基于机器学习的洪水灾害预测模型

摘　要

洪水是全球造成重大经济损失和人员伤亡的自然灾害之一。有效预测和减轻其影响，本文综合分析了多项指标数据，并构建了**预测模型**。基于季风强度、地形排水等因素，通过数据清洗和相关性分析筛选出关键指标。运用各种神经网络模型，构建洪水预警模型，洪水概率预测模型。在数据分析部分，我们采用 Python 软件的 Seaborn 库，制作热力图、柱状图、折线图等**可视化**工具展示不同指标之间的相关性和分布特征。

针对问题一，基于 `train.csv` 数据集，分析 20 个不同指标与洪水概率的相关性。通过**相关性分析**以及可视化，提出了一系列洪水提前预防的建议，包括加强降水量和河流流量的实时监测、土壤湿度监测与调节、气象预报与警报系统的优化、防洪基础设施建设以及公众教育和应急演练。这些措施将有助于提高洪水预警的准确性和及时性，减轻洪水灾害的影响。

对于问题二，我们将 `train.csv` 中的洪水发生概率进行聚类分析，以识别高、中、低风险的洪水事件。首先，使用 K-means 聚类算法将数据分为三类，接着用 XGBoost 算法构建预警模型。最后，通过交叉验证的方法，进行模型的灵敏度分析，得出验证出的模型准确率为 96.515%。。

针对问题三，基于问题 1 中的指标分析结果，我们从 20 个指标中选取 5 个关键指标，建立洪水概率的预测模型。本组采用 MLP 神经网络模型来对选取的指标数据进行分析。通过利用测试集中的数据，**MLP 神经网络**使用反向传播算法来优化网络参数，以最小化预测输出与实际输出之间的误差。最终，我们验证模型的准确性，达到 99.97%，证明其能够有效地预测洪水发生的概率，为实际防洪提供了科学依据。

对于问题四，基于问题 3 中建立的洪水发生概率预测模型，我们对 `test.csv` 中的所有事件进行了洪水概率预测，并将预测结果填入 `submit.csv`。进一步绘制了 74 万件事件的洪水发生概率的直方图和折线图，我们根据 Kolmogorov-Smirnov 方法检验预测值洪水概率是否服从正态分布。

本文的研究结果不仅揭示了影响洪水发生的主要因素，还提供了一种有效的洪水预测方法，对防灾减灾具有重要的指导意义。未来的工作将进一步优化模型，并结合实时监测数据，提高预测的实时性和精确度。

关键词： 洪水灾害预测　可视化　相关性分析　K-means 聚类　MLP 神经网络

目　录

一、 问题重述

1.1 问题背景

洪水是世界上最严重的自然灾害之一，洪水事件对人类社会自然环境都形成了强烈的冲击。随着全球变暖，气温上升，人类活动的日益影响下，洪水的发生频率和强度都在显著上升，为此，对于洪水灾害的数据分析和预测是十分有必要的。为了有效进行分析和预测，我们需要依据各项指标，通过科学精确的方法建立洪水预测模型，从而提前采取相应措施来降低自然灾害带给人类的影响。

1.2 问题提出

根据以上背景，以及给出的三个附件，需要解决以下问题：

1. 分析附件'train.csv'中洪水概率与各指标的相关性，提出合理建议和措施。
2. 将洪水概率聚类成不同类别，选取合适指标，建立预警模型。
3. 基于相关性分析，选取合适指标建立概率预测模型，并简化该模型。
4. 基于预测模型，预测'test.csv'中的洪水概率，并分析结果1. 分析附件'train.csv'中洪水概率与各指标的相关性，提出合理建议和措施。
2. 将洪水概率聚类成不同类别，选取合适指标，建立预警模型。
3. 基于相关性分析，选取合适指标建立概率预测模型，并简化该模型。
4. 基于预测模型，预测 `test.csv` 中的洪水概率，并分析结果。

其中，`train.csv` 中给出了100多万组季风强度、地形排水等20个指标得分与洪水概率的数据；`test.csv` 中仅给出74万组20个指标得分。

二、 基本假设

1. 洪水发生的概率可以被明确区分为高、中、低风险类别
2. 过去洪水事件的特征和概率在未来将保持不变。
3. 附件'train.csv'和'test.csv'给出的数据准确无误，且涵盖多种不同的洪水事件，能够代表洪水发生的典型情况和异常情况
4. 所选的统计和机器学习模型适用于洪水灾害的概率预测。
5. 模型能够通过选择适当的指标和降维技术处理潜在共线性问题，使最终使用的指标在模型中是相对独立的。
6. 在计算指标权重和模型参数时，这些值在不同的样本和时间点上是稳定的，不会出现剧烈波动。

