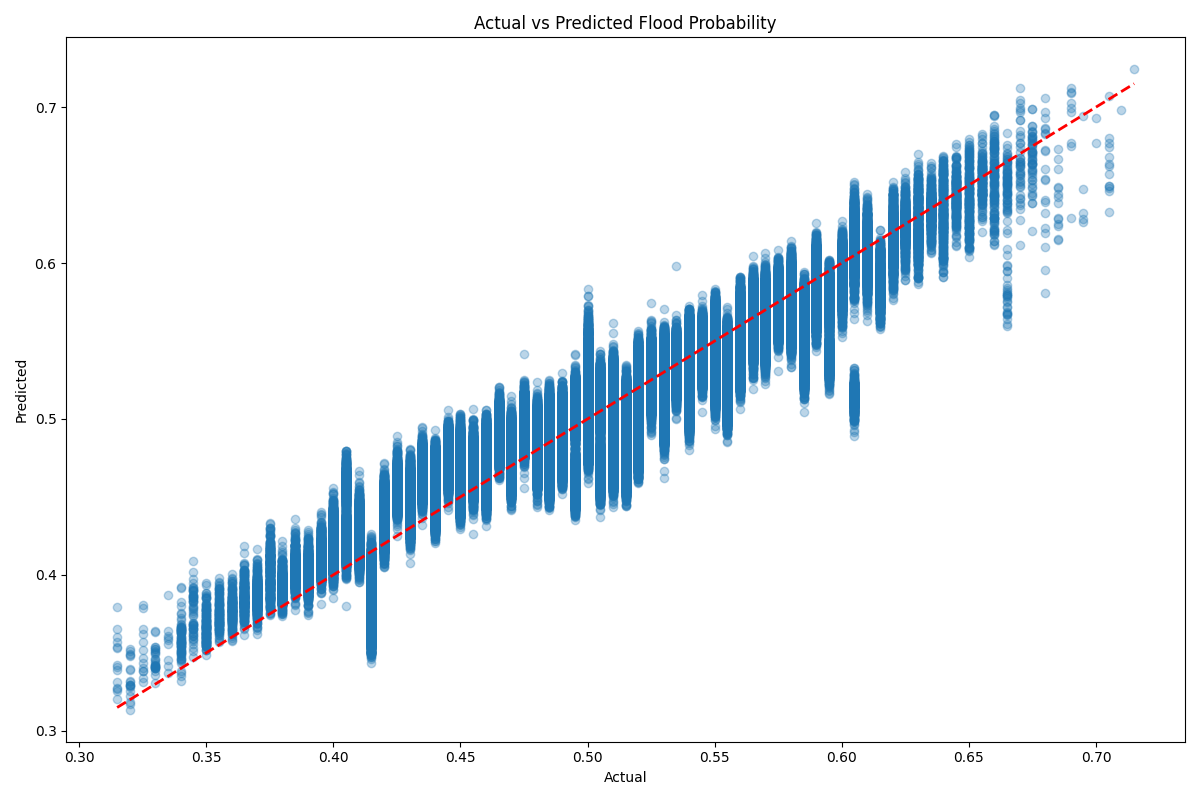


图 15:基于梯度提升决策树的洪水预警评价模型测试示意图

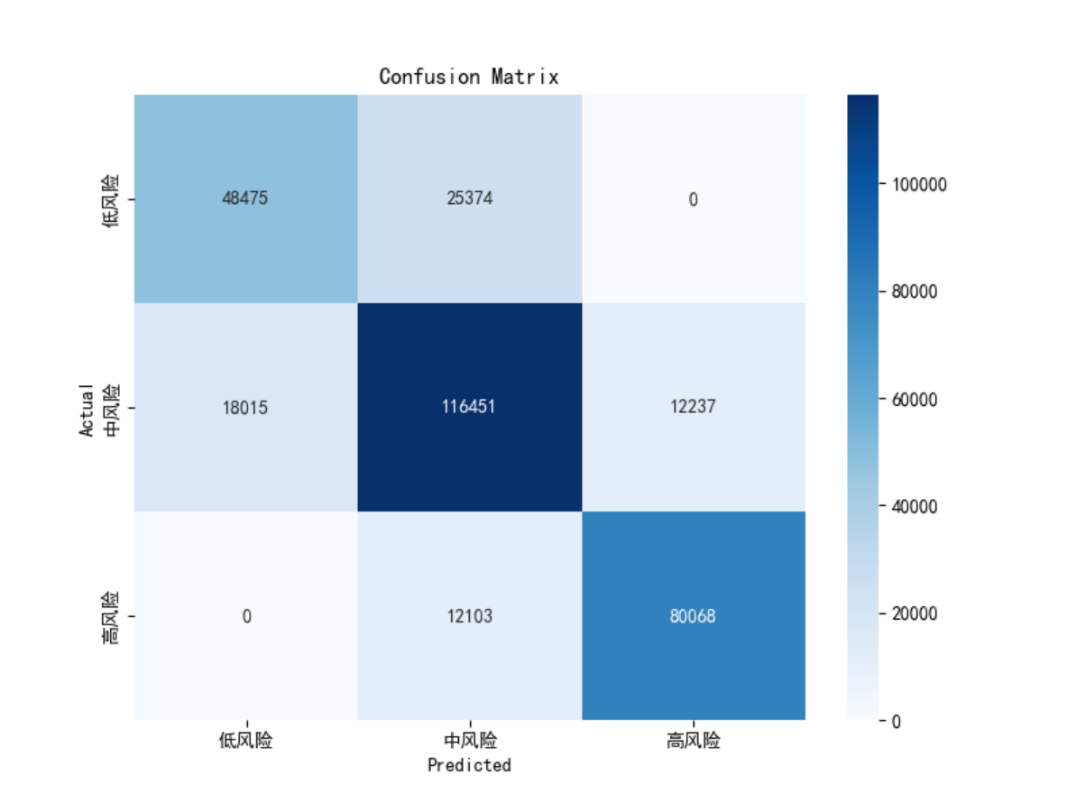


图 16：基于梯度提升决策树的洪水预警评价模型测试混淆矩阵

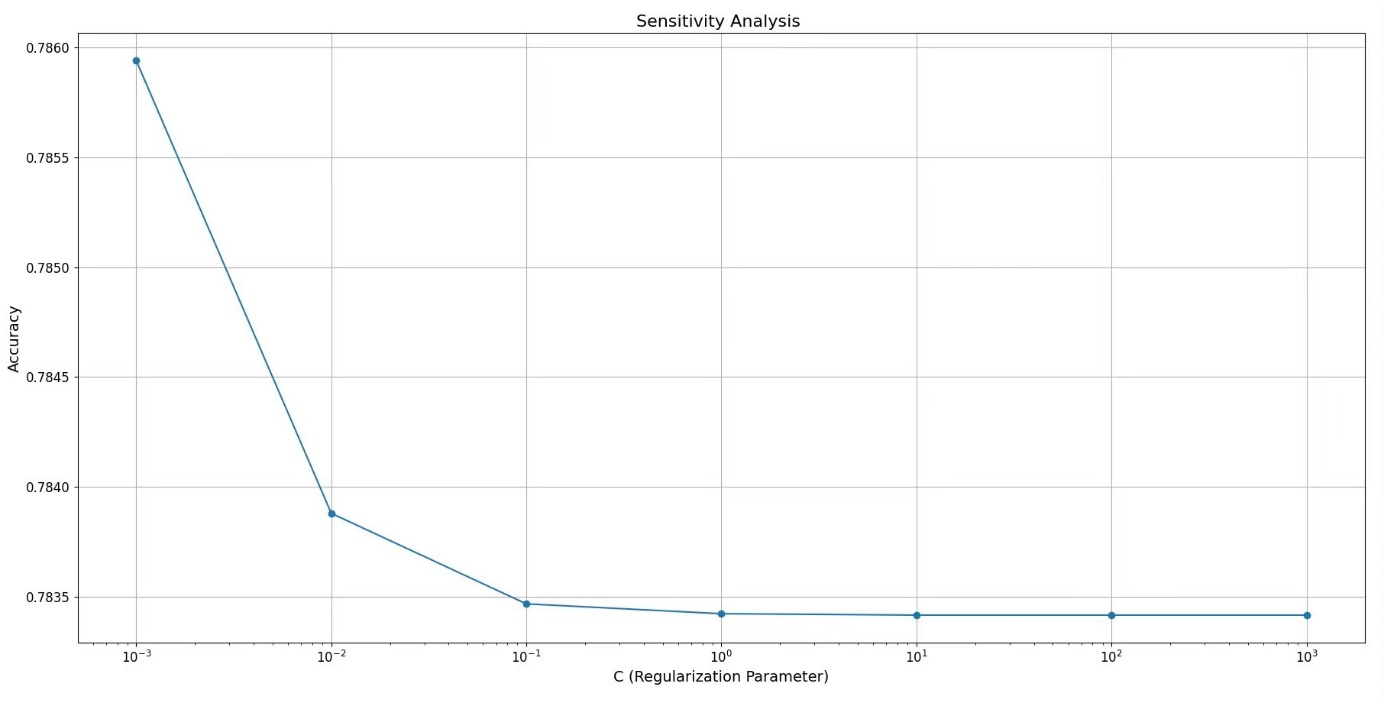


图 16:基于梯度提升决策树的灵敏度分析

我们对梯度提升决策模型的正则化参数进行灵敏度分析，发现，当参数变化时，模型的预测精度变化不大，说明我们的模型有较好的稳定性。

根据混淆矩阵热力图可知，模型对中、高风险类别的预测准确率较高，这说明当洪水发生概率较大时，模型能够很好的做出预警，使相关部门提前做好预防措施。同时模型的总体预测准确率较高，说明该预测评价模型具有良好的性能。

为检验模型的优越性，我们将模型与经典的多元logits回归模型的结果进行比较，逻辑回归通过对线性回归模型的输出应用一个逻辑函数（sigmoid函数），将线性输出转换为概率输出，对于多分类问题，对于任一输入特征向量**,**逻辑回归将函数假设为：

