****  ****



**中国研究生创新实践系列大赛**

**“华为杯”第二十一届中国研究生**

**数学建模竞赛**

|  |  |
| --- | --- |
| **学 校** |  |
| **参赛队号** |  |
| **队员姓名** | **1.** |
| **2.** |
| **3.** |

大数据驱动的地理综合问题

**摘要**

地理系统是由自然和人文等多种要素共同作用的复杂巨系统。地理学研究中，学者们常通过地理综合的方法对地理系统的主导特征进行表达与分析。例如，通过三大阶梯的概括揭示中国地形的总体特征，借助秦岭—淮河一线以及其他地理区划的方式来说明中国在水平和垂直方向上气温、降水、植被、土壤及生态环境的地带性和非地带性规律。此外，利用胡焕庸线和T型开发结构等手段勾勒出我国人口、社会和经济发展的总体空间格局。这些方法在早期研究中以宏观结构和定性分析为主，为我国生态保护、社会经济发展及国家安全战略提供了重要支持

对于问题一，首先分析降水，使用马氏距离法预处理数据，本题旨在通过大数据驱动的分析手段，降水量作为具有连续变化特征的变量，需要利用平均降水量、极值和变异系数等统计指标，分析1990-2020年间中国范围内的降水时空演化特征，并通过折线图、热力图等方式可视化其时空分布变化。同时，借助地理信息系统（GIS）技术绘制等降水量线图，并且选取河北和广东一北一难着重分析，揭示不同区域的降水趋势。其次，土地利用/土地覆被类型作为离散变量，参赛者需统计其变化比例和变化率，结合GIS分析其时空演化特征，并通过面积图或饼图呈现不同覆被类型的变化情况。

对于问题二，首先要使用无监督聚类方法（如K-means或层次聚类）对洪水发生概率进行分类，分为高、中、低三种风险等级，并通过频率分布直方图可视化分析各类别的特征，找出关键影响因素。其次，根据这些指标的重要性确定权重，并利用梯度提升决策树模型来建立一个洪水风险预警系统，通过交叉验证和数据集分割来优化和评估模型的表现。

对于问题三，TODO

对于问题四，TODO

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、决策树回归模型对地理数据进行了深入分析，建立了有效的暴雨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。并且对于中国土地利用提出了有力的建议。

**关键词：地理系统；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**

地理系统是由自然和人文等多种要素共同作用的复杂巨系统。地理学研究中，学者们常通过地理综合的方法对地理系统的主导特征进行表达与分析。例如，通过三大阶梯的概括揭示中国地形的总体特征，借助秦岭—淮河一线以及其他地理区划的方式来说明中国在水平和垂直方向上气温、降水、植被、土壤及生态环境的地带性和非地带性规律。此外，利用胡焕庸线和T型开发结构等手段勾勒出我国人口、社会和经济发展的总体空间格局。这些方法在早期研究中以宏观结构和定性分析为主，为我国生态保护、社会经济发展及国家安全战略提供了重要支持。

随着对地观测技术的快速进展，当前已经积累了海量的对地观测数据。如何利用先进的技术手段对已有的数据分析和建模，用以分析复杂的地理系统，成为了当前地球科学研究的核心议题之一。这也为数学建模和数据分析在探讨全球气候变化背景下中国地理环境的演变趋势提供了广阔的舞台。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分析预测各种地理环境变量对极端天气的影响成为一种可行的手段。

本文运用大数据分析技术，全面剖析了中国在全球气候变化背景下的地理环境演变趋势。通过对海量数据的清洗与量化分析，揭示了降水量、土地利用/覆被类型等关键要素在中国境内的时空变化特征。进一步，采用逻辑回归与随机森林算法，深入探讨了地形与气候相互作用对极端天气（尤其是暴雨）形成的贡献，精准定位了国内暴雨灾害防御的薄弱环节。研究还评估了降水量、人口、GDP等多维度因素对土地利用变化的综合效应，为国土空间规划与管理提供了科学依据。最终，本文的工作不仅加深了对中国地理环境动态变化的理解，也为制定有效防灾减灾策略、提升灾害应对能力奠定了坚实基础，推动了灾害管理向智能化、精准化方向迈进。

* 1. **问题重述**

问题 1. 在众多描述地理环境的变量中，一些简单的指标背后蕴藏了深厚的内涵，对人类的生存发展具有重大深远的影响，如大气中二氧化碳的浓度、全球年平均气温等。降水量是一个连续变化的变量，而土地利用/土地覆被类型则是一个存在突变和离散分布的变量。同时，它们都具有时空分布不均匀的特征。请从附件数据中选取相关数据集，为这两个变量分别构建一套描述性统计方法，用1~3个较为简洁的统计指标或统计图表，对这两个变量在1990~2020年间中国范围内的时空演化特征进行描述和总结。

问题 2. 近年来，以暴雨为代表的极端天气事件对人类的生产生活造成了越来越难以忽视的影响。请结合附件中所给的数据，建立数学模型，说明地形-气候相互作用在极端天气形成过程中的作用。

问题 3. 降雨、地形和土地利用对于暴雨等极端天气灾害的形成都具有不可忽视的影响。这其中，降雨的时空变异性和不可控性都最强；土地利用作为自然条件和人类活动的综合结果，虽然也随时空演化，但具有一定可控性；地形是最为稳定、不易改变的因素。请考虑第2问所反映的从“暴雨”到“灾害”中上述三方面因素的角色及其交互作用，确定暴雨成灾的临界条件；并结合第1问中降雨量和土地利用/土地覆被变化的历史时空演化特征，对2025~2035年间中国境内应对暴雨灾害能力最为脆弱的地区进行预测。请以地图的形式呈现你们的预测结果。

问题 4. 在中国级别的尺度上，描述自然地理特征的地形可以概括为“三级阶梯”，而降水中具有标志性意义的“800mm等降水量线”则与区分我国南北方的“秦岭—淮河”一线大体重合；描述人文地理特征的人口分布及其社会经济活动总量等指标，则被由连接黑龙江黑河与云南腾冲的“胡焕庸线”清晰地划分成东密西疏的两部分。那么，对于自然地理和人文地理交汇点的土地利用/土地覆被情况，结合其在前三问中描述、估计和预测任务中的“特性”，利用地理大数据，建立相应的数学模型，对数据进行简化和综合，描述中国土地利用变化的特征与结构。从准确性和有用性两个方面解释验证你们的总结。

# 问题分析

**2.1 问题一分析**

本问题旨在通过描述性统计方法，对1990年至2020年间中国范围内的降水量和土地利用的时空演化特征进行系统描述与总结。具体目标包括：

降水量：分析年平均降水量的变化趋势。衡量降水量的年度波动性。描述降水量的空间分布特征及其演变。土地利用：计算各类土地利用类型的面积比例及其变化趋势。分析不同土地利用类型的年度变化率。描述土地利用的空间分布特征及其动态变化。

**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 TODO**

**2.4 问题四分析 TODO**

# 模型假设

在进行降水量和土地利用的描述性统计分析过程中，为确保分析结果的科学性和可靠性，基于以下假设：

1. **数据完整性假设**：所选取的降水量和土地利用数据在1990-2020年间无显著缺失，且经过必要的预处理（如插值、平滑）以保证数据质量。
2. **时间稳定性假设**：在研究期间内，降水量和土地利用数据的采集方法和标准保持一致，确保数据的可比性。
3. **空间一致性假设**：不同数据集的空间分辨率和坐标系经过统一处理，确保空间分析的准确性。
4. **变量独立性假设**：在描述性统计分析中，降水量与土地利用作为独立变量进行分析，尽管实际中两者可能存在一定的相互影响。
5. **代表性假设**：所选取的数据集能够全面、准确地反映中国地理环境在1990-2020年间的时空演化特征。

四、符号说明

|  |  |
| --- | --- |
| **符号** | **说明** |
|  | 坐标i，j在t时刻的降水量 |
|  | 第类土地利用类型在第 j 年的面积比例，范围：[ 0 ， 1 ] |
|  | 第 j 年的全国平均降水量，单位：毫米 |
|  | 第 j 年降水量的标准差，单位：毫米 |
|  | 第类土地利用类型在第 j 年的总面积，单位：平方公里 |
|  | 第左类土地利用类型在第 j 年的年度变化率，单位：％ |
|  | 第 j 年降水量的空间分布特征（通过空间热力图表示） |
|  | 第 j 年土地利用类型的空间分布特征（通过地理分布图表示） |
|  | 地形高度 |
|  | 坡度 |
|  | 气温 |
|  | 相对湿度 |
|  | 风速 |
|  | 风向 |
|  | 坡向 |

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

降水量数据预处理：取1990-2020年间的降水量数据。处理缺失值和异常值，确保数据完整性。将0.25°×0.25°的降水量数据重采样至统一分辨率（如1°×1°或其他适当分辨率）。按年度对降水量数据进行归一化，确保不同年份数据的可比性。

土地利用数据预处理：提取1990-2020年间的土地利用数据。处理缺失值和分类错误，确保数据准确性。将0.5°×0.5°的土地利用数据重采样至统一分辨率。分类统一确保不同年份土地利用类型分类一致，便于时间序列分。

年最大降水量是找出每年中降水量最大的网格值，可用于分析极端天气事件，其计算公式如下：

年平均降水量是衡量某一年度内全国范围内平均降水水平的指标，计算公式如下：

其中：

为第j年的全国平均降水量

为全国范围内单元的总数

为第i个空间在第j年的降水量

降水量标准差：

其中：

为 年的降水量标准差

为第j年的全国平均降水量

为全国范围内单元的总数

为第i个空间在第j年的降水量

空间变异系数：

其中：

是在经纬度位置的降水量，和分别是经度和纬度的数量。

降水量的空间分布通过空间热力图进行可视化展示，表示公式如下：

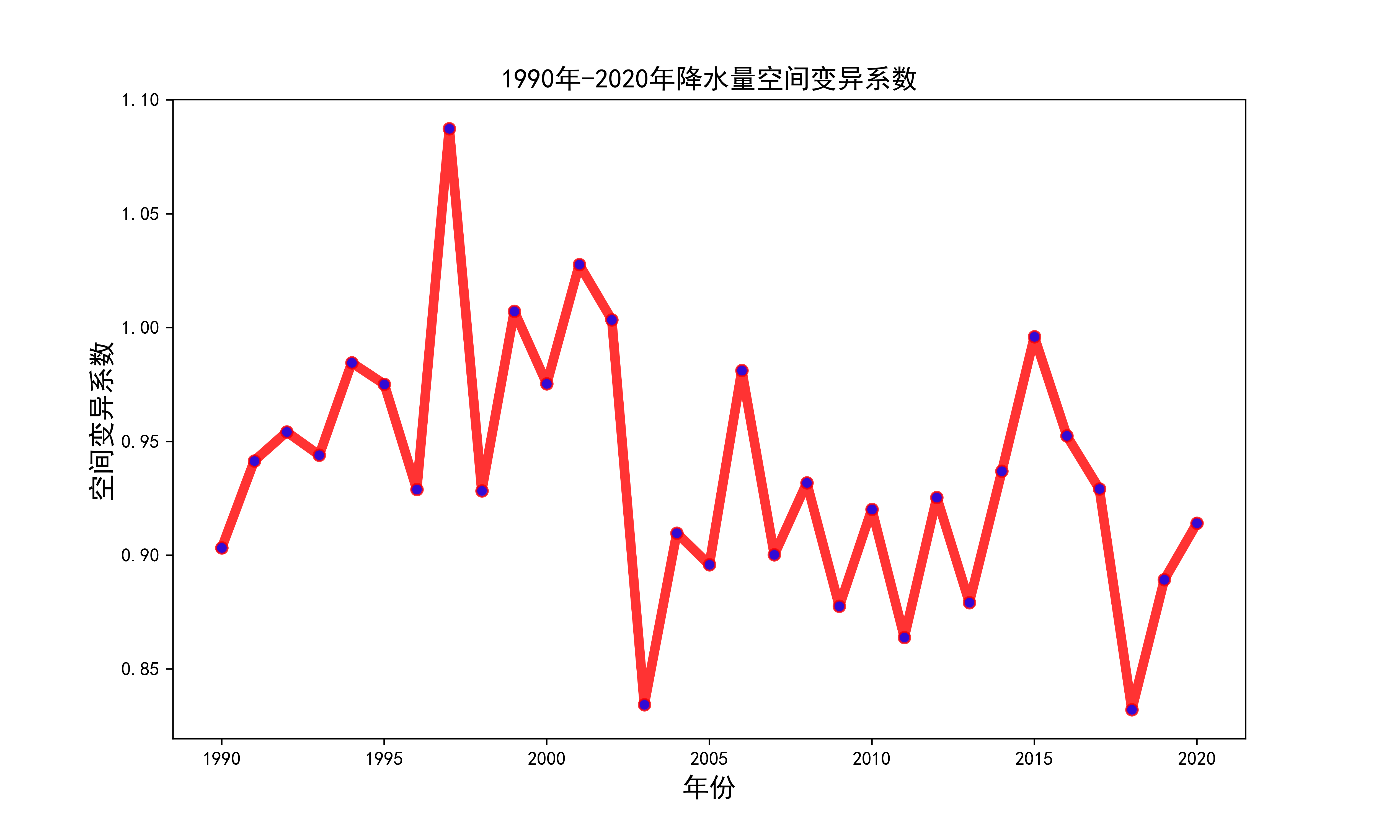
其中：

表示第 年降水量的空间分布热力图。

**5.1.2 降水量**

为了量化1990年至2020年间中国降水量的空间不均匀性，我们建立了降水量的空间变异系数模型。具体而言，空间变异系数 定义为当年降水量在所有空间单元中的标准差 与年平均降水量 的比值，并乘以100%以表示百分比，即：

其中， 反映了降水量在空间上的波动性，而 则表示全国范围内的平均降水量。通过计算每年的空间变异系数，我们能够量化降水分布的相对不均匀性，进而分析降水模式的变化趋势。这一指标有助于识别降水分布的集中或分散趋势，为理解气候变化对降水分布的影响提供了定量依据。同时，结合年平均降水量、年总降水量及其标准差，空间变异系数为降水量的时空演化特征提供了全面的描述，支持进一步的生态保护、资源管理和政策制定。