三、 符号说明

序号	变量名	所示含义
1	x_i	指标
2	Y	洪水发生概率
3	X	训练集中的数据集

序号	变量名	所示含义
4	G	3个不同风险等级簇
5	w_i	预测模型权重

四、 问题一的模型建立与求解

4.1 问题分析

通过全面而深入的定量分析，我们研究了二十个关键因素与洪水发生概率之间的关联性，这些因素包括季风强度、地形排水特性、河流管理措施、森林砍伐程度、城市化进程、气候变化趋势、大坝的质量状况、河流淤积现象、农业实践的影响、土地侵蚀程度、防灾措施的有效性、排水系统的效能、海岸线的脆弱性、滑坡风险、流域管理状况、基础设施的老化程度、人口分布及密度、湿地生态系统的健康状态、城乡规划的合理性以及政府政策导向。我们运用统计学工具，计算了这些因素与洪水发生概率之间的皮尔逊相关系数，这是一种衡量两个变量之间线性关系强度和方向的指标。通过这种方法，我们能够量化评估每一个因素对洪水风险的贡献度，确定哪些是主要的驱动因素，哪些影响较弱，从而揭示了它们在洪水生成机制中的相对重要性。

在数据分析的基础上，我们深入探究了各因素影响洪水概率的内在机理。例如，季风强度的增加可能会导致降雨量的显著上升，进而增加洪水发生的可能性；城市化的扩张往往伴随着绿地的减少，减少了水的自然渗透能力，增加了地表径流，从而加剧了洪水风险；而有效的河流管理和合理的流域规划，则可以降低洪水的概率。通过对这些因素的综合考量，我们得出了关于洪水风险的全新认识。

4.2 指标相关性判断

首先将20个指标分别设为随机变量$x_{i(i=1,2,...,20)}$，设洪水发生概率为Y。随机变量之间的相关系数矩阵或者协方差矩阵可以反应随机变量之间的相关，如果相关系数越大，那么随机变量之间的相关性越强；或$cov(x_i,Y),cov(Y,x_i)$越趋近于$\sqrt{D(x_i)}\sqrt{D(Y)}$，x_i与Y的相关性越强。

根据相关系数计算公式：（x_i,Y为两个随机变量）

$$R=\frac{cov(x_i,Y)}{\sqrt{D(x_i)}\sqrt{D(Y)}}$$

得到相关系数矩阵：$\begin{pmatrix}1 & R\\ R & 1\end{pmatrix}$，通过观察随机变量$x$与$Y$的相关系数$R$的大小就能够判断两者的相关性，即第$i$项指标与洪水发生概率的相关性。其中相关系数$R$值越大，则表明第$i$项指标与洪水发生概率的相关性越强。根据协方差中$cov(x_i,Y)$或$cov(Y,x_i)$越趋近于$\sqrt{D(x_i)}\sqrt{D(Y)}$，则随机变量$x_i$与$Y$的相关性越强。

现基于 SPSS 软件可分析得到各项指标之间的相关性：

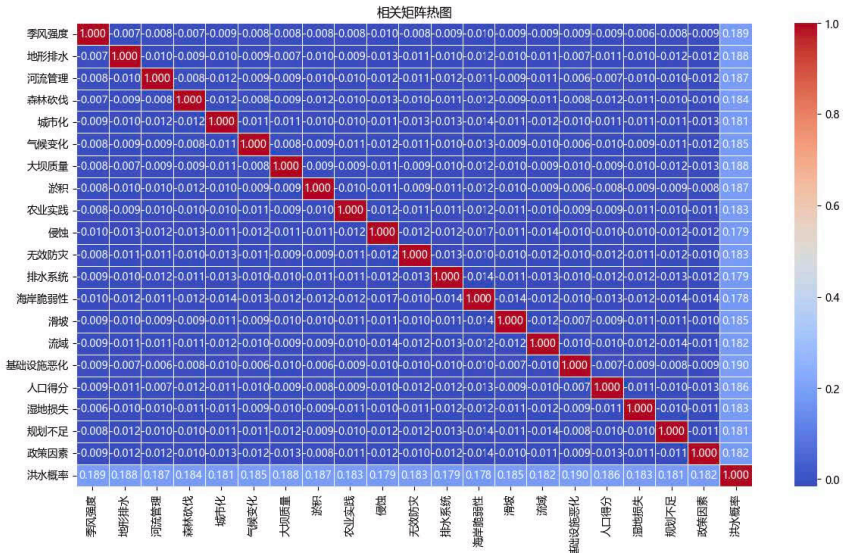


图 1　相关矩阵热力图

通过对表中详尽的数据进行细致分析，我们可以得出结论，各项指标之间的相关性表现得相当微弱。这意味着，在统计意义上，这些指标可以被视为几乎互不影响的独立变量。这种独立性表明，每一项指标的变化似乎并不直接影响其他指标的数值，它们各自遵循着自己的变化规律和模式，没有显示出明显的相互作用或连锁反应。这一发现对于理解洪水风险评估模型中的变量关系具有重要意义，它提示我们在构建预测模型时，可以将这些指标视为独立的输入因子，简化模型结构，提高计算效率。