其中 是参数， 是输入特征向量， 是类别数量。

我们采用最大似然估计进行逻辑回归模型的训练，即通过最大化似然函数来求解模型参数。对于多分类问题，模型的损失函数通常是对数损失函数，也称为交叉摘损失函数，表示为：

其中， 是样本 属于类别 的指示函数 (one-hot 编码)。

在训练过程中，通过计算梯度并使用梯度下降法更新参数 :

其中， 是学习率， 是损失函数关于参数的梯度。

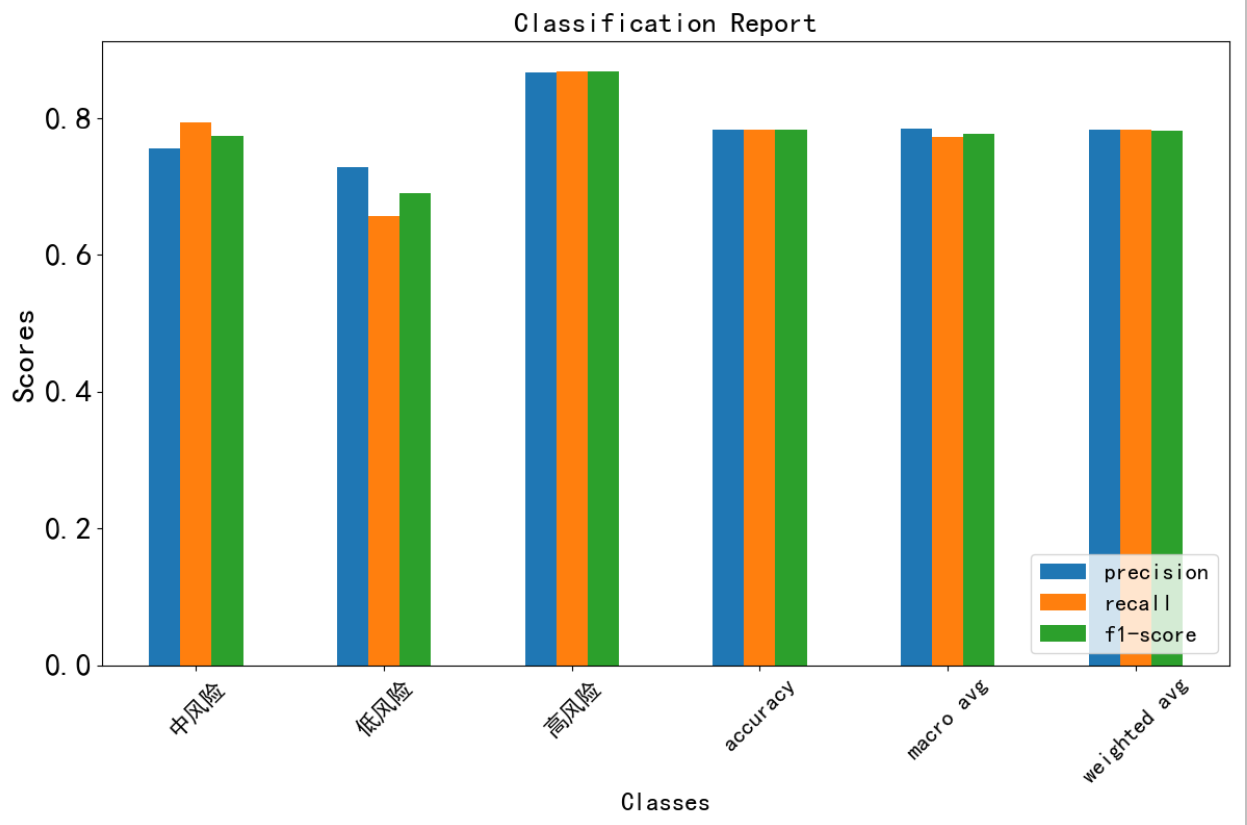


图 17：多元逻辑回归模型测试示意图

在训练完后，我们得到如图16所示的测试结果，逻辑回归模型在测试数据集上高、中、低三类风险的预测精度的分别为79.3%、76%、73%，平均预测精度达到76.1%，小于我们建立的基于梯度提升决策树（GBDT）的洪水预警评价模型。比较两个模型的测试平均精度可知，我们的模型更加性能更加优越。

**5.3 问题三的模型建立与求解**

为了预测洪水发生的概率，我们构建一个基于统计学习的预测模型，假设洪水发生的概率 与一系列环境和人为因素具有相关性。我们选取的主要影响因素包括季风强度、地形排水条件、河流管理水平、森林砍伐情况、城市化速度、气候变化情况、大坝质量、淤积情况、农业实践、地面侵蚀、防灾措施效果、排水系统效能、海岸脆弱性、滑坡发生频率、流域条件、基础设施老化、人口密度、湿地损失状况、规划充分性和政策支持力度。

我们假设模型为一个多变量逻辑回归，其中洪水发生的对数几率是这些因素线性组合的函数：

其中， 表示上述各个指标的量化值， 是模型参数，需要通过数据拟合得到。我们进一步假设这些变量之间相互独立，即每个因素对洪水的影响不受其他因素的影响。此外，我们将使用交叉验证方法来优化模型参数，确保模型在未见数据上也能保持稳定的预测性能。

**5.3.1 全部指标**

模型建立：

我们团队针对第三问洪水预测问题，开发了一个名为 **FloodRiskPredictor** 的预测模型。本模型基于随机森林算法，使用了超过一百万条由季风强度、地形排水情况、河流管理状况等多达20个不同指标组成的洪水数据。我们的目标是构建一个能够准确预测洪水发生概率的统计模型，从而帮助政府和相关机构在洪水季节前采取更有效的预防措施。

在本问中，我们首先进行了数据的详细清洗和预处理，确保模型输入的数据质量。此外，特征工程的应用帮助我们识别并选择了最具预测力的指标，以提高模型的预测精度和泛化能力。通过这种方法，**FloodRiskPredictor** 不仅可以预测洪水的发生概率，还能为洪水管理提供科学的决策支持，显著增强防灾减灾的效率和效果。

其流程图如下

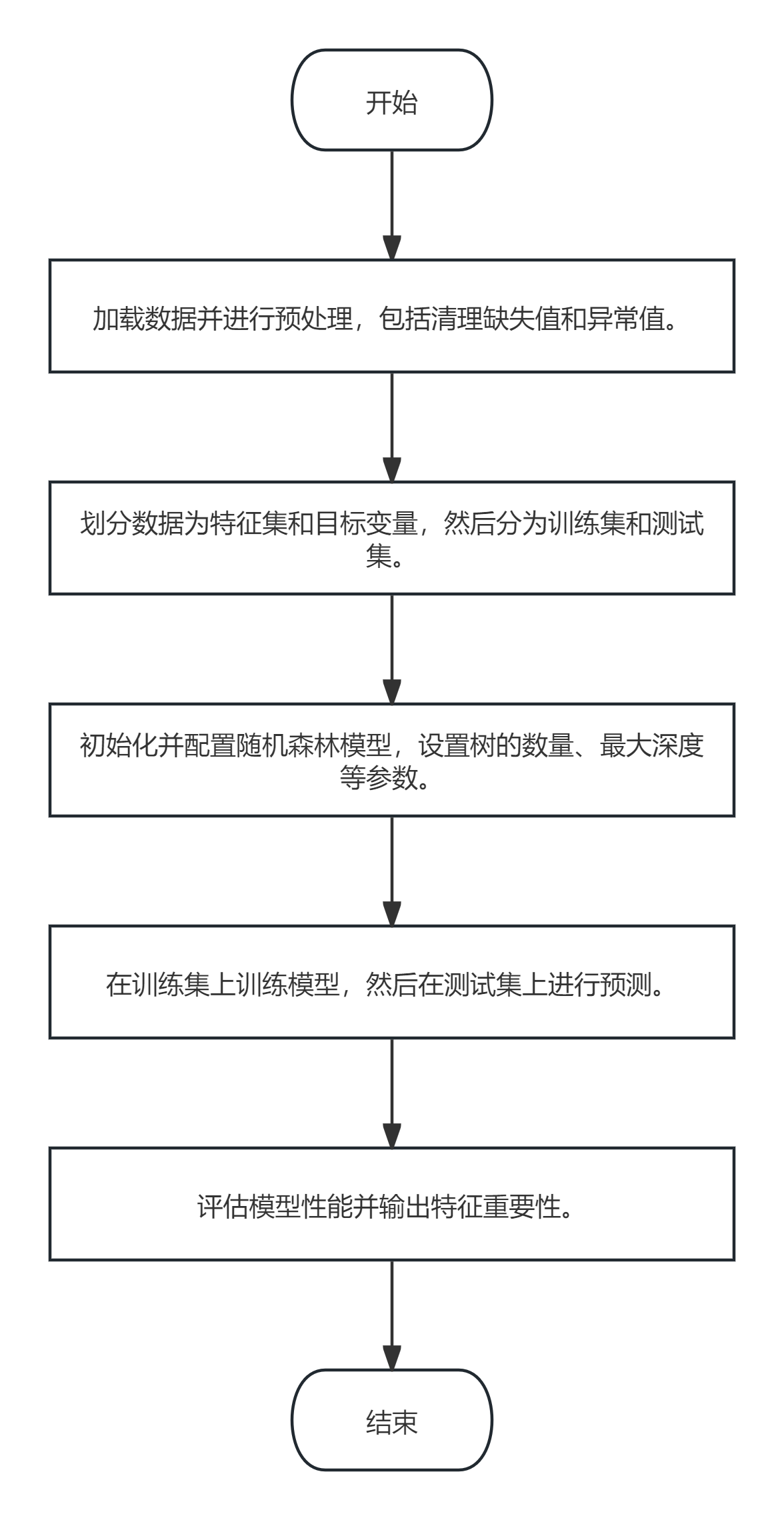


图 18： FloodRiskPredictor预测模型流程图

对预处理后的数据，我们选择随机森林算法进行模型训练和预测，因为随机森林是一种强大的机器学习方法，它能有效处理高维数据并提供良好的准确性和鲁棒性。随机森林通过构建多个决策树，并让它们独立地进行决策，最后通过投票机制来决定最终的预测结果。这种方法的优势在于它可以自然地评估特征的重要性，并且对于未见过的数据具有很好的泛化能力,模型结构如图18所示，