图：1990年-2020年降水量空间变异系数

CV 大 (通常超过 50%)：表示降水量在不同空间点之间的变化非常显著。不同地区的降水量差异较大。

CV 小 (通常低于 20%)：表示降水量在不同空间点之间的变化较小，空间分布相对均匀。

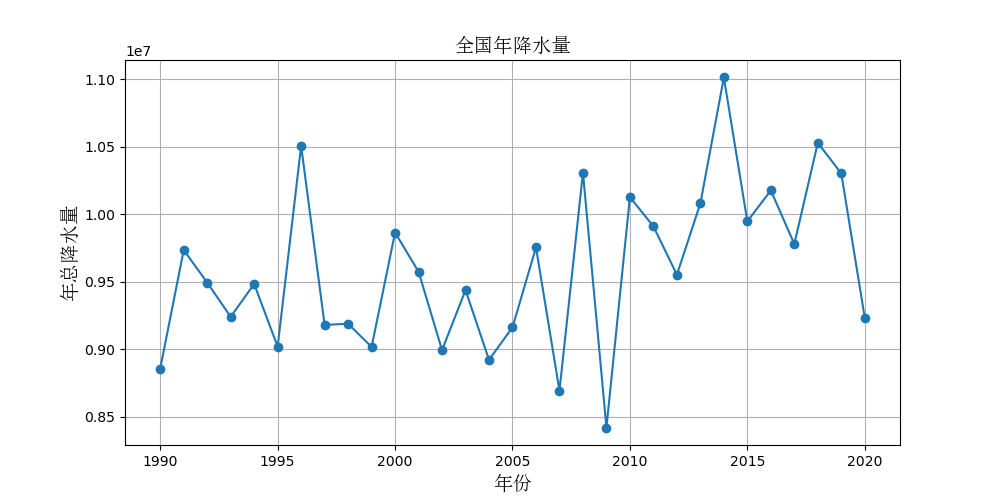


图1：全国全年总降水量

年总降水量是衡量某一年度内全国范围内总降水水平的指标，计算公式如下：

其中：

为第j年的全国平均降水量

为全国范围内单元的总数

为第i个空间在第j年的降水量

年最大降水量是找出每年中降水量最大的网格值，可用于分析极端天气事件，其计算公式如下：

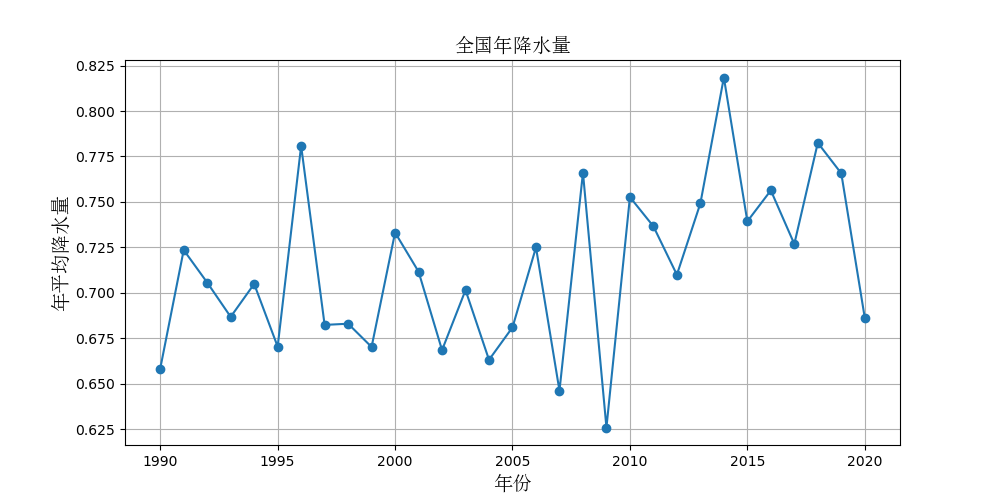
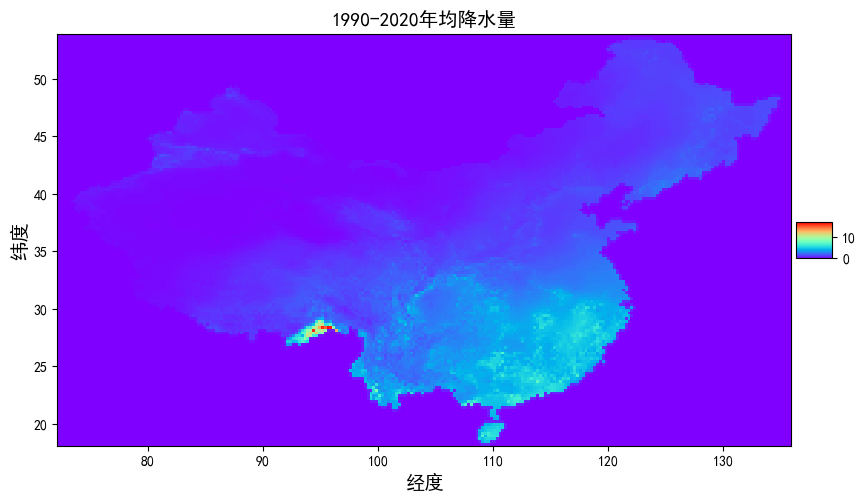
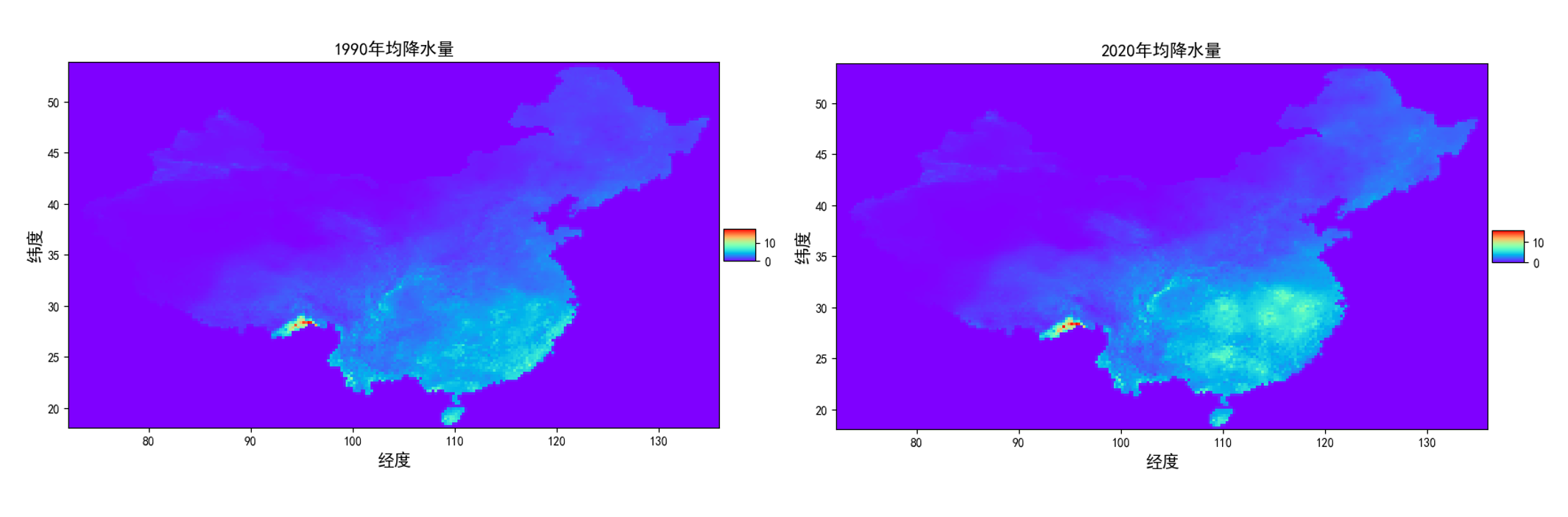


图2：全国年均降水量

GIS生成的热力图，通过颜色深浅表示降水量的高低。直观展示降水量在不同地区的分布差异。分析高降水和低降水区域的空间变化情况。



图：1990-2020年总均降水量



图：1990年和2020年均降水量

年平均降水量是衡量某一年度内全国范围内平均降水水平的指标，计算公式如下：

其中：

为第j年的全国平均降水量

为全国范围内单元的总数

为第i个空间在第j年的降水量

图：1990年-2020年降水量标准差

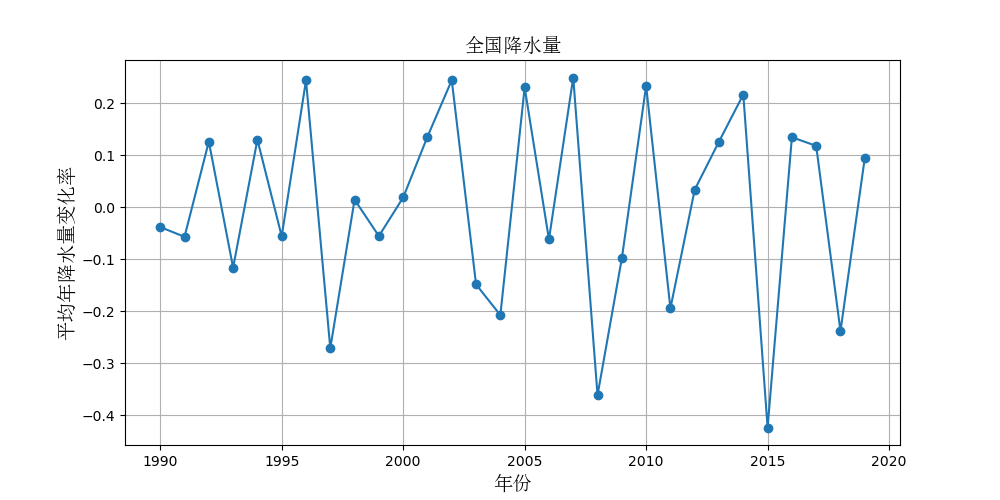


图3：全国全年平均降水年变化率



图4：全国年降水量标准差和最大值

全国年降水量的标准差公式可以按照以下方式计算：

则全国年降水量的标准差  计算公式为：



其中：

 表示第  个地区的年降水量。

 表示全国所有地区的年降水量的平均值。

 是全国地区的总数。

这个公式反映了每个地区的降水量相对于全国平均值的离散程度。

年最大降水量是找出每年中降水量最大的网格值，可用于分析极端天气事件，其计算公式如下：

然后我们在那河北和广东省进行分析，一个选择我国北方，一个选择我国南方。

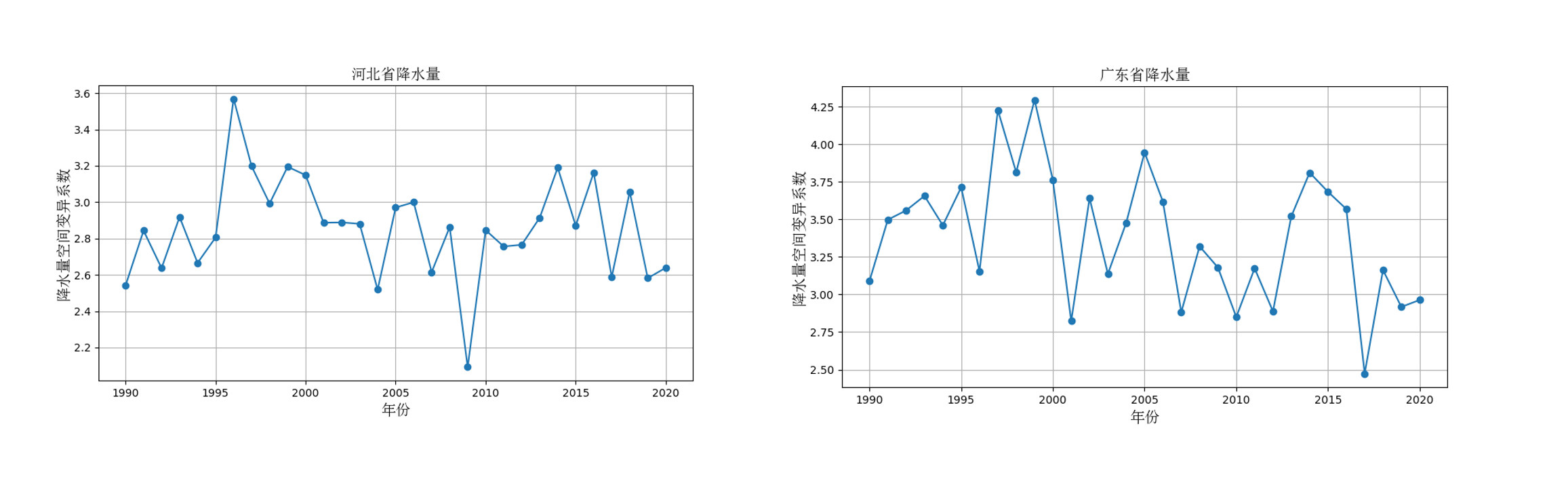


图6 河北-广东降水量空间变异系数

中间值：表示降水量在空间上有一定程度的变化，但不至于过大

时间序列图内容：展示1990-2020年间全国年平均降水量 的变化趋势。

形式：折线图，辅以趋势线，以便直观展示降水量的上升或下降趋势。

空间热力图

内容：选择若干典型年份（如1990年、2005年、2020年），绘制降水量空间分布热力图。

形式：GIS生成的热力图，通过颜色深浅表示降水量的高低。

分析：

直观展示降水量在不同地区的分布差异。

分析高降水和低降水区域的空间变化情况。

年度变化箱线图

内容：展示各年降水量 P(i,j,t) 的分布情况，包括中位数、四分位数及异常值。

形式：箱线图。

分析：

评估降水量分布的集中趋势和离散程度。

识别降水量分布中的异常值年份。

**5.1.3 土地**

为了系统地分析1990年至2020年间中国土地利用/土地覆被类型（以下简称“土地利用”）的时空演化特征，我们建立了以下统计模型。首先，应用马氏距离法对土地利用数据进行预处理，以识别和剔除数据中的异常值，确保数据质量和分析的可靠性。接着，采用相关性分析和主成分分析（PCA）方法，识别影响土地利用变化的关键因素。这些关键因素可能包括地形特征（如海拔、坡度）、气候变量（如降水量、温度）、人口密度及社会经济活动等。通过PCA，可以将高维数据降维，提取主要成分，从而简化模型并突出主要影响因素。

在模型构建阶段，我们采用多元线性回归模型来量化各关键因素对不同土地利用类型面积比例变化的影响。具体而言，设第类土地利用类型的面积比例为，其年度变化可表示为：

其中，为第年的关键影响因素，为回归系数，为误差项。通过最小二乘法估计回归系数，评估各因素对土地利用变化的显著性和影响程度。

此外，为了捕捉土地利用类型之间的相互关系和动态变化，我们引入时间序列分析，利用自回归移动平均（ARIMA）模型预测未来土地利用变化趋势。结合空间分析方法，如地理加权回归（GWR），可以进一步探讨不同地理区域内影响因素的异质性，揭示空间上土地利用变化的差异性。

最后，通过可视化手段（如堆积柱状图、地理分布图和变化率折线图），直观展示土地利用类型的面积比例变化、空间分布特征及其动态演变过程。模型结果表明，土地利用变化与地形特征、气候变量及社会经济活动密切相关，具体表现为与人口增长、经济发展、城市化进程等因素显著相关。通过该模型，我们能够全面理解和预测土地利用的时空演化规律，为生态保护、资源管理和城市规划提供科学依据。

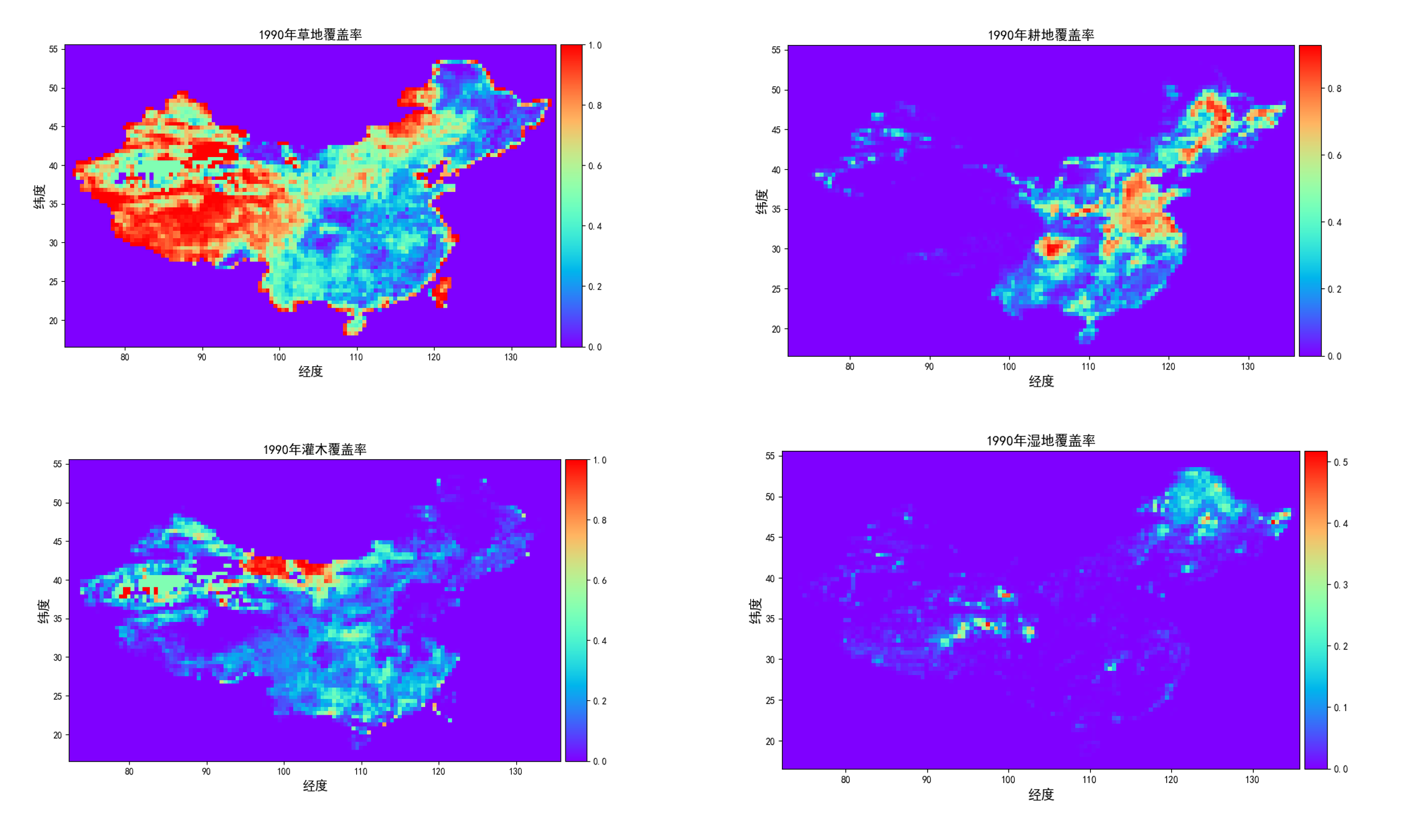
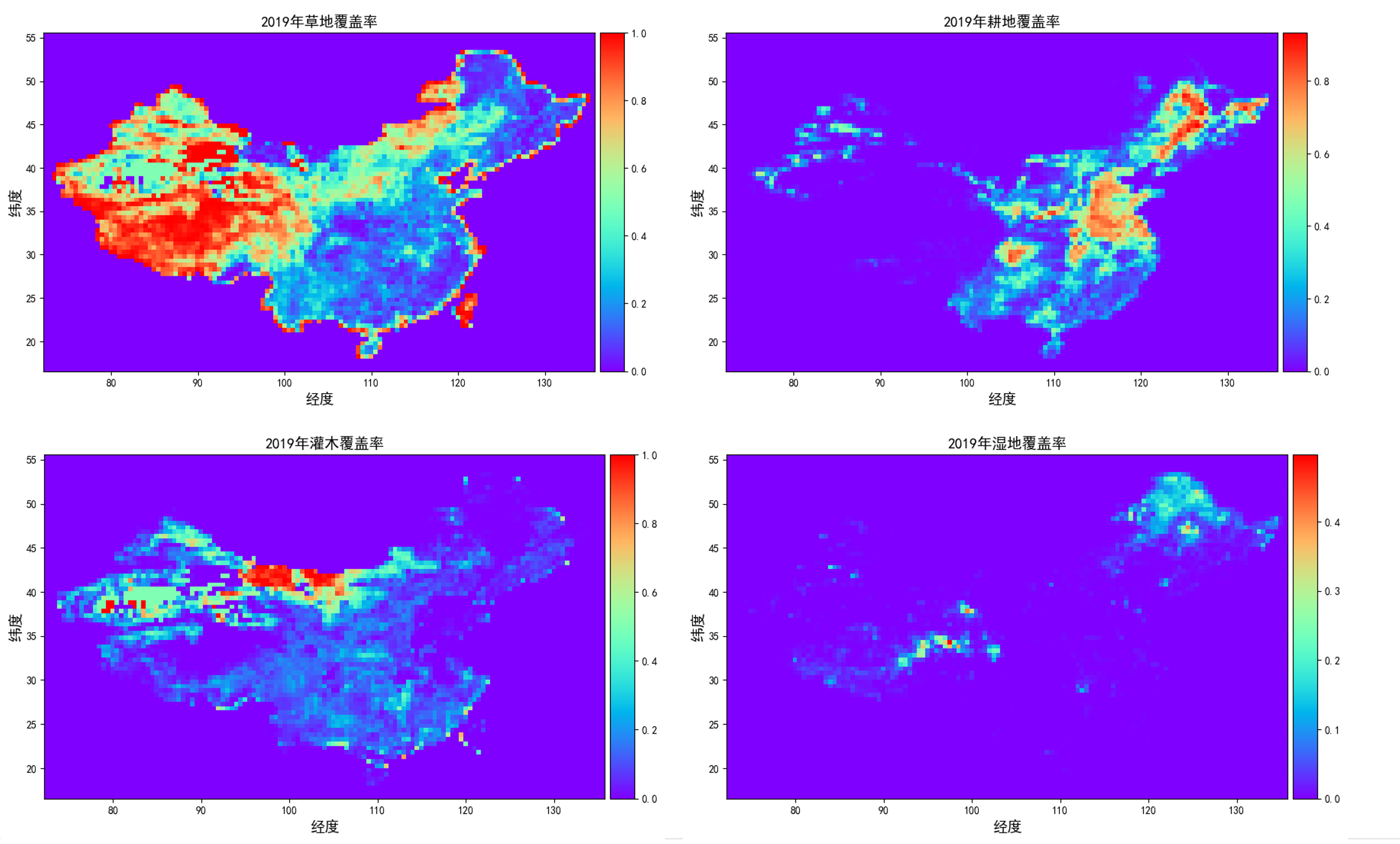