然而，尽管各项指标彼此之间的直接联系较为松散，但它们与洪水发生概率之间的关联性却呈现出不同的面貌。通过计算各项指标与洪水概率之间的相关系数，我们发现某些指标与洪水的发生存在正相关或负相关的关系。例如，降雨量的增加可能与洪水概率呈正相关，而有效的排水系统则可能与洪水概率呈负相关。这些发现有助于我们识别出那些对洪水风险有显著影响的关键因素，为进一步的风险管理和减灾措施提供了科学依据。

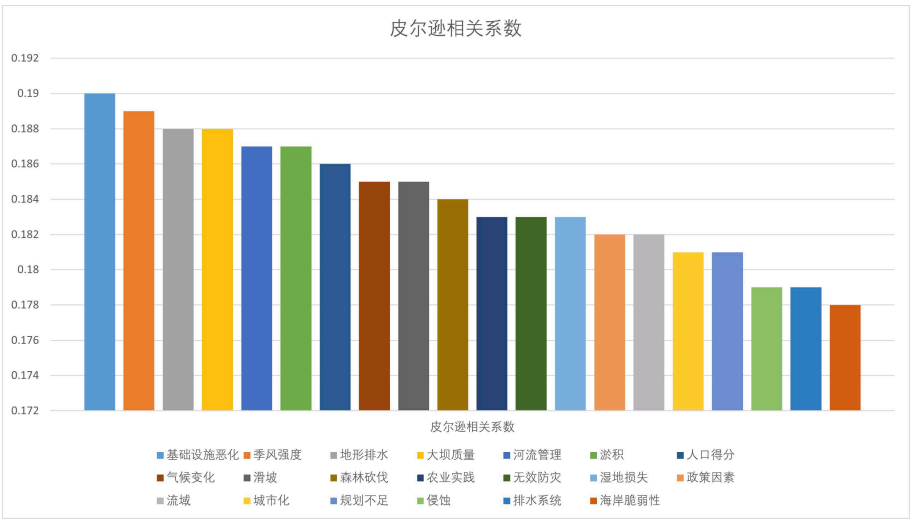


图 2　相关系数

结合上述分析，基础设施恶化、季风强度、地形排水这三个因素对洪水概率影响相对最大；而海岸脆弱性、排水系统、侵蚀这三个因素对洪水概率影响相对最小。

4.3 合理建议与措施

针对基础设施恶化、季风强度、地形排水等问题，现给出一下建议和措施：

1. 加强基础设施建设和维护：

定期检查和修复堤坝、水库、排水系统和其他水利工程设施。

升级老旧的排水系统，增加其容量和效率。

在关键区域建立防洪墙或提高现有防洪墙的高度和强度。

确保水闸和泵站的正常运行，以便在需要时迅速排放积水。

2. 改善城市规划和管理：

限制在洪水易发区域的开发建设，保持足够的绿地和开放空间以促进雨水渗透。

实施雨水收集和利用系统，减少对下水道系统的压力。

采用透水性材料铺设道路和人行道，增加地面的雨水吸收能力。

3. 生态保护和恢复：

保护和恢复湿地、河流沿岸和上游森林，这些自然生态系统可以减缓洪水流速并吸收多余水分。

实施梯田耕作、植被覆盖等措施，减少山坡地的水土流失，降低下游洪水风险。

4. 地形改造：

在地形允许的情况下，进行土地平整和地形调整，以改善排水条件。

在低洼地区建立蓄水池或人工湿地，用于暂时储存洪水，减轻主排水系统的压力。

五、 问题二的模型建立与求解

5.1 问题分析

为了深入探讨洪水概率的分布特性，本组采用了$K-means$聚类算法对洪水概率进行了分析。通过设定$K=3$，算法将数据集划分为三个具有不同风险等级的集群，并确定了相应的中心点，从而形成了三个等风险级。这一步骤为后续的风险评估提供了坚实的基础。