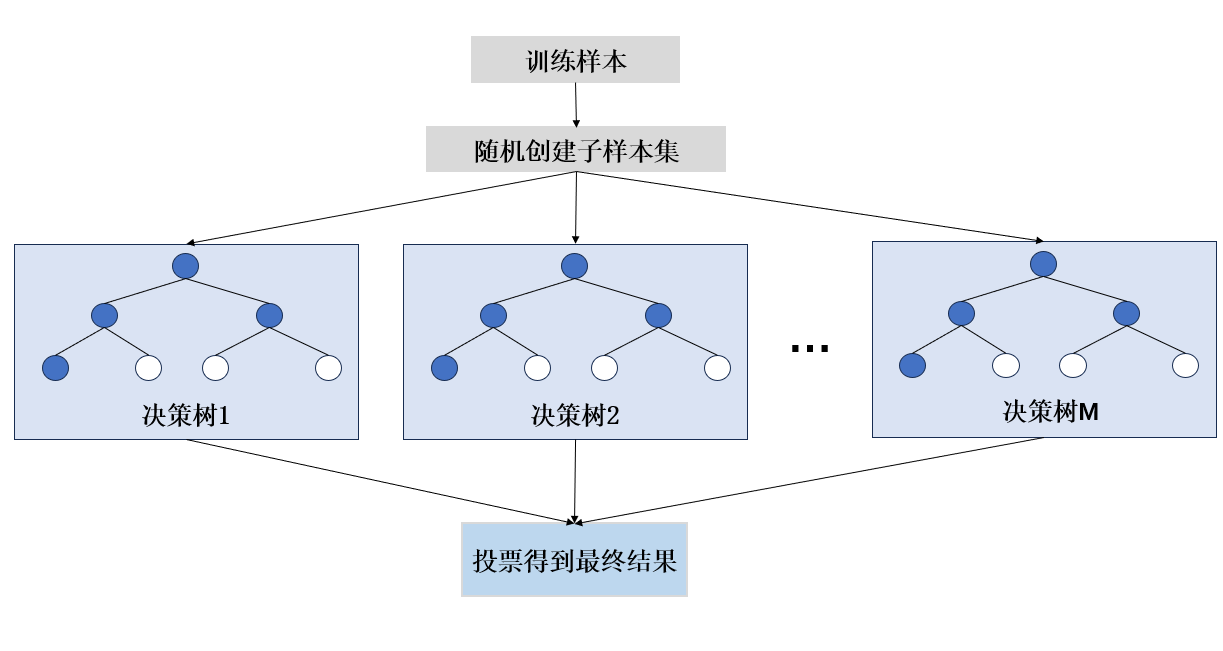


图 19：随机森林算法示意图

我们对随机森林模型的主要参数进行了调整，包括树的数量、树的最大深度、以及划分时考虑的特征数量等，以达到最佳的训练效果。

表 3：随机森林模型主要参数设置

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **n\_estimators** | **max\_depth** | **max\_features** | **min\_samples\_split** | **min\_samples\_leaf** |
| **500** | **15** | **‘sqrt’** | **10** | **4** |

具体参数设置详述在附录。

随机森林预测洪水发生概率:

假设我们有一个包含 个样本和 个指标的数据集，每个样本 可以表示为 ，其中 是样本 在指标 上的值。

随机森林模型构建公式，包含参数细节

模型构建与预测：

其中：

是树的总数，

- 是参数化设置下第 棵树对样本 的预测， 包括树的最大深度、划分时考虑的特征数量等参数。

决策树分割：

每棵树的构建基于以下参数化决策：

最大深度 ,

划分时考虑的特征数量 ,

最小样本分割 ,

最小样本叶节点 。

树分割的决策点选择基于最大化信息增益：

其中， 可能是基尼不纯度或熵，根据 和 控制节点分割和生长。

在完成对533,551个训练样本的模型训练后，随后利用训练完成的模型对133,388个独立测试样本进行了性能验证。

模型验证：

使用随机森林模型**[1]**预测洪水发生概率，首先使用 MSE和RMSE两个指标对模型进行了评估。模型在测试集上达到了较高的AUC值，显示出良好的分类能力。

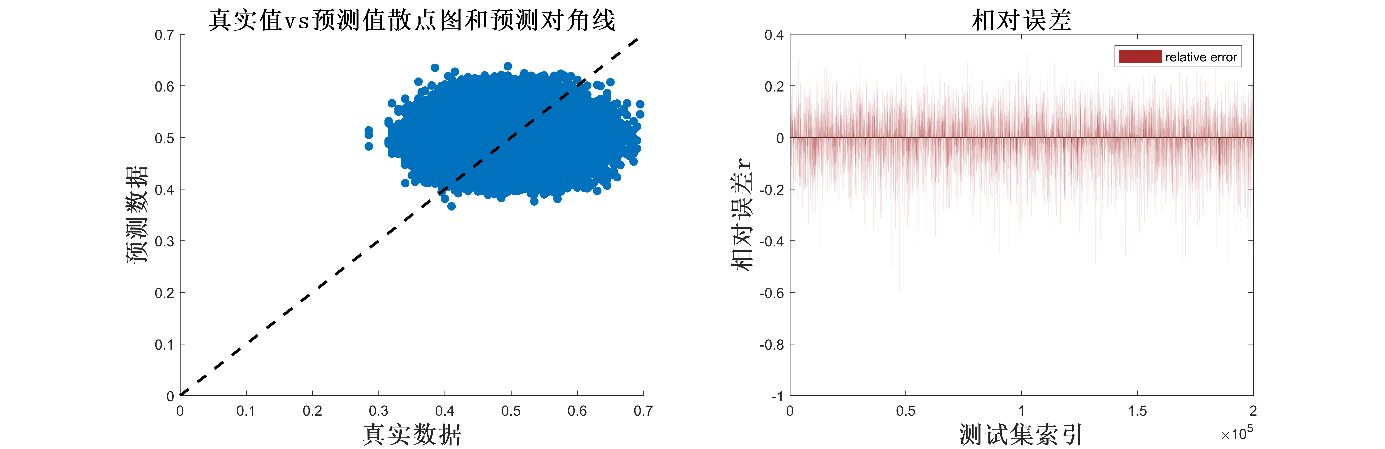
MSE 和 RMSE的数据如下。

表 4：MSE 和 RMSE结果图

|  |  |
| --- | --- |
|  | 0.003168 |
|  | 0.056285 |

MSE为0.003168，这表明模型在整体上具有较高的预测准确性。RMSE 是 MSE 的平方根，它提供了误差的估计值与实际观测值的单位一致的度量，值0.056285显示了模型的预测误差在绝对尺度上较小，这进一步证明了模型的有效性。

我们又对预测值与真实值之间做了回归图和相对误差可视化，可以使我们更加直观地的观察模型的准确性。



图表 18

图 20：真实值与预测值之间的回归与相对误差图

从回归图中可以看出散点很接近斜率为1的这条线，说明模型的预测值和实际值成正相关，并且从右边的误差图中可以看出相对误差较小，说明本模型预测较为准确。

选取的是滑坡和流域两个指标，我们对于其增加一定的随机数上下增减，来验证模型的稳定性。

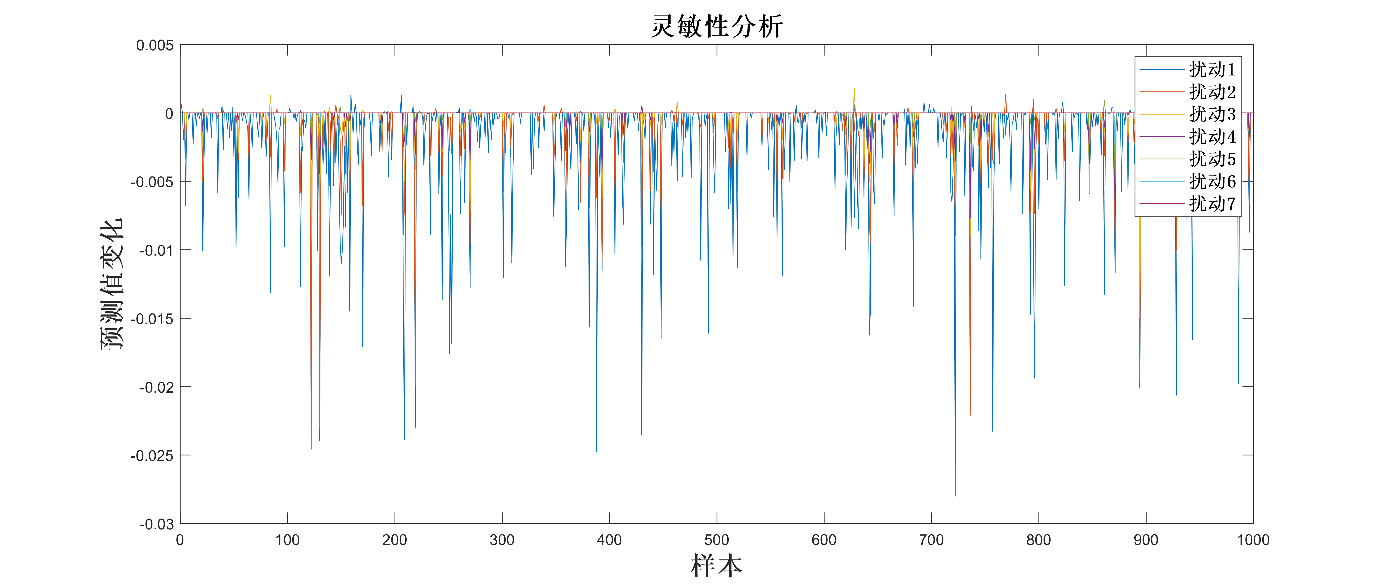


图 21：灵敏性分析示意图

从上述灵敏性分析中我们可以看出，模型对于输入扰动的响应显示出较小的预测值变化，通常在0附近。这表明我们的洪水发生概率预测模型具有较高的稳定性和可靠性。对于扰动的不同输入参数，模型输出的波动非常有限，这证明了模型在面对输入数据微小变化时能够保持其预测准确性和一致性。此结果为进一步应用模型提供了信心，确保在实际操作中即使在数据波动或不确定性条件下，也能够得到可靠的预测结果。