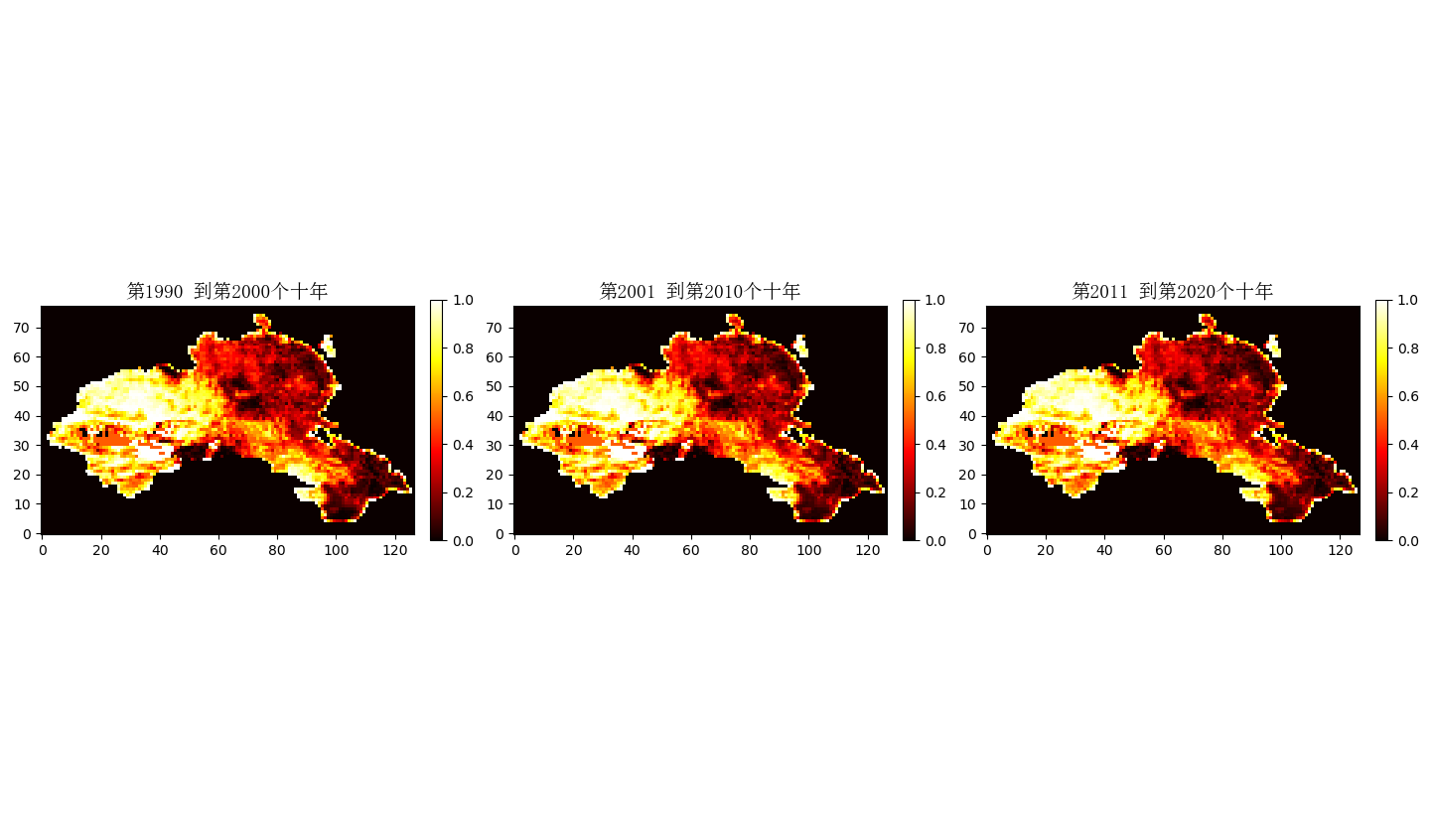
图1990年覆盖率



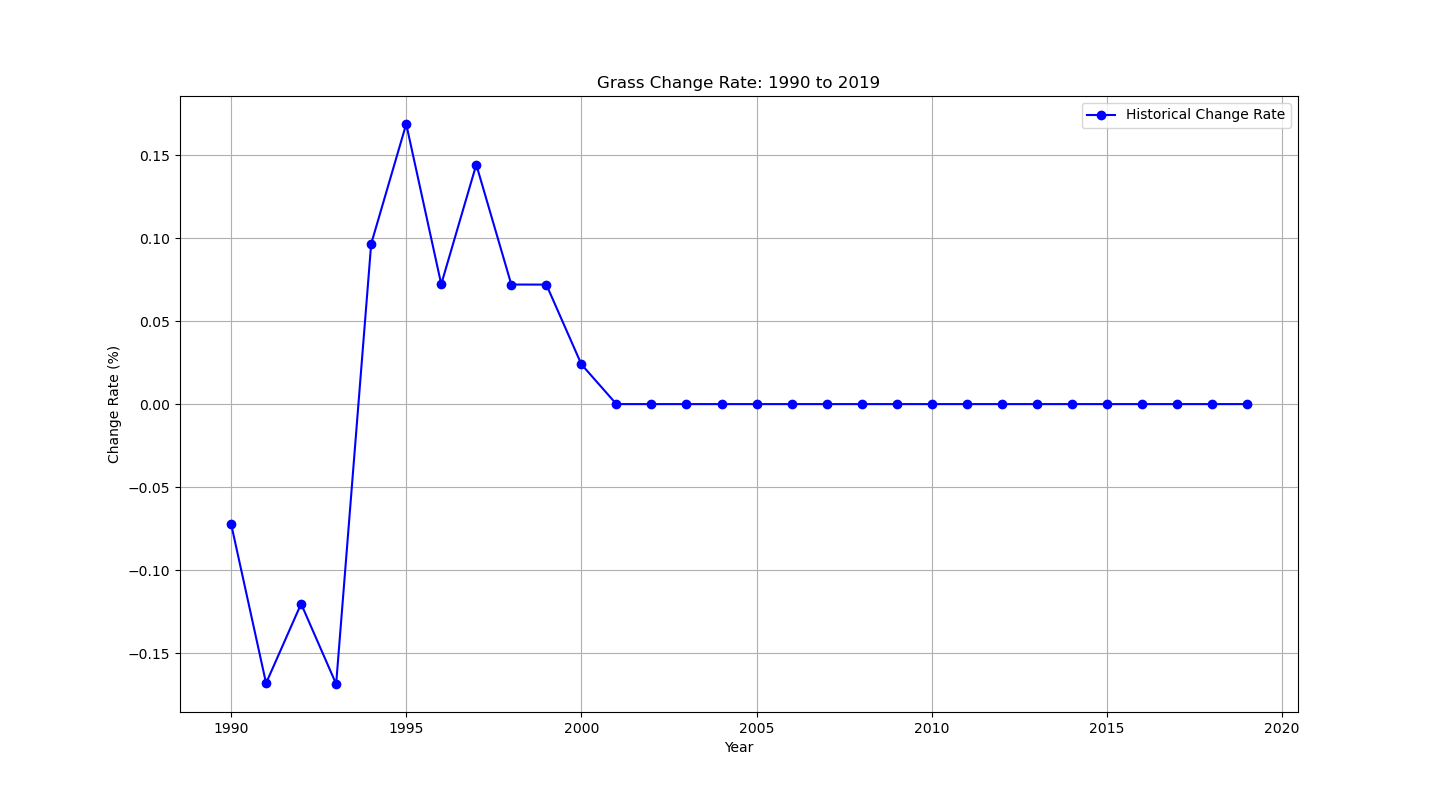
图：2019年覆盖率

堆积柱状图

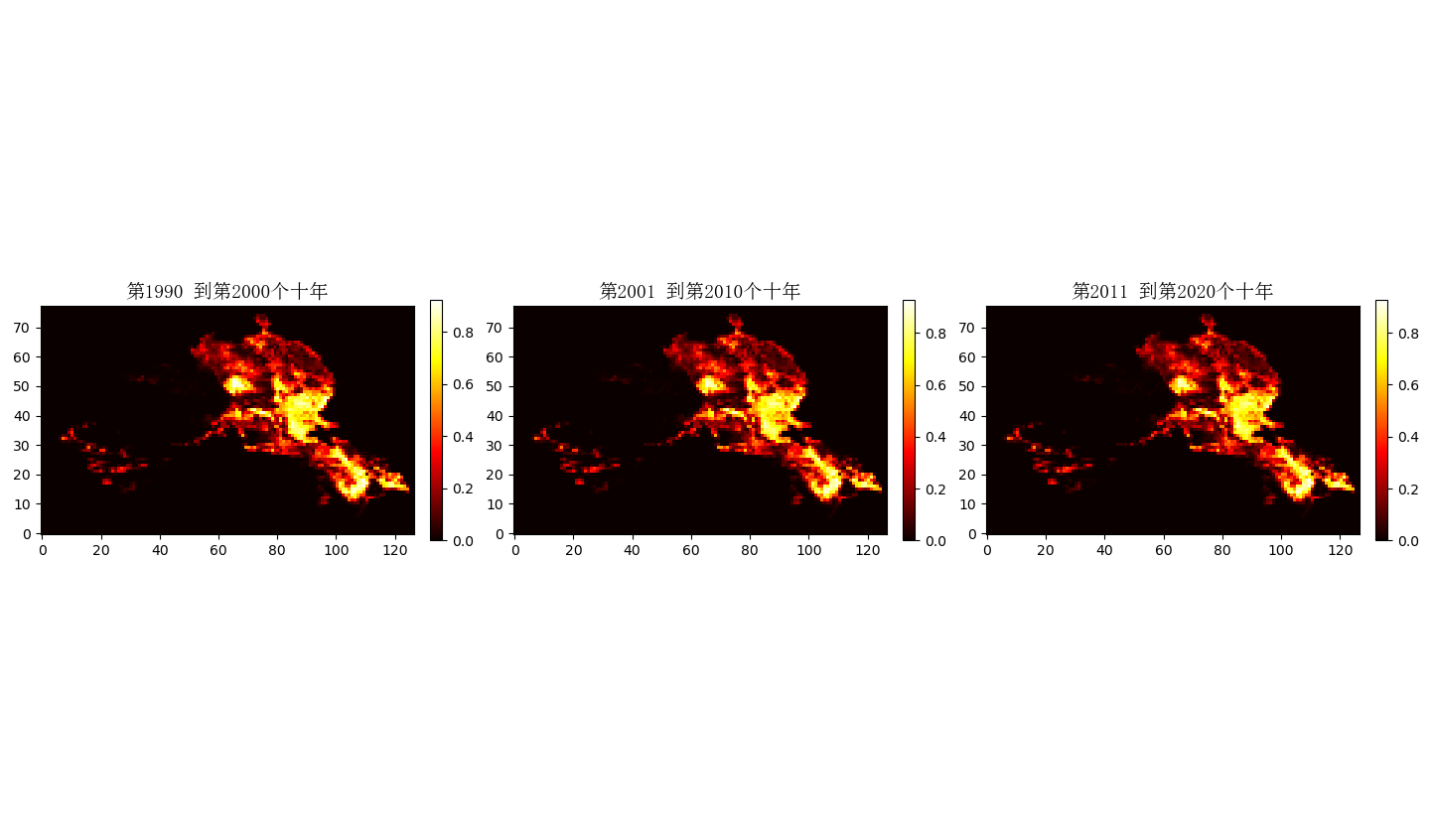
内容：展示1990-2020年间各类土地利用类型 的年度面积比例变化。

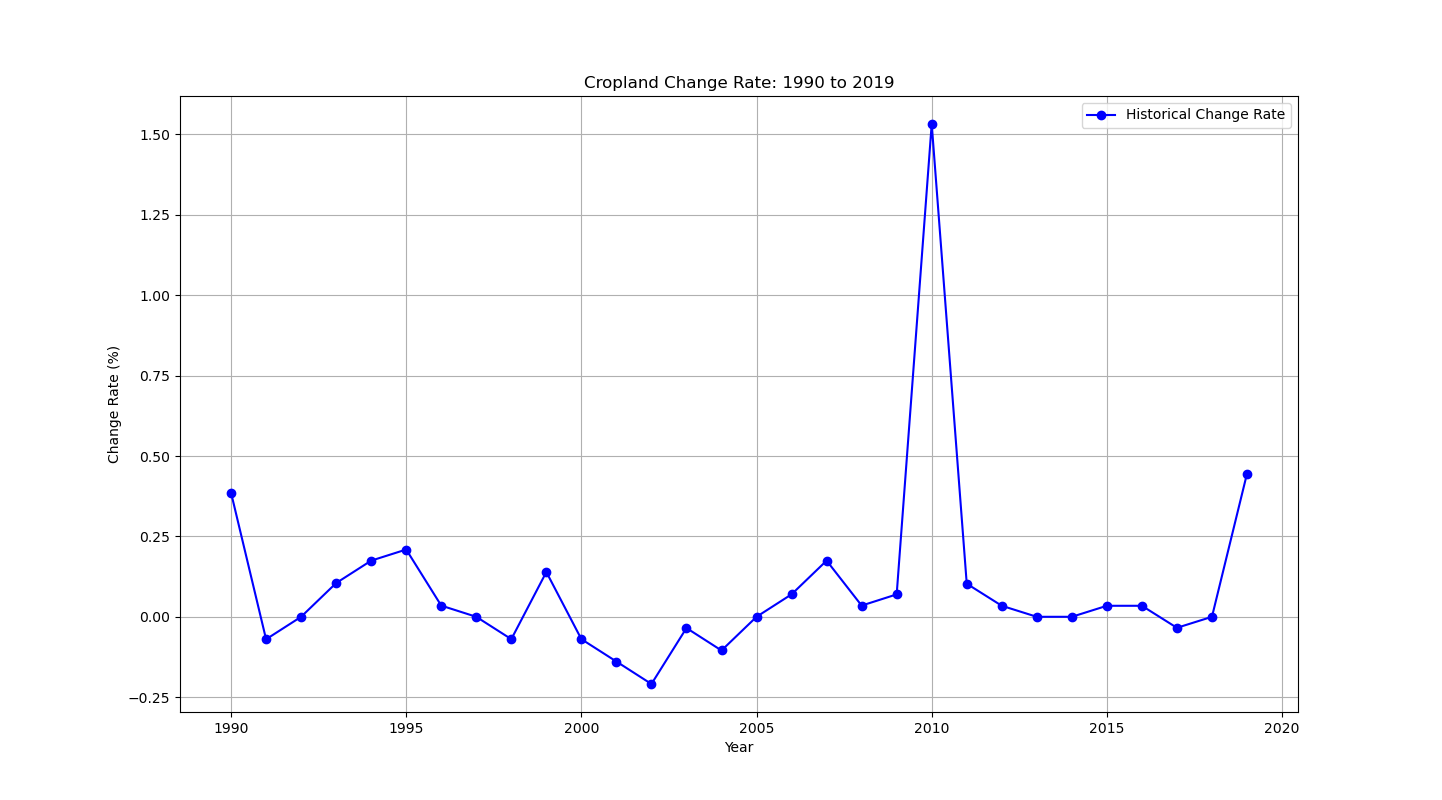


图：草地的占地面积变化热力图

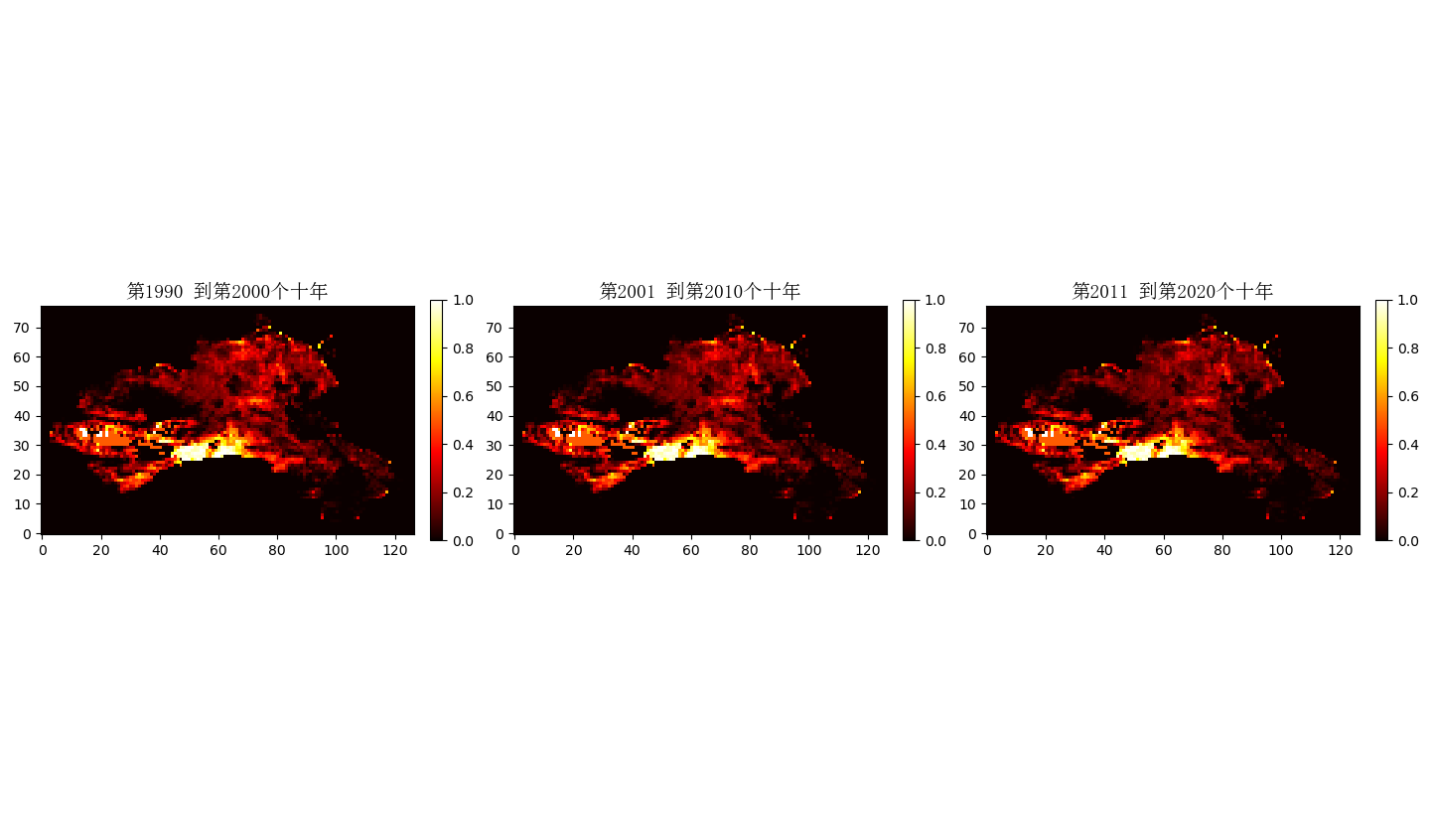


图x:草地变化率

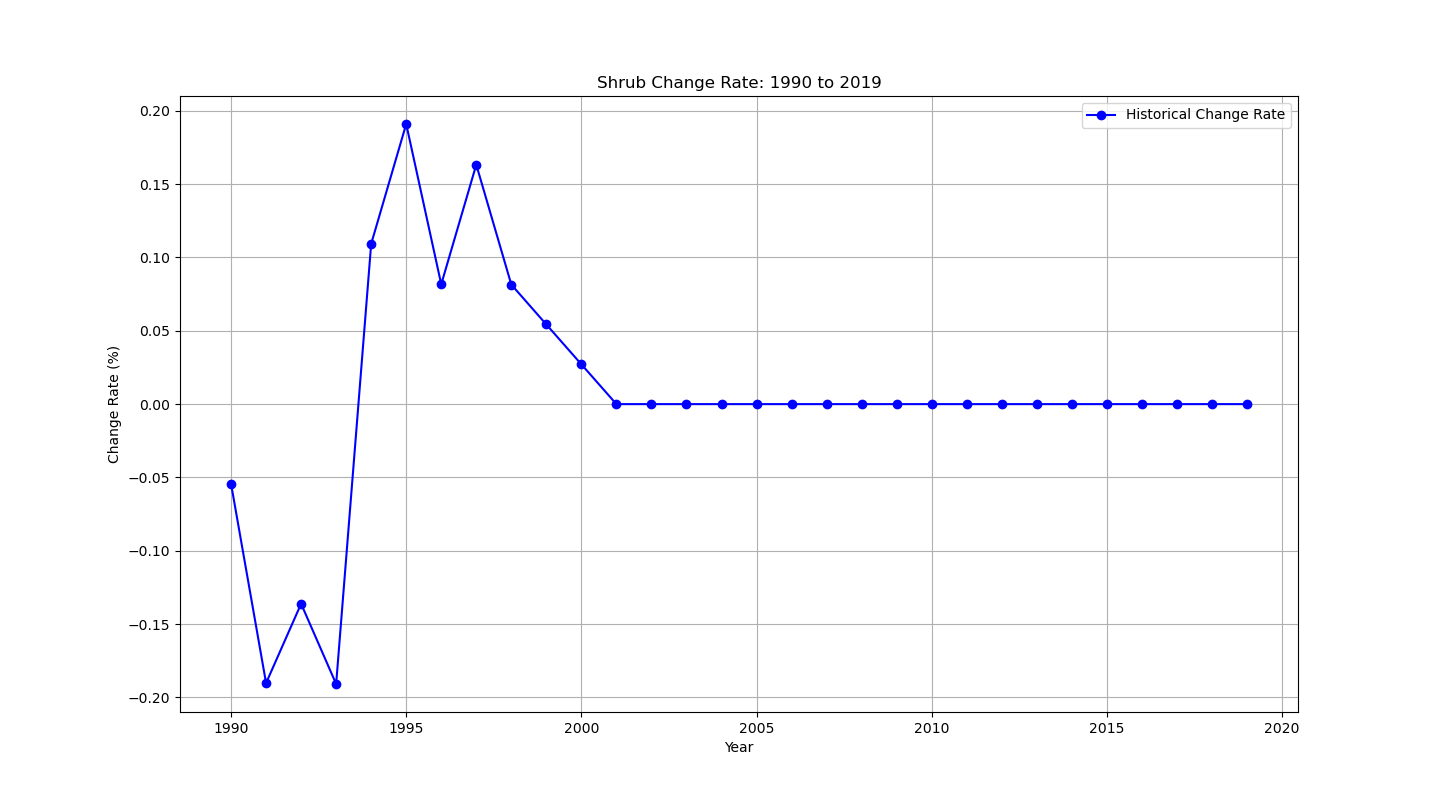


图x：中国耕地面积变化热力图

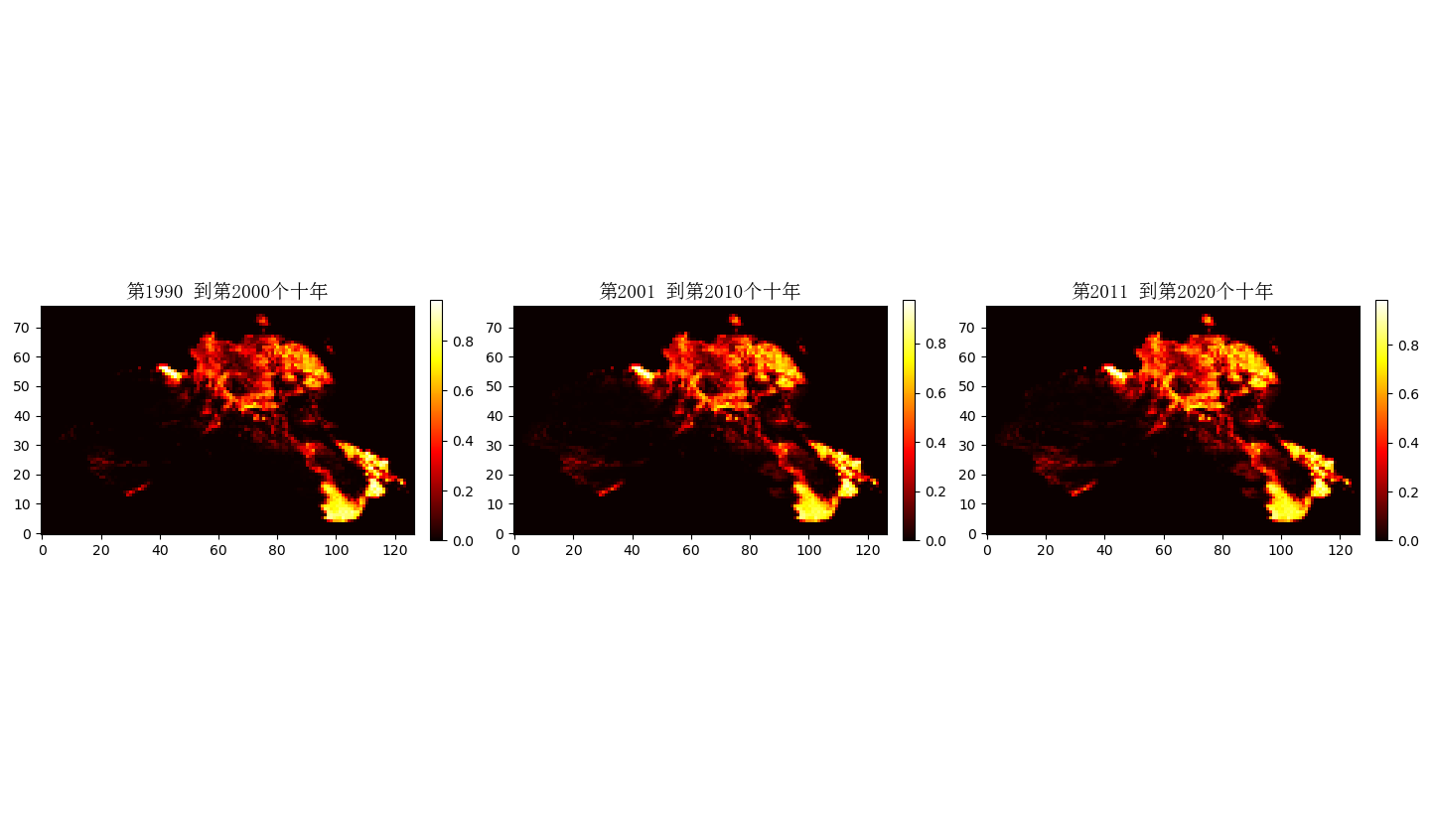
图：耕地面积变化率

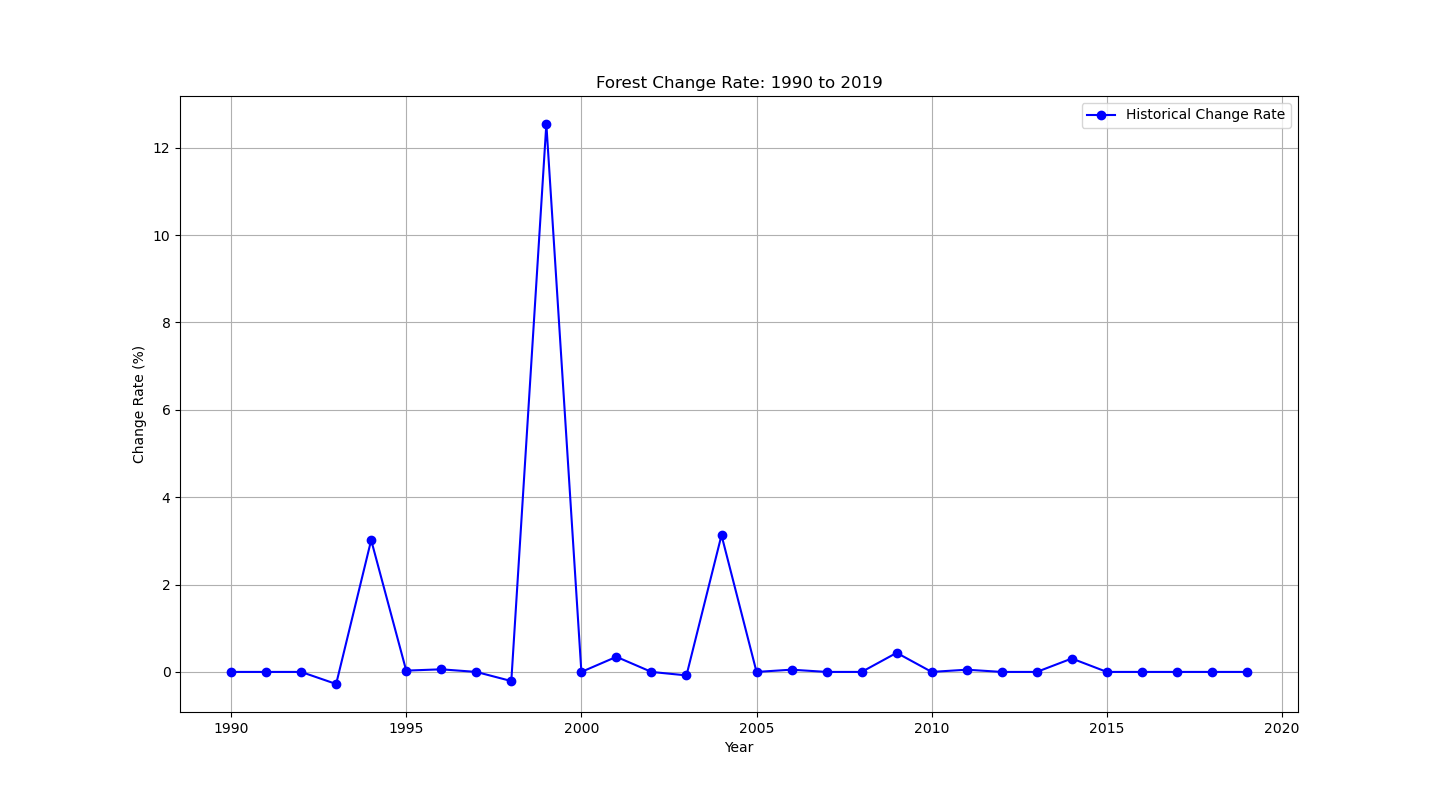


图：灌木丛面积拜热力图

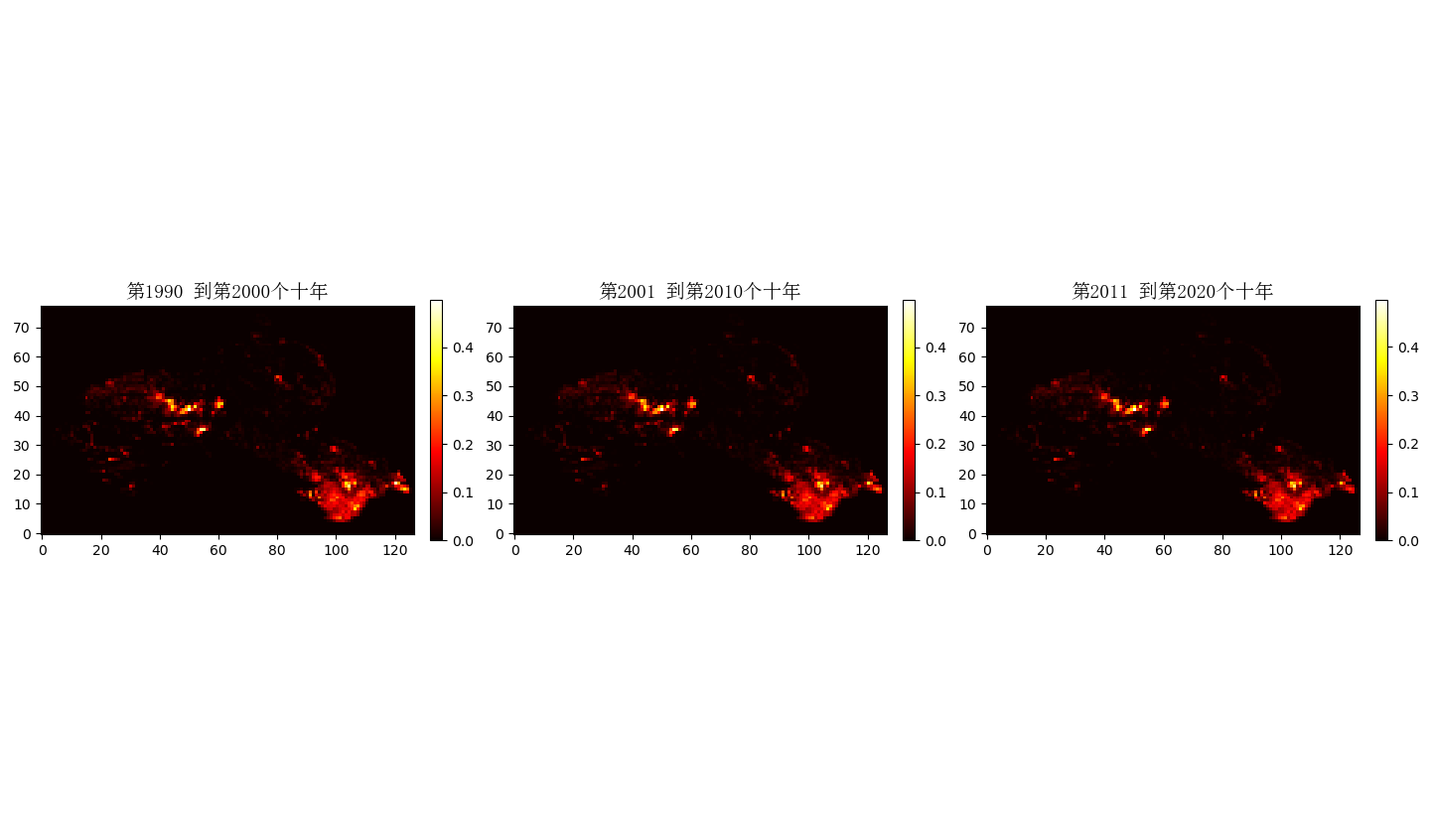


图：灌木丛变化率图

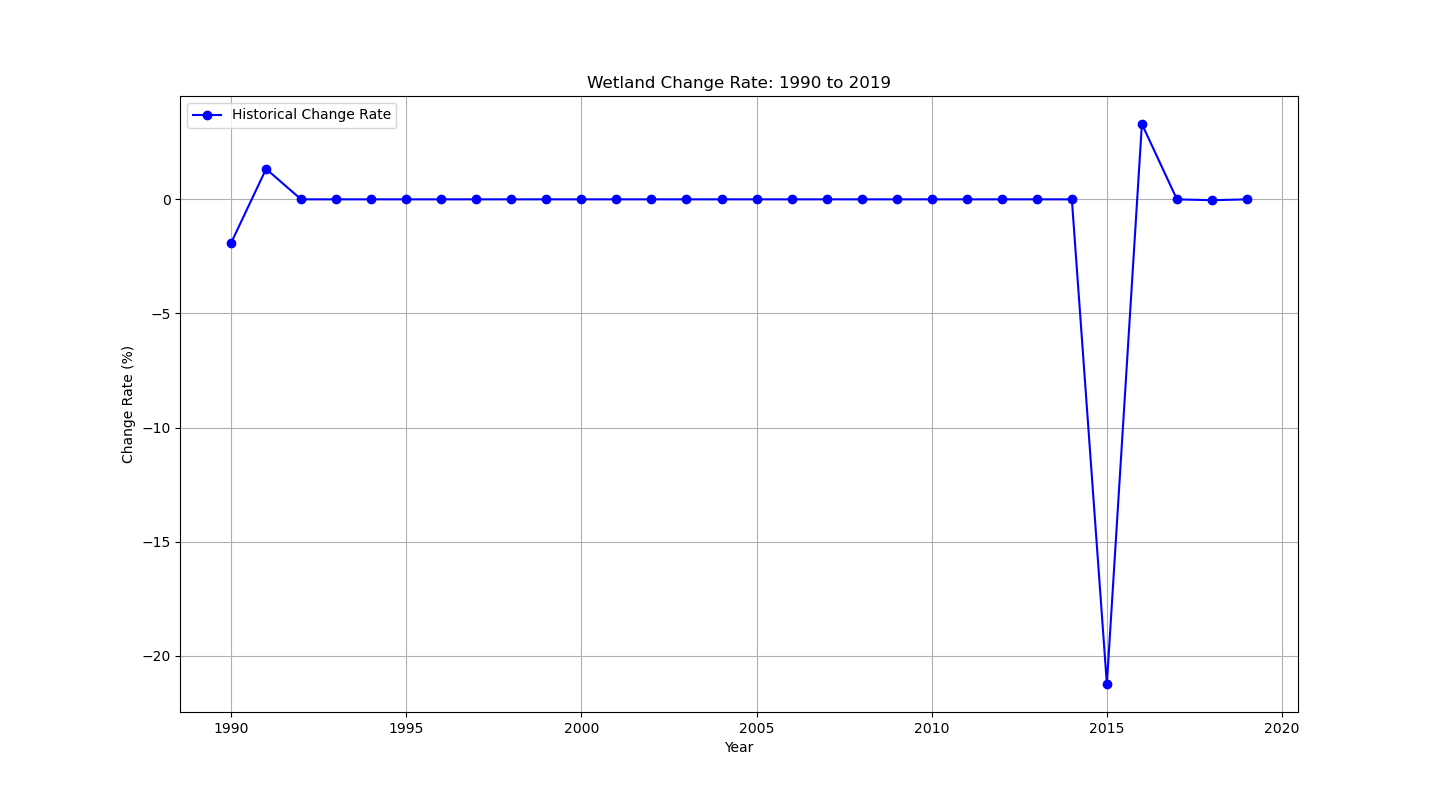


图：森林面积变化热力图

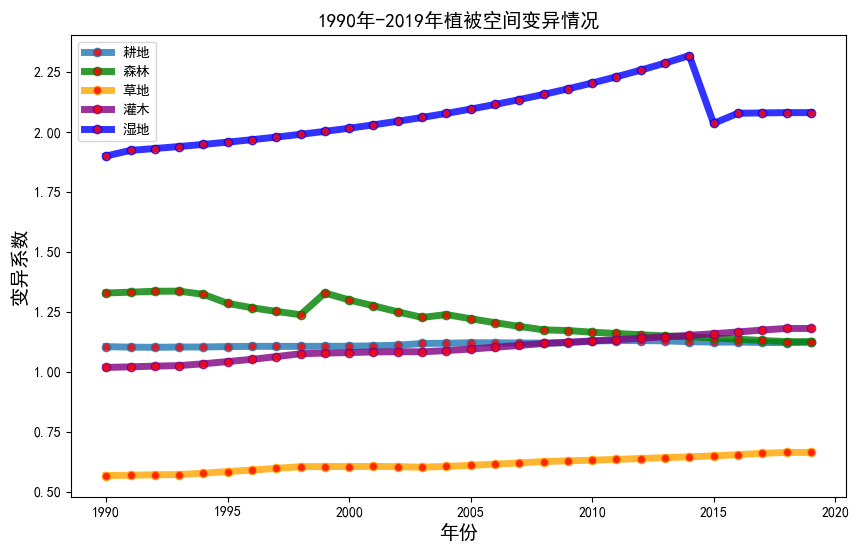
图：森林变化率



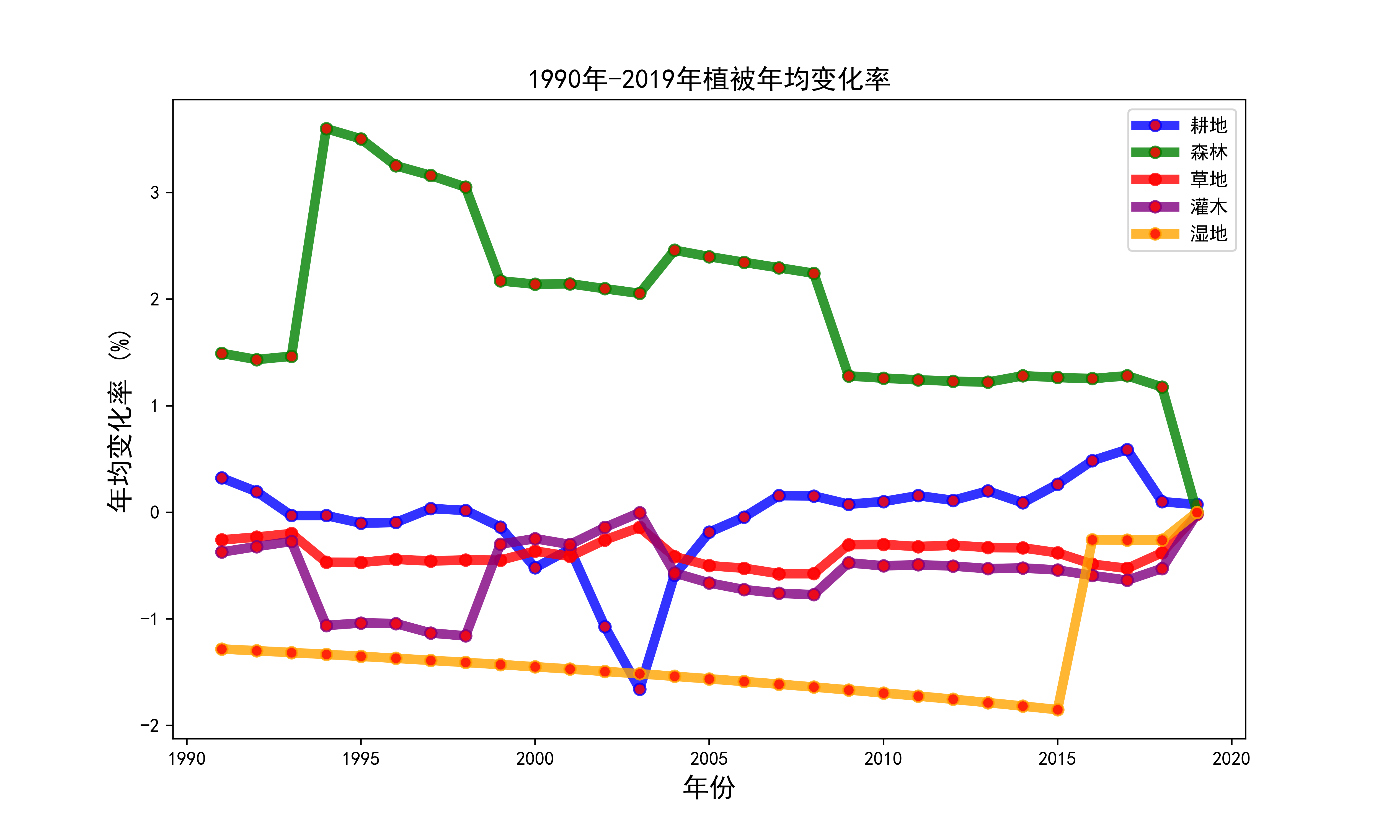
图：湿地面积变化热力图



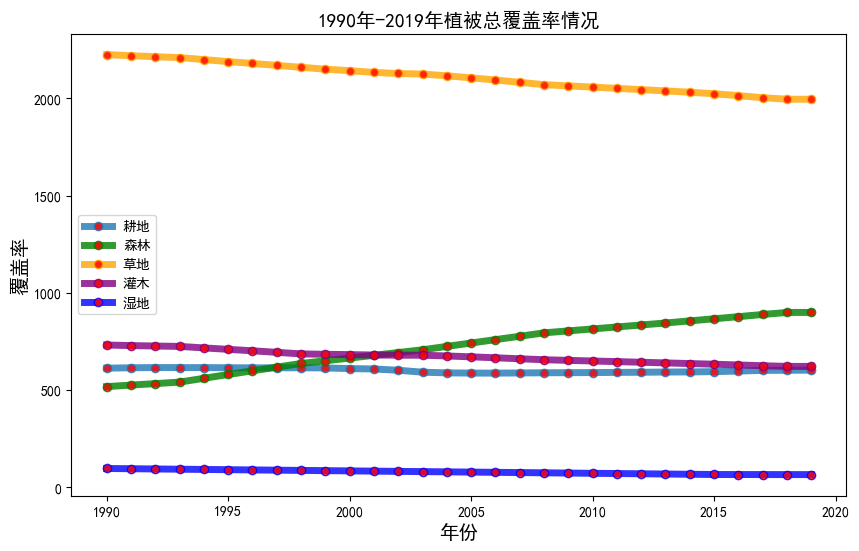
湿地变化率



图：1990年-2019年植被空间变异情况

图：

图：1990年-2019年植被年均变化率

图：1990年-2019年植被总覆盖率情况

形式：堆积柱状图。

分析：

直观比较不同土地利用类型的变化趋势和占比变化。

识别主要土地利用类型的扩展或缩减趋势。

地理分布图

内容：选择若干典型年份（如1990年、2005年、2020年），绘制各类土地利用类型的空间分布图 。

形式：GIS生成的地理分布图，不同颜色代表不同土地利用类型。

分析：

展示土地利用类型在空间上的分布特征。

分析土地利用类型在不同地理区域的动态变化。

变化率折线图

内容：展示各类土地利用类型的年度变化率 的变化趋势。

形式：折线图。

分析：

评估不同土地利用类型的变化速率。

识别变化速率较快的土地利用类型及其时间段。

通过建立多元线性回归模型和时间序列分析模型，结合马氏距离法预处理和主成分分析识别关键影响因素，我们系统地量化了1990-2020年间中国土地利用/土地覆被类型的时空演化特征。模型结果显示，土地利用变化主要受地形特征、气候变量及社会经济活动的驱动，且不同土地利用类型对这些因素的响应存在显著差异。通过可视化手段，进一步揭示了土地利用变化的空间分布模式及其动态演变趋势。该模型不仅提升了对土地利用变化机制的理解，也为未来的生态保护、资源管理和城市规划提供了科学的预测和决策支持。