基于$K-means$聚类的结果，本研究进一步采用$XGBoost$算法对三个风险等级进行了模型构建。$XGBoost(eXtremeGradientBoosting)$是一种高效的梯度提升决策树算法，它能够集成多个弱学习器，通过迭代优化，逐步减少模型的偏差和方差，最终形成强学习器[1]。该算法在处理分类问题中展现出了强大的性能，因此非常适合用于构建洪水不同风险的预警评价模型。在构建模型的过程中，我们针对每个风险等级选取了合适的指标，确保模型能够准确地反映各个等级的风险特征。随后，通过将数据集代入模型进行训练，不断调整模型[2]，我们得到了针对不同风险等级的预警评价模型。

为了评估模型的有效性和稳定性，本研究还进行了模型的灵敏度分析。灵敏度分析是模型评估中的重要环节，它可以帮助研究者了解模型对于输入数据的微小变化的敏感程度，从而判断模型的稳健性。在本研究中，我们通过调整数据集中的参数，观察模型输出的变化情况，以评估模型的灵敏度。

其具体思路流程图如下：

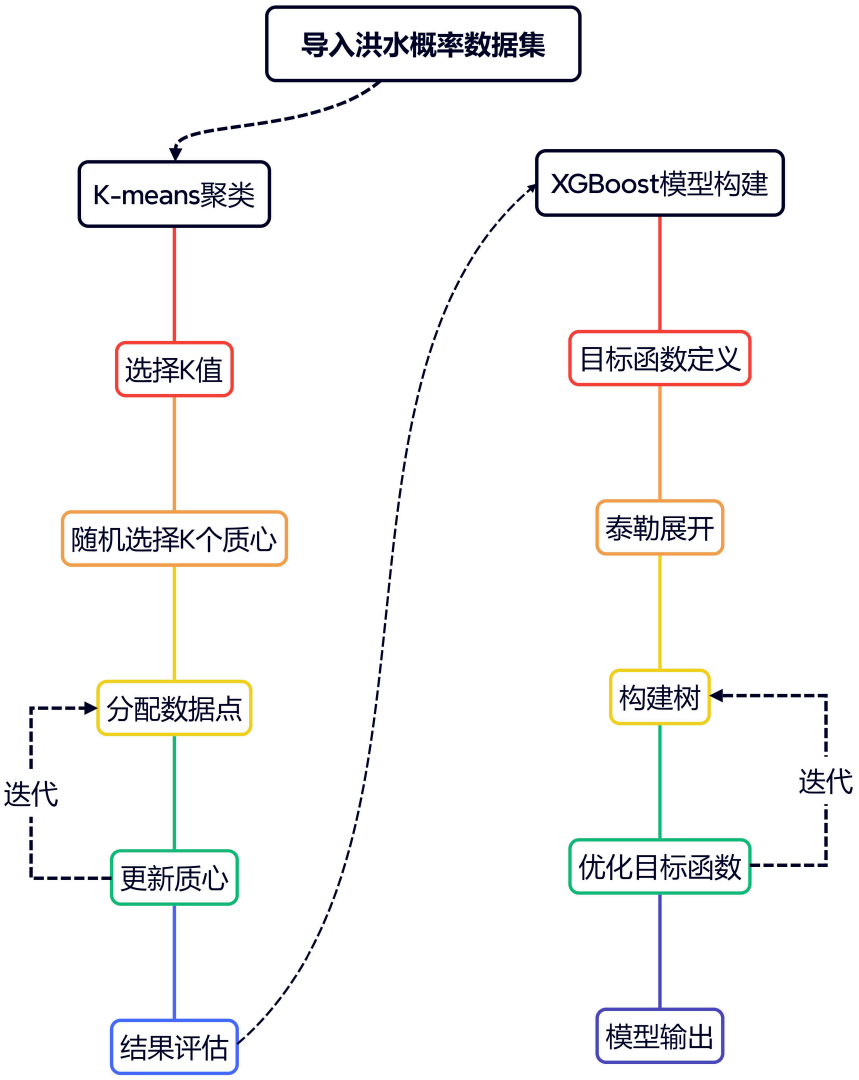


图 3　预警模型构建流程图

5.2 洪水风险预警模型构建

5.2.1 洪水概率分级

首先提取附件中提供的 `train.csv` 中的洪水发生概率的数据集：$X = \{x_1, x_2, ..., x_n\}$，并进行相应的数据处理，检验并排除异常数据后，将得到的数据集导入$K-means$聚类算法中，将洪水分为A、B、C三类，分别对应洪水发生概率：高风险、中