5.3.2 五个指标

本问从前面的随机森林**[4]**中我们只需要对20个指标重要性程度排序，取前五个即可。

重要性定义：特征重要性是指特征在预测模型中的有效性和贡献程度，即这个特征对模型的预测性能的影响有多大。在决策树和随机森林等模型中，特征重要性通常通过计算特征分裂时导致的纯度增益（如信息增益）来评估。其他模型如逻辑回归则可以通过系数的大小来判断特征的重要性。

重要性高的特征在模型中扮演着关键角色，对模型的预测结果有显著影响。它们的变化可以显著改变模型的输出。

特征的重要性受到所使用的模型类型的影响，所以我们采用随机森林和梯度提升决策树模型两种模型来分析其重要性。

特征重要性的计算公式如下

反映了使用特征 进行分裂后的不纯度变化量。

通过随机森林和上述公式，我们得到了二十种指标的重要程度，如下图所示。

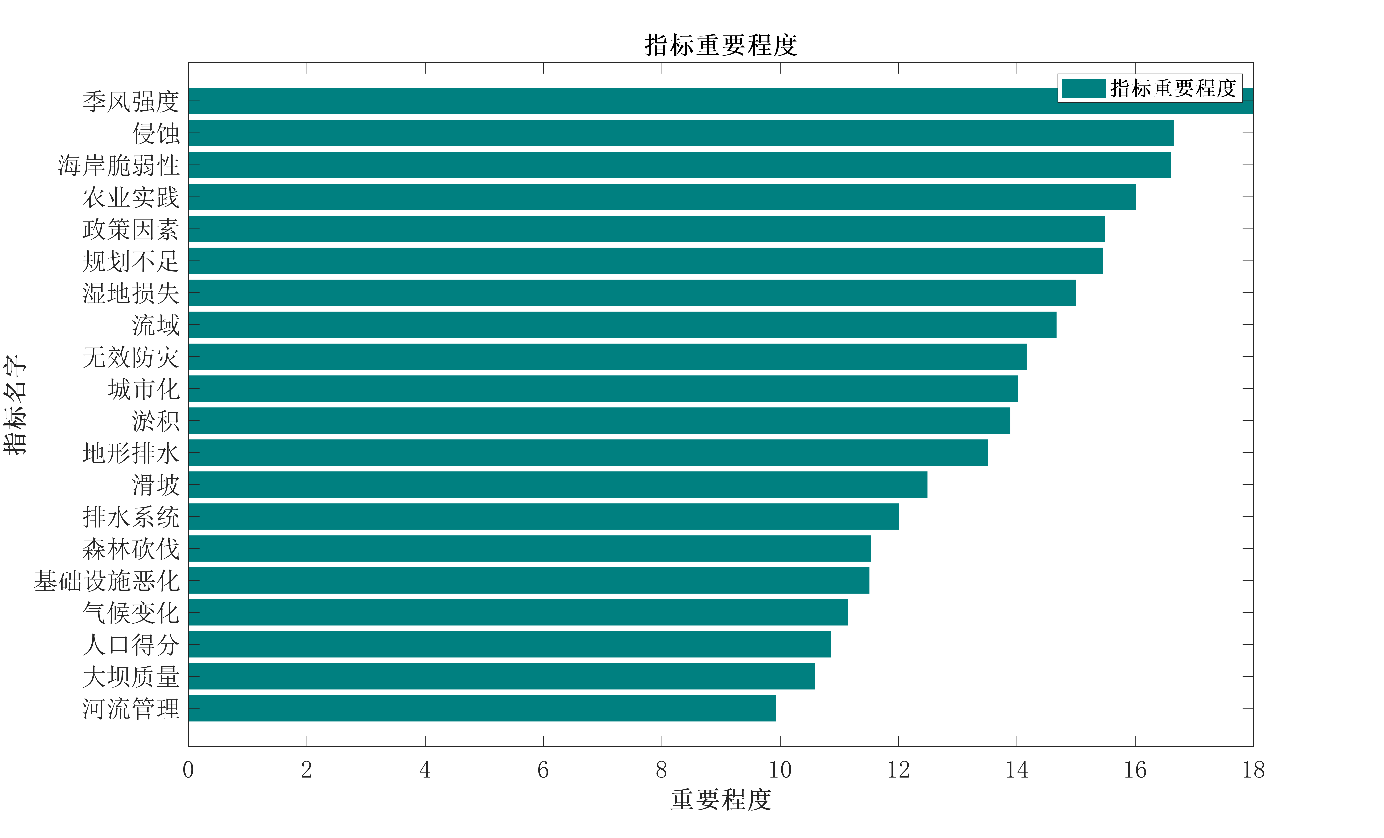


图 22：指标重要程度直方图

由上图得到的指标重要程度，我们选择季风强度、侵蚀、海岸脆弱性、农业实践和政策因素五个为我们选取的五个指标建立FloodRiskPredictor模型。

表 5：五个指标的MSE和RMSE结果

|  |  |
| --- | --- |
|  | **0.0020** |
|  | **0.044718** |

从这个表格中我们同样可以看出在模型只选取五个重要指标之后的MSE和RMSE还是很小，表明模型仍具有很准确预测。

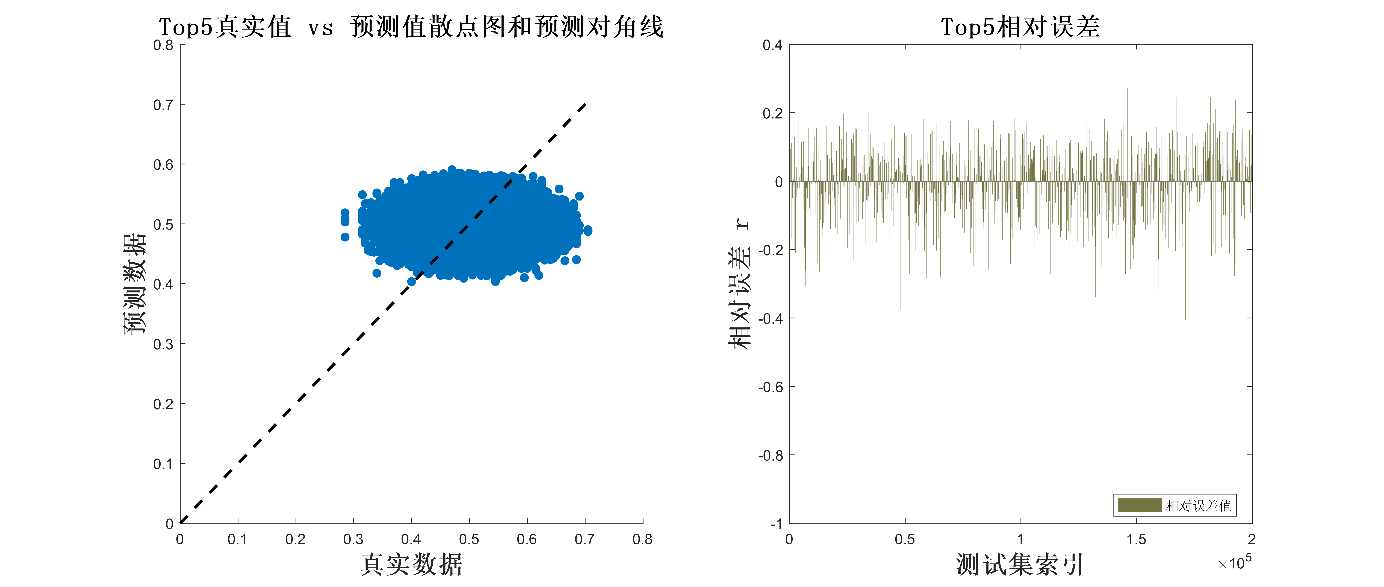


图 23：重要指标的回归与相对误差图

从图中我们看出选取五个重要指标后的回归图和相对误差仍符合预期，模型在五个指标的情况下依旧是预测准确的。

我们采用交叉验证的方法来检验模型的稳定性，我们采用10折，循环每一折进行训练和测试然后计算所有折的平均MSE再通过柱状图23表示。

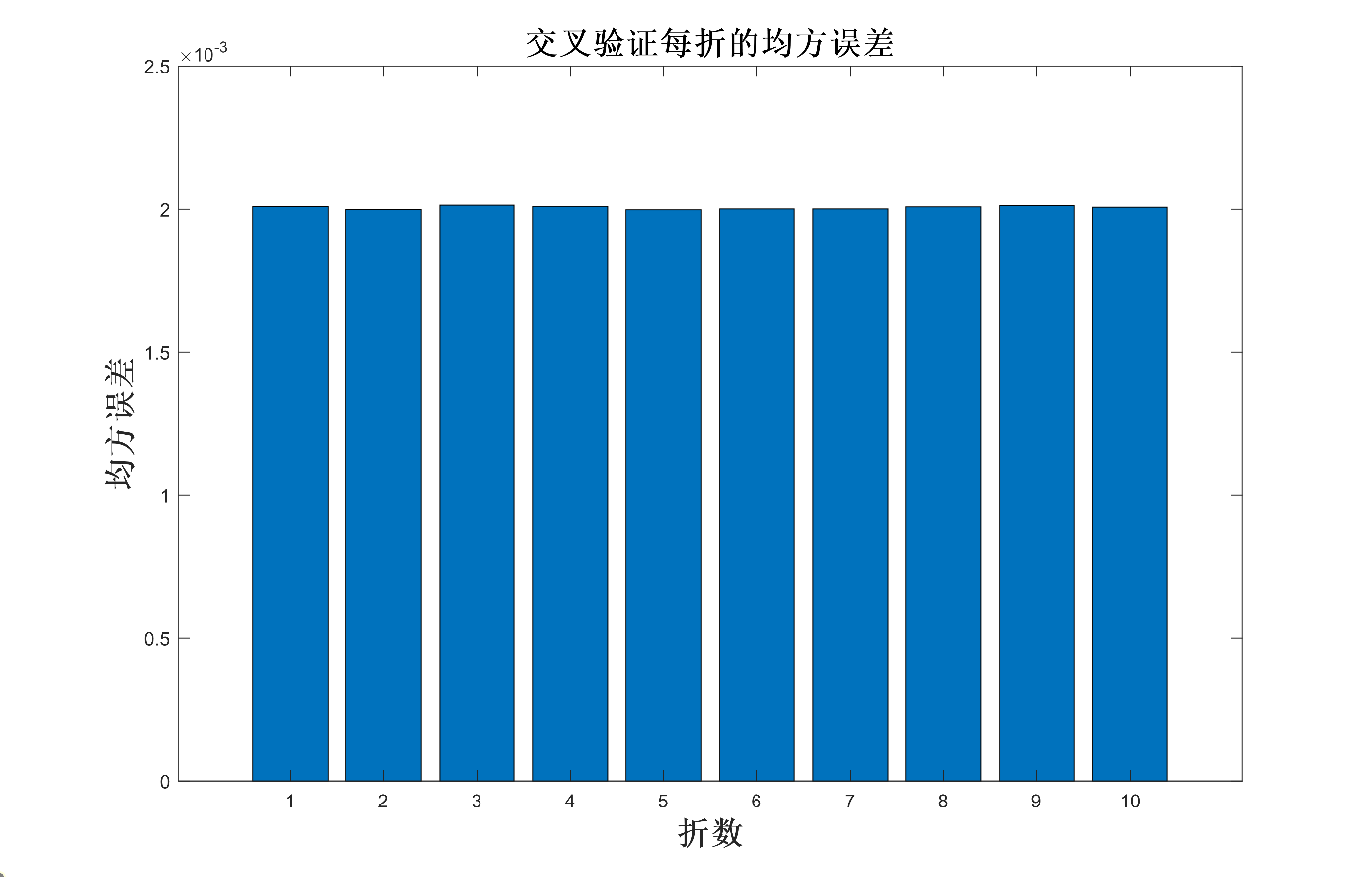


图 24：交叉验证每折的均方误差图

由图中我们可以看出在这图表显示的均方误差（MSE）在各个折(fold)上的值相对均匀，没有出现异常高或低的折，这表明模型在不同的数据子集上具有一致的性能。误差的一致性表明模型对于不同的数据分割具有良好的稳定性。

**5.4 问题四的模型建立与求解**

问题重述：

问题要求基于问题 3 中建立的洪水发生概率预测模型，对 test.csv 文件中所有事件的洪水发生概率进行预测，并将这些预测结果填入 submit.csv 文件中。完成预测后，需要绘制这些预测结果745,305项的直方图和折线图，以分析这些数据的分布情况，特别是是否服从正态分布。

我们将采取以下步骤：

数据准备与验证：首先，确认 test.csv 文件已经被正确读取，并对其进行必要的预处理，以确保数据格式与训练模型时所使用的一致。同时，检查 submit.csv 文件，确保其结构适合填入预测结果。

模型应用：使用在问题 3 中开发的洪水发生概率预测模型对测试数据进行预测。这一步骤涉及加载已训练的模型，并使用它对新数据集中的所有实例进行洪水概率预测。

结果填充与保存：将预测得到的洪水概率填充到 submit.csv 的相应列中，确保每个 ID 对应的预测概率正确无误地录入。

数据可视化：绘制预测结果的直方图和折线图，用于分析洪水发生概率的分布特征。这包括评估数据的集中趋势、离散程度以及是否服从正态分布。

结果分析与报告：基于可视化和统计分析的结果，编写详细的分析报告，解释模型的预测行为及其潜在的统计特性。这将有助于进一步理解模型的性能，并为实际应用提供依据。

通过这些系统化的步骤，我们不仅能确保模型预测的准确性和可靠性，还还能深入分析洪水发生概率的统计特征，从而为后续的决策提供科学的依据。

|  |
| --- |
| Flood Probability Prediction and Analysis Algorithm |
| Require: Test dataset test.csv, Prediction model Model, Submission template submit.csv |
| 1: Initialize:Read the test data from test.csv.Ensure the Model is correctly loaded and ready for prediction. |
| 2. Use the Model to predict flood probabilities for each event in the test dataset. |
