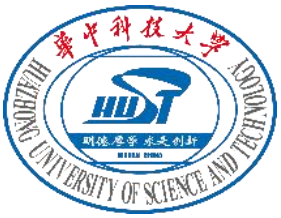


中国研究生创新实践系列大赛

中国光谷·“华为杯”第十九届中国研究生数学建模竞赛

学　　校	北京石油化工学院
参赛队号	**22100170002**
队员姓名	**1.邢晓龙**
	2.王利猛
	3.张婧

中国研究生创新实践系列大赛

中国光谷·“华为杯”第十九届中国研究生数学建模竞赛

题 目 基于草原放牧策略研究的优化建模

摘 要：

草原放牧策略存在较严重的不合理性，系统破坏严重，采取合理的放牧管理策略，确定适当的放牧方式和放牧强度，使得获得更高的放牧收益的同时又达到可持续发展的目的。为了给放牧的稳定和持续性提供理论依据和方法，在放牧过程中，根据研究的因素找到放牧方式、放牧强度与环境之间的关系，找到可以稳定草原生态平衡又能保持经济增长的方法。建立微分方程模型，利用微分方程稳定性理论，研究平衡状态的稳定性，并且数据可视化作图分析得到结论通过合理放牧来维持草原生态平衡，并提出了有效的放牧措施。同时根据附件文件中基本数据和监测点数据的宏观层面的规律性、微观层面的差异性，建立具有针对性的多个数学模型。本文所做的工作可概括以下几点：

问题一：针对问题一的要求，以放牧方式和放牧强度为自变量，土壤湿度和植被生物量作为因变量进行机理分析。查阅相关文献了解放牧策略对土壤湿度和植被生物量影响，主要使用文献数学模型和数据来建立数学模型。其中放牧方式的影响体现在整体上设为系数,不同放牧方式根据文献中的对有机物影响效果进行取值。放牧强度对植被生物量的关系，通过植被指数间接建立数学模型。放牧强度与土壤湿度的关系，则是通过简化土壤-植被-大气系统的水平衡基本方程建立数学模型。

问题二：按年月份整合附件 3，4，8 为一张表，经查阅资料选定平均气温(附件 8)、平均降雨量（附件 8）、蒸散量(附件 4)，平均风速(附件 8)和平均站点气压(附件 8)为不同深度土壤湿度的气候因素影响量。这些量与年份序列相关。针对不同的月份组，可以利用此特性预测出该月所需年份(1-3 月份需要 2023 年，4-12 月份需要 2022 和 2023 年)的气候因素影响量。进一步，利用已经预测得到的气候因素影响量。继续预测出不同深度的土壤湿度。本题在实现的过程中使用随机森林和提升决策树同时进行预测，最终得到的两份结果。同时给定了一个基于欧式距离定义的评判指标a因子及标准，经比较得知随机森林预测效果更好，将预测结果填入表 5.12。

问题三：本题预先做出假设如下：放牧策略的放牧方式固定为选择划区轮牧，仅用 12 个放牧小区号表征；放牧策略的放牧强度分为 4 类。选择附件 14 15，按共有的 16 18 20 年份及放牧小区号进行合并。进行数据清洗后保留放牧强度，放牧小区，5 个化学性质列。对离散变量放牧小区进行 one-hot 编码，放牧强度规定为固定数值(0 2 4 8)，构建放牧小区和放牧强度对 5 个化学性质的 5 个回归决策树模型。使用 K 折交叉验证及网格搜索算法进行模型优化超参。对该模型 r2_score 进行分析认定满足题干所需。模型预测结果见表 7.11。

问题四：根据题目要求根据，首先使用主成分分析法计算沙漠化影响因素的权重，使用牧畜量因素除以草原面积作为放牧强度，构建带有放牧强度自变量的沙漠化程度指数预测模型表达式。由于土壤湿度与放牧强度没有明确的对应关系，使用多元回归方法，获得放牧强度与土壤湿度的关系。构建带有放牧强度自变量的板结化指数预测模型。模型设定好参数后进行实践，检验模型额正确性。最后通过预测不同放牧强度下的沙漠化指数和板

结化指数，选取使两者之和最小的放牧强度作为结果

问题五：根据本题要求，需要在给定降水量（300mm，600mm、900 mm 和 1200mm）情形下，在环境可持续发展的情况下找到一个放牧羊数量的最大阈值。衡量环境可持续发展的定量指标可以使用问题 4 中的综合指数。本文选用 2019 年的已知数据代入到公式，降水量和放牧强度作为自变量对评价指数进行预测，设定 0.5 作为归一化后指数的阈值然后利用枚举法，求得放牧强度的最大阈值为 7.9。

问题六：本题基于问题 4 放牧方案，即放牧强度为中度放牧(MGI)，放牧方式为划区轮牧(没有轮次，仅有放牧小区 G8 G11 G16)。并且假设附件 13 对问题的求解没有影响。土地状态用问题 4 得到的有机物含量和土壤湿度来表征。生成各放牧小区在中度放牧的情况下历年来 9 月份的土地状态。土地状态也与年份序列相关。本题用 LSTM 多变量模型进行预测，同时输出评价指标 MAPE，RMSE，MAE 对模型进行打分。最后以二维点图方式表示 2023 年九月份的土地状态见 10.3。

关键词：放牧策略；土壤湿度；植被生物量；沙漠化与板结化；

目录

一、问题重述

1.1 问题背景

草原作为世界上分布最广的重要的陆地植被类型之一，分布面积广泛。中国的草原面积为 3.55 亿公顷，是世界草原总面积的 6%~8%，居世界第二。此外，草原在维护生物多样性、涵养水土、净化空气、固碳、调节水土流失和沙尘暴等方面具有重要的生态功能。自 2003 年党中央、国务院实施“退牧还草”政策以来，在保护和改善草原生态环境、改善民生方面取得了显著成效。“退牧还草”并不是禁止放牧，除了部分区域禁牧外，很多草原实行划区轮牧以及生长季休牧。合理的放牧政策是带动区域经济、防止草原沙漠化及保障民生的关键，放牧优化问题的研究也为国家、政府制定放牧政策和草原管理决策提供科学的依据。

中国草原主要分为温带草原、高寒草原和荒漠草原等类型。内蒙古锡林郭勒草原是温带草原中具有代表性和典型性的草原，是中国四大草原之一，位于内蒙古高原锡林河流，地理坐标介于东经 110° 50′ ~119° 58′ ,北纬 41° 30′ ~46° 45′ 之间，年均降水量 340mm。内蒙古锡林郭勒草原不仅是国家重要的畜牧业生产基地，同时也是重要的绿色生态屏障，在减少沙尘暴和恶劣天气的发生方面发挥着作用，也是研究生态系统对人类干扰和全球气候变化响应机制的典型区域之一和国际地圈—生物圈计划（IGBP）陆地样带—中国东北陆地生态系统样带（NECT）的重要组成部分。

1.2 问题提出

问题 1. 从机理分析的角度，建立不同放牧策略（放牧方式和放牧强度）对锡林郭勒草原土壤物理性质（主要是土壤湿度）和植被生物量影响的数学模型。

问题 2. 请根据附件 3 土壤湿度数据、附件 4 土壤蒸发数据以及附件 8 中降水等数据，建立模型对保持目前放牧策略不变情况下对 2022 年、2023 年不同深度土壤湿度进行预测，并完成表 1。

