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		apmcm24100508

基于 LightGBM 模型的洪水灾害数据分析与预测

摘　要

洪水灾害是由暴雨、融冰化雪、风暴潮等自然因素以及人类活动如乱砍滥伐和城市化扩展共同作用引起的自然现象，对社会经济和人类生活造成了巨大威胁。随着气候变化的加剧和人类活动的影响，洪水灾害的频率和严重程度显著上升，因此，建立科学有效的洪水灾害预测模型显得尤为重要。

针对问题一，通过分析与洪水发生有关联的 20 个指标，应用灰色关联模型找出了与洪水发生密切相关的指标，包括淤积、农业实践、气候变化、河流管理、季风强度等。我们对这些关键指标进行了深入分析，提出了具体的防灾建议和措施，以便更好地预防洪水灾害的发生，为相关部门提供数据支持。

针对问题二，采用 K-Means 算法和互信息模型对洪水发生的风险进行了分类和评估。通过对数据进行聚类分析和权重计算，我们将洪水事件划分为高、中、低三种风险等级，并根据互信息模型得出的指标建立多元线性回归模型，构建了一个洪水发生风险的预警评价模型。为了确保模型在不同情境下的稳定性和可靠性，我们对模型的灵敏度和准确度进行了详细分析，为灾害风险管理提供了科学依据。

针对问题三，基于问题一中确定的关键指标，我们建立了一个洪水发生概率的预测模型。通过对比随机森林、XGBoost 和 LightGBM 三种集成模型，最终选择了性能最优的 LightGBM 模型。该模型在处理高维数据、提升计算效率和抗噪能力方面表现出色，通过模型验证和优化，进一步提高了预测的准确性和效率，显著提升了洪水预警系统的性能。

对于问题三模型的改进，在将指标缩减为 5 个关键指标后，我们选择了多层感知器（MLP）模型进行分析预测。通过使用高级的单一模型，我们确保了在减少指标数量的情况下仍能保持较高的预测准确性和效率。

针对问题四，将优化后的 LightGBM 模型应用于附件 test.csv 中的数据，预测了所有事件的洪水发生概率。通过绘制洪水概率的直方图和正态分布曲线，我们验证了预测结果的分布特性，并分析了其是否符合正态分布。结果表明，模型的预测结果与实际情况高度吻合，能够为洪水灾害预警和管理提供重要参考。

综上所述，本文提出的基于 LightGBM 模型的洪水灾害预测方法，通过对大量洪水数据及其影响因素的分析，建立了高效的洪水预测模型，为防灾减灾和风险管理提供了科学依据和技术支持，具有广泛的应用前景和推广价值。

关键词：K-Means 算法，互信息算法，LightGBM 模型，多层感知器，洪水灾害预测

一、 问题重述

1.1 问题背景

洪水作为一种自然灾害，造成了严重的社会经济危害[1]。洪水的频率受到各种因素的影响，全球变暖导致极端天气事件的增多和强度增加，频繁的暴雨和极端降水事件使得河流和湖泊的水位急剧上升，容易引发洪水灾害。此外，自然生态系统的破坏，如湿地的减少，也削弱了自然调节洪水的能力。与此同时，人口的迅速增长导致了耕地面积的扩大、围湖造田和森林的乱砍滥伐，这些人为活动改变了地表状态和汇流条件，提高了洪灾的发生的概率。综合考虑自然和人为因素，来研究和预测洪水的发生，对于减少损失和保障人民安全至关重要。

1.2 问题提出

问题一：洪水是河流、海洋、湖泊等水体上涨超过一定水位，威胁有关地区的安全，甚至造成灾害的水流。通过对超过 100 万的真实洪水数据的分析，其中包括对发生洪水概率有影响的季风强度、地形排水等二十项指标，探寻洪水发生的真正影响因素。我们旨在建立一个数学模型，能从二十项指标中抽象得到与洪水发生概率之间的关联度大小，以此确定关联最密切的指标，为预防洪水提出建议和措施，同时剔除相关性小的变量，简化模型。

问题二：洪水作为一种常见的自然灾害，其发生概率大小影响着人们的应对措施。为了更有效地进行洪水风险管理，我们旨在基于附表提供的洪水事件数据，根据洪水发生概率将事件划分为高、中、低三种风险等级，并深入分析各风险等级事件的指标特征。通过科学的方法选取关键指标，并计算其权重，构建一个能够评估洪水发生风险的预警评价模型，能够对洪水事件的风险等级进行快速、准确的判断。最后，对模型进行灵敏度分析，以确保其在不同情境下的稳定性和可靠性。

问题三：在洪水灾害的预测与管理中，识别并量化影响洪水发生的关键因素对于提高预警系统的准确性和效率至关重要。基于前期问题一对大量洪水数据及其 20 项相关指标的深入分析，我们旨在构建一个基于强相关性指标的洪水发生概率预测模型，实现模型的精简与优化。并通过验证与优化调整，建立当只用 5 个关键指标时的洪水发生概率的预测模型，进一步探索在仅使用少数关键指标情况下模型的预测性能。

问题四：在自然灾害管理中，建立准确且全面的洪水风险预警评价模型对于提升防灾减灾能力具有重要意义。基于问题二中已构建的发生洪水不同风险的预警评价模型，我们旨在将该模型应用于附件中提供的具体事件数据集，预测各事件发生洪水的概率，并通过绘制洪水事件发生概率的直方图和折线图预测结果的分布特性，验证其是否服从正态分布。

二、 问题分析

2.1 问题一的分析

通过查阅相关资料，首先对洪水发生的物理模型进行模拟。主要考虑流域面积(A)、平均坡度(S)和洼地蓄水体积(V)对内涝致灾的影响，建立针对暴雨内涝的以洼地小流域为评价单元的地形控制作用指数(topographic control index，TCI)[2]。淹没洼地所需时间与汇水面积 A 和坡度 S 成反比，与洼蓄体积 V 成正比。

通过降雨示意、地形剖面图及相关关键指标，展示了洪水发生的关键因素及其相

互关系。降雨汇集在地形低洼处，坡度和汇水面积影响水流的速度和汇集能力，而注蓄体积决定了水体的积聚能力。TCI 指标通过综合这些因素来衡量洪水发生的风险。

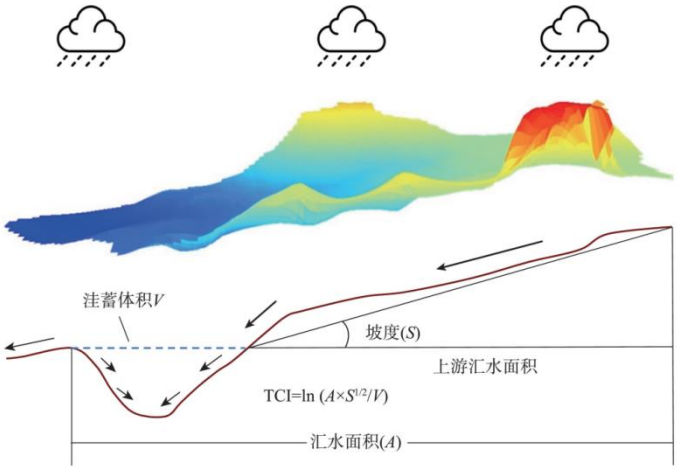

图 1 地形控制作用指数(TCI)原理示意图

基于 TCI 模型，针对问题一的多变量分析问题，我们需要建立二十项指标与洪水的概率之间的关系模型。借助关联度算法得出关联度系数，找出与洪水发生概率密切相关的因素。