| 3. Load the submission file template submit.csv and verify its format. |
| 4. Insert the prediction results into the appropriate column in submit.csv (assumed to be the 'Flood Probability' column). |
| 5. Check that the prediction data has been correctly filled in submit.csv. |
| 6. Save the updated submit.csv. |
| Visualization and Analysis:  7. Plot histograms and line graphs of the prediction results.  8. Analyze the distribution of the prediction results to check if they approximate a normal distribution.  9. Ensure the analytical results meet expectations, such as the normality of distribution, etc.  10. Make further decisions or adjustments based on the analysis results.  11. The final updated submit.csv file and an analytical report. |

根据上述步骤我们得到预测值部分如下所示，左侧是没有排序过的，右侧是排序过的洪水概率。

**| ID (Unsorted) | Flood Probability (Unsorted) | | ID (Sorted) | Flood Probability (Sorted) |**

**|---------------|------------------------------|-|---------------|----------------------------|**

**| 1117957 | 0.545006 | | 1598008 | 0.643804 |**

**| 1117958 | 0.465801 | | 1747210 | 0.642458 |**

**| 1117959 | 0.476207 | | 1620852 | 0.638162 |**

**| 1117960 | 0.475460 | | 1550526 | 0.638060 |**

**| 1117961 | 0.481554 | | 1627551 | 0.633093 |**

**| 1117962 | 0.504508 | | 1653142 | 0.632731 |**

**| 1117963 | 0.552804 | | ... | ...**

**| 1117964 | 0.508404 | | 1160311 | 0.384813 |**

**| 1117965 | 0.491374 | | 1146909 | 0.383639 |**

**| 1117966 | 0.512914 | | 1564227 | 0.382605 |**

**| ... | ... | | 1484416 | 0.382365 |**

**| 1863261 | 0.486589 | | 1411045 | 0.381773 |**

**| 1863260 | 0.532341 | | 1234077 | 0.378113 |**

详细数据在附录的submit.csv中。

根据本问我们绘制了直方图和直线图，还绘制了密度曲线。

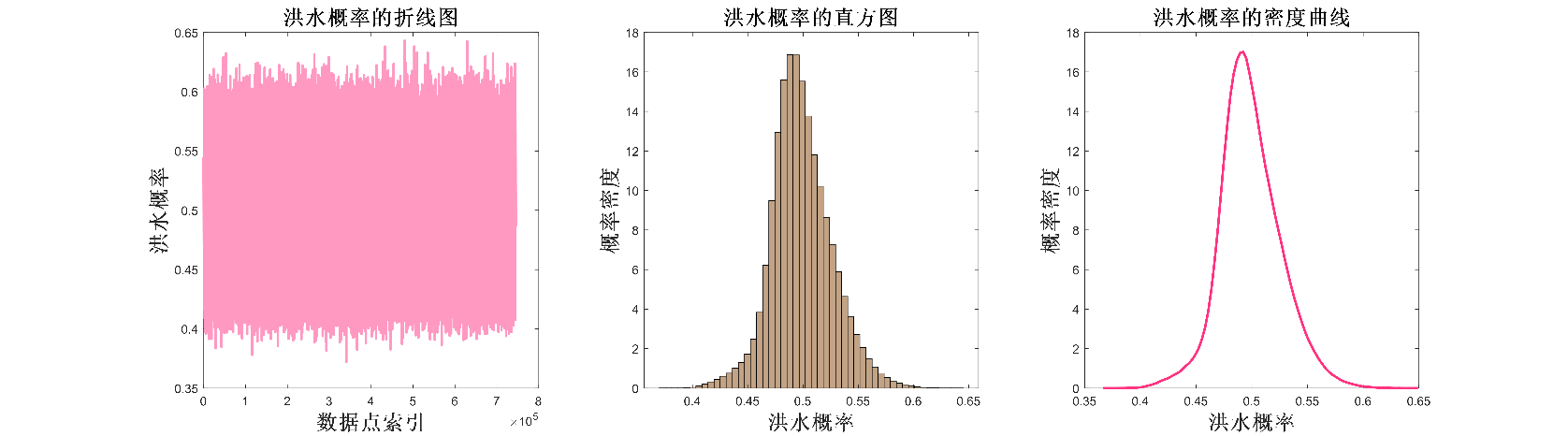


图 25：洪水概率分布图

洪水概率的直方图显示概率密度最高的区域集中在0.45附近，洪水概率的分布存在明显的波动，并非均匀分布，大部分数据点接近于一条直线，两端的数据存在偏离直线的现象，因此洪水概率的分布并非完全的正态分布，存在一定的偏差。

我们根据结果数据绘制了QQ图和概率累积分布函数。QQ图对于任何样本大小都是一个有用的视觉工具，可以直观地显示数据是否偏离正态分布。对于大数据集，QQ图可以帮助识别数据中的异常值或偏离正态分布的趋势。