问题 3.从机理分析的角度，建立不同放牧策略（放牧方式和放牧强度）对锡林郭勒草原土壤化学性质影响的数学模型。并请结合附件 14 中数据预测锡林郭勒草原监测样地(12 个放牧小区)在不同放牧强度下 2022 年土壤同期有机碳、无机碳、全 N、土壤 C/N 比等值，并完成表 2。

问题 4、利用沙漠化程度指数预测模型和附件提供数据（包括自己收集的数据）确定不同放牧强度下监测点的沙漠化程度指数值。并请尝试给出定量的土壤板结化定义，在建立合理的土壤板结化模型基础上结合问题 3，给出放牧策略模型，使得沙漠化程度指数与板结化程度最小。

问题 5、锡林郭勒草原近 10 的年降水量（包含降雪）通常在 300 mm ~1200 mm 之间，请在给定的降水量（300mm，600mm、900 mm 和 1200mm）情形下，在保持草原可持续发展情况下对实验草场内（附件 14、15）放牧羊的数量进行求解，找到最大阈值。（注：这里计算结果可以不是正整数）

问题 6、在保持附件 13 的示范牧户放牧策略不变和问题 4 中得到的放牧方案两种情况下，用图示或者动态演示方式分别预测示范区 2023 年 9 月土地状态（比如土壤肥力变化、土壤湿度、植被覆盖等）。

二、问题分析

2.1 问题一

针对问题一的要求，需要查阅相关文献对放牧策略对土壤湿度和植被生物量影响进行理分析,主要使用文献数学模型和数据来建立数学模型。放牧策略主要通过影响有机物含量进而影响植被生物量以及土壤湿度。经分析放牧方式和放牧强度为自变量，土壤湿度和植被生物量作为因变量。其中放牧方式的影响体现在整体上设为系数,不同放牧方式根据文献中的对有机物影响效果进行取值。放牧强度的影响更加复杂，需要利用所查文献公式以及数据对植被生物量和土壤湿度建立数学模型。同时使用多元回归模型，用作对照实验。

2.2 问题二

结合题目，查看并理解附件 3，4，8 各列数据含义。可以发现：附件 3,4 存在 4 个共同列，分别是年份、月份、经度和纬度。附件 8 虽然是以文件夹的方式给出的，但是以年份命名的单独文件，单独文件内也包含了以上四列。并且附件 3,4,8 四列数据完全对应，即记载了自 2012 年 1 月至 2022 年 3 月的所有数据。由此可以将所有数据整合到一张表，除去经纬度，以月份为组，其中 1-3 月每月共有 11 条即 11 年数据，4-12 月份每月共有 10 条即 10 年数据，共 123 条数据。

在该表的每个月份组内，数据按年份升序排列。经查阅资料可知，每个月份的土壤蒸发及降水等数据应该与年份序列相关。所以针对不同的月份组，可以利用此特性预测出该月所需年份(1-3 月份需要 2023 年，4-12 月份需要 2022 和 2023 年)的土壤蒸发及降水等数据。进一步，利用已经推测得到的数据继续预测出不同深度的土壤湿度。

为了得到更优的湿度预测数据，本题在实现的过程中使用两种不同的分类器同时进行预测，并且查阅资料，选出五个与湿度相关的属性值作为模型的输入特征。对于最终得到的两份结果，基于欧式距离定义评判指标及标准，选取最优的一份结果填入表 5.5 中。

2.3 问题三

本题采用的附件为 14,15。其中附件 14 包含对放牧策略和化学性质的信息。但是它的数据量比较少，单用一个很有可能会导致模型欠拟合。所以考虑采用附件 15。附件 15 包含放牧策略。后续会将二者合并以扩充数据量。

观察到附件 14 包含放牧小区，放牧强度两列，附件 15 包含放牧小区，轮次，处理 3 列。这些都是描述放牧策略的数据。所以可以假设本题放牧策略的放牧方式为选择划区轮牧，用附件 14 15 的放牧小区和 15 的轮次表征；放牧策略的放牧强度用附件 14 的放牧强度和 15 的处理表征。注意到放牧强度(NG LGI MGI HGI)和处理(无牧（0 天） 轻牧（3 天）中牧（6 天）重牧（12 天）)完全对应，故在后续的合并操作中仅保留前者作为放牧强度的唯一表征即可。

为了数据有效，取附件 14 15 共有的 16 18 20 年数据，以“放牧小区+年份”为索引进行合并。处理缺失值及无关项。为了补全表中的数据，人为保留放牧策略和化学性质的相关项。其中放牧策略仅为放牧强度和小区(取消轮次)，化学性质为'SOC 土壤有机碳', 'SIC 土壤无机碳','STC 土壤全碳', '全氮 N', '土壤 C/N 比'。使用 5 个回归决策树进行拟合预测。并将结果填入表格中。

2.4 问题四

本题是由两小问构成，第一问求解沙漠化程度指数值，根据扩展阅读第三条沙漠化程度指数预测模型表达式确定主要影响因素进行求解，分别为风速、降水、气温、植被盖度、地表水资源、地下水位、人口数量、牲畜数量、社会经济水平，由于附件中没有直接提供植被盖度、地下水位和地表水资源的数据，分别选取植被指数、10cm 湿度代替植被盖度和地下水位，未找到合适的地下水位代替数据，因此忽略其影响。根据附件 3、4、6、8、9 提供的数据，首先对数据进行遍历，检验其是否存在如数据值缺失、超出操作变量范围等异常值并剔除，然后将得出的结果保存在 Q4data.csv 文件中。第二问则根据土壤板结化公式确定主要影响因素，利用主成分分析法(PCA)分别确定各个因素权重，建立土壤板结化指数模型。最终得到沙漠化程度指数与板结化程度最小时的放牧策略。

2.5 问题五

据本题要求，需要在给定降水量（300mm，600mm、900 mm 和 1200mm）情形下，在环境可持续发展的情况下找到一个放牧羊数量的最大阈值。衡量环境可持续发展的定量指标可以使用问题 4 中的综合指数。求解该问题可以通过设定一个综合指数阈值，在不超过阈值的约束条件下，规划处四种降水量的情形下放牧强度最大值。本文利用枚举法，进行求解。同也可以考虑时空因素，分析得到各个月份适合放牧的月份有哪些。

2.6 问题六

本题基于问题 4 放牧方案，即放牧强度为中度放牧(MGI)，放牧方式为划区轮牧(没有轮次，仅有放牧小区 G8 G11 G16)。并且假设附件 13 对问题的求解没有影响。土地状态用问题 4 得到的有机物含量和土壤湿度来表征。

依次选取问题四输出的在中度放牧情况下不同放牧小区 9 月份的所有数据并按年份进行合并。这样得到的数据即为该放牧小区在中度放牧的情况下历年来 9 月份的土地状态。和问题 2 类似，土地状态也与年份序列相关。这次该用 LSTM 多变量模型进行预测，同时输出评价指标 MAPE，RMSE，MAE 对模型进行打分。最后以二维图方式表示 2023 年九月份的土地状态。