风险、低风险三种风险等级。即将数据集X分成3个不同的簇$G=\{g_1,g_2,g_3\}$，使得簇心位置不发生显著变化。

风险等级	A级	B级	C级
质心点	0.568401	0.504591	0.442312

5.2.2 分析各级指标特征

针对每个风险级别的洪水事件，本组首先计算了各个指标的平均值、中位数和标准差等基本统计量。这些统计量的计算为后续的风险评估提供了数据支持。

在模型构建过程中，我们首先计算了每个聚类的平均洪水概率。这一步骤使得我们能够了解不同聚类在洪水概率上的分布情况。随后，我们对聚类按照平均洪水概率进行了排序，从而确定了各个聚类在风险等级上的排列顺序。为了进一步细化风险评估，我们定义了一个风险等级列表，并设置了选择的特征数量。在本问题中，我们决定对每个风险级别选取个特征数量。这一决策是基于特征选择的复杂性和模型准确性之间的权衡考虑。最后，我们遍历了排序后的聚类索引，并对每个聚类找到了其特征值最大的前个指标特征。这一步骤确保了我们所选的特征能够有效地代表各个聚类的风险特点，从而提高了模型的预测能力。

通过以上步骤，我们选取了若干个洪水预测的指标：

风险等级	A级	B级	C级
指标一	大坝质量	海岸脆弱性	海岸脆弱性
指标二	基础设施恶化	大坝质量	侵蚀
指标三	河流管理	侵蚀	排水系统
指标四	季风强度	无效防灾	城市化

5.2.3 建立预警模型

基于聚类分析的结果，基于2.2.2节所选指标特征，选取与洪水发生概率密切相关的指标。接着，我们利用$XGBoost$算法（梯度提升决策树算法），来计算模型中各指标的权重。该算法的优势在于它能自动学习特征的重要性，从而准确地为每个指标赋予合适的权重。在$XGBoost$模型训练过程中，算法会评估每个指标对模型预测贡献的大小，并据此分配权重。权重较大的指标对洪水发生概率的预测具有更大的影响力，从而在预警模型中占据更重要的位置。结合选定的指标和通过$XGBoost$计算出的权重，我们建立了一个用于预测不同风险等级的洪水预警模型。该模型通过整合指标的加权值来评估洪水发生的风险等级，为决策者提供了一个量化的、基于数据驱动的决策工具。

$$Y=\sum_{i=1}^{m}w_i\cdot x_i$$

5.2.4 灵敏度分析

为评估洪水风险预警模型的灵敏性，采用了交叉验证的方法，将数据集`train.csv`导入训练模型中。通过这种方法，我们能够确保模型在不同子集上的表现具有一致性，并

且减少模型过拟合的风险。基于先前的$K-means$聚类分析识别出的三个不同风险等级，对数据集进行分层抽样，以确保每个风险等级都得到充分的代表性。

在交叉验证的过程中，数据集被分为多个子集，每个子集都有机会作为测试集来评估模型的性能，而其余的子集合并为训练集用于模型的训练。这种反复的训练和测试，使我们能够全面地评价模型的预测能力。

在完成交叉验证后，我们将统计模型的精准度。这包括计算模型在各个风险等级上的准确率、召回率以及F_1分数等指标。这些指标能够为我们提供模型精准度的量化度量，从而让我们可以比较模型在不同风险等级上的表现差异。

基于上述灵敏度分析，最终得出模型准确率为96.515%。

六、 问题三的模型建立与求解

6.1 问题分析

题目要求在第一问的相关性基础上，建立洪水概率模型。而洪水发生与20个指标均有关系，模型过于庞大，故从中选取相关性较强的几组指标建立模型。则洪水概率可以直接由各项指标的值进行线性运算得出。紧接着，为进一步简化模型，仍是依据相关系数仅选取5项指标，对模型进行修改。同时，我们考虑到由于选取指标的个数差异会导致建立的洪水发生概率模型的精度产生不同，即分析选取5组指标建立的模型与原来选取的多组指标所建立的模型进行精度比较。从而可以得到选取相关性较强的指标对于模型的影响。

6.2 基于MLP神经网络模型建立

6.2.1 指标选取

根据第一问分析所得的相关系数大小来选取与洪水发生概率相关性相对强的指标x_i。在第二小问中仅选取出前5个相关性最强的指标。即在第一小问中选取：基础设施恶化、季风强度、地形排水、大坝质量、河流管理、淤积、人口得分、气候变化、滑坡9个指标；在第二小问中选取：基础设施恶化、季风强度、地形排水、大坝质量、河流管理5个指标。

6.2.2 MLP神经网络

MLP神经网络即是对已有的数据进行处理，拟合出来一个比较好的模型，再通过添加一些其他的数据来对拟合出来的模型进行误差分析，修改原来拟合出的模型，使模型的误差降低。其工作原理分为两个部分：