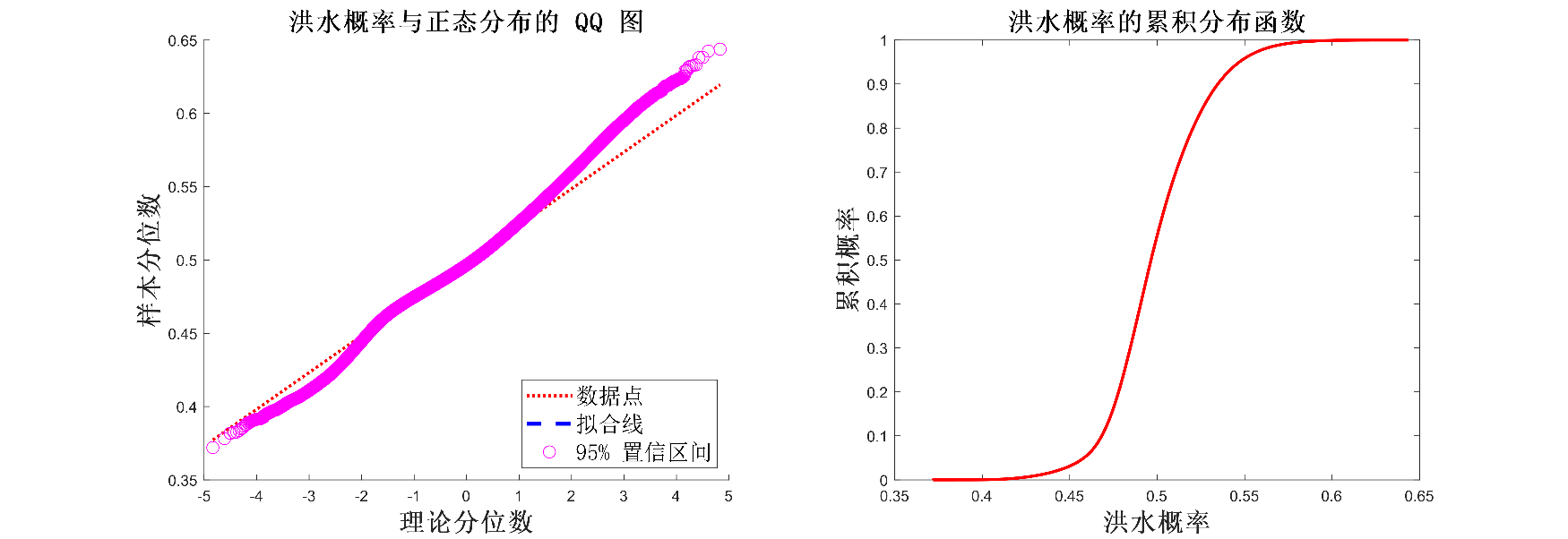


图 26：洪水概率与正态分布的QQ图

洪水概率与正态分布的QQ图直观地比较了洪水概率的样本数据与正态分布的理论分布，大部分样本数据点沿45度线分布，表明洪水概率的样本数据大致符合正态分布，但有部分数据点存在偏离，说明样本数据与理论分布存在细微的差异。

洪水概率的累计分布函数直观地反应了洪水概率随其数值变化的累计频率，这张图进一步反应了洪水概率的样本数据在整体上符合正态分布，但存在细微的差异。

本次数据量较大有74万项，为了更加严格的论证其是否符合正态分布，我们选取进行了Kolmogorov-Smirnov**、**Shapiro-Wilk和偏度和峰度检验来进行验证其是否符合正态分布。

Shapiro-Wilk 检验通过比较数据样本的排列次序与正态分布的期望次序来检验数据的正态性。它计算样本值与期望正态分布值之间的差异，并基于这些差异来确定样本是否来自正态分布。

K-S 检验通过比较经验累积分布函数（ECDF）与理论累积分布函数（CDF）之间的最大差异来评估数据是否符合特定分布。通常用于检验数据是否来自特定分布（例如正态分布）。

偏度（Skewness）和峰度（Kurtosis）检验是用于检验数据分布是否符合正态分布的两种统计方法。偏度检验主要用于检测数据分布的对称性，而峰度检验用于检测数据分布的峰态，即数据集中程度。

以下是 Shapiro-Wilk 检验和 Kolmogorov-Smirnov (K-S) 检验的公式和计算步骤。

Shapiro-Wilk 检验的统计量公式：

其中：

表示样本数据按升序排列后的第 个值。

表示样本数据的均值。

是预先计算的常数，通常通过期望正态分布的协方差矩阵的特征值和特征向量计算得到。

Kolmogorov-Smirnov 检验的统计量公式：

其中：

表示经验累积分布函数 (ECDF)。

表示理论累积分布函数 (CDF)。

表示取最大值。

代码编写之后结果如下。

表 6：正态分布检验因素

|  |  |
| --- | --- |
|  | 0.0000 |
|  | 0.0005 |
| 偏度 | 0.267 |
| 峰度 | 3.755 |

Kolmogorov-Smirnov Test p-value = 0.0000: KS检验的p值为0，表示在统计上显著拒绝数据符合正态分布的假设。KS检验对样本中的任何偏离正态分布都非常敏感，尤其是在大样本情况下。

Anderson-Darling Test p-value = 0.0005: AD检验也显示了类似的结果，其p值虽然略高于KS检验，但仍然远低于任何常用的显著性水平（如0.05），从而同样拒绝了数据符合正态分布的假设。AD检验特别注重数据尾部的行为，其结果指出数据尾部可能与正态分布不符。

偏度值是0.267:偏度值接近0，显示数据的对称性较好，但仍轻微向右偏斜。峰度值是3.755: 正态分布的峰度理论值为3（excess kurtosis为0）。您的数据峰度略高，表明数据分布比正态分布稍尖，显示出比正态分布更厚的尾部。

从上述五个图的分析和三个检验方法我们可以看出洪水趋近于正态分布，不严格符合正态分布。

六、模型评价

## 6.1 模型的优点

处理大规模数据能力强：随机森林能够有效地处理大规模数据集，它可以通过随机选择特征来降低数据维度，减少计算复杂度。

特征选择能力强：随机森林能够自动计算每个特征的重要性得分，从而帮助识别出对模型预测性能有重要贡献的特征。这一特点有助于简化模型，提高模型的泛化能力。

抗噪声能力强：随机森林对噪声和异常值不敏感，因为每个决策树都是基于不同的训练子集和特征子集构建的，它们之间的预测结果存在差异，这种差异有助于减少噪声和异常值对最终预测结果的影响。

## 6.2 模型的不足

对决策树参数敏感：随机森林中决策树的数量和深度等参数需要仔细调节。如果决策树的数量太少，可能导致模型欠拟合；如果数量太多，尽管数据集进行了分割和交叉验证方法，但仍可能导致模型过拟合。同样地，决策树的深度也需要适中，以捕捉复杂的数据模式而不至于过拟合。

计算复杂度和内存需求高：机森林需要训练多个决策树模型，并且需要存储这些模型，因此计算复杂度和内存需求相对较高。在资源有限的情况下，可能需要考虑其他算法。

**七、模型的改进与推广**

## 7.1 模型的推广

模型的推广可以从跨领域应用、多尺度预测、集成多种模型、国际合作等方面入手，实现更广泛应用于更大的社会效益。

跨领域应用

环境监测：将洪水灾害预测模型扩展到其他自然灾害的预测中，如地震、滑坡、台风等。这些自然灾害的数据集通常也包含多种影响因素，随机森林模型能够很好地处理这类复杂数据。

城市管理：将模型应用于城市水资源管理、排水系统设计等领域，通过分析城市内涝的风险因素，优化城市基础设施的布局和管理策略。

农业生产：结合气象数据和农业实践数据，预测农作物受灾风险，帮助农民制定防灾减灾措施，提高农业生产效益。

多尺度预测

短期预测：结合实时气象数据和卫星遥感数据，实现洪水灾害的短期预警，为应急响应提供及时信息。

长期预测：基于历史气候数据和长期趋势分析，进行洪水灾害的长期预测，为城市规划和防洪工程建设提供科学依据。

集成多种模型

将随机森林模型与其他机器学习算法（如支持向量机、神经网络等）进行集成，通过投票或加权平均的方式提高预测精度和稳定性。

结合专家系统和规则引擎，将机器学习模型的预测结果与专家的经验知识相结合，提高模型的实用性和可信度。

这些推广措施能够为防洪减灾工作提供更加科学、有效的支持和保障，推动洪水灾害预测模型的研究与应用取得更大的进展和成果。

## 7.2 模型的改进

为了提升模型的预测性能以及适用性，可以从数据增强、模型优化、特征工程深化、实时预测等方面入手。首先通过采用更先进的数据清洗技术，如自动异常值检测、缺失值插补等，减少数据噪声和缺失对模型性能的影响。引入专家标注机制，对部分数据进行人工标注和验证，提高数据标签的准确性和可靠性。其次，模型优化是提高模型性能的关键，采用Stacking方法，将多个不同的机器学习模型（如随机森林、梯度提升树、神经网络等）作为基模型，再用一个元模型（如逻辑回归）来整合这些基模型的预测结果，以提高整体的预测性能。

八、参考文献

1. 罗俊,李博阳,黄琳,等.基于随机森林模型的近岸海域水质污染预测预警方法研究[J].黑龙江科学,2024,15(12):22-25.
2. 马赛赛,张瑞新.基于随机森林算法的露天矿抛掷爆破影响因素分析[J].露天采矿技术,2024,39(03):11-14.DOI:10.13235/j.cnki.ltcm.2024.03.003.
3. 龚利平,李建,陈忠英,等.基于决策树法构建深静脉置管并发导管相关性感染的风险预测模型及防控策略分析[J].军事护理,2024,41(06):52-54.
4. 郭丽,孙华.基于K-means和支持向量机SVM的电力数据通信网络流量分类方法[J].网络安全技术与应用,2024,(04):64-66.
5. 王子轩,陈德辉,欧斌,等.基于KPCA降维分析的特高拱坝监测模型[J/OL].人民长江,1-11[2024-07-08].http://kns.cnki.net/kcms/detail/42.1202.TV.20240619.1158.006.html.
6. 刘肖冰.基于主成分分析和聚类分析不同品种猕猴桃品质评价[J/OL].现代食品科技,1-10[2024-07-08].https://doi.org/10.13982/j.mfst.1673-9078.2024.10.1242.
7. 赵磊.脉冲噪声中基于斯皮尔曼简捷相关系数的信号检测算法[D].广东工业大学,2022.DOI:10.27029/d.cnki.ggdgu.2022.001595.
8. 贾科,杨哲,魏超,等.基于斯皮尔曼等级相关系数的新能源送出线路纵联保护[J].电力系统自动化,2020,44(15):103-111.