三、模型假设

1. 问题二假设：

假设所有数据有效，不存在缺失值。

2. 问题三假设：

放牧策略的放牧方式为选择划区轮牧，仅由 12 个放牧小区(G6 G8 G9 G11 G12 G13 G16 G17 G18 G19 G20 G21)决定，与轮次无关；放牧强度为 4 种(对照 NG，轻度放牧强度 LGI，中度放牧强度 MGI，重度放牧强度 HGI)。

3. 问题六假设：

附件 13 的使用对解题没有实质性的影响，算法实现中可以忽略对附件 13 的使用。土地状态可用问题 4 得到的有机物含量和土壤湿度来表征。

四、符号说明

本文所使用的符号系统及其解释如表 4.1 所示。

表 4.1 本文所使用的部分符号说明

符号	符号说明
NDVI	植被指数
R	分别为入和出径流量
T	平均气温(℃)
PL	降水量(mm)
KN	平均风速(knots)
V	畜牧量
RW	人口
G	经济收入
Q1	10cm 湿度(kg/m2)
Q2	40cm 湿度(kg/m2)
Q3	100cm 湿度(kg/m2)
Q4	200cm 湿度(kg/m2)
E	土壤蒸发量(mm)
LAI	叶面积指数
G	径流量
O	有机物含量
NDVI	植被指数
R	分别为入和出径流量

注：考虑到全文连续性，其他未在表 4.1 中列出的符号将在建模和求解过程中给出解释说明。

五、问题一模型的建立与求解

针对问题一分析得到，放牧方式和放牧强度作为自变量，草原土壤物理性质和植被生物量作为因变量，查阅相关文献对自变量进行定量分析。在查找相关文献遇到数据集不开源，公式推导不完整的原因，决定使用现有的公式和数据进行分析建模。最后编写相应的matlab 程序实现数学模型。

5.1 模型的建立

先考虑自变量放牧方式和放牧强度，其中放牧方式和放牧强度中的轻度放牧因素重复，所以不重复考虑轻度放牧的影响。

放牧方式根相关文献得知，碳（C）、氮（N）和磷（P）是组成有机体的基础元素,也是植物生长所必需的元素和土壤养分的主要组成部分[1]。因此选用土壤中碳（C）、氮（N）和磷（P）的含量作为不同放牧方式的效果指标。

选用呼伦贝尔新巴尔虎左旗 4 种不同放牧制度下草地土壤化学计量特征相关数据作为依据,对不同放牧方式对植被生物量影响进行分析[3]。相关数据如下图 5.1 所示。

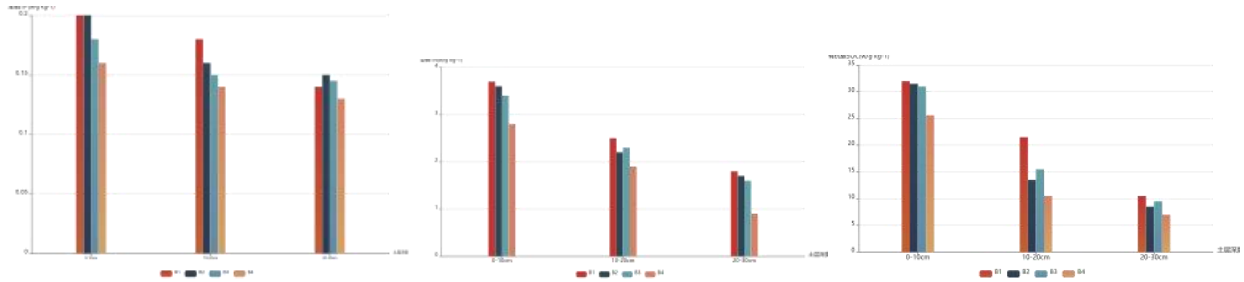


图 5.1 不同放牧方式下的土壤碳、氮磷含量

不同的放牧方式下，禁牧区的 SOC、TN 和 TP 的含量最高，植被生长状况相对最好，而常牧区最低，休牧区和轮牧区介于二者之间. 土壤碳和氮含量随着土层深度增加显著降低，而磷含量受土壤深度的影响较小.因此设放牧方式影响系数为 BE，取值范围为[0,1]，根据不同放牧方式 SOC、TN 和 TP 三者总量进行归一化得到相应的系数取值。

归一化公式：

$$BE=\frac{x-x_{min}}{x_{max}-x_{min}} \tag{1}$$

其中 x 为不同方式下的 SOC、TN 和 TP 三者的和，x_{min} 是最小值，x_{max} 是最大值。

得到四种放牧方式影响系数 BE 的取值为

$$BE=\begin{cases}1 & B1\text{禁牧}\\ 0.85 & B2\text{休牧}\\ 0.89 & B3\text{轮牧}\\ 0.68 & B4\text{常牧}\end{cases} \tag{2}$$

对于放牧强度，设放牧强度为F，表示每公顷牧羊数。根据题目做如下的划分：对照（NG，0 羊/天/公顷）、轻度放牧强度（LGI，1-2 羊/天/公顷）、中度放牧强度（MGI，3-4 羊/天/公顷）和重度放牧强度（HGI，5-8 羊天/公顷）。S 的取值范围为[0,8]。

5.2 放牧策略对植被生物量的影响

设因变量植被生物量为w，根据题目中拓展阅读中的放牧与植物生长之间方程可知，植被生物量受到载畜率的影响。载畜率可以用附件 2 中统计年检中获得 2012 年到 2020 年的家畜量除以草场面积获得，放牧强度F本质近似载畜率 V。题目给数据中没有植被生物量w数据，但具有植被指数$NDVI$。查阅相关文献可知植被指数$NDVI$与植被生物量w密切相关，利用 NDVI 三次多项式模型模拟典型草原植被生物量是一种实用的方法[4]。其转换公式为：

$$w = 3000 \cdot NDVI^3 + 7252 \cdot NDVI^2 + 4164 \cdot NDVI - 1594 \tag{3}$$

其中w是植被生物量，$NDVI$为植被指数

因此可以分析放牧强度对植被指数$NDVI$的数学模型，进而得到对植被生物量的数学模型。根据相关数据进行线性拟合得到载畜率 V 与植被指数的关系：

$$NDVI = 0.4485\text{V}-0.02457 \tag{4}$$

同时实际的载畜率与放牧强度密切正相关，这里简化模型设其相同[5]。可以得到放牧强度 F 与植被指数 NDVI 的关系：

$$V = F \tag{5}$$

$$NDVI = 0.4485\text{F}-0.02457 \tag{6}$$

再考虑到放牧方式影响系数 B 最终获得放牧策略对植被生物量影响的数学模型:

$$w = BE(270.6504F^3 + 1444.1134F^2 - 79.1022F - 1691.9766) \tag{7}$$

5.3 放牧策略对土壤湿度的影响

设土壤湿度为Q，根据土壤含水量-降水量-地表蒸发模型和土壤-植被-大气系统的水平衡基本方程可知，土壤湿度变化，主要受降水量、实际蒸发量和植被截流量影响,因此设初始土壤湿度为Q_1,增加量为ΔQ，简化模型将渗透和径流量设为常数 CL，得到土壤湿度的方程式:

$$Q = Q_1 + \Delta Q = Q_1 + P\text{L} + \text{CL} - (E + I\text{C}) \tag{8}$$

其中受降水量 PL、实际蒸发量 E 和植被截流量 IC，CL 为常数。

为简化模型，植被截流量 IC 仅考虑叶面积指数 L 和植被覆盖度 c_p 因素，$k \cdot R_{cum}$ 设为常数 CR。其关系表达式为

$$IC = c_p \cdot I_{\max} \cdot \left[1 - \exp\left(-CR / I_{\max}\right)\right] \tag{9}$$

$$I_{\max} = 0.935 + 0.498 \cdot LAI - 0.00575 \cdot LAI^2 \tag{10}$$

同时查阅相关文献叶面积指数 LAI 和植被覆盖度 C_p 可由植被指数 $NDVI$ 计算得到[6][7]：

$$c_p = \frac{NDVI - NDVIsoil}{NDVIveg - NDVIsoil} \tag{11}$$

$$LAI = 7.67NDVI - 4.01 \tag{12}$$

其中 $NDVIsoil$ 为 $NDVI$ 最小值，$NDVIveg$ 是 $NDVI$ 的最大值

带入放牧策略可以影响植被指数公式（6）可得模型：

$$IC = B\left(\frac{NDVI - NDVIsoil}{NDVIveg - NDVIsoil} \cdot (-0.0689F^2 + 1.8756F - 1.251) \cdot \left(1 - \exp\left(\frac{-CR}{-0.0689F^2 + 1.8756F - 1.251}\right)\right)\right) \tag{13}$$

$$QL = Q_1 + CL \tag{14}$$

最后得到最终的公式

$$Q = QL + PL - (E + IC) \tag{15}$$

其中未知变量为 QL 和 CR，使用 2019 年 6 月和 12 月的牧畜量除以草原面积作为放牧强度数据，对应的是 2020 年 6 月和 12 月的土壤湿度数据进行求解。根据深度不同土壤湿度可以得到对应的 4 个公式。

六、问题二模型的建立与求解

按年月份整合附件 3，4, 8 为一张表，以月份为组，组内按年份升序排列。经查阅资料选定平均气温(附件 8)、平均降雨量（附件 8）、蒸散量(附件 4)，平均风速(附件 8)和平均站点气压(附件 8)为不同深度土壤湿度的气候因素影响量。这些量与年份序列相关。针对不同的月份组，可以利用此特性预测出该月所需年份(1-3 月份需要 2023 年，4-12 月份需要 2022 和 2023 年)的气候因素影响量。进一步，利用已经预测得到的气候因素影响量。继续预测出不同深度的土壤湿度。

为了得到更优的湿度预测数据，本题在实现的过程中使用随机森林和提升决策树同时进行预测，最终得到的两份结果。同时本题给定了一个基于欧式距离定义的评判指标a因子及标准，经比较得知随机森林预测效果更好，并将预测结果填入表 6.5。

6.1 模型输入的选取

土壤水分状况是由气候因素、地形因素、土壤因素等共同决定的，其中地形因素和土壤因素基本是稳定的，只有气候因素是动态变化的，故而气候因素对土壤水分具有重要的影响。除了平均气温(附件 8)、平均降雨量（附件 8）、蒸散量(附件 4)外，考虑到草原生态系统主要分布在干旱和半干旱地区 ，而风蚀现象多发生在这些地区，风对土壤资源起着再分配和搬运的作用。并且的地面气温、湿度、气压和风速会构造区域高时空分辨率的大气强迫场继而影响不同深度的土壤湿度。因此。风速的变化(选取平均风速 附件 8)和气压变化(选取平均站点气压 附件 8)也对考虑作为模型的输入。

许多学者对不同地区土壤水分动态分布规律进行了深入研究和探讨，但基本都是通过短期观测数据来揭示土壤水分动态变化，而本题利用长期气象数据(2012-2022 年)和土壤湿度数据来探讨气候因子对土壤湿度的影响，训练的模型更为泛化，具有更高的可信性。

6.2 算法流程

本题选用分别选用随机森林和提升决策树模型。分三步完成，共得到 2 组结果：

表 6.1 算法步骤

过程：
1.构建不同月份的年份时间序列和湿度影响量(蒸发量 zf、气温 qw、降水量 js、气压 qy、风速 fs)之间的模型，共 5 个。对于 1-3 月份，将时间序列扩充 1 组后代入模型，预测得到 12 组湿度影响量数据(仅最后 1 组即为 2023 年的湿度影响量有效，其余相对于真实值有偏差)；对于 4-12 月份，将时间序列扩充 2 组后代入模型，预测得到 12 组湿度影响量数据。(仅最后 2 组即为 2022 2023 年的湿度影响量有效，其余相对于真实值有偏差)。 2.构建不同月份的湿度影响量(蒸发量、气温、降水量、气压 、风速)和不同深度土壤湿度之间的模型，共 4 个。对于 1-3 月份，将 12 组湿度影响量代入模型，取预测得到的该深度土壤湿度最后 1 组即为该深度下 2023 年的土壤湿度值；对于 4-12 月份，将 12 组湿度影响量代入模型，取预测得到的该深度土壤湿度最后 2 组即为该深度下 2022 2023 年的土壤湿度值。 3.对两种分类器得到的两份结果分别计算a因子，定义a因子最小的结果为最佳。

6.3 模型说明

6.3.1Bagging 与随机森林

Bagging 把个体预测器当做黑盒子处理，不进行进一步修改，所以，它的个体预测器可以是任何机器学习算法。它对训练数据集进行随机取样，并使用取样后的数据子集，训练每一个个体预测器。在预测时，每一个预测器都会做出预测，整体结果则是每个预测的平均。

为了增加个体预测器的多样性，随机森林在 bagging（决策树）的基础上更进一步，在每棵树的建造过程中增加了随机性。在每棵随机树的生长过程中，节点分裂时，所用的特征不再是所有特征中最好的，而是特征的一个子集中最好的。这种随机性降低了运算量，略微提升了 bias，极大提升了决策树的多样性，在取平均数之后可以大幅减小 variance。整体而言，随机森林是一个非常强大的算法。

表 6.2 Bagging 算法流程

过程：

输入为样本集 $D=\{(x,y_1),(x_2,y_2),\cdots(x_m,y_m)\}$，弱学习器算法，弱分类器迭代次数 T。

输出为最终的强分类器 f(x)

(1)对于 t=1,2…,T:

(a)对训练集进行第 t 次随机采样，共采集 m 次，得到包含 m 个样本的采样集 Dt

(b)用采样集 Dt 训练第 t 个弱学习器 Gt(x)

(2)如果是分类算法预测，则 T 个弱学习器投出最多票数的类别或者类别之一为最终类别。如果是回归算法，T 个弱学习器得到的回归结果进行算术平均得到的值为最终的模型输出。

随机森林既可以用于分类，也可以用于回归。一般适用于数据维度较低，同时对准确性要求较高的场景中。本题两步预测涉及输入数据的特征数都很少(1 维或 5 位)，比较适合用其进行回归。

表 6.3 随机森林算法流程

算法：

输入为样本集 $D=\{(x,y_1),(x_2,y_2),\cdots(x_m,y_m)\}$，弱学习器算法，弱分类器迭代次数 T。

输出为最终的强分类器 f(x)

(1)对于 t=1,2…,T:

(a)对训练集进行第 t 次随机采样，共采集 m 次，得到包含 m 个样本的采样集 Dt

(b)用采样集 Dt 训练第 t 个弱学习器 Gt(x)

(2)如果是分类算法预测，则 T 个弱学习器投出最多票数的类别或者类别之一为最终类别。如果是回归算法，T 个弱学习器得到的回归结果进行算术平均得到的值为最终的模型输出。

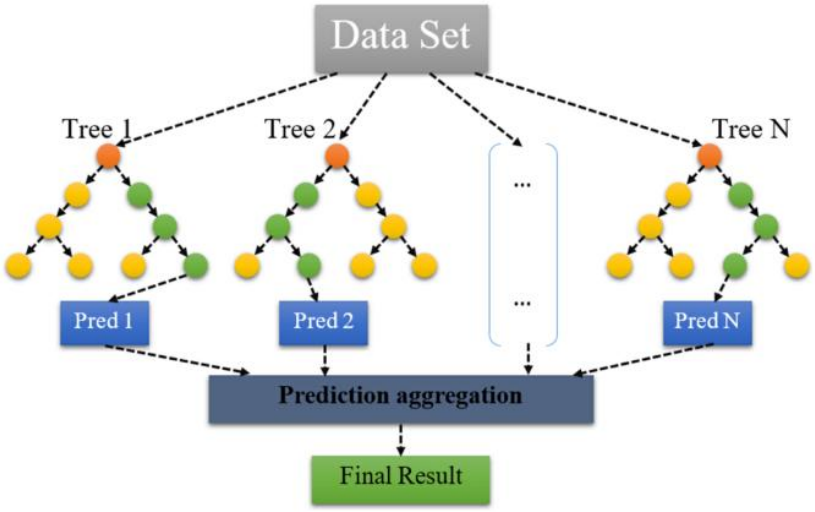


图 6.4 随机森林拓扑图

6.3.2 提升决策树

它能解决各种分类、回归和排序问题，能优秀地处理定性和定量特征，针对 outlier 的鲁棒性很强，数值不需要 normalize。在本质上，决策树类型的算法几乎不对数据的分布做任何统计学假设，这使它们能拟合复杂的非线性函数。

本题用到的接口是 GradientBoostingClassifier()，依赖于 GBDT(gradient boosting decision tree)算法。算法流程如表 6.5：

表 6.5 GBDT 算法流程

输入： 训练数据集 T={(x1,y1),(x2,y2),...,(xN,yN)},xi∈X⊆Rn，yi∈Y⊆R；损失函数 L(y,f(x))；

输出： 回归树 $\hat{f}(x)$

1.初始化

$$f_0(x) = \arg\min_c \sum_{i=1}^{N} L(y_i, c)$$

2.对 m=1,2，...，M

(a)对 i=1,2，...，N，计算

$$r_{mi} = -\left[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)}\right]_{f(x)=f_{m-1}(x)}$$

(b)对 r_{mi} 拟合一个回归树，得到第 m 棵树的叶结点区域 R_{mj}，j=1,2，...，J。

(c)对 j=1,2，...，J。计算

$$c_{mj} = \arg\min_{x_i \in R_{mj}} \sum L(y_i, f_{m-1}(x_i + c))$$

(d)更新 $f_{\mathrm{m}}(x)=f_{\mathrm{m-1}}(x)+\sum_{j=1}^{J}c_{mj}I(x\in R_{mj})$

(3)得到回归树

$$\hat{f}(x)=f_M(x)=\sum_{m=1}^{M}\sum_{j=1}^{J}c_{mj}I(x\in R_{mj})$$

6.4 算法实现

6.4.1 年份序列预处理

整合后的表中，年份序列是[2012 2013 ...2022]"的序列数据。使用机器学习模型应该尽量避免使用大值。所以将该序列编码为[0 1 2 ... 10]的序列数据。作为模型原始数据输入。

6.4.2 构建年份序列与不同深度土壤湿度模型并预测

因为 1-3 月份与 4-12 月份需要预测的值数量不同，所以将模型拟合与推理分两种情况进行讨论。

Case1：对于 1-3 月份每月

表 6.6

输入：年份序列编码 x=0,1,...,10。

2012-2022 蒸发量 zf、气温 qw、降水量 js、气压 qy 、风速 fs 以及不同深度的土壤湿度 sdi

注：蒸发量 zf,预测的蒸发量(zf_pred)
气温 qw,预测的气温(qw_pred)
降水量 js,预测的降水量(js_pred)
气压 qy,预测的气压(qy_pred)
风速 fs,预测的风速(fs_pred)
不同深度的土壤湿度 sdi,预测的不同深度的土壤湿度(sdi_pred)，i=10,40,100,200。

过程：

1.分别构建 x 与 2012-2022 年 zf、qw、js、qy、fs 的模型 a、b、c、d、e

2.分别构建 2012-2022 年 zf、qw、js、qy、fs 与 2012-2022 年 sdi 的模型 A、B、C、D

3.更新年份序列编码 x=0,1,...,11。

4.x 代入模型 a、b、c、d、e 输出 2012-2023 年的 zf_pred、qw_pred、js_pred、qy_pred、fs_pred。

5. zf_pred、qw_pred、js_pred、qy_pred、fs_pred代入ABCD得到2012-2023的sdi_pred，取sdi_pred最后1列即为本月份在2023年的预测值。

输出：本月份在2023年的sdi_pred(i=10,40,100,200)

Case2：对于4-12月份每月

表6.7

输入：年份序列编码x=0,1,...,10。

2012-2022蒸发量zf、气温qw、降水量js、气压qy 、风速fs以及不同深度的土壤湿度sdi

注：蒸发量zf,预测的蒸发量(zf_pred)

气温qw,预测的气温(qw_pred)

降水量js,预测的降水量(js_pred)

气压qy,预测的气压(qy_pred)

风速fs,预测的风速(fs_pred)

不同深度的土壤湿度sdi,预测的不同深度的土壤湿度(sdi_pred)，i=10,40,100,200。

过程：

1.分别构建x与2012-2021年zf、qw、js、qy、fs的模型a、b、c、d、e

2.分别构建2012-2021年zf、qw、js、qy、fs与2012-2022年sdi的模型A、B、C、D

3.更新年份序列编码x=0,1,...,11。

4.x代入模型a、b、c、d、e输出2012-2023年的zf_pred、qw_pred、js_pred、qy_pred、fs_pred。

5. zf_pred、qw_pred、js_pred、qy_pred、fs_pred代入ABCD得到2012-2023的sdi_pred，取sdi_pred最后2列即为本月份在2023年的预测值。