1. 前向传播：输入数据通过网络的输入层，经过隐藏层中的加权求和和激活函数处理，最终得到输出层的结果。
2. 反向传播：MLP神经网络通常使用反向传播算法来优化网络参数，以最小化预测输出与实际输出之间的误差。反向传播通过计算损失函数的梯度，然后沿着梯度的反方向调整权重和偏置。

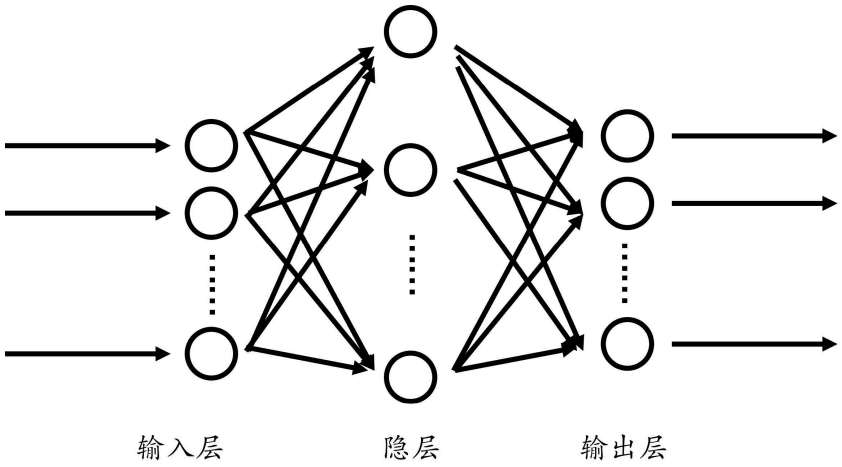


图 4　MLP 神经网络示意图

6.2.3 MLP模型建立与评估

首先导入 `train.csv` 数据集，并将其划分为80%的训练集和的20%测试集，同时设置随机状态为1以保证结果的可重复性；接着通过计算数据的均值与标准差，将特征数据进行标准化，以保证模型训练过程中的数值稳定性；然后创建MLP模型，设置三层隐藏层分别为200、100、50，并将双曲正切函数tanh作为激活函数，并确定学习率初始化值、最大迭代次数、正则化参数、随机状态和开启早停策略以防止过拟合。基于上述灵敏度分析，最后得出评估模型精度为99.97%。

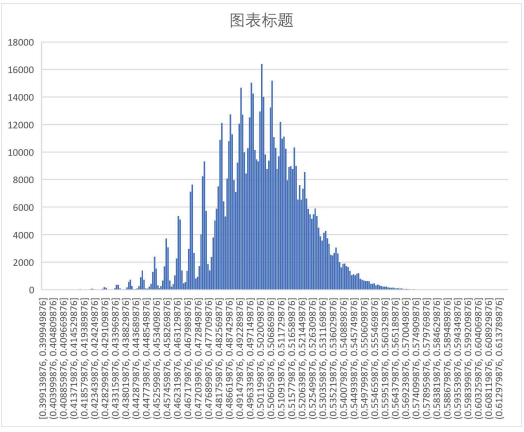
七、 问题四的模型建立与求解

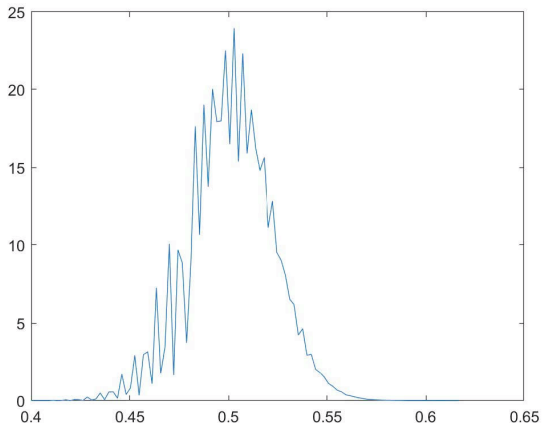
7.1 问题分析

根据题目要求利用第三问中所建立的模型进行预测，将所得到的预测结果用直方图和折线图的方式直观地表示出来，同时分析是否符合正态分布。从一般生活中的情形中的随机事件均服从正态分布，因此先默认预测出的结果服从正态分布，再验证其服从正态分布。

7.2 图像绘制

概率直方图　　概率折线图





7.3 基于$Kolmogorov-Smirnov$正态分布检验

7.3.1 $Kolmogorov-Smirnov$正态分布检验

$Kolmogorov-Smirnov(KS)$检验是一种非参数统计检验方法，用于比较一个样本分布与一个理论分布或两个样本之间的差异。它基于累积分布函数(CDF)的差异来评估两个分布的相似性或者一个样本是否来自于一个特定的分布。