**5.1.4 结果分析**

通过上述描述性统计方法，可以系统地描述1990-2020年间中国范围内降水量和土地利用的时空演化特征。具体来说：

降水量：年平均降水量的时间序列分析揭示了降水水平的长期变化趋势。降水量标准差的计算反映了降水分布的均匀性和波动性。空间热力图和箱线图直观展示了降水量的空间分布特征及其异常情况。

土地利用：各类土地利用类型的面积比例分析揭示了土地利用结构的变化趋势，如耕地减少、林地增加等。年度变化率的计算衡量了不同土地利用类型的变化速率，识别了变化剧烈的类型。

地理分布图和变化率折线图展示了土地利用类型在空间上的分布特征及其动态演变。

通过这些统计指标和图表，可以全面理解中国地理环境在过去三十年间的变化，为后续的深入分析和建模提供坚实的基础。这些描述性统计结果不仅有助于揭示降水量和土地利用的变化趋势，还为生态保护、资源管理和政策制定提供了科学依据。

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:地形-气候相互作用在极端天气形成过程中的作用**

1. 问题背景

近年来，极端天气事件（如暴雨、台风、高温等）频发，对人类生产生活和社会经济发展造成了严重影响。特别是在中国，复杂的地形与多变的气候条件相互作用，成为极端天气形成和发展的重要因素。因此，理解地形与气候之间的相互作用机制，对预测和防范极端天气事件具有重要意义。

2. 研究目标

本问题旨在通过数学建模，揭示地形与气候相互作用在极端天气（以暴雨为例）形成过程中的作用机制。具体目标包括：

分析地形特征对暴雨发生频率和强度的影响。

探讨气候因素（如温度、湿度、风速等）在极端天气形成中的作用。

建立地形-气候相互作用模型，定量描述其对暴雨形成的影响。

3. 研究意义

通过建立地形-气候相互作用模型，可以更准确地预测极端天气事件的发生，提升灾害预警和防范能力，进而减少人员伤亡和财产损失，促进社会经济的可持续发展。

首先定义了极端天气种类：

1.暴雨 (Heavy Rainfall)

暴雨通常是指在短时间内降水量显著超过常规水平。暴雨的定义可以根据日降水量来设置，具体标准如下:

定义:24小时内的累计降水量超过某个阈值，通常这个值根据地区平均降水量来设定，参考标准:

中等暴雨:24小时降水量>50毫米:

强暴雨:24小时降水量>100毫米。

特大暴雨:24小时降水量>200享米，数据来源:利用你所收集的降水数据按时间维度计算累计降水量，提取暴雨发生频率。

2.热浪(Heatwave)

热浪定义为连续多天的高温天气。具体标准可以根据气温的历史百分位数和日最高温度来确定