输出：本月份在2022 2023年的sdi_pred(i=10,40,100,200)(3)得到回归

6.4.3*a*因子计算准则及判别

对于1-3月份

$$\beta=[\sum_{i}^{4}(x_i-\bar{d_i})]^{1/2} \tag{16}$$

$x_1x_2x_3x_4$为预测得到的2023年该月份10，40,100,200cm土壤湿度

$d_1d_2d_3d_4$为原表中计算得到的该月份11年的10，40,100,200cm土壤湿度均值

对于 4-12 月份

$$y=[\sum_{i}^{4}[(x_i-\bar{d_i})+(x_{i+4}-\bar{d_i})]]^{1/2} \tag{17}$$

$x_1x_2x_3x_4$为预测得到的 2022 年该月份 10，40,100,200cm 土壤湿度

$x_5x_6x_7x_8$为预测得到的 2023 年该月份 10，40,100,200cm 土壤湿度

$d_1d_2d_3d_4$为表中计算得到的该月份 10 年的 10，40,100,200cm 土壤湿度均值

$\alpha=\beta+\gamma$

计算得到随机森林α=51.99147005，提升决策树α =553.1955855。所以选择随机森林模型得到的结果并填入 6.12。

6.5 结果分析

2012-2022 1-3 月份(以 3 月份为例),如图 6.8 所示。

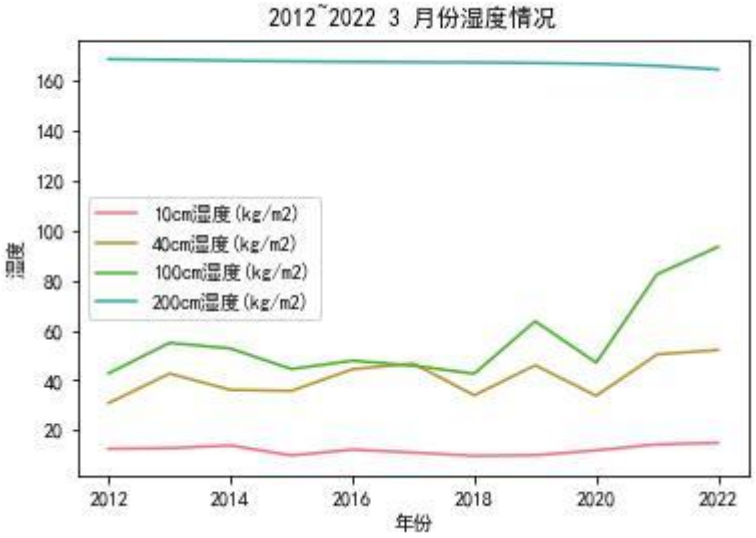


图 6.8 11 年不同深度土壤湿度的变化趋势

蓝色线代表 3 月份用年份序列输入随机森林模型得到的预测不同深度下的土壤湿度，红线代表 3 月份用年份序列输入标准时序分析 LSTM(见问题 6)模型得到的预测不同深度下的土壤湿度。可见土壤深度越深两条线越接近，200cm 深度时最佳，此时随机森林模型与标准时序分析模型结果最接近，预测最准确可信，如图 6.9 所示。

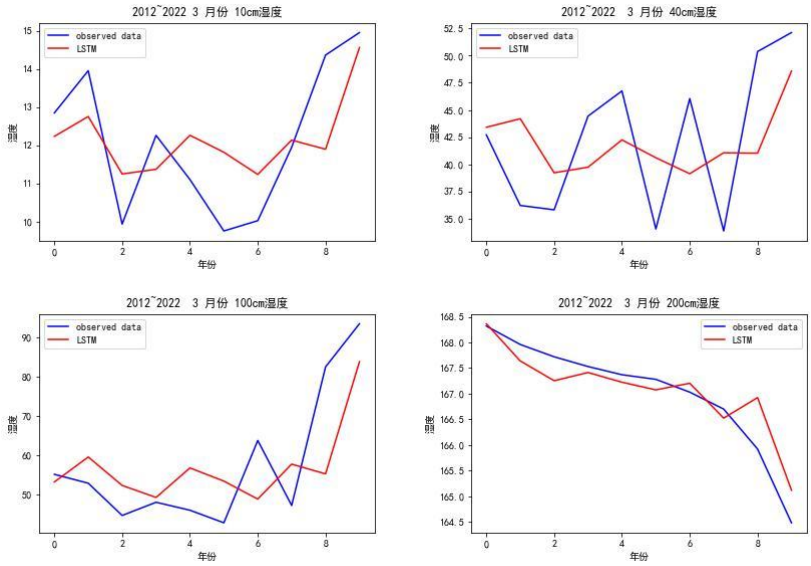


图 6.9 随机森林模型与标准时序分析模型结果对比

2012-2022 4-12 月份(以 4 月份为例)，如图 6.10 所示

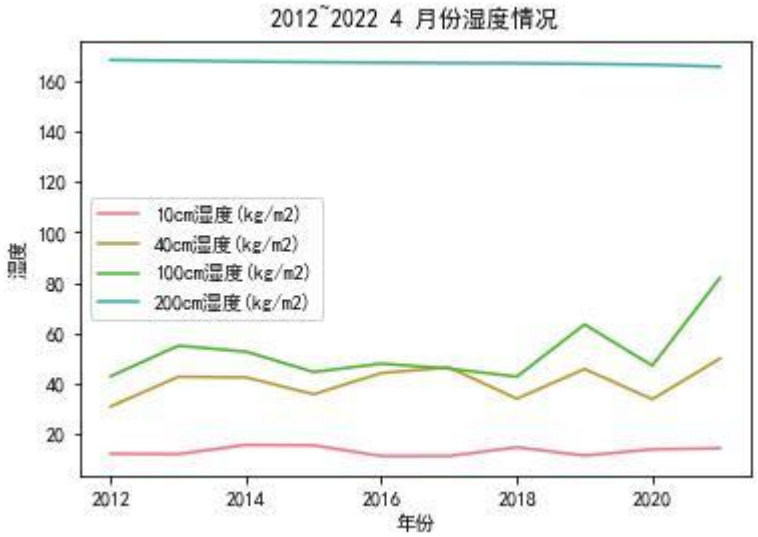


图 6.10 11 年不同深度土壤湿度的变化趋势

以 4 月份为例，蓝色线代表 4 月份用年份序列输入随机森林模型得到的预测不同深度下的土壤湿度，红线代表 3 月份用年份序列输入标准时序分析 LSTM(见问题 6)模型得到的预测不同深度下的土壤湿度。可见两条线一直比较接近，随机森林模型与标准时序分析模型结果比较接近，预测比较准确可信，如图 6.11 所示。