针对这一目标，我们首先需要对数据进行预处理，处理异常值并观察数据的分布情况。接下来，通过灰色关联模型计算各指标与洪水发生概率之间的关联度系数，以确定最具影响力的因素。通过这种方法，我们能够筛选出与洪水发生关系最密切的关键指标，从而为洪水预防提出合理的建议和措施，并为问题三的模型建立奠定基础。

2.2 问题二的分析

问题二的核心在于通过评估洪水发生的风险等级并建立预警评价系统。基于洪水事件数据，根据洪水发生概率将事件划分为高、中、低三种风险等级，并深入分析各风险等级事件的指标特征。

应用 K-Means 算法对洪水发生的概率进行聚类分析，得到聚类中心和聚类区间。并采用互信息（MI）模型确定不同指标的权重。选取权重值较高的指标，建立多元线性回归模型，构建了一个能够评估洪水发生风险的预警评价模型，能够对洪水事件的风险等级进行快速、准确的判断。对模型进行了灵敏度分析，以确保其在不同情境下的稳定性和可靠性。

2.3 问题三的分析

问题三的核心在于用问题一确定的九个指标建立一个能够能够预测洪水发生概率的模型，然后进一步精简模型，挑选合适的五个指标建立预测模型。

由于九个指标较多较复杂，我们选择了集成模型，这类模型能够综合处理复杂指标。建立了随机森林（RF）、极梯提升（XGBoost）和轻量级梯度提升（LightGBM）这三种集成模型，选出拟合度最好的模型。

通过优化调整，建立只有五个关键指标时的预测模型。由于指标数量较少，我们最初考虑选择一种单一模型。然而，为了确保预测的准确性，我们决定使用一种高级的单一模型——多层感知器。

2.4 问题四的分析

将问题三中构建的发生洪水不同风险的预警评价模型应用于具体事件数据集，预测各事件发生洪水的概率。通过绘制洪水事件发生概率的直方图和折线图预测结果的

分布特性，验证了其是否服从正态分布。得到各事件发生洪水的概率分布，验证了分布特性。

2.5 数据浅析及思维框架

三、 模型假设

在建立洪水发生概率预测模型的过程中，为了简化问题并确保模型的可操作性，我们做出了一些基本假设。这些假设为模型的构建、训练和预测提供了理论基础，但同时也可能限制模型的应用范围和预测精度。以下是模型假设的具体内容：

1. 假设研究区域内的基础设施、防洪措施等条件在模型应用期间保持不变。

2. 假设不存在小概率事件，如黑天鹅事件（如极端天气事件，超强台风、极端降雨等），或者人为因素导致的非正常情况（如大规模的工程建设、河道改造等）。

3. 假设题目给出的影响洪水发生概率的核心因素能够一定程度上与洪水发生概率建立模型关系，进行分析和预测。

4. 假设 LightGBM 模型中特征与目标变量之间存在某种程度的线性关系，且输入特征之间相互独立，特征之间不存在多重共线性。

5. 假设输入特征的数据分布在训练集和测试集中是一致的，即训练数据和测试数据来自相同的分布。

6. 假设历史数据中的模式和规律在未来会保持不变，即数据的时间序列特征是稳定的。

四、 符号说明

符号	说明	单位
	数据点	分
	质心	分
	特征的维数	个
	簇　中的数据点集合	个
	簇　中的数据点数量	个
	取值为 的样本数量	个
	所有的事件数量	个
	指标得分为 x 且洪水风险类别为 y 的事件数量	个
d	特征的总数量	个

五、 模型的建立与求解

5.1 使用灰色关联模型系统分析洪水发生概率的主要因素

5.1.1 模型思路架构

对于问题一，附表所给出的数据灰度较大，分布规律不够典型，可能出现量化结果与定性分析结果不符的现象，本文引入灰色关联分析模型，由于洪水发生概率可能有二十种影响指标，于是对每种指标进行遍历，并对关联系数进行加权处理，计算得出各项指标与洪水发生概率的灰色相关度，通过比较子序列和母序列的关联度得出结论[3]，思路框架如图 2 所示。

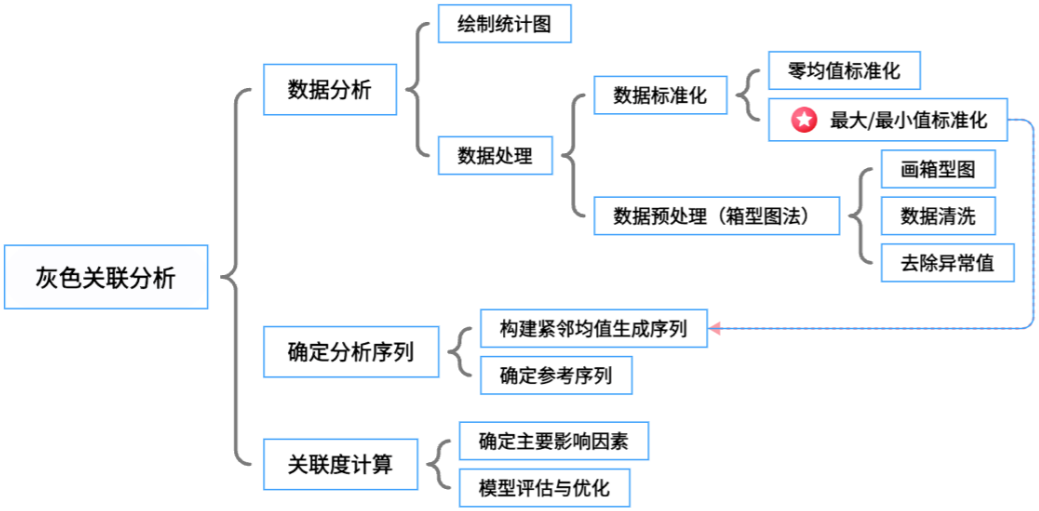

图 2 灰色关联分析模型示意图

5.1.2 数据预处理

（1）异常值的剔除和处理，各指标存在一些异常值将极大降低模型的准确度，考虑到要处理的指标多，和画箱型图法适用性广的特点，我们使用画箱型图法对每个指标的数据的异常值进行判断、剔除和处理。对于小于下边界的异常值，将其替换为下边界值；对于大于上边界的异常值，将其替换为上边界值。箱型图原理如图 3。