**附录（另起一页）**

**树的数量（n\_estimators）**：为了提高模型的稳定性和预测精度，我们设置了 n\_estimators=500。这意味着随机森林模型将包括500棵决策树，增加树的数量通常能提高模型的性能，但也会相应增加计算成本。

**树的最大深度（max\_depth）**：我们设置了 max\_depth=15。限制树的深度可以帮助防止模型过拟合，特别是在样本数据量不是非常大的情况下。这个深度是基于交叉验证结果选择的，以确保每棵树的生长既能充分学习数据的特征，又不会过度复杂化。

**划分时考虑的特征数量（max\_features）**：设置为 max\_features='sqrt'，这意味着在寻找最佳分割时，每个分割最多考虑总特征数平方根的数量。这个参数的设置可以帮助提高树的多样性，降低模型的方差，同时还能提升训练速度。

**最小样本分割（min\_samples\_split）**：我们设置了 min\_samples\_split=10，这表示一个节点必须至少有10个样本才允许进一步分割。这有助于防止树变得过于复杂和过拟合。

**最小样本叶节点（min\_samples\_leaf）**：设置 min\_samples\_leaf=4，确保每个叶节点至少有4个样本，进一步控制模型的过拟合，提高模型在未见数据上的表现。

问题二代码：

风险等级划分代码：

1. import pandas as pd
2. from sklearn.cluster import KMeans
3. import matplotlib.pyplot as plt
4. from matplotlib.colors import ListedColormap
5. import numpy as np
6. from pylab import mpl
7. *# 设置显示中文字体*
8. plt.rcParams['font.sans-serif'] = ['SimHei']
9. mpl.rcParams['font.serif'] = ['Times New Roman']
10. *# 加载数据*
11. data = pd.read\_csv('train.csv',encoding='gbk')
12. *# 由于我们需要基于洪水发生的概率进行聚类，我们可以只使用这个列*
13. X = data[['洪水概率']].values.reshape(-1, 1)  *# 将数据调整为二维形式*
14. *# 使用 K-means 聚类算法，因为需要分为3类（高风险、中风险、低风险）*
15. kmeans = KMeans(n\_clusters=3, random\_state=0).fit(X)
16. *# 预测每个洪水事件的类别*
17. labels = kmeans.labels\_
18. *# 将类别标签添加回原始数据框*
19. data['Risk\_Level'] = labels
20. *# 为了将聚类结果映射到具体的风险等级（高风险、中风险、低风险），我们可以按聚类中心的值排序*
21. centers = kmeans.cluster\_centers\_.flatten()
22. sorted\_centers = sorted(centers)
23. risk\_levels = ['低风险', '中风险', '高风险']
24. risk\_mapping = {i: risk\_levels[sorted\_centers.index(center)] for i, center in enumerate(centers)}
25. *# 映射类别标签到具体的风险等级*
26. data['Risk\_Level\_Mapped'] = data['Risk\_Level'].map(risk\_mapping)
27. *# 查看结果*
28. print(data[['洪水概率', 'Risk\_Level', 'Risk\_Level\_Mapped']])
29. *# # 创建颜色映射*
30. cmap\_bold = ListedColormap(['#66BB6A', '#FFC107', '#EF5350'])
31. *# 使用 DataFrame 的索引作为横坐标*
32. index\_values = data.index
33. flood\_probabilities = data['洪水概率']
34. plt.rcParams.update({'font.size': 12})
35. *# 绘制散点图*
36. plt.figure(figsize=(10, 6))
37. plt.scatter(index\_values[labels==2], flood\_probabilities[labels==2], c=cmap\_bold.colors[0], label='Low Risk', alpha=0.7)
38. plt.scatter(index\_values[labels==0], flood\_probabilities[labels==0], c=cmap\_bold.colors[1], label='Medium Risk', alpha=0.7)
39. plt.scatter(index\_values[labels==1], flood\_probabilities[labels==1], c=cmap\_bold.colors[2], label='High Risk', alpha=0.7)
40. centers\_index = np.arange(len(kmeans.cluster\_centers\_))
41. *# 绘制聚类中心*
42. plt.scatter(centers\_index, kmeans.cluster\_centers\_, color='black', marker="x", s=150, label='Centroids')
43. *# 添加标题和标签*
44. plt.title('Clustering of Flood Probability',fontsize=16)
45. plt.xlabel('index',fontsize=14)
46. plt.ylabel('Flood Probability',fontsize=14)
47. plt.grid(True, linestyle='--', linewidth=0.5, alpha=0.5)
48. plt.legend(loc='upper center', bbox\_to\_anchor=(0.5, -0.08), fancybox=True, shadow=True, ncol=5)
49. *# 显示图表*
50. y\_min = 0.25
51. y\_max = 0.75
52. plt.yticks(np.arange(y\_min, y\_max, 0.05))
53. plt.tight\_layout()
54. plt.show()

数据划分文件生成

1. *# 根据Risk\_Level列的值对数据进行分组*
2. grouped\_data = data.groupby('Risk\_Level')
3. *# 遍历每个分组，并将其写入单独的CSV文件*
4. for name, group in grouped\_data:
5. *# 选择需要的列*
6. columns\_to\_write = ['洪水概率', 'Risk\_Level', 'Risk\_Level\_Mapped'] + list(
7. data.columns.difference(['洪水概率', 'Risk\_Level', 'Risk\_Level\_Mapped']))
8. result\_group = group[columns\_to\_write]
9. *# 写入CSV文件，文件名包含风险等级*
10. file\_name = f'risk\_level\_{name}\_data.csv'
11. result\_group.to\_csv(file\_name, index=False, encoding='utf-8')
12. print(f'Data for risk level {name} has been written to {file\_name}')

特征雷达图

1. indicators = [
2. '季风强度', '地形排水', '河流管理', '森林砍伐',
3. '城市化', '气候变化', '大坝质量', '淤积', '农业实践',
4. '侵蚀', '无效防灾', '排水系统', '海岸脆弱性', '滑坡',
5. '流域', '基础设施恶化', '人口得分', '湿地损失', '规划不足', '政策因素'
6. ]
7. mid\_risk\_data = data[data['Risk\_Level\_Mapped'] == '中风险']
8. *# 计算低风险等级下所有相关指标的平均值*
9. low\_risk\_average = mid\_risk\_data[indicators].mean()
10. values = low\_risk\_average.values.tolist()
11. *# 准备雷达图*
12. N = len(indicators)
13. angles = [n / float(N) \* 2 \* math.pi for n in range(N)]
14. angles += angles[:1]
15. *# 创建雷达图*
16. fig, ax = plt.subplots(figsize=(8, 8), subplot\_kw=dict(polar=True))
17. ax.set\_theta\_offset(np.pi / 2)
18. ax.set\_theta\_direction(-1)
19. *# 绘制雷达图*
20. ax.fill(angles, values + values[:1], 'b', alpha=0.25)
21. ax.plot(angles, values + values[:1], linewidth=2, linestyle='solid')
22. *# 添加文本注释*
23. for angle, value, indicator in zip(angles[:-1], values, indicators):
24. *# 调整角度，使其指向正确的指标位置*
25. adjusted\_angle = angle \* 180 / math.pi
26. ax.text(angle, value + 0.5, f"{value:.2f}", ha='center', va='bottom', fontsize=15, rotation=0)
27. ax.set\_yticklabels([])
28. ax.set\_xticks(angles[:-1])
29. ax.set\_xticklabels(indicators)
30. ax.set\_title('中风险事件的指标特征雷达图', y=1.13)
31. ax.tick\_params(axis='x', which='both', pad=25)
32. ax.set\_theta\_offset(math.pi / 2)
33. ax.set\_theta\_direction(-1)
34. ax.set\_rlabel\_position(0)
35. ticks = np.linspace(0, 6.5, 5, endpoint=True)
36. ax.set\_rticks(ticks)
37. ax.set\_rlabel\_position(90)
38. *# 显示图表*
39. plt.tight\_layout()
40. plt.rcParams['font.sans-serif'] = ['SimHei']
41. plt.show()

指标特征频率分布直方图

1. indicators = [
2. '季风强度', '地形排水', '河流管理', '森林砍伐',
3. '城市化', '气候变化', '大坝质量', '淤积', '农业实践',
4. '侵蚀', '无效防灾', '排水系统', '海岸脆弱性', '滑坡',
5. '流域', '基础设施恶化', '人口得分', '湿地损失', '规划不足', '政策因素'
6. ]
7. num\_rows = 5
8. num\_cols = 4
9. cmap = plt.get\_cmap('tab10')
10. fig, axs = plt.subplots(num\_rows, num\_cols, figsize=(12, 9))
11. axs = axs.flatten()
12. xlim\_range = (0, 10)  *# 横坐标范围*
13. low\_risk\_data = data[data['Risk\_Level\_Mapped'] == '低风险']
14. for i, ind in enumerate(indicators[:20]):
15. data = low\_risk\_data[ind].dropna().values  *# 删除NaN值*
16. if len(data) > 0:
17. *# 拟合正态分布*
18. mu, std = norm.fit(data)
19. *# 创建一个从最小值到最大值的x轴数据点*
20. x = np.linspace(min(data), max(data), 1000)
21. *# 计算正态分布的概率密度函数*
22. pdf = norm.pdf(x, mu, std)
23. *# 绘制直方图*
24. sns.histplot(data, linewidth=2, kde=False, ax=axs[i], stat='density', color='#87CEEB', bins=15, alpha=0.5,
25. edgecolor='black')
26. *# 绘制概率密度函数曲线*
27. axs[i].plot(x, pdf, 'r-', lw=2)
28. axs[i].set\_title(f'{ind}')
29. axs[i].set\_xlabel('特征值')
30. axs[i].set\_ylabel('概率密度')
31. *# 设置横坐标范围为数据的最小值和最大值*
32. axs[i].set\_xlim(min(data), 10)
33. fig.suptitle('低风险洪水事件的指标特征频率分布直方图', fontsize=16)
34. plt.tight\_layout()
35. plt.show()
36. *# 显示数据框中低风险级别的洪水概率及其对应的风险等级*
37. print(low\_risk\_data[['洪水概率', 'Risk\_Level', 'Risk\_Level\_Mapped']].head())