定义:日最高温度连续数天超过某个高温阈值，o

参考标准:

”日最高温度:连续3天以上的日最高温度超过当地历史日最高温度90或95百分位，绝对高温:日最高温度连续超过35°C，视为较强的热浪。数据来源:使用温度数据，计算每一天的最高温度，识别连续高温事件并设定持续天数的条件。

3.寒潮(Cold Wave)

襄潮通常指气温在短时间内急剧下降并维持较低温度的现象。定义寒潮时可以参考温度的急剧下降和低温持续时间,

。定义:短期内气温急剧下降，且最低气温持续低于某个阈值。

参老标准:。急剧降温:24小时内气温骤降超过8°C，或48小时内气温下降超过10°C。，低温:最低温度连续2-3天低于某个极端低温值。例如，最低温度连续3天低于0°℃，或当地历史最低气温10%以下数据来源:从温度数据中提取最低温度变化，并识别温度骤降的时间段。

模型假设

在建立地形-气候相互作用模型时，为简化问题并确保模型的可操作性，需做出以下假设：

地形稳定性假设：在研究期间（1990-2020年），地形特征（如海拔、坡度、坡向）保持相对稳定，不发生显著变化。

气候数据代表性假设：所选取的气候变量（温度、湿度、风速等）能够全面、准确地反映影响暴雨形成的主要气候条件。

暴雨定义明确：暴雨事件按照国家气象标准进行定义，例如24小时降水量超过某一特定阈值（如50毫米）即为暴雨。

变量独立性假设：在一定程度上，地形特征与气候变量之间的相互作用可以被建模为独立的影响因素，尽管实际中存在一定的耦合关系。

数据完整性假设：所使用的数据集完整、无显著缺失，且经过必要的预处理（如插值、平滑等）以保证数据质量。

5.2.1 数据处理与模型建立：

为了避免不同量纲对模型准确性的影响，对所有地形数据进行了归一化处理。同样地，气候数据也采用了类似的归一化方法，但归一化过程以年度为单位进行，即对每个文件夹中的所有图像像素进行独立归一化。

可视化：对数据集1中的WGS84地理坐标系数据进行可视化，以了解地形的空间分布特征。

归一化：对地形数据和气候数据进行归一化处理，确保不同量纲的数据不会影响模型的准确性。气候数据按年度独立归一化，处理每个文件夹下所有图像的像素。

数据对齐：将地形数据沿时间维度复制，使其时间维度与降水量数据的时间维度22645对齐。选择2008年至2018年期间的每日数据，进行空间维度上的对齐，通过大图像取平均值减少像素数量，统一空间维度为（144，256）。

具体来说，数据集1中的地形数据具有维度 ，其中 和 ；数据集2中的气温数据为 ，维度为 和 ；数据集3中的降水量数据表示为 ，其维度为 和 。

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降水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** |