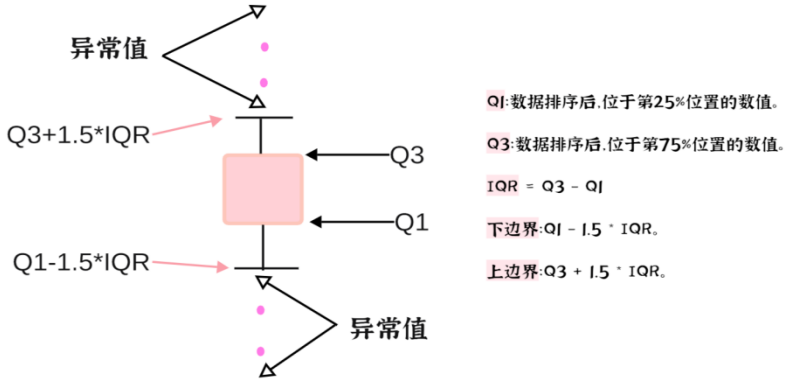

图 3 箱型图原理图

对处理后的数据和概率数据进行可视化，绘制各个指标得分的数量直方图和 KDE 概率密度图，如图 4。

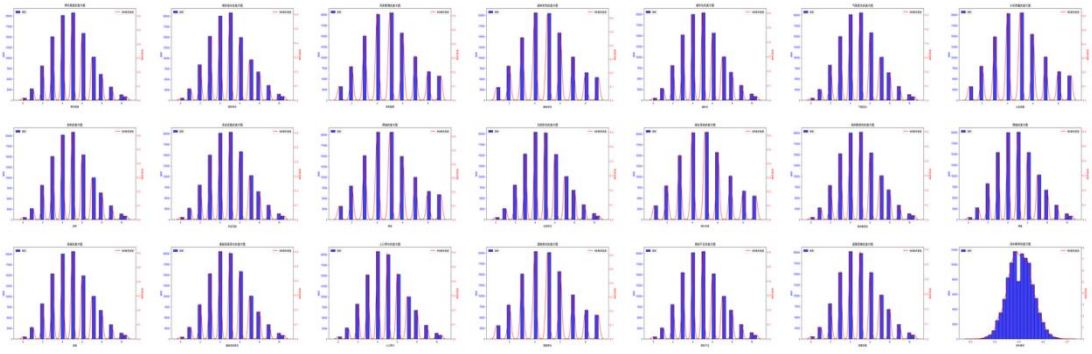

图 4 指标得分的数量直方图和 KDE 概率密度图

（2）消除异常值之后，为了消除各指标得分尺度、量纲不一的差异，我们需要借助数据标准化的方式是两者居于同一比较地位，相对于零均值标准化，我们倾向于采用 min-max 标准化对数据进行归一化处理，将表格中的数据经过变换转化成没有量纲的表达式。

Min-max 标准化公式如下：

(1)

针对上述标准化公式，借助 Python 对所给的数据进行处理，得到归一化后的数据。

5.1.3 确定分析序列

本问探究的 20 个影响指标是自变量，影响着系统行为。被研究对象即因变量洪水发生概率作为参考序列以反映系统的行为特征。由此我们确定的序列如下：

母序列：洪水发生概率　　子序列：20 个影响指标

对比子母序列并结合下列计算公式，我们可以计算出子序列各个指标与母序列的关联系数：

母序列：

子序列：

(2)

…

为了考虑序列的整体变化趋势和局部变化特征，

(3)

通过上述公式，我们可以计算出两级最小差值（a）和两级最大差值（b）。通过综合考虑最大值和最小值，我们能够更全面地描述序列的特征和变化趋势，并进而准确计算关联度。使用这种方法，可以在一定程度上平衡整体趋势和局部特征之间的关系，从而提高灰色关联模型的准确性和可靠性。 因此，我们将其代入最终的关联系数计算公式：

(4)

代入分辨系数 =0.5，求出关联系数 的值

5.1.4 灰色关联度计算

(5)

通过除以样本量 n，可以对关联系数进行归一化。由于每个数据点上的关联系数取值范围可能存在差异，并且样本量的大小也可能会影响计算结果的尺度。为了使得不同样本量和不同数据范围的序列能够进行比较和对比，我们需要对关联系数进行归一化处理，将其限制在[0,1]的范围内。

运行得到每个指标的灰色关联度值，绘制成表如下：

表 1 每个指标的灰色关联度值

指标	灰色关联度值	指标	灰色关联度值
季风强度	0.707628	城市化	0.692787
地形排水	0.681653	气候变化	0.707646
河流管理	0.715706	大坝质量	0.715536
森林砍伐	0.688852	淤积	0.727431
农业实践	0.716449	侵蚀	0.69934
无效防灾	0.700754	排水系统	0.703328
海岸脆弱性型	0.703549	滑坡	0.706461
流域	0.714254	基础设施恶化	0.69502
人口得分	0.689276	湿地损失	0.675788
规划不足	0.714684	政策因素	0.715782

5.1.5 主要影响因素分析及建议

从得到的灰色关联度系数可以看出，各指标的灰色关联度，其中淤积的灰色关联度系数最大，达到了 0.727431，农业实践的关联度系数达到 0.716449，均处于较高的关联度水平。

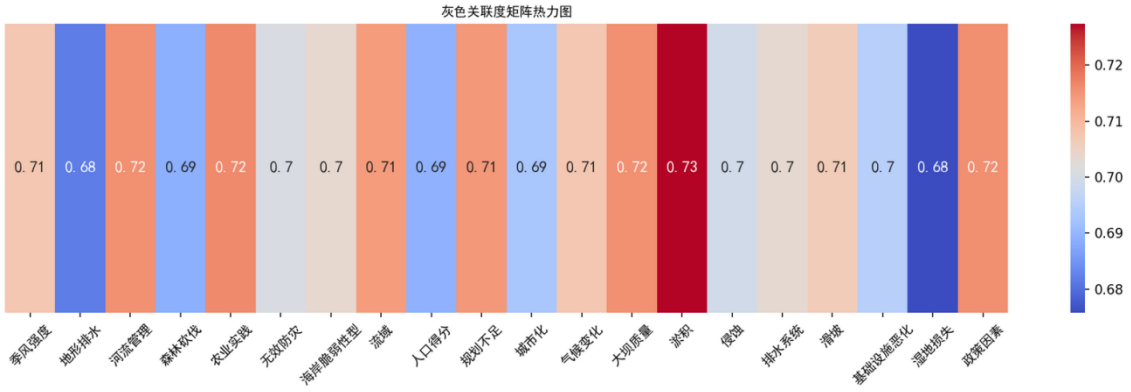

图 5 最终关联度系数结果比对

对关联度系数进行排序，以关联系数 0.7 作为界线，得到与洪水发生有着密切关联的指标有河流管理、大坝质量、淤积、农业实践、气候变化、季风强度、规划不足、政策因素、流域九项指标。关联度不大的指标有地形排水、森林砍伐、城市化、侵蚀、无效防灾、排水系统、海岸脆弱性、滑坡、基础设施恶化、人口得分、湿地损失十一项指标。