7.3.2 数据检验

根据$Kolmogorov-Smirnov$检验中的单样本(KS)检验来判断预测值洪水概率是否服从正态分布。单样本(KS)检验时，先得到一个标准正态分布随机变量：

$$Z \sim N(0,1)$$

假设洪水概率预测值为随机变量X，由于默认$X \sim N(E(X), D(X))$，由预测出来的数据可以得到洪水概率预测值的方差$E(X)$和期望$D(X)$，将洪水概率X转化为标准正态分布随机变量：

$$Y = \frac{X - E(X)}{\sqrt{D(X)}}$$

不妨令随机变量$W = Y - Z$计算得到W的期望$E(W)$和方差$D(W)$大小可以判断洪水概率是否服从正态分布。可以观察得到$E(W)$和$D(W)$的大小都趋近于0，则可以判断出X服从正态分布。通过 `matlab` 中 `kstest` 函数检验得到：KS统计量（H值）为0，p值为1，这两个结果表明预测出的样本数据非常接近正态分布，而且差异小到不足以拒绝正态分布的原假设[3]。

八、 模型评价

8.1 模型优点

在问题二中，我们采用$K-means$聚类算法将洪水概率分成三类，这类算法简单、易于实现且时间复杂度$\mathcal{O}(tkmn)$适中，在处理大量数据时又较好的伸缩性。在后续预警预测模型中，采用$XGBoost$算法，该算法作为一种经过优化的分布式梯度提升库，有较高的可移植性，且对缺失数据有较高的预测能力。

在问题三中，采用了MLP神经网络模型，可通过多个隐藏层和非线性激活函数，学习到复杂的数据特征，从而提高模型的表达能力，且能逼近任何连续函数，且该模型提供丰富的参数设置，可根据模型需求进行灵活调控，从而优化模型性能。

8.2 模型缺点

在模型建立过程中，存在大量假设，使得该预测模型在实践生活使用过程中，存在一项模型预测的偏差；并且，所采用的模型，对数据初始值比较敏感，不具有较强的泛用性，故对极端天气或人为导致的洪水灾害无法进行准确预测。

九、 参考文献

[1] 王宇宁, 周凯, 沈守枫. 一种基于 XGBoost 的状态转移预测模型[J]. 浙江工业大学学报, 2024, 52(3): 275-279.

[2] 刘明锦, 张智涌, 王宾. 基于机器学习和大数据的山洪预测模型研究[J]. 四川水利, 2022, 42(6): 107-113.

[3] MATHWORKS. One-sample Kolmogorov-Smirnov test - MATLAB kstest - MathWorks 中国[EB/OL]. https://ww2.mathworks.cn/help/stats/kstest.html?s_tid=doc_ta,2024.8.1.

十、 附录

附录 1: 问题一相关矩阵热图

```
import matplotlib
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
matplotlib.rcParams['font.family']='Microsoft YaHei'
file_path = r"C:\Users\典总\Desktop\train.xlsx"
data = pd.read_excel(file_path)
plt.figure(figsize=(10, 8))
correlation_matrix = data.corr()
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm',
fmt='.3f',linewidths=.5)
plt.title('相关矩阵热图')
plt.show()
sns.pairplot(data)
plt.suptitle('散点图矩阵', y=1.02)
plt.show()
```

附录 2: 问题二聚类及预警模型构建

```
import pandas as pd
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
import numpy as np
import xgboost as xgb
from sklearn.metrics import accuracy_score
from sklearn.model_selection import GridSearchCV
from sklearn.model_selection import StratifiedKFold
import matplotlib.pyplot as plt
import seaborn as sns
df = pd.read_csv('E:/python/train.csv', encoding='GBK')
columns_to_analyze=['季风强度', '地形排水', '河流管理', '森林砍伐',
'城市化', '气候变化', '大坝质量', '淤积', '农业实践', '侵蚀', '无效防
灾','排水系统', '海岸脆弱性', '滑坡', '流域', '基础设施恶化', '人口得
分', '湿地损失', '规划不足', '政策因素', '洪水概率']
sc = StandardScaler()
df_sc = sc.fit_transform(df[columns_to_analyze])
kmeans = KMeans(n_clusters=3, random_state=1,
n_init=10).fit(df_sc)
df['risk_cluster'] = kmeans.labels_
cluster_features = df.groupby('risk_cluster')
[columns_to_analyze].mean()
cluster_centers = kmeans.cluster_centers_
cluster_centers = sc.inverse_transform(cluster_centers)
cluster_probabilities = df.groupby('risk_cluster')['洪水概率'].
mean()
sorted_clusters = np.argsort(cluster_probabilities)
risk_levels = ['C', 'B', 'A']
select_features_count = 4
for i, cluster_idx in enumerate(sorted_clusters):
select_features =
```

```
cluster_features.columns[np.argsort(cluster_features.loc[cluste
r_idx].values)[-select_features_count:]]
print(f"风险等级:{risk_levels[i]}")
print(cluster_features.loc[cluster_idx,
select_features].to_string(), "\n")
X = df[columns_to_analyze]
y = df['risk_cluster']
model_train = xgb.DMatrix(X, label=y)
params = {
'max_depth': 3,
'eta': 0.1,
'objective': 'multi:softmax',
'num_class': 3,
'eval_metric': 'mlogloss'
}
bst = xgb.train(params, model_train, num_boost_round=100)
predict = xgb.DMatrix(X)
y_predict = bst.predict(predict)
df['predicted_risk_cluster'] = y_predict
accuracy = accuracy_score(df['risk_cluster'],
df['predicted_risk_cluster'])
print(f"模型准确率:{accuracy:.3%}")
```