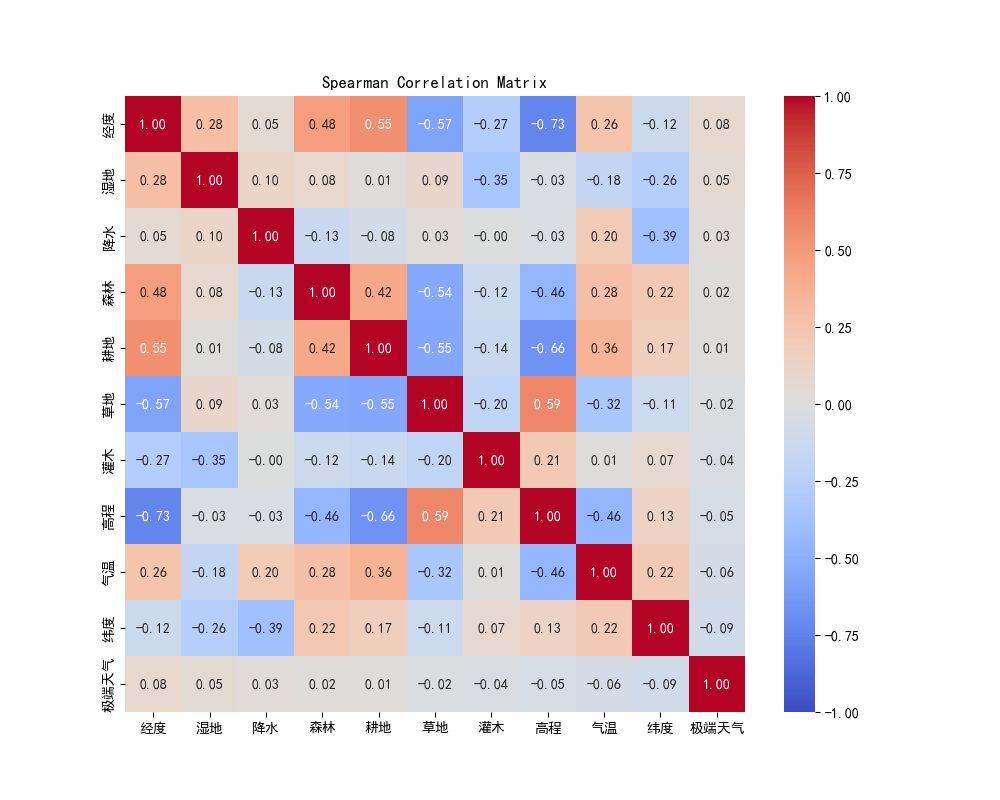


图 2：斯皮尔曼相关系数热力图

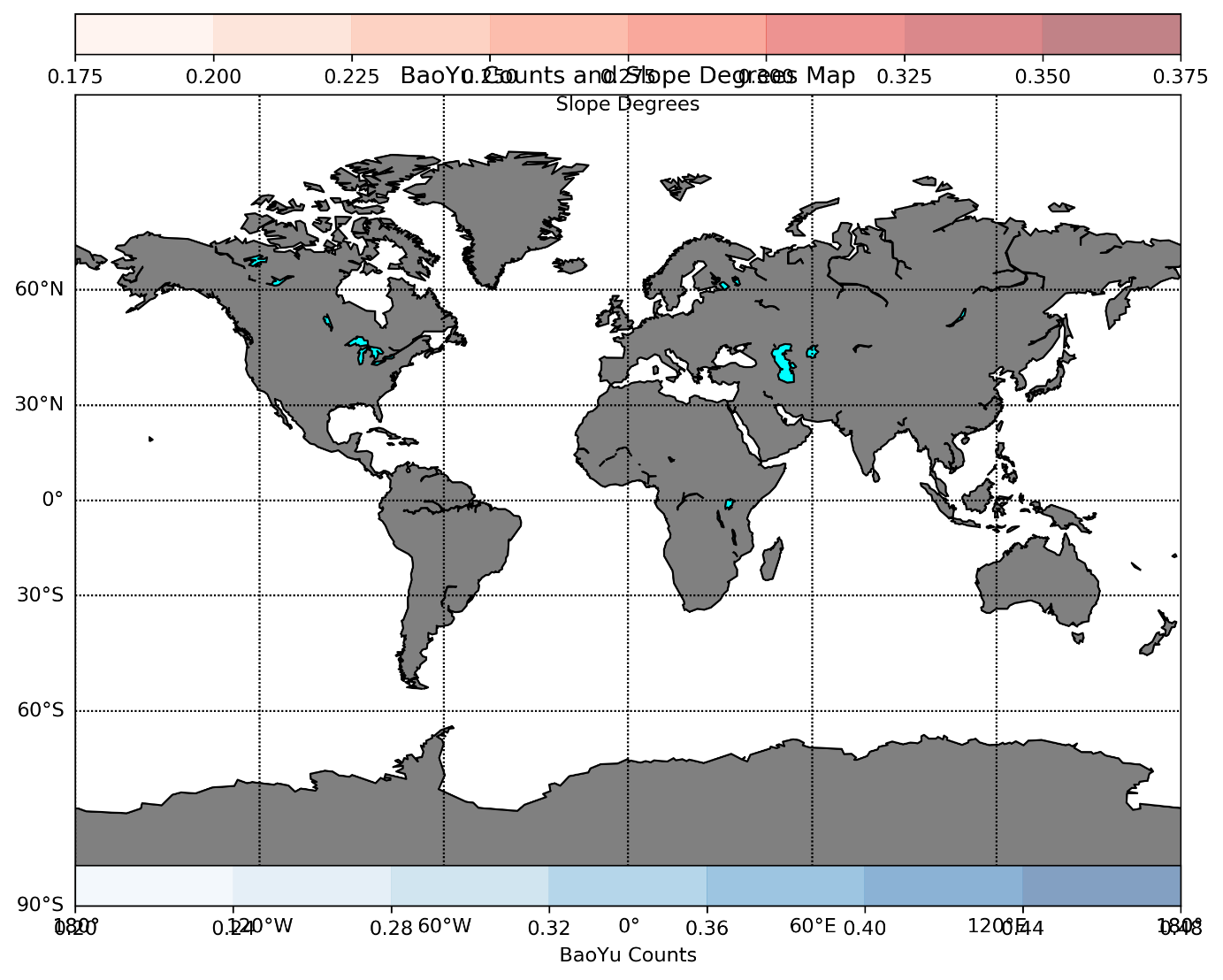


图: 暴雨次数和坡度关系地图

针对问题二，我们需要建立一个模型来量化地形与气候相互作用在极端天气（如暴雨）形成过程中的作用，具体目标是确定函数 的表达式：

在此过程中，已量化的地形数据沿时间维度进行复制，以使时间维度与降水量数据的时间维度（共22645个时间点）对齐。考虑到长期数据中地形与气候相互作用对极端天气形成的贡献相对较小，本文选择使用2008年至2018年期间（每年1月1日至12月31日）的每日数据进行分析。

在空间维度上，为了简化计算并提高处理效率，我们采用了对大尺寸图像进行平均处理以减少像素数量，从而使所有数据的空间维度统一为（xxx，xxxx）。如果需要更高的精度，可以采用将小尺寸图像对齐至大尺寸图像的方法，使所有数据的空间维度达到（xxxx，xxxx）。本文选择了前者方法，即通过降低空间分辨率来提高算法的计算效率，同时保持模型的有效性。

5.2.2**逻辑回归模型**

为了量化地形与气候对暴雨形成的影响，我们建立了以下逻辑回归模型：

其中：

表示暴雨事件发生的概率。

为截距项。

为各自自变量的回归系数。

为误差项，假设服从正态分布 。

模型假设

线性关系假设：假设地形数据 、气温数据 和降水量数据 与暴雨发生的对数几率存在线性关系。

独立性假设：各个观测值之间相互独立。

同方差性假设：误差项的方差在所有观测值中保持恒定。

正态性假设：误差项服从正态分布。

模型求解

为了求解上述逻辑回归模型中的参数 ，我们采用了模拟退火算法。这种算法能够有效地在大规模搜索空间中找到全局最优解，特别适用于处理复杂的优化问题。具体步骤如下：

初始化：设定初始温度和退火速率，随机生成初始参数 。

邻域搜索：在当前参数的邻域内随机生成新参数 。

接受准则：计算新旧参数对应的目标函数值（如对数似然），根据模拟退火的接受概率决定是否接受新参数。

温度更新：逐步降低温度，减少接受劣质解的概率。

终止条件：达到预设的最低温度或达到最大迭代次数时终止算法。

通过上述步骤，最终得到最优的参数 ，从而确定地形与气候对暴雨形成的量化影响函数 。

Step-by-Step Procedure

1. Initialization

1.1 数据可视化

对数据集1中的WGS84地理坐标系数据进行可视化处理，结果见图7。

1.2 数据归一化

对所有地形数据进行归一化处理，确保不同量纲不影响模型准确性。

对气候数据（气温和降水量）按年度进行归一化，即对每个文件夹下的所有图像像素进行独立归一化处理。

1.3 数据对齐

将地形数据沿时间维度复制，使其时间维度与降水量数据的时间维度22645对齐。

采用2008年至2018年期间（每年1月1日至12月31日）的每日数据进行分析。

在空间维度上，通过对大图像取平均值，减少像素点数量，使所有数据的空间维度统一为（144，256）。

2. 模型构建

2.1 定义变量

地形数据：，其中 ，

气温数据：，其中 ，，

降水量数据：，其中 ，，

2.2 量化函数定义

建立函数 来量化地形与气候对降水量的影响：

其中， 为待确定的量化函数，描述地形 和气候 在时间 上对降水量 的影响。

3. 模拟退火算法实施

3.1 初始化参数

设置初始温度 （例如，）

设置冷却速率 （例如，）

初始化模型参数 （例如，通过随机初始化）

3.2 计算初始适应度

定义适应度函数 ，衡量模型拟合优度：

3.3 迭代优化过程

Repeat until 或达到最大迭代次数:

3.3.1 生成新解

在当前参数 的基础上，随机微调部分参数，生成新参数

3.3.2 计算新适应度

计算新参数下的适应度函数 \( F' \):

3.3.3 决定是否接受新解

如果 ，则接受新参数 ，即 ，

否则，以概率 接受新参数

3.3.4 更新温度

按照冷却速率更新温度：

4. 模型输出

当满足终止条件时，输出最优参数

确定量化函数 的最优表达式，用于描述地形与气候对降水量的影响

1. 适应度函数

解释：该公式表示模型预测的降水量与实际降水量之间的误差平方和，作为适应度函数，用于衡量模型拟合的优度。

2. 接受概率

解释：在模拟退火算法中，当新解的适应度 较差时，以概率 接受该解，以避免陷入局部最优。

3. 温度更新公式

解释：按照冷却速率 更新温度，逐步降低接受较差解的概率。模型假设

1. 地形稳定性假设：在研究期间（2008-2018年），地形特征（如海拔、坡度、坡向）保持相对稳定，不发生显著变化。

2. 气候数据代表性假设：所选取的气候变量（气温、降水量）能够全面、准确地反映影响暴雨形成的主要气候条件。

3. 暴雨定义明确：暴雨事件按照国家气象标准进行定义，例如24小时降水量超过某一特定阈值（如50毫米）即为暴雨。

4. 变量独立性假设：地形特征与气候变量之间的相互作用可以通过模型中的参数 量化，尽管实际中存在一定的耦合关系。

5. 数据完整性假设：所使用的数据集完整、无显著缺失，且经过必要的预处理（如插值、平滑）以保证数据质量。

3. 模型优化

- 模拟退火算法：采用模拟退火算法优化模型参数 ，以最小化适应度函数 。

邻域搜索：在当前参数基础上，随机调整部分参数，生成新的参数集 。

适应度评估：计算新参数下的适应度函数 ，与当前适应度 进行比较。

接受准则：如果 ，接受新参数；否则，以概率 接受较差解。

温度更新：按冷却速率 逐步降低温度，减少接受较差解的概率，逐步趋近全局最优解。

4. 模型验证

交叉验证：将数据分为训练集和测试集，评估模型的预测能力。

性能指标：采用均方误差（MSE）、决定系数（）等指标衡量模型拟合效果。

结果分析：分析地形与气候变量对降水量的影响程度，验证模型的合理性和准确性。

总结

通过模拟退火算法优化的逻辑回归模型，成功量化了地形与气候相互作用在暴雨形成过程中的影响。数据预处理阶段通过归一化和数据对齐，确保了不同数据源的兼容性和模型输入的有效性。模型优化过程中，模拟退火算法有效地搜索了参数空间，避免了陷入局部最优，提升了模型的拟合精度。最终，模型能够准确描述地形与气候因素对降水量的影响，为极端天气预测和防灾减灾提供了科学依据。

数据集说明

数据集1：中国数字高程图（1km）

内容：包含两种坐标系的数据，WGS84地理坐标系下的GeoTIFF图像，每个像素代表地表海拔高度，单位为米。

处理：筛选2008-2018年间的数据，进行空间维度统一处理，减小像素数量至（144，256）。

数据集2：中国0.1°近地表气温数据集（1979-2018年）

内容：包含每日气温平均值的GeoTIFF图像，单位为摄氏度。

处理：筛选2008-2018年间的数据，按年度进行归一化处理。

数据集3：中国大陆0.25°逐日降水数据集（1961-2022年）

内容：包含每日降水量的NetCDF文件，单位为毫米。

处理：筛选2008-2018年间的数据，空间维度统一为（144，256）。

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 |
| 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 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