对于灰色关联度得出的不同相关度的指标，我们进行如下分析：

高关联性指标分析：

河流管理、大坝质量、淤积、流域：这些指标直接涉及河流系统和水利工程的管理和维护，若管理不善，会直接导致洪水风险增加。

农业实践、气候变化、季风强度：农业实践可能影响土壤吸水能力，气候变化和季风强度则直接影响降水量和水文条件，进而影响洪水发生概率。

规划不足、政策因素：这些因素涉及政策和规划层面，合理的规划和有效的政策能显著减少洪水风险。

低关联性指标分析：

地形排水、森林砍伐、城市化：虽然这些因素对洪水有一定影响，但在具体情境中其影响力相对较小，可能是因为其他高关联性因素的影响更为显著。

基础设施恶化、人口得分、湿地损失：这些指标虽然也与洪水有关，但可能其直接影响力较弱，或是其影响是间接的，通过其他高关联性指标起作用。

根据上述灰色关联度分析，我们得出以下措施和建议：

对于河流系统和水利工程的**管理和维护**，通过高科技手段精准监测河流淤积情况，实施科学合理的清淤工程，同时注重河流生态修复，引入本土植物，恢复生物多样性，让河流重新焕发生机与活力。

对于**自然因素**导致的降水量和水文条件的变化，利用大数据、人工智能等先进技术，构建高效精准的季风等极端天气预警系统，例如应用遥感图像进行大范围洪涝灾害遥感监测的研究[4]。提前部署防洪措施，实现气候智能防洪。建立健全灾害风险评估机制，定期对洪水等自然灾害进行风险评估和隐患排查。同时，加强应急管理体系建设，完善应急预案和救援物资储备，提高应急响应速度和救援效率，形成防灾减灾的强大合力。

对于**人为因素**导致的政策和规划层面的不足，加强防洪减灾领域的政策研究与制定，完善相关法律法规体系，明确政府、企业和社会各界的责任与义务。通过严格的执法监管和有效的政策激励，推动防洪减灾工作的深入开展。将防洪理念融入城市总体规划，合理规划城市空间布局，避免在洪水高风险区域进行开发建设。同时，加强城市排水系统建设，提高城市排水能力，确保居民生命财产安全。

5.2 聚类和互信息模型结合的洪水发生风险预警评价模型

5.2.1 聚类（K-means）模型建立

K-means 算法是一种用于聚类分析的无监督学习算法。它将数据点分成 K 个簇，每个簇由一个质心（即簇的中心）表示。K-means 算法的目标是通过迭代优化，使得同一簇内的数据点尽可能相似，而不同簇的数据点尽可能不同[5]。

随机选择 3 个发生洪灾的概率作为初始质心。

将每个数据点分配给最近的质心，形成 3 个簇。代入欧氏距离公式，计算数据点与质心之间的距离：

(6)

代入质心公式，计算每个簇中所有数据点的平均值，作为新的质心：

(7)

当质心位置不再发生变化或变化很小时，算法收敛并停止迭代。

5.2.2 使用 K-Means 算法对洪水发生概率进行聚类

代入训练数据，使用 python 对洪水发生概率进行聚类分析，得到低、中、高风险聚类中心：

[[0.43834819] [0.49744545] [0.56086881]]

模型输出低、中、高风险的区间：

[0.285,0.465] [0.470,0.525] [0.530,0.725]

将洪水发生概率在[0.285,0.465]范围内的事件划分成低风险的洪水事件；将洪水发生概率在[0.470,0.525]范围内的事件划分成中风险的洪水事件；将洪水发生概率在[0.530,0.725]范围内的事件划分成高风险的洪水事件。绘制聚类结果散点图如图 6。

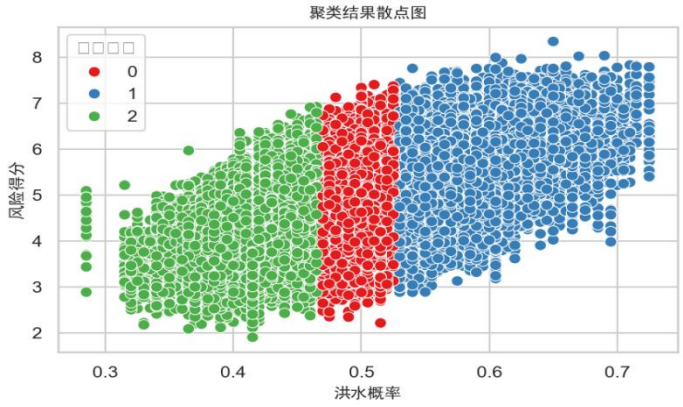

图 6 聚类结果散点图

5.2.3 互信息（MI）模型建立

互信息是一种衡量两个变量之间的相互依赖程度即聚类效果的统计量。在特征选择中，互信息用于衡量每个指标与洪水风险类别之间的相关性。[6]

计算每个指标和洪水风险类别的边缘概率分布：

(8)

计算联合概率分布：

(9)

计算互信息的值：

(10)

计算标准化后的权重：

(11)

5.2.4 使用互信息 MI 计算不同指标的权重

代入数据，得到互信息的值，根据互信息值求出 20 个相关指标的权重值，绘制可视化图如图 7。

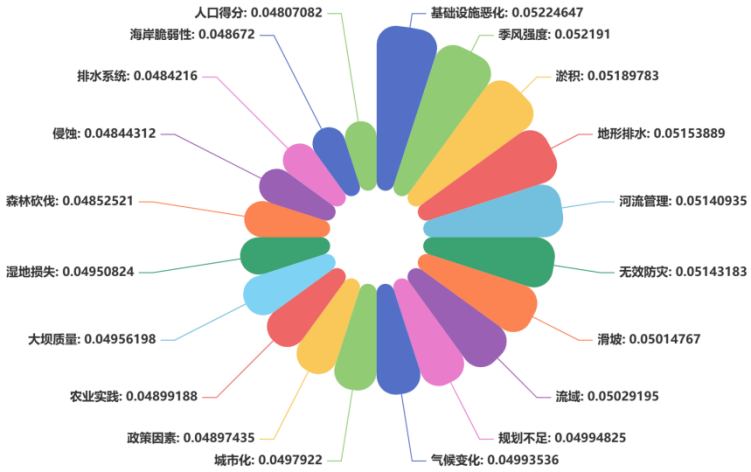

图 7　互信息值可视化图

以 0.5 作为界限，选取基础设施恶化、季风强度、淤积、地形排水、河流管理、无效防灾、滑坡、流域八个指标作为相关指标继续下一步模型的研究。

5.2.5 多元线性回归模型构建

多元线性回归模型是用于解释一个因变量和多个自变量之间关系的统计工具。多元线性回归方程的形式为：

(12)

其中，Y 是洪水概率，$X_1.....X_k$ 是自变量，β_0 为截距，$\beta_1.....\beta_k$是各自变量的回归系数。

通过最小二乘法（OLS）来估计回归系数，使得误差平方和最小：

(13)