问题三代码：

1. import pandas as pd
2. from sklearn.ensemble import RandomForestRegressor
3. from sklearn.model\_selection import train\_test\_split
4. from sklearn.metrics import mean\_squared\_error
5. *# 加载数据*
6. data = pd.read\_csv('train.csv')
7. *# 数据预处理*
8. def preprocess\_data(data):
9. *# 删除包含缺失值的行*
10. data.dropna(inplace=True)
12. *# 计算四分位数和IQR*
13. Q1 = data.quantile(0.25)
14. Q3 = data.quantile(0.75)
15. IQR = Q3 - Q1
17. *# 根据IQR过滤异常值*
18. data = data[~((data < (Q1 - 1.5 \* IQR)) | (data > (Q3 + 1.5 \* IQR))).any(axis=1)]
20. return data
21. *# 应用数据预处理*
22. cleaned\_data = preprocess\_data(data)
23. *# 分离特征和目标变量*
24. features = cleaned\_data.drop('FloodProbability', axis=1)
25. target = cleaned\_data['FloodProbability']
26. *# 划分训练集和测试集*
27. features\_train, features\_test, target\_train, target\_test = train\_test\_split(
28. features, target, test\_size=0.2, random\_state=42)
29. *# 初始化随机森林模型*
30. rf = RandomForestRegressor(n\_estimators=500, max\_depth=15, max\_features='sqrt',
31. min\_samples\_split=10, min\_samples\_leaf=4, random\_state=42)
32. *# 训练模型*
33. rf.fit(features\_train, target\_train)
34. *# 进行预测*
35. predictions = rf.predict(features\_test)
36. *# 评估模型*
37. mse = mean\_squared\_error(target\_test, predictions)
38. print("Mean Squared Error:", mse)
39. *# 特征重要性*
40. importances = rf.feature\_importances\_
41. print("Feature Importances:", importances)
42. import pandas as pd
43. import numpy as np
44. import matplotlib.pyplot as plt
45. import seaborn as sns
46. *# 从CSV文件读取数据*
47. df = pd.read\_csv('/home/xinyuan/train.csv')
48. *# 确保数据包含需要的列*
49. required\_columns = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u']
50. if not all(column in df.columns for column in required\_columns):
51. raise ValueError(f"CSV文件缺少必要的列: {required\_columns}")
52. *# 对除'u'之外的特征进行指数变换*
53. df\_exp = df.copy()
54. df\_exp[required\_columns[:-1]] = df[required\_columns[:-1]].apply(np.exp)
55. *# 计算皮尔逊相关性系数*
56. pearson\_corr\_exp = df\_exp.corr(method='pearson')
57. *# 计算斯皮尔曼相关性系数*
58. spearman\_corr\_exp = df\_exp.corr(method='spearman')
59. *# 绘制皮尔逊相关性系数热力图*
60. plt.figure(figsize=(12, 10))
61. sns.set(font\_scale=1.2)
62. sns.heatmap(pearson\_corr\_exp, annot=True, cmap='coolwarm', fmt=".2f")
63. plt.title('Pearson Correlation Coefficients (Exponential Transformed Data Except "u")')
64. plt.tight\_layout()
65. plt.savefig('/home/xinyuan/pearson\_corr\_exp\_heatmap.png')
66. plt.show()
67. *# 绘制斯皮尔曼相关性系数热力图*
68. plt.figure(figsize=(12, 10))
69. sns.set(font\_scale=1.2)
70. sns.heatmap(spearman\_corr\_exp, annot=True, cmap='coolwarm', fmt=".2f")
71. plt.title('Spearman Correlation Coefficients (Exponential Transformed Data Except "u")')
72. plt.tight\_layout()
73. plt.savefig('/home/xinyuan/spearman\_corr\_exp\_heatmap.png')
74. plt.show()
75. *# 保存皮尔逊相关性系数到CSV文件*
76. pearson\_corr\_exp.to\_csv('/home/xinyuan/pearson\_corr\_exp.csv')
77. *# 保存斯皮尔曼相关性系数到CSV文件*
78. spearman\_corr\_exp.to\_csv('/home/xinyuan/spearman\_corr\_exp.csv')

上面为斯皮尔曼

下面为基于梯度提升的预测评价模型

1. import pandas as pd
2. import matplotlib.pyplot as plt
3. import seaborn as sns
4. from sklearn.model\_selection import train\_test\_split, ParameterGrid
5. from sklearn.ensemble import GradientBoostingRegressor
6. from sklearn.metrics import mean\_squared\_error, r2\_score, accuracy\_score
7. from tqdm import tqdm
8. import joblib
9. *# 从CSV文件读取数据*
10. df = pd.read\_csv('/home/xinyuan/train\_cleaned.csv')
11. *# 确保数据包含需要的列*
12. required\_columns = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't',
13. 'u']
14. if not all(column in df.columns for column in required\_columns):
15. raise ValueError(f"CSV文件缺少必要的列: {required\_columns}")
16. *# 准备特征和目标变量*
17. X = df.drop('u', axis=1)
18. y = df['u']
19. *# 划分数据集*
20. X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=42)
21. *# 定义参数网格*
22. param\_grid = {
23. 'n\_estimators': [50, 100, 150],
24. 'learning\_rate': [0.05, 0.1, 0.2],
25. 'max\_depth': [3, 4, 5]
26. }
27. *# 创建梯度提升树回归模型*
28. gbdt = GradientBoostingRegressor(random\_state=42)
29. *# 使用tqdm显示网格搜索交叉验证的进度*
30. progress\_bar = tqdm(total=len(ParameterGrid(param\_grid)), desc='Grid Search Progress', position=0)
31. best\_score = None
32. best\_params = None
33. for params in ParameterGrid(param\_grid):
34. gbdt.set\_params(\*\*params)
35. gbdt.fit(X\_train, y\_train)
36. *# Calculate validation score*
37. score = gbdt.score(X\_test, y\_test)
38. *# Update best score and params if better score found*
39. if best\_score is None or score > best\_score:
40. best\_score = score
41. best\_params = params
42. *# Update progress bar*
43. progress\_bar.update(1)
44. progress\_bar.set\_postfix({'Best Score': best\_score, 'Best Params': best\_params})
45. progress\_bar.close()
46. *# 输出最佳参数和最佳得分*
47. print("Best Parameters:", best\_params)
48. print("Best Score (R^2):", best\_score)
49. *# 使用最佳参数的模型进行预测*
50. best\_gbdt = GradientBoostingRegressor(random\_state=42, \*\*best\_params)
51. best\_gbdt.fit(X\_train, y\_train)
52. y\_pred = best\_gbdt.predict(X\_test)
53. *# 计算均方误差（MSE）*
54. mse = mean\_squared\_error(y\_test, y\_pred)
55. print(f"MSE: {mse}")
56. *# 计算决定系数R²*
57. r2 = r2\_score(y\_test, y\_pred)
58. print(f"R^2: {r2}")
59. *# 可视化预测结果与实际值的对比*
60. plt.figure(figsize=(12, 8))
61. plt.scatter(y\_test, y\_pred, alpha=0.3)
62. plt.plot([y\_test.min(), y\_test.max()], [y\_test.min(), y\_test.max()], 'r--', lw=2)
63. plt.xlabel('Actual')
64. plt.ylabel('Predicted')
65. plt.title('Actual vs Predicted Flood Probability')
66. plt.tight\_layout()
67. plt.savefig('/home/xinyuan/actual\_vs\_predicted.png')
68. plt.show()
69. *# 保存最佳模型*
70. model\_filename = '/home/xinyuan/best\_gbdt\_model.pkl'
71. joblib.dump(best\_gbdt, model\_filename)
72. print(f"Best Gradient Boosting Regressor model saved to {model\_filename}")

下面为基于多分类逻辑回归的预测评价模型代码：

1. import pandas as pd
2. import matplotlib.pyplot as plt
3. import seaborn as sns
4. from sklearn.model\_selection import train\_test\_split
5. from sklearn.linear\_model import LogisticRegression
6. from sklearn.metrics import accuracy\_score, classification\_report, confusion\_matrix
7. from sklearn.preprocessing import StandardScaler
8. *# 从CSV文件读取数据*
9. df = pd.read\_csv('/home/xinyuan/train\_cleaned.csv')
10. *# 确保数据包含需要的列*
11. required\_columns = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u']
12. if not all(column in df.columns for column in required\_columns):
13. raise ValueError(f"CSV文件缺少必要的列: {required\_columns}")
14. *# 准备特征和目标变量*
15. X = df.drop('u', axis=1)
16. y = df['u']
17. *# 将目标变量转换为多分类标签*
18. def classify\_risk(probability):
19. if probability > 0.53:
20. return '高风险'
21. elif probability >= 0.47:
22. return '中风险'
23. else:
24. return '低风险'
25. y\_class = y.apply(classify\_risk)
26. *# 划分训练集和测试集*
27. X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y\_class, test\_size=0.3, random\_state=42, stratify=y\_class)
28. *# 特征标准化*
29. scaler = StandardScaler()
30. X\_train\_scaled = scaler.fit\_transform(X\_train)
31. X\_test\_scaled = scaler.transform(X\_test)
32. *# 训练多分类逻辑回归模型*
33. log\_reg = LogisticRegression(multi\_class='multinomial', solver='lbfgs', max\_iter=500)
34. *# 模型训练*
35. print("Training Logistic Regression model...")
36. log\_reg.fit(X\_train\_scaled, y\_train)
37. *# 在测试集上进行预测*
38. y\_pred = log\_reg.predict(X\_test\_scaled)
39. *# 评估模型性能*
40. accuracy = accuracy\_score(y\_test, y\_pred)
41. report = classification\_report(y\_test, y\_pred, zero\_division=0, output\_dict=True)
42. print(f"Accuracy: {accuracy}")
43. print(f"Classification Report:\n{classification\_report(y\_test, y\_pred, zero\_division=0)}")
44. *# 绘制混淆矩阵*
45. cm = confusion\_matrix(y\_test, y\_pred, labels=['低风险', '中风险', '高风险'])
46. plt.figure(figsize=(8, 6))
47. sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=['低风险', '中风险', '高风险'], yticklabels=['低风险', '中风险', '高风险'])
48. plt.xlabel('Predicted')
49. plt.ylabel('Actual')
50. plt.title('Confusion Matrix')
51. plt.show()
52. *# 绘制分类报告条形图*
53. report\_df = pd.DataFrame(report).transpose()
54. report\_df = report\_df.drop(columns=['support'])  *# 去除support列，只保留precision, recall, f1-score*
55. report\_df.plot(kind='bar', figsize=(12, 8))
56. plt.title('Classification Report')
57. plt.xlabel('Classes')
58. plt.ylabel('Scores')
59. plt.xticks(rotation=45)
60. plt.legend(loc='lower right')
61. plt.tight\_layout()
62. plt.show()