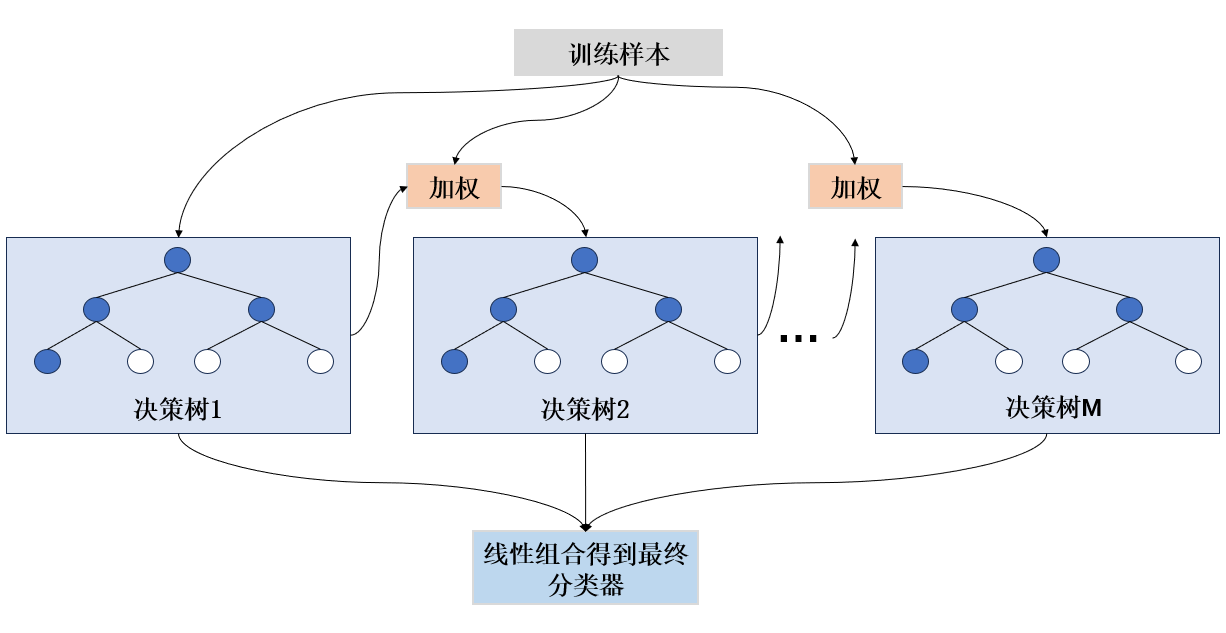


图 13:梯度提升决策树模型框架图

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

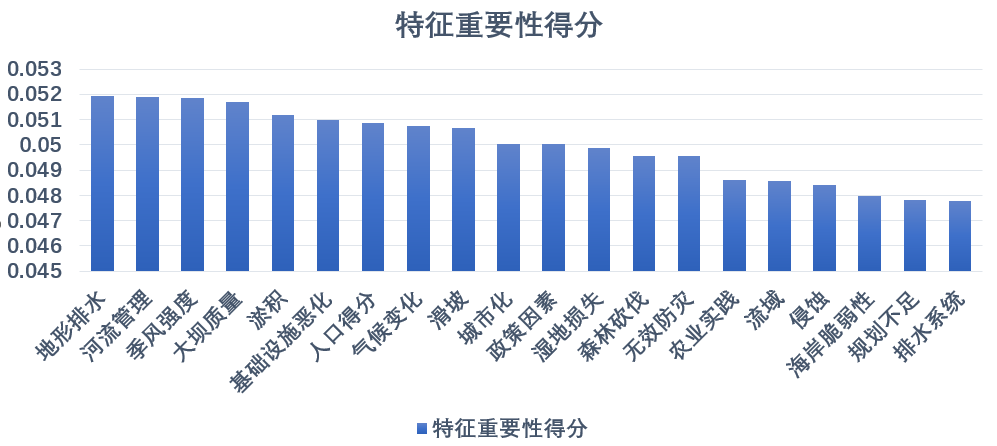


图 14：基于梯度提升决策树的特征重要性得分

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。

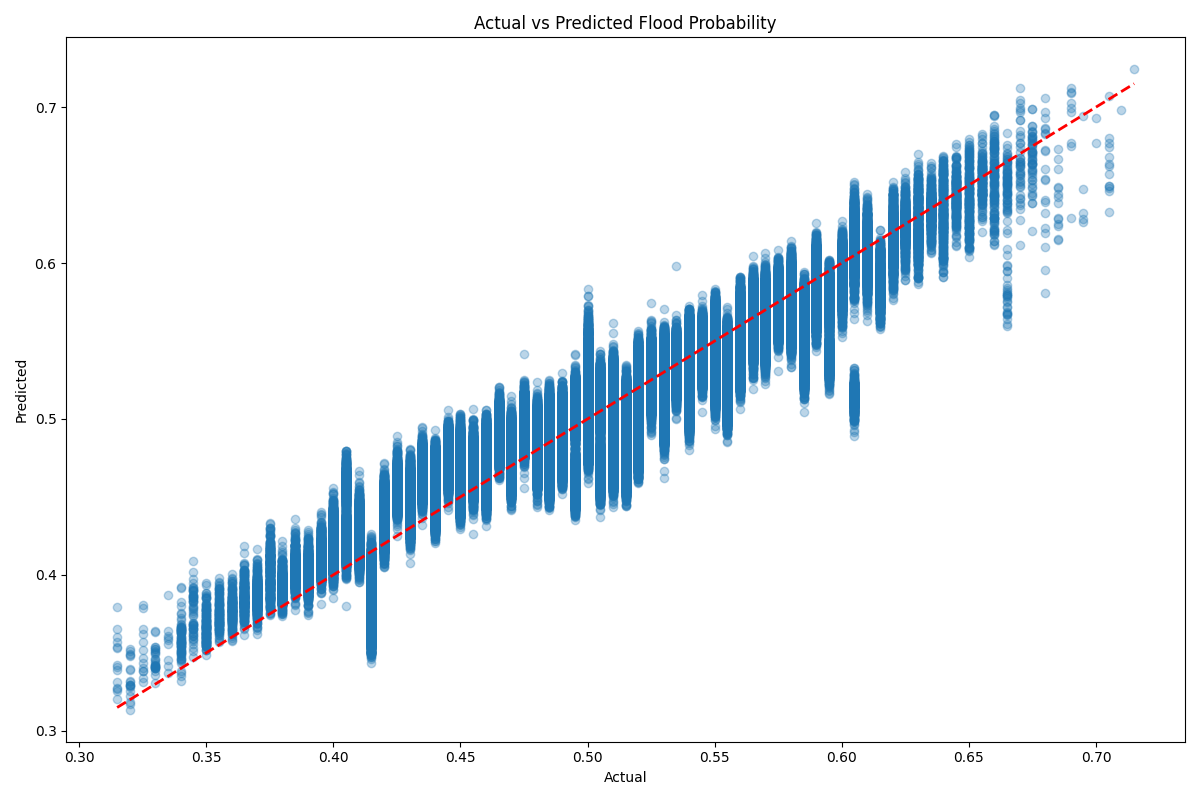


图 15:基于梯度提升决策树的洪水预警评价模型测试示意图

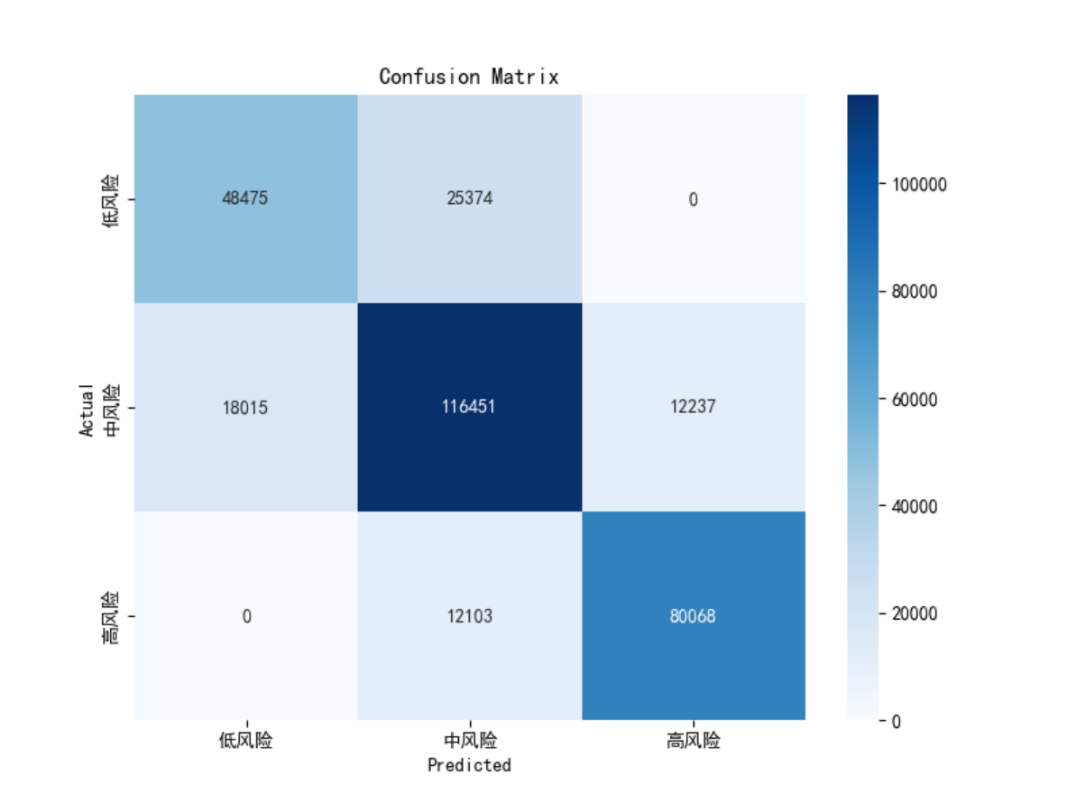


图 16：基于梯度提升决策树的洪水预警评价模型测试混淆矩阵

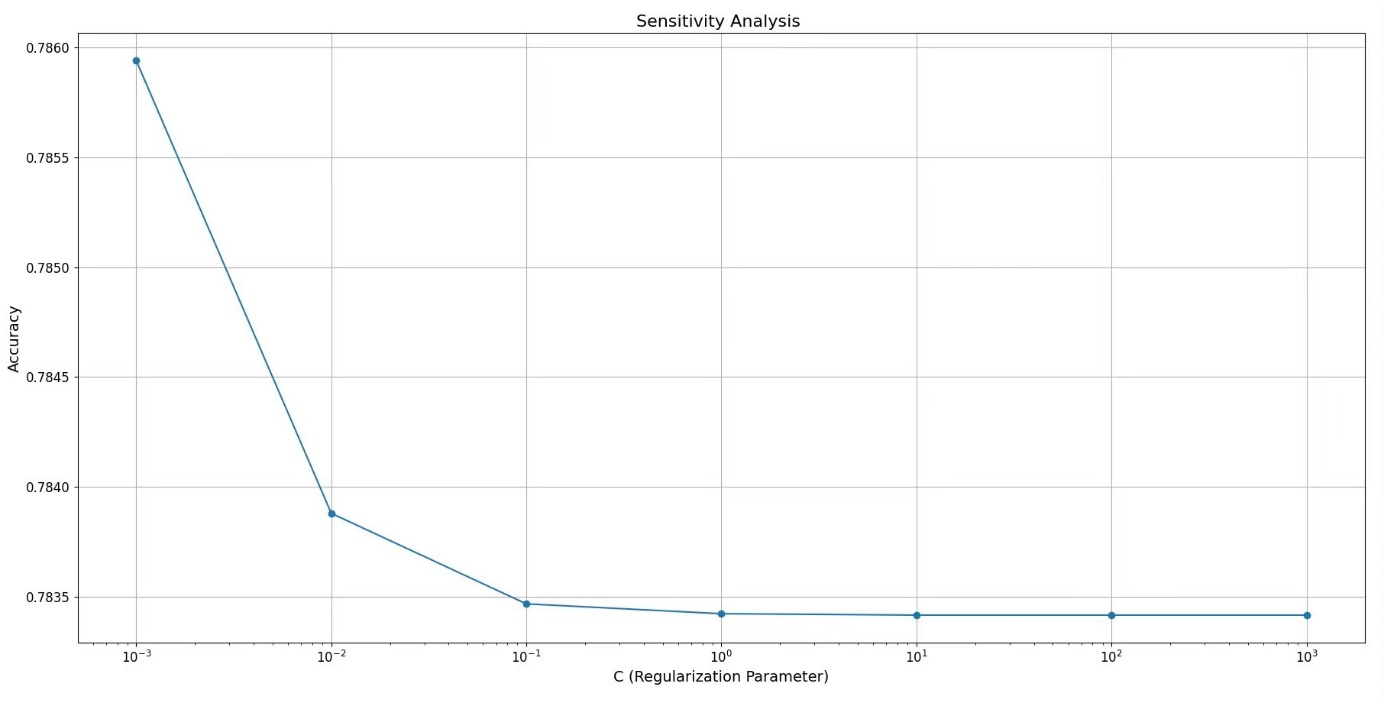


图 16:基于梯度提升决策树的灵敏度分析

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。

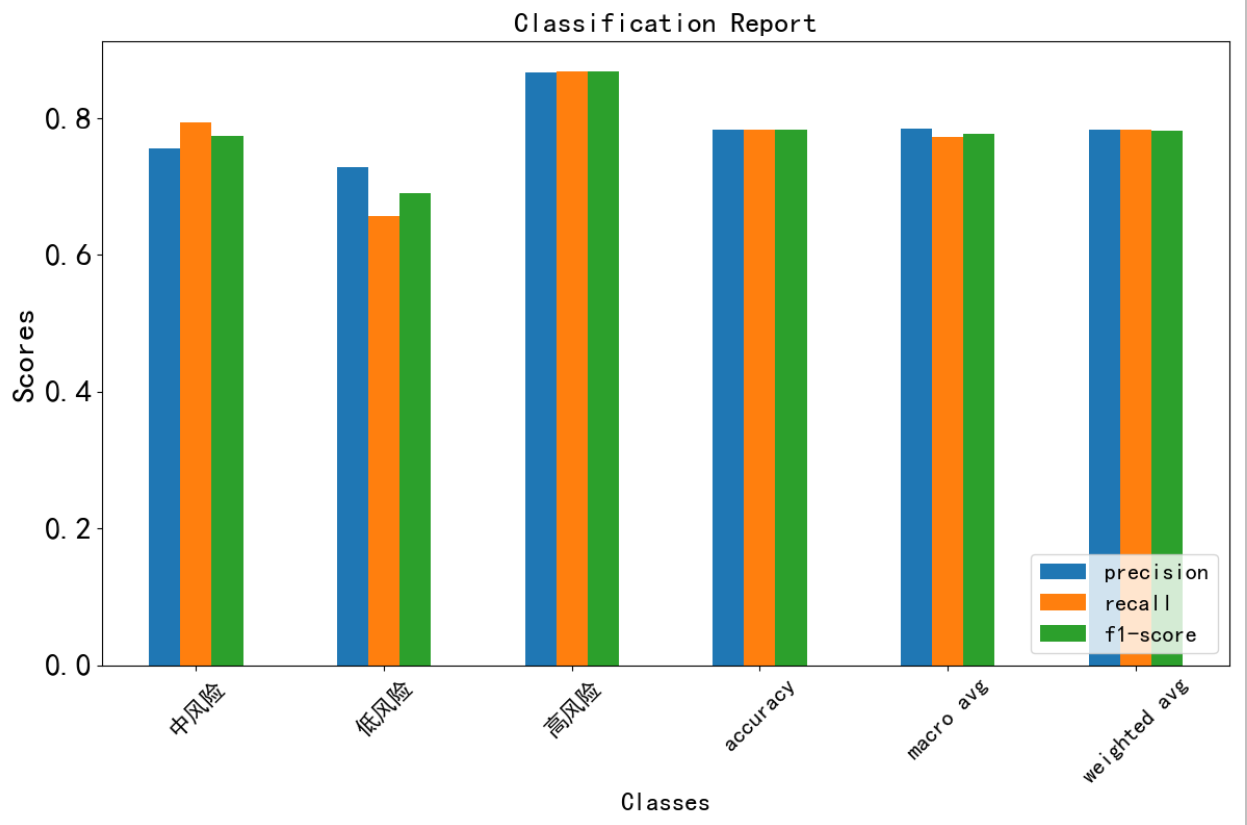


图 17：多元逻辑回归模型测试示意图

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

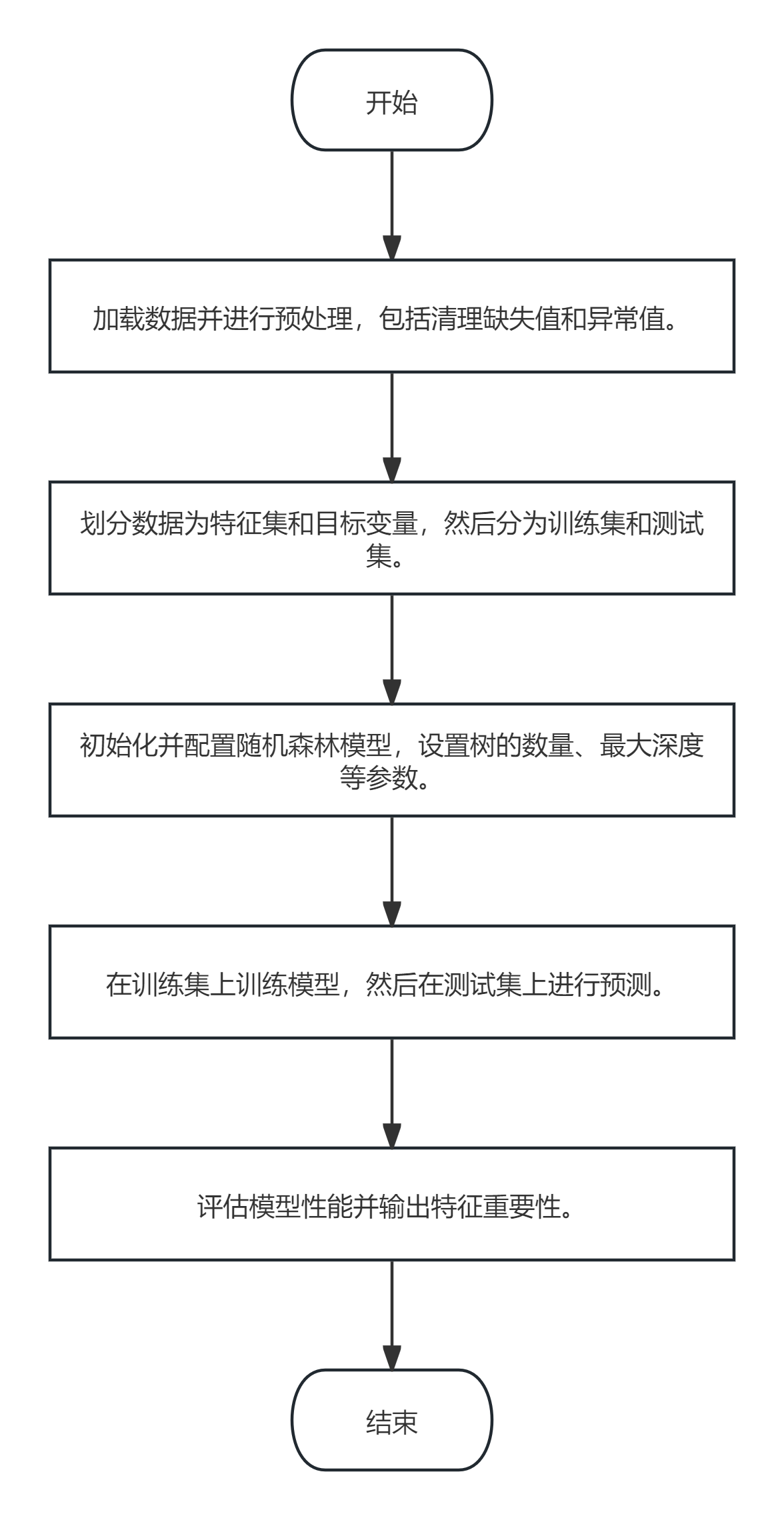


图 18： FloodRiskPredictor预测模型流程图

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，

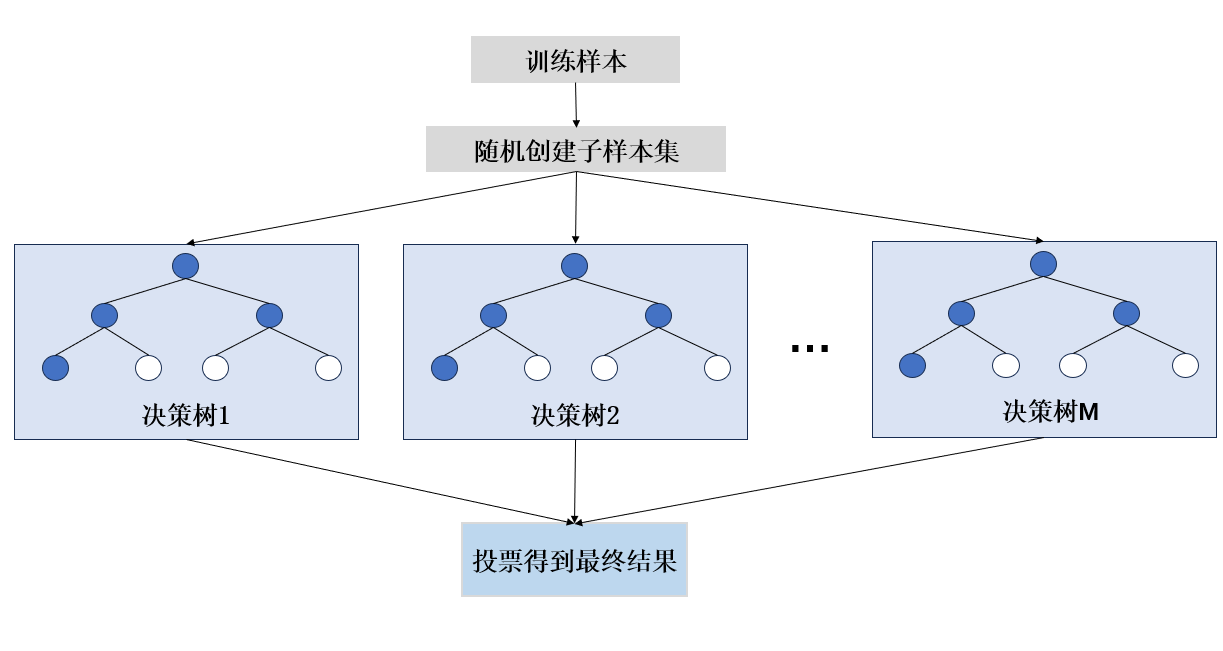


图 19：随机森林算法示意图