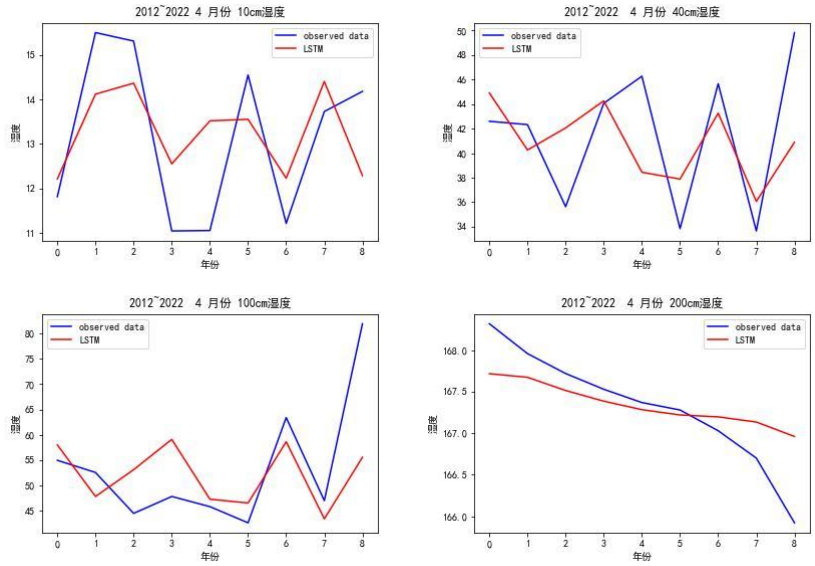


图 6.11 随机森林模型与标准时序分析模型结果对比

综上，随机森林模型在 1-3 月份对比较浅层的土壤(10cm 40cm)湿度预测不佳，其余情况皆在可信范围内。

由随机森林模型预测得到的问题 2 答案见下表 5.12 所示：

表 5.12 问题 2 结果

年份	月份	10cm 湿度(kg/m2)	40cm 湿度(kg/m2)	100cm 湿度(kg/m2)	200cm 湿度(kg/m2)
2022	04	14.48	41.29	69.39	166.54
	05	15.17	41.80	56.88	166.58
	06	17.26	45.13	63.53	166.43
	07	21.30	58.44	62.70	166.52
	08	18.26	47.48	69.78	166.32
	09	18.83	49.25	61.12	166.81
	10	15.39	45.51	73.49	165.91
	11	13.37	47.03	73.36	166.02
	12	13.49	48.37	64.74	166.62
2023	01	12.63	40.89	50.73	167.19
	02	11.75	45.88	76.49	165.88
	03	14.37	46.63	71.17	166.12
	04	14.48	41.29	64.74	166.62
	05	15.17	41.80	69.39	166.54
	06	17.26	45.13	56.88	166.58
	07	21.30	58.44	63.53	166.43
	08	18.26	47.48	62.70	166.52
	09	18.83	49.25	69.78	166.32
	10	15.39	45.51	61.12	166.81
	11	13.37	47.03	73.49	165.91

	12	13.49	48.37	73.36	166.02

七、问题三模型的建立与求解

本题预先做出假设如下：放牧策略的放牧方式固定为选择划区轮牧，仅用 12 个放牧小区号表征；放牧策略的放牧强度分为 4 类，即对照(NG)、轻度放牧(LGI)、中度放牧(MGI)以及重度放牧(HGI)。选择附件 14 15，按共有的 16 18 20 年份及放牧小区号进行合并。进行数据清洗，保留放牧强度，放牧小区，化学性质（'SOC 土壤有机碳'，'SIC 土壤无机碳','STC 土壤全碳'，'全氮 N'，'土壤 C/N 比'）列。对离散变量放牧小区进行 one-hot 编码，放牧强度规定为固定数值(0 2 4 8)，构建放牧小区和放牧强度对 5 个化学性质的 5 个回归决策树模型。并使用 K 折交叉验证及网格搜索算法进行模型优化超参。经对该模型 r2_score 进行分析认定满足题干所需。模型预测结果见表 7.3。

7.1 算法流程及实现

表 7.1

过程：

1. **数据合并：**

   分别选取 16、18、20 年的全部数据，按小区-年份为索引进行合并得到 T。

2. **数据清洗：**

   对 T 处理缺失值及无关项。

3. **数据剔除：**

   人为保留 T 的相关项,其中放牧小区（plot）和轮次表示放牧方式；放牧强度（intensity）表示放牧强度；SOC 土壤有机碳，SIC 土壤无机碳，STC 土壤全碳'，全氮 N'，土壤 C/N 比表示土壤的化学性质。

4. **数据量化：**

   放牧方式是离散值，其值不代表数的大小。为了应用机器学习模型，需要将小区号和轮次号都进行 one-hot 编码。放牧强度量化方式直接采取题干信息设定为常量。具体为：NG=0,LGI=2,MGI=4,HGI=8

5. **模型训练：**

   导入决策树模型，应用 k 折交叉验证和网格化搜索算法进行超参数搜索以获得最优回归决策树。进而使用该树进行推理，并将结果写在表 7.10 中。

5. zf_pred、qw_pred、js_pred、qy_pred、fs_pred 代入 A B C D 得到 2012-2023 的 sdi_pred，取 sdi_pred 最后 2 列即为本月份在 2023 年的预测值。