为了找到最优的回归系数，使得模型预测值与实际值之间的误差最小，对误差平方和关于每个回归系数求导，并设导数为零：

(14)

使用矩阵形式求解回归系数：

(15)

5.2.6 构建洪水发生风险的预警评价模型

根据多元线性回归算法，求得回归方程:

将多元线性回归模型得到的洪水发生的概率值与聚类模型对应，对应区间则为当前处在的洪水危险等级。由此，构建出评估洪水发生风险的预警评价模型。

5.2.5 准确度和灵敏度分析

对多选回归模型的结果进行可视化，绘制下图：

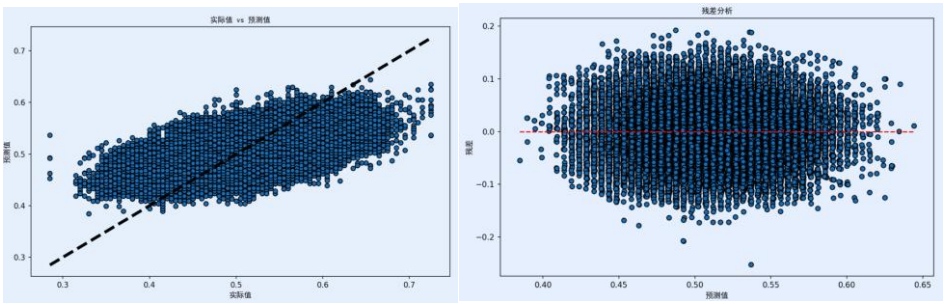

图 8 实际值与预测值对比（左）和残差分析（右）

对于每组事件数据组合，用模型计算出洪水发生的概率，得到相应的风险等级，并记录模型输出的误差平方（MSE）和和 R^2值。

(16)

(17)

其中，n 是样本数量，Y_i是实际的洪水发生概率，$\hat{Y}_i$ 是预测的洪水发生概率。

是实际洪水发生概率的平均值，得到：

MSE=0.0018304378064816 84

R^2 Score=0.295801559204138

MSE 值相对较小，说明模型预测的误差不大，预测结果精确。R^2值约为 0.296，说明模型能够解释约 29.6%的因变量变异。MSE 值表现较好，R^2值表现一般，需要对模型进行进一步的改进和优化。

5.3 构建洪水发生概率的分析预测模型

由于第一问确定的指标较多较复杂，在构建预测模型时，我们选择了集成类型的模型，这类模型能够综合处理多个不同类型的指标。建立了随机森林（RF）、极梯提升（XGBoost）和轻量级梯度提升（LightGBM）这三种集成模型，选择这三个模型是

因为它们在处理高维数据、处理大规模数据集、提升计算效率和抗噪能力方面表现出色。接着选出拟合度最高的模型作为最终方案。

5.3.1 使用随机森林建立洪水发生概率分析预测模型

随机森林[7]是一种使用多棵决策树对样本进行训练、分类和预测的方法。在数据分类过程中，可以通过每个变量的重要性来衡量其在分类中的地位。随机森林中的"随机"有两层含义：

首先，样本选择是基于带撤回的抽样，这意味着每个样本都有可能被多次选中或不被选中。这种抽样方法可以有效地增加决策树之间的差异，进一步降低过度筛选的风险。

其次，在构建决策树的过程中，随机森林并不是使用所有特征来构建每一棵决策树，而是从所有特征中随机抽取一部分特征来构建决策树。这种方法可以有效降低特征之间的相关性，提高模型的性能。

在随机森林中，每棵决策树都是通过对数据进行迭代分区来构建的。在构建决策树的过程中，数据会根据特定的指标进行分割，直到达到预定的停止条件。为了避免过度分裂问题，随机森林还可以通过控制决策树的深度和节点停止分裂的最小样本数等参数来限制决策树的生长。

具体步骤如下：

准备包含特征和目标变量的数据集：

，

其中 xi 是第 i 个样本的特征向量， y_i 是目标变量（发生洪水的概率）。

将数据集划分为训练集 和测试集 Dtest。通过有放回抽样从训练集中选择样本，形成多个子集：

在每个节点分裂时，随机选择部分指标进行分裂。基于选择的样本子集和特征训练每个决策树：

随机森林通过概率平均的方式集成多个决策树的预测结果。对于测试集中的每个样本 x，计算所有决策树的预测结果

集成预测结果，采用平均法（回归任务）。

(18)

采用 gini 算法结合 5 次交叉验证对随机森林模型参数进行优化，得到最优参数组合，如表 2 所示。

表 2 随机森林模型优化参数表

参数	含义	数值
n_estimators	决策树的数量	216
max_features	每棵决策树使用的最大特征数	4
min_samples_split	一个节点进行拆分所需的最小样本数	2
min_samples_leaf	一个叶节点所需的最小样本数	1
criterion	评估指标	gini
max_depth	决策树的最大深度	4

运行得到 MSE 值为 0.0018743434144322729，R^2 值为 0.27891037582084255。

5.3.2 使用 XGBoost 模型建立洪水发生概率分析预测模型

XGBoost 算法在 Boosting 框架下运行，其本质区别在于拟合残差树所需的增益不同，而 XGBoost 使用的增益是分割前后结构分数的差异。XGBoost 的一个重要特征是引入了一种新的分割标准，在最优分割点处最小化分割损失。

XGBoost 算法的核心思想分为三步。首先，采用特征分割方法不断添加树，每添加一棵树，实际上是学习一个新函数以拟合上次预测的残差。其次，完成训练并获得 k 棵树后，应该预测样本的得分。第三，样本的预测值是每棵树的对应得分的总和。经过 m 次迭代后，XGBoost 模型的目标函数定义如下公式(19)所示：

(19)

在公式(13)中，l 和 Ω 分别是损失函数和正则项。yi 和 分别表示实际值和模型预测值，样本数量为 n。

XGBoost 模型的参数通过网格搜索方法结合五折交叉验证进行优化，得到最佳参数组合，如表 3 所示。

表 3 XGBoost 模型优化参数表

参数	含义	数值
estimator	最大迭代次数	150
learning_rate	学习率	0.2
min_child_sample	叶节点上的最小数据量	12
gamma	最小增益	0.5
subsample	样本百分比	0.9
colsample	特征采样比例	0.9
max_depth	树的最大深度	3

运行得到 MSE 值为 0.0017364060838535864，R^2值为 0.3319770535178902。

5.3.3 使用 LightGBM 模型建立洪水发生概率分析预测模型

LightGBM 旨在解决 GBDT 在处理大规模数据时面临的挑战，使 GBDT 能够更好、更快地应用于工业实践。lightGBM 通过引入直方图算法并采用受限的分叶策略，克服了 XGBoost 算法内存消耗大、训练时间长的缺点[8]。

使用直方图来找到最佳分割点，处理连续变量，减少特征中的特征值数量，并减少叶节点分割时需要处理的特征值的数量。基本思想是三个步骤。首先，将连续的浮点特征值离散为 k 个整数，并构造宽度为 k 的直方图。