附录 3: 问题三洪水概率预测模型构建

```
import pandas as pd
from sklearn.neural_network import MLPRegressor
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_absolute_percentage_error
from sklearn.preprocessing import StandardScaler
df = pd.read_csv('E:/python/9.csv', encoding='GBK')
columns_to_analyze = ['季风强度', '地形排水', '河流管理', '气候变化',
'大坝质量', '淤积', '基础设施恶化', '人口得分', '洪水概率']
X = df[columns_to_analyze[:-1]]
y = df['洪水概率']
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=0)
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
model = MLPRegressor(hidden_layer_sizes=(200, 100, 50),
activation='tanh',
learning_rate_init=0.01, max_iter=500,
alpha=0.0001, random_state=1, early_stopping=True)
model.fit(X_train, y_train)
y_predict = model.predict(X_test)
error_rate = mean_absolute_percentage_error(y_test, y_predict)
accuracy = 100-error_rate
print(f"准确率为: {accuracy:.2f}%")
```

附录 4: 问题三简化模型指标为 5

```
import pandas as pd
from sklearn.neural_network import MLPRegressor
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_absolute_percentage_error
from sklearn.preprocessing import StandardScaler
df = pd.read_csv('E:/python/5.csv', encoding='GBK')
columns_to_analyze = ['季风强度', '地形排水', '河流管理', '大坝质量',
'基础设施恶化', '洪水概率']
X = df[columns_to_analyze[:-1]]
```

```
y = df['洪水概率']
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=0)
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
model = MLPRegressor(hidden_layer_sizes=(200, 100, 50),
activation='tanh',
learning_rate_init=0.01, max_iter=500,
alpha=0.0001, random_state=1,
model.fit(X_train, y_train)
y_predict = model.predict(X_test)
error_rate = mean_absolute_percentage_error(y_test, y_predict)
accuracy = 100-error_rate
print(f"准确率为：{accuracy:.2f}%")
```

附录 5: 问题四概率预测

```
import pandas as pd
from sklearn.neural_network import MLPRegressor
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_absolute_percentage_error
from sklearn.preprocessing import StandardScaler
df = pd.read_csv('E:/python/5.csv', encoding='GBK')
columns_to_analyze = ['季风强度', '地形排水', '河流管理', '大坝质量',
'基础设施恶化', '洪水概率']
X = df[columns_to_analyze[:-1]]
y = df['洪水概率']
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=0)
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
model = MLPRegressor(hidden_layer_sizes=(200, 100, 50),
activation='tanh',
learning_rate_init=0.01, max_iter=500,
alpha=0.0001, random_state=1, early_stopping=True)
model.fit(X_train, y_train)
y_predict = model.predict(X_test)
error_rate = mean_absolute_percentage_error(y_test, y_predict)
accuracy = 100-error_rate
print(f"准确率为：{accuracy:.2f}%")
new_data = pd.read_csv('E:/python/test.csv', encoding='GBK')
new_data = new_data[columns_to_analyze[:-1]]
new_data_scaled = scaler.transform(new_data)
new_predictions = model.predict(new_data_scaled)
new_data['洪水概率'] = new_predictions
new_data.to_csv('E:/python/predict.csv', index=False)
```

附录 6: 正态分布检验

```
clc,clear,close all
data = table2array(readtable("predict.csv",'Range'  ,'F2:F1048576'));
[h_ks,p_ks]=kstest(data);
disp(h_ks);
disp(p_ks);
```