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。

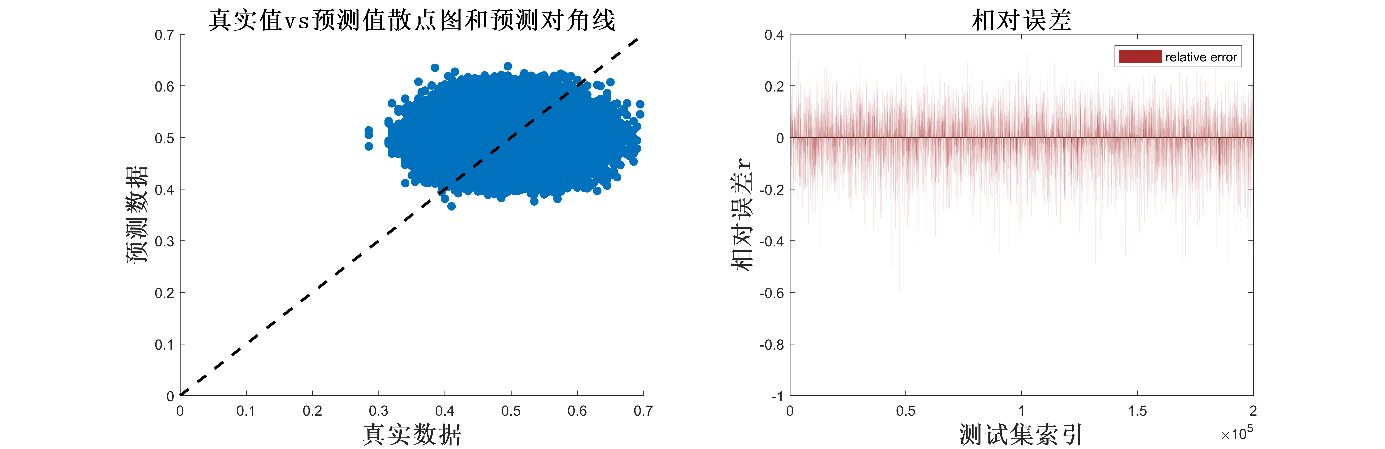
MSE 和 RMSE的数据如下。

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。

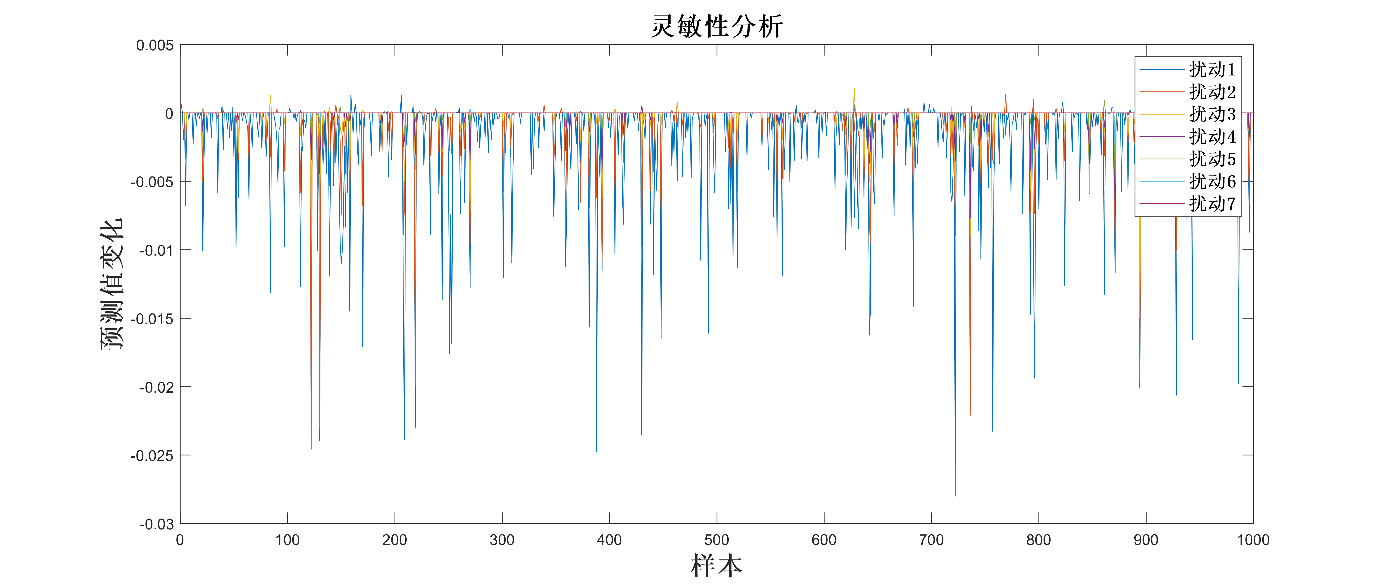


图 21：灵敏性分析示意图

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。

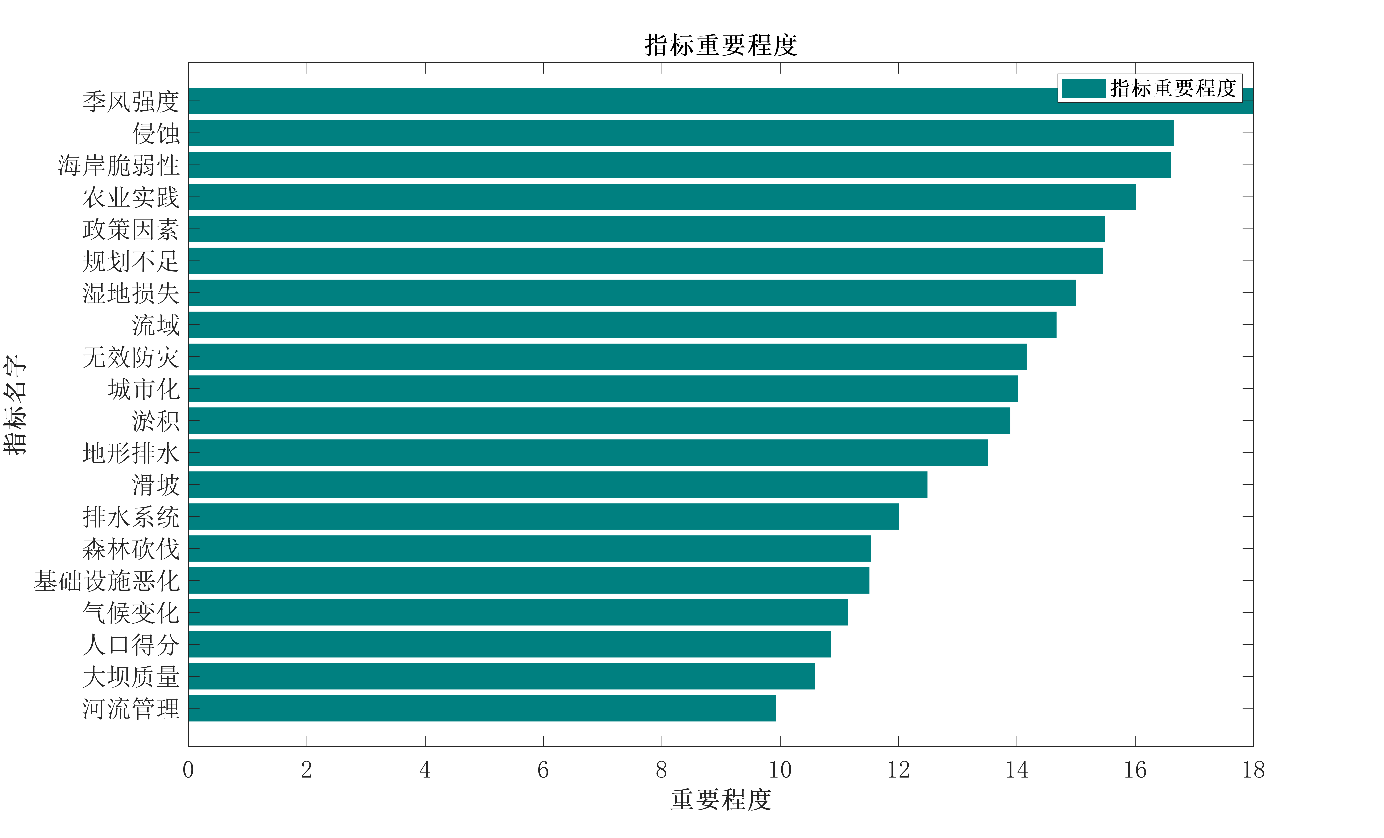


图 22：指标重要程度直方图

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。

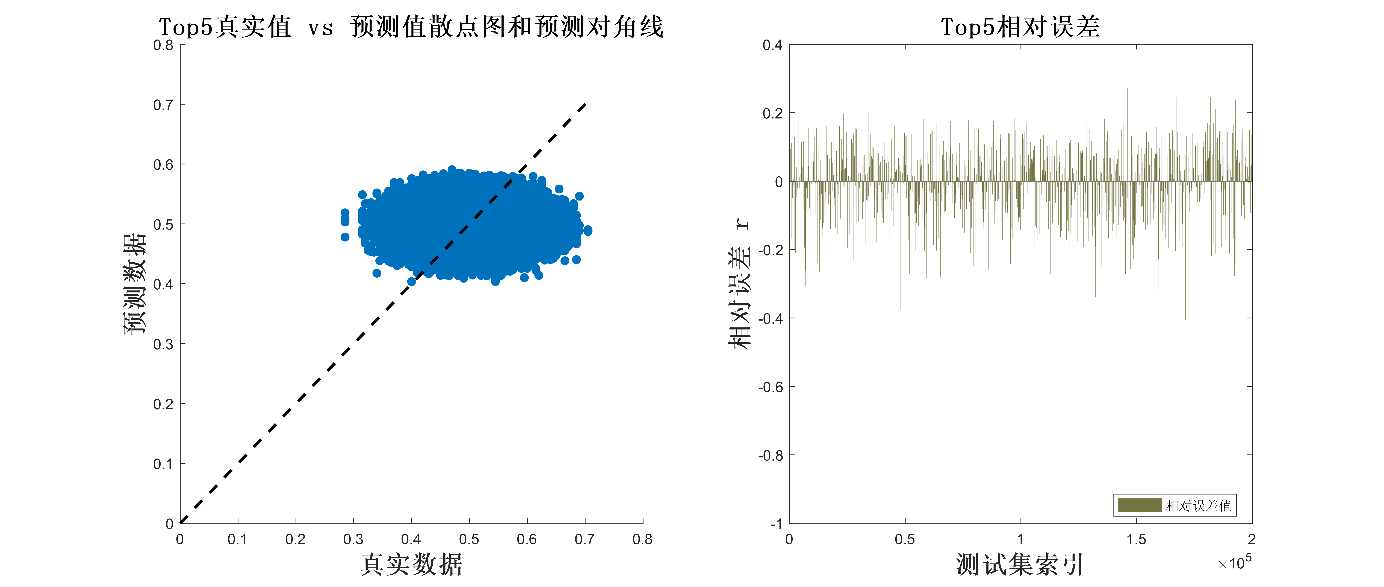


图 23：重要指标的回归与相对误差图

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。

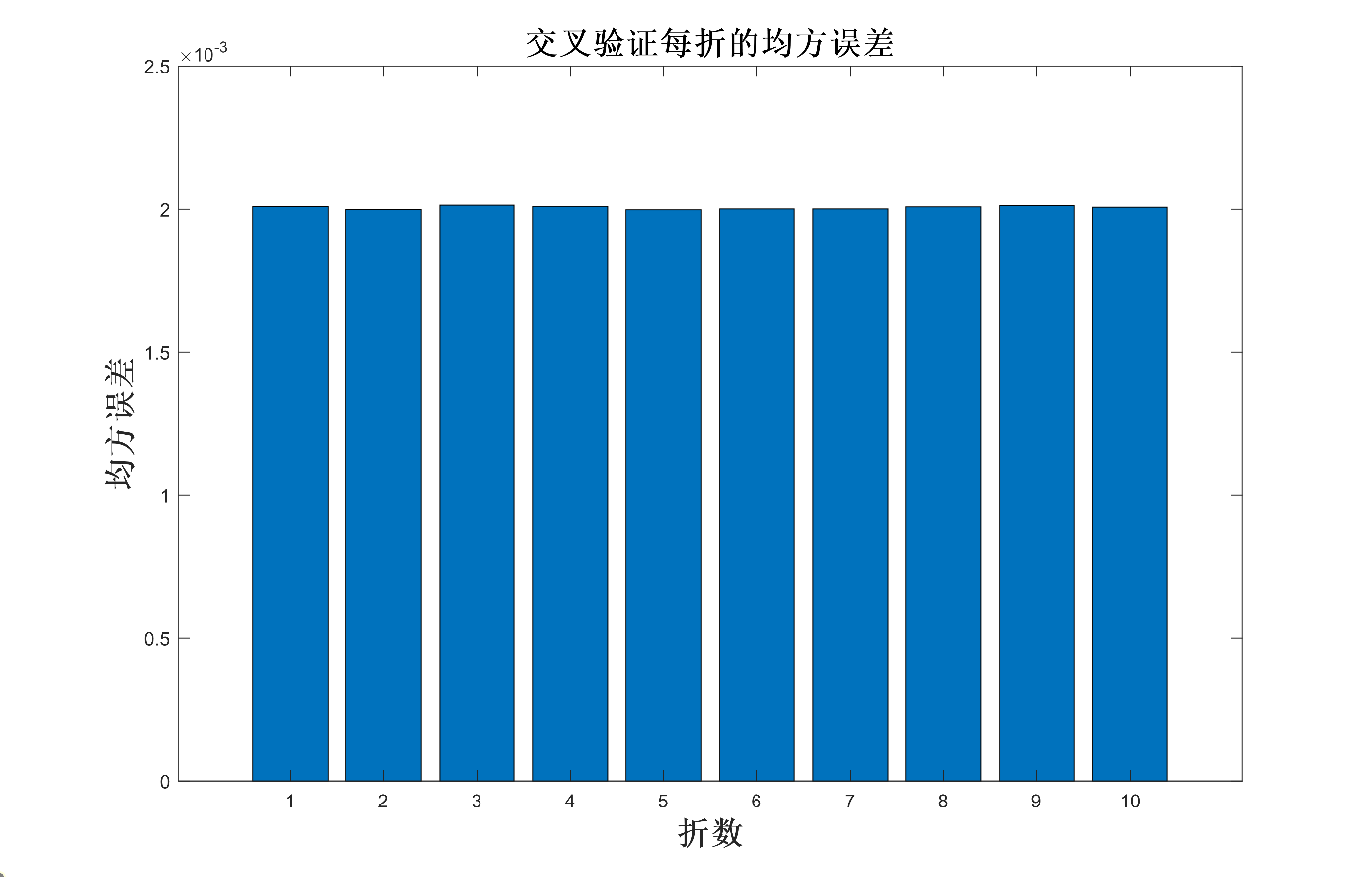


图 24：交叉验证每折的均方误差图

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。

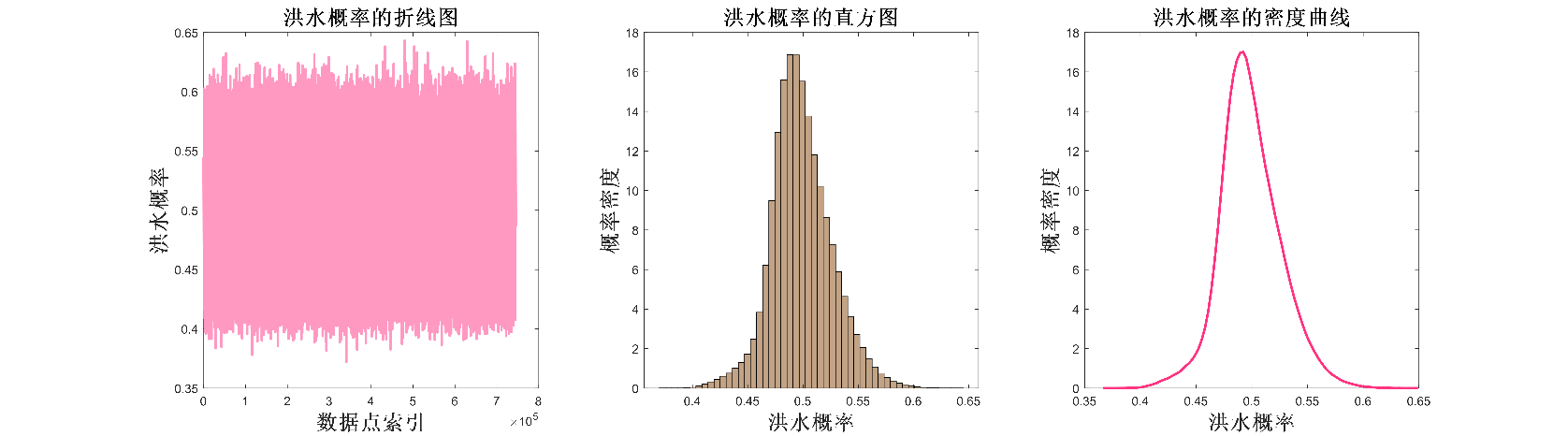


图 25：洪水概率分布图

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。

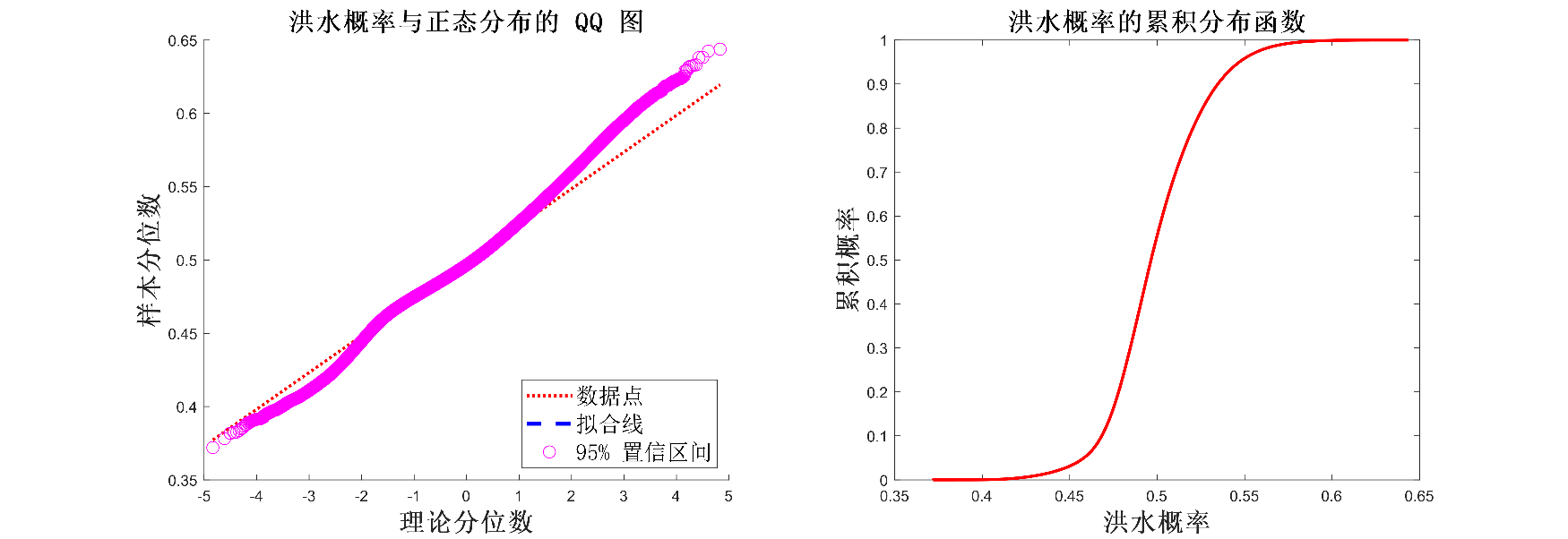


图 26：洪水概率与正态分布的QQ图

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()