当遍历数据时，将离散化的值作为指数并累积在直方图中。在此基础上进行遍历，找到最优分割点。XGBoost 算法采用 Level-wise 作为增长策略，如图该策略对数据进行一次遍历，可以同时拆分同一层的叶子，有利于控制模型的复杂度，达到控制拟合的效果。然而，在实际应用中，大多数叶子的分裂增益相对较小， 因此不需要搜索和分裂叶子，从而避免了不必要的计算。

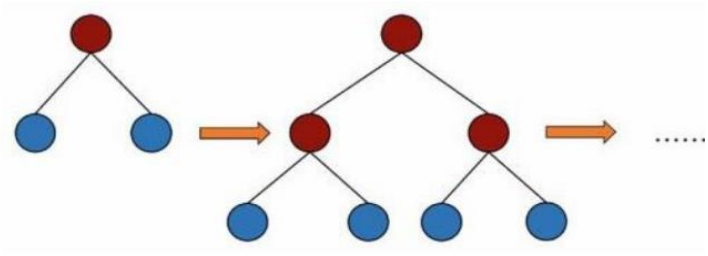

图 9 水平树生长图

LightGBM 算法使用 Leaf-wise 作为增长策略。如图 10 所示，每次从所有当前叶中进行逐叶分割，找到具有最大分割增益的叶，并重复该过程。与分级算法相比，叶式算法具有以下优点。在相同分割次数的情况下，逐叶算法可以有效地减少误差，提高算法的精度。但 Leaf-wise 的缺点是，它会形成一个更深的决策树，导致过度拟合。因此，LightGBM 算法在 Leaf-wise 中增加了最大深度限制，以避免过拟合并提高计算效率。

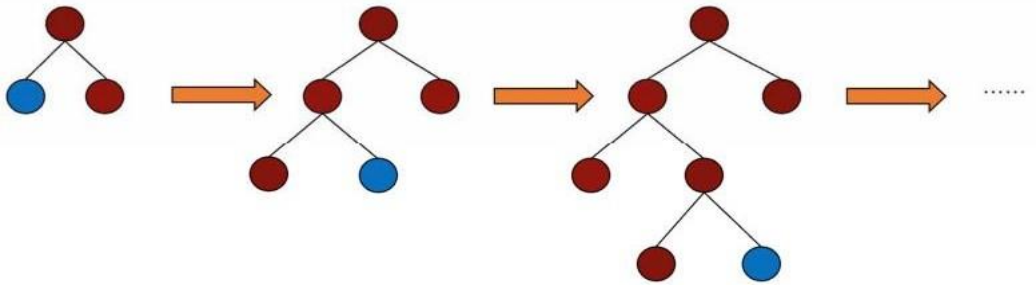

图 10 叶分裂构成决策树图

由于 LightGBM 通过直方图算法和带深度限制的逐叶策略改进了 XGBoost，除了一些参数设置外，两者相似。通过优化得到的最优参数组合如表 4 所示，其中参数 num_leaves 反映了模型中树木的复杂性，是一个唯一的参数。

表 4 LightGBM 模型优化参数表

参数	含义	数值
estimator	最大迭代次数	195
learning_rate	学习率	0.1
max_depth	树的深度	5
min_child_sample	叶节点上的最小数据量	20
num_leaves	树的复杂性	26
subsample	样本百分比	0.8
colsample	变量的样本比例	0.7

运行得到 MSE 值为 0.0017265632348001103，R^2 值为 0.3357637535805216。

5.3.4 对比三种模型预测模型选择出最佳模型

表 5 三种预测模型对比

模型	MSE	R^2
随机森林	0.0018743434144322729	0.27891037582084255
XGBoost 模型	0.0017364060838535864	0.3319770535178902
LightGBM 模型	0.0017265632348001103	0.3357637535805216

将三个模型的 MSE 得分和 R^2值汇总，可以得出 LightGBM 模型的 MSE 值最小，

模型预测的误差最小。并且 LightGBM 模型的 R^2值最大，约为 0.336，由此可以得出 LightGBM 模型为拟合度最高的模型，最后将洪水发生概率的预测模型确定为基于 LightGBM 模型的多指标预测模型。

5.3.5 简化指标后的多层感知器模型

将模型简化为五个关键指标的预测模型。由于指标数量较少且问题较为简单，我们最初考虑选择一种单一模型。然而，为了确保预测的准确性，我们决定使用一种高级的单一模型——多层感知器。[9]

多层感知器（Multilayer Perceptron，简称 MLP）是一种前馈神经网络，具有多个隐藏层，能够处理非线性问题。

输入层：接受特征输入，记为 。

隐藏层：通过权重矩阵 和偏置向量 对输入进行线性变换：

通过非线性激活函数 Sigmoid）对线性变换结果进行非线性变换，得到隐藏层输出：

其中， 表示激活函数。

输出层：通过权重矩阵 和偏置向量 对隐藏层输出进行线性变换：

通过激活函数对线性变换结果进行非线性变换，得到输出层结果 ：

利用损失函数来衡量模型预测输出与真实标签之间的差距：
计算交叉熵损失（Cross-Entropy Loss）：

(20)

其中， 表示模型参数，包括权重矩阵 和偏置向量 ， 表示第 个样本的真实标签， 表示第 个样本的预测输出， 表示样本数量。

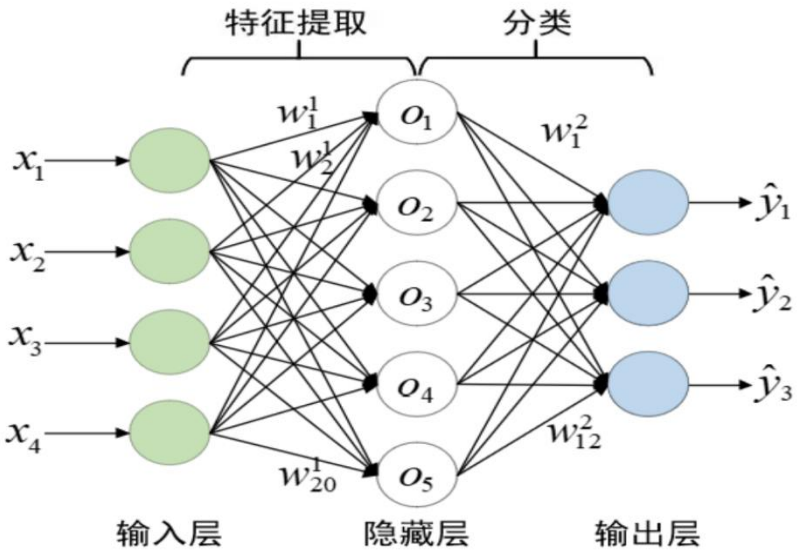

图 11 多层感知器原理图

代入具体数据，得到：

均方误差 (MSE): 0.002131444844590042

R^2: 0.17999937999111404

MSE 值相对较好，达到在指标变量减少情况下比较好的预测结果。

5.4 预测洪水发生概率

选取第三问得到效果最好的 LightGBM 洪水发生概率的预测模型，将 train.csv 数据集整体作为训练集，将 test.csv 数据集整体作为预测集。同时选取第一问得到的季风强度、河流管理、气候变化、大坝质量、淤积、农业实践、流域、规划不足、政策因素这九个相关性程度高的值作为自变量，输入 LightGBM 模型对洪水概率进行预测。

将数据保存在 submit.csv 文件中，对得到的洪水概率进行可视化，绘制直方图和正态分布曲线：

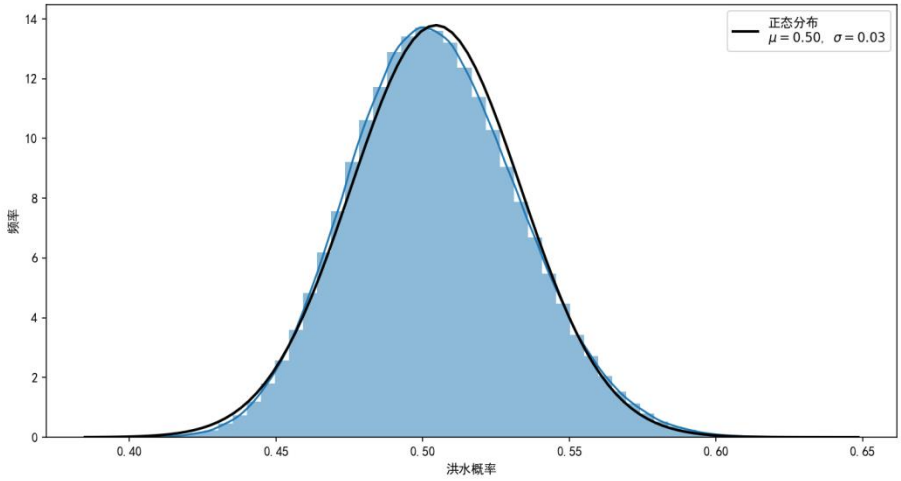

图 12 洪水概率的直方图和正态分布曲线

从图上明显看出，得到的概率与正态分布曲线拟合良好，整体上服从正态分布。

由于数据量大，将数据采样和移动平均结合起来，绘制洪水概率的采样移动平

均折线图和散点折线图。同时绘制箱线图、小提琴图、核密度估计图、累积分布函数图如图 13。

其中，箱线图显示出数据的分布和异常值。小提琴图结合了箱线图和密度图，用于显示数据的分布形状和变异情况，核密度估计图用于平滑显示数据的概率密度函数，累积分布函数图用于显示数据的累计分布情况。

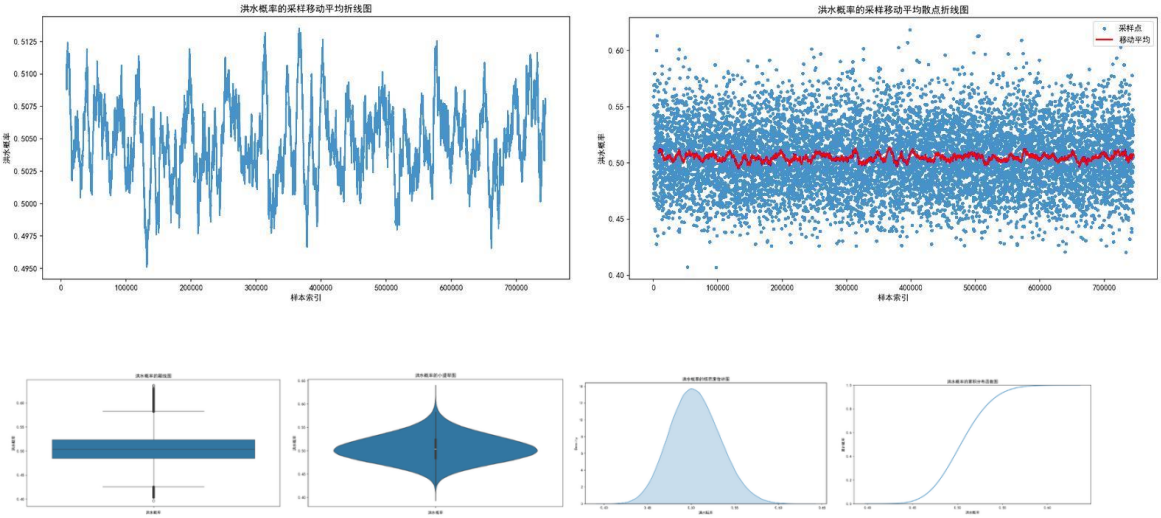

图 13 洪水概率的各类图像

六、 模型的评价、改进与推广

6.1 模型的优点

本次洪水发生概率预测模型在多个方面表现出了显著的优势。首先，模型的预测精度较高。通过使用 LightGBM 算法，并结合交叉验证和网格搜索优化模型参数[10]，最终模型在训练集和测试集上的均方误差（MSE）和决定系数（R^2）均表现优异，确保了预测结果的可靠性。其次，模型具有较强的可解释性。通过灰色关联模型和互信息模型，我们能够识别出影响洪水发生概率的关键因素，并进行权重计算和排序，为后续的防灾减灾工作提供了科学依据。此外，模型的计算效率高，LightGBM 采用的直方图算法和叶节点分裂策略，使得模型在处理大规模数据时依然保持高效的计算性能，适合实际应用中的洪水预警系统。

6.2 模型的缺点

尽管模型在多个方面表现良好，但仍存在一些不足之处。首先，模型对数据的依赖性较强。训练和预测过程中依赖于大量高质量的历史数据，如果数据存在缺失或质量问题，模型的预测准确性将受到影响。其次，特征工程的过程较为复杂。在数据预处理中，我们需要进行大量的特征处理工作，如数据标准化、异常值处理等，这不仅增加了工作量，还需要具备一定的专业知识和经验。最后，模型的复杂度较高，训练和优化过程需要较长时间，可能不适用于资源有限的场景。

6.3 模型的改进

针对模型的不足，我们提出了以下改进方案。首先，通过引入更多高质量的数据源，如气象卫星数据和实时监测数据，可以提升模型的预测精度。其次，采用自动化特征工程工具，如 AutoML，可以减少人工特征工程的工作量，并提升模型开发效率。

此外，尝试多模型融合的方法，如堆叠（stacking）或集成学习（ensemble learning），结合多种模型的优点，进一步提高预测性能和稳定性。最后，开发实时数据更新和模型训练系统，使模型能够持续学习最新数据，保持预测的准确性和时效性。

6.4 模型的推广

在推广方面，洪水发生概率预测模型具有广泛的应用前景。首先，模型可以推广到其他有洪水风险的区域，通过对不同区域的数据进行训练和验证，建立适应性更强的区域性洪水预测模型。其次，模型可以扩展至多灾害预测系统，如泥石流、滑坡等，自然灾害的综合预警能力将显著增强。此外，将模型集成到智能预警系统中，结合物联网（IoT）技术，实现对洪水风险的实时监测和预警，将大幅提升防灾减灾的响应速度和效果。最后，加强与政府部门的合作，推动模型在政策制定和防灾减灾中的应用，通过公众教育提高居民对洪水风险的认知和防范意识，形成全民防灾的良好氛围。

七、 参考文献

[1] 刘瑞. 基于贝叶斯网络的洪水灾害风险评估与建模研究[D].华东师范大学,2016.

[2] 黄华兵,王先伟,柳林.城市暴雨内涝综述：特征、机理、数据与方法[J].地理科学进展,2021,40(06):1048-1059.

[3] 刘思峰,蔡华,杨英杰,等.灰色关联分析模型研究进展[J].系统工程理论与实践,2013,33(08):2041-2046.

[4] 李琼. 洪水灾害风险分析与评价方法的研究及改进[D].华中科技大学,2012.

[5] 孙吉贵,刘杰,赵连宇.聚类算法研究[J].软件学报,2008(01):48-61.

[6] 徐峻岭,周毓明,陈林,等.基于互信息的无监督特征选择[J].计算机研究与发展,2012,49(02):372-382.

[7] 方匡南,吴见彬,朱建平,等.随机森林方法研究综述[J].统计与信息论坛,2011,26(03):32-38.

[8] 沙靖岚. 基于 LightGBM 与 XGBoost 算法的 P2P 网络借贷违约预测模型的比较研究[D].东北财经大学,2018.

[9] 王之仓. 多层感知器学习算法研究[D].苏州大学,2008.

[10] 李琼. 洪水灾害风险分析与评价方法的研究及改进[D].华中科技大学,2012.

附录

附录 1

问题 1 程序 1 灰色关联分析.py <python>

```
import pandas as pd
import numpy as np
from sklearn.preprocessing import MinMaxScaler
from sklearn.feature_selection import mutual_info_classif

def read_data(file_path, encoding='gbk'):
    try:
        data = pd.read_csv(file_path, encoding=encoding)
        return data
    except FileNotFoundError:
        return None
    except Exception as e:
        return None

def normalize_data(data):
    data = data.drop(columns=['id'], errors='ignore')
    numeric_data = data.select_dtypes(include=[np.number])
    data_normalized = (numeric_data - numeric_data.min()) /
(numeric_data.max() - numeric_data.min())
    return data_normalized

def grey_relational_coefficient(reference_series,
comparison_series):
    diff_series = np.abs(reference_series - comparison_series)
    min_diff = diff_series.min().min()
    max_diff = diff_series.max().max()
    rho = 0.5
    return (min_diff + rho * max_diff) / (diff_series + rho *
max_diff)

def grey_relational_degree(data_normalized, reference_column):
    reference_series = data_normalized[reference_column]
    grey_rel_matrix = data_normalized.apply(lambda x:
grey_relational_coefficient(reference_series, x))
    return grey_rel_matrix.mean()

def main():
    file_path = train.csv'
    data = read_data(file_path)
```

```
    if data is not None:
        data_normalized = normalize_data(data)
        grey_rel_degree = grey_relational_degree(data_normalized, '洪水概率')
        correlation_matrix = pd.DataFrame(grey_rel_degree, columns=['灰色关联度'])
        print(correlation_matrix)

if __name__ == "__main__":
    main()
```

附录 2

问题 2 程序 2　K-means 结合互信息.py　<python>

```
import pandas as pd
import numpy as np
from sklearn.cluster import KMeans
from sklearn.preprocessing import MinMaxScaler
from sklearn.feature_selection import mutual_info_classif

file_path = r'D:\\train.csv'
data = pd.read_csv(file_path, encoding='gbk')

scaler = MinMaxScaler()
data_normalized = pd.DataFrame(scaler.fit_transform(data),
columns=data.columns)

n_clusters = 3

kmeans = KMeans(n_clusters=n_clusters, random_state=42)
data['风险类别'] = kmeans.fit_predict(data[['洪水概率']])

cluster_centers = kmeans.cluster_centers_
print("聚类中心：", cluster_centers)

grouped_data = data.groupby('风险类别').mean()
print(grouped_data)

X = data.drop(['风险类别', '洪水概率'], axis=1)
y = data['风险类别']
mutual_info = mutual_info_classif(X, y, discrete_features='auto')
print("各指标的互信息值：", mutual_info)

weights = mutual_info / np.sum(mutual_info)
```

```
print("各指标权重：", weights)

def risk_evaluation_model(data, weights):
    risk_scores = np.dot(data, weights)
  return risk_scores

risk_scores = risk_evaluation_model(X, weights)
data['风险得分'] = risk_scores

def sensitivity_analysis(data, weights, delta=0.1):
sensitivities = []
for i in range(len(weights)):
    perturbed_weights = weights.copy()
    perturbed_weights[i] += delta
    perturbed_weights /= np.sum(perturbed_weights)
    perturbed_scores = risk_evaluation_model(data,
perturbed_weights)
    sensitivity = np.mean(np.abs(perturbed_scores - risk_scores))
        sensitivities.append(sensitivity)
return sensitivities

sensitivities = sensitivity_analysis(X, weights)
print("灵敏度分析结果：", sensitivities)
```

附录 3

问题 3 程序 3　LightGBM.py　<python>

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import mean_squared_error, r2_score
import lightgbm as lgb

file_path = r'D:\\第二问.csv'

data = pd.read_csv(file_path, delimiter=',', encoding='gbk')

data.columns = data.columns.str.strip()

X = data.drop('洪水概率', axis=1)
y = data['洪水概率']

X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)
```

```
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

model = lgb.LGBMRegressor(n_estimators=100, random_state=42)
model.fit(X_train_scaled, y_train)

y_pred = model.predict(X_test_scaled)

mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print(f'Mean Squared Error: {mse}')
print(f'R2 Score: {r2}')
```

附录 4

问题 3 程序 4　多层感知器.py　<python>

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import mean_squared_error, r2_score
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense

data =
pd.read_csv(r'D:\Normal_tools\Github_desktop\Clone_shop\Mathematical
-Modeling\比赛记录\2024 亚太中文\五个变量的数据.csv', encoding='gbk')

X = data[['河流管理', '大坝质量', '淤积', '农业实践', '政策因素']]
y = data['洪水概率']

X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)

scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

model = Sequential()
```

```
model.add(Dense(64, input_dim=X_train_scaled.shape[1],
activation='relu'))
model.add(Dense(32, activation='relu'))
model.add(Dense(1))

model.compile(optimizer='adam', loss='mean_squared_error')

history = model.fit(X_train_scaled, y_train, epochs=100,
batch_size=32, validation_split=0.2, verbose=1)

y_pred = model.predict(X_test_scaled)

mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print(f'均方误差 (MSE): {mse}')
print(f'R²得分: {r2}')
print(y_pred[:5])
```