输出：本月份在 2022 2023 年的 sdi_pred(i=10,40,100,200)(3)得到回归

数量分布直方图如图 7.2、7.3、6.4 所示。

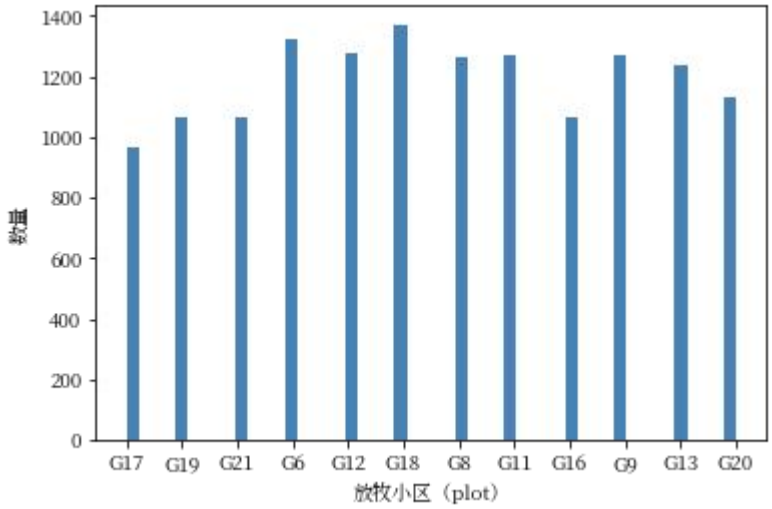


图 7.2 不同放牧小区数量分布

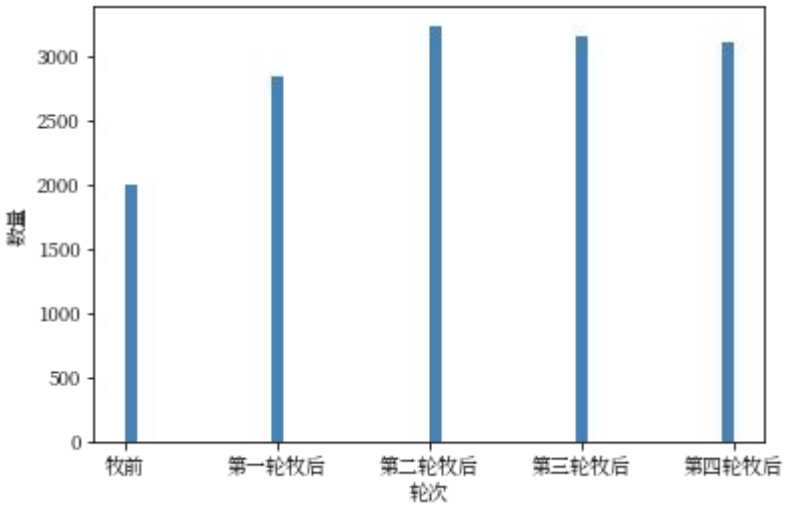


图 7.3 不同放牧轮次数量分布

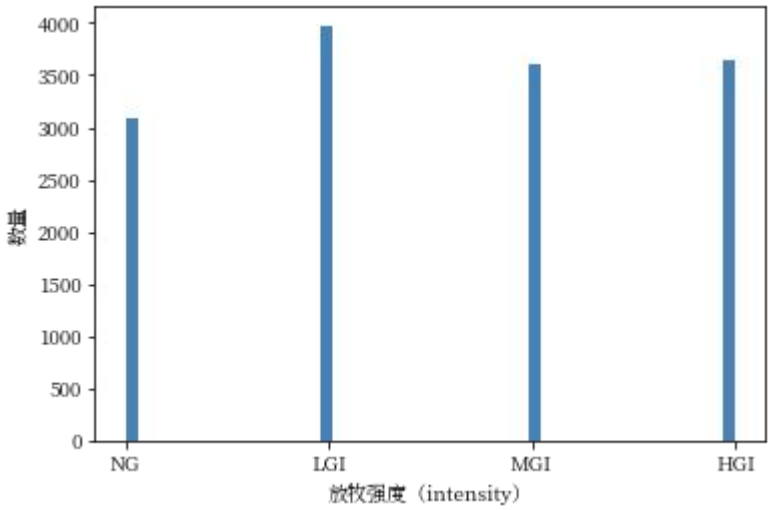


图 7.4 不同放牧强度数量分布

7.2 模型说明

7.2.1 回归决策树

决策树是表示基于特征对实例进行分类的树形结构。从给定的训练数据集中，依据特征选择的准则，递归的选择最优划分特征，并根据此特征将训练数据进行分割，使得各子数据集有一个最好的分类的过程。

本题用到的接口为 DecisionTreeRegressor。以基尼指数为分类依据，CART 算法构建决策树，并对构建的决策树进行后剪枝处理。

基尼指数：基尼指数（基尼不纯度），表示在样本集合中一个随机选中的样本被分错的概率。

$$Gini(p)=\sum_{k=1}^{K}p_k(1-p_k)=1-\sum_{k=1}^{K}p_k^2 \tag{18}$$

1.p_k表示选中的样本属于 k 类别的概率，则这个样本被分错的概率是(1- p_k)

2.样本集合中有 K 个类别，一个随机选中的样本可以属于这 k 个类别中的任意一个，因而对类别就加和。

3. 当为二分类是，Gini(p)=2p(1-p)

CART 算法：

第一步：输入数据集 X=（x1，x2，.....，xn)，设变量为 V =（v1，v2，.....，vk），设定最小 Gini 值的阈值，在 V 中选出目标变量 v*；

第二步：处理变量，将变量 V 中每一 vj 都处理成二元属性变量的形式。其中，若 vj 已经是一元变量，则不用更变：如果 vj，不是元元属性变量，是多元分类 （属性）变量，或者是离散、连续数值型变量，则需要通过最小 Gini 法确定最优分割，即在某一个分割条件下,计算得到的 Gini 值最小，Gini 值的公式为：

$$Gini(v_j)=\frac{n_1}{n}(1-\sum_{i=1}^{2}p_i)+\frac{n_2}{n}(1-\sum_{i=1}^{2}q_i) \tag{19}$$

该公式表示目标变量 v*按变量 vj 划分后的 Gimi 值，其中 n1 和 n2 表示变量 vj,二元化后两个类别(设为 1 和 2）所包含的数据点个数，p1，p2 表示 vj 第 1（2）个类别中属于目标变量 v*的第 1(2)个类别的概率，同理 q1，q2。选择日标变量 v*按变量 vj 划分后的 Gini 值最小的变量 vj 作为最优划分变量;

第三步：对划分好的决策树重复进行第二步，直到无法继续划分，或满足收敛条件：

第四步：输出最终的 CART 回归二叉树，作为最终的决策树，用于分类预测。

后剪枝算法：

后剪枝则是先从训练集生成一颗完整的决策树，然后自底向上地对非叶结点进行考察，若将该结点对应的子树替换为叶结点能带来决策树泛化性能的提升，则将该子树替换为叶结点。后剪枝决策树通常比预剪枝决策树保留了更多的分支。一般情况下，后剪枝决策树生长更为充分，因此欠拟合风险很小，泛化性能也往往优于预剪枝决策树。当然，由于对树的生长过程不加限制，并且要自底向上地对所有叶结点进行一一考察，所以后剪枝决策树的

训练时间和复杂程度都要大于预剪枝决策树。

7.2.2 K 折交叉验证和网格化搜索

Kfold:将所有的样例划分为 k 个组，称为折叠 (fold) ，每组数据都具有相同的大小。每一次分割会将其中的 K-1 组作为训练数据，剩下的一组用作测试数据，一共会分割 K 次。

GridSearch 和 CV:即网格搜索和交叉验证网格搜索算法。是一种通过遍历给定的参数组合来优化模型表现的方法。超参数选择不恰当，就会出现欠拟合或者过拟合的问题

网格搜索指在指定的参数范围内，按步长依次调整参数，利用调整的参数训练学习器，从所有的参数中找到在验证集上精度最高的参数，这其实是一个训练和比较的过程。本题用其来搜索最佳的 Fold 和决策树深度(1-11)。

GridSearchCV 存在的意义就是自动调参,可以保证在指定的参数范围内找到精度最高的参数，但是这也是网格搜索的缺陷所在，他要求遍历所有可能参数的组合，在面对大数据集和多参数的情况下，很难得出结果

7.3.3 R2_score MAE MSE RMSE

决定系数：R2（R-Square）

$$R^2 = 1 - \frac{\sum_i (\hat{y}_i - y_i)^2}{\sum_i (\bar{y}_i - y_i)^2}$$

分子部分表示真实值与预测值的平方差之和，类似于均方差 MSE；分母部分表示真实值与均值的平方差之和，类似于方差 Var。

根据 R-Squared 的取值，来判断模型的好坏，其取值范围为[0,1]：

如果结果是 0，说明模型拟合效果很差；

如果结果是 1，说明模型无错误。

一般来说，R-Squared 越大，表示模型拟合效果越好。R-Squared 反映的是大概有多准，因为，随着样本数量的增加，R-Square 必然增加，无法真正定量说明准确程度，只能大概定量。

均方误差：MSE（Mean Squared Error）

$$MSE=\frac{1}{\mathrm{m}}\sum_{\mathrm{i}=1}^{\mathrm{m}}(y_{\mathrm{i}}-\hat{y}_i)^2$$ 其中，$y_i-\hat{y}_i$为测试集上真实值-预测值。

均方根误差：RMSE（Root Mean Squard Error）(RMSE=sqrt（MSE）)

$$RMSE=\sqrt{\frac{1}{\mathrm{m}}\sum_{\mathrm{i}=1}^{\mathrm{m}}(y_{\mathrm{i}}-\hat{y}_i)^2}$$

平均绝对误差：MAE（Mean Absolute Error）

$$MAE=\frac{1}{m}\sum_{i=1}^{m}\left|(y_i-\hat{y}_i)\right|$$

7.3 结果分析

由 13 个特征预测（12 个特征为放牧小区的 one-hot 编码，1 个特征为放牧强度）出 5 个土壤化学性质的回归决策树模型分别如下：

表 7.5　预测 5 个土壤化学性质的回归决策树模型评价指标

	max_depth	r2_score
SOC	9	0.19
C/N	7	0.08
SIC	8	0.14
STC	6	0.45
全 N	8	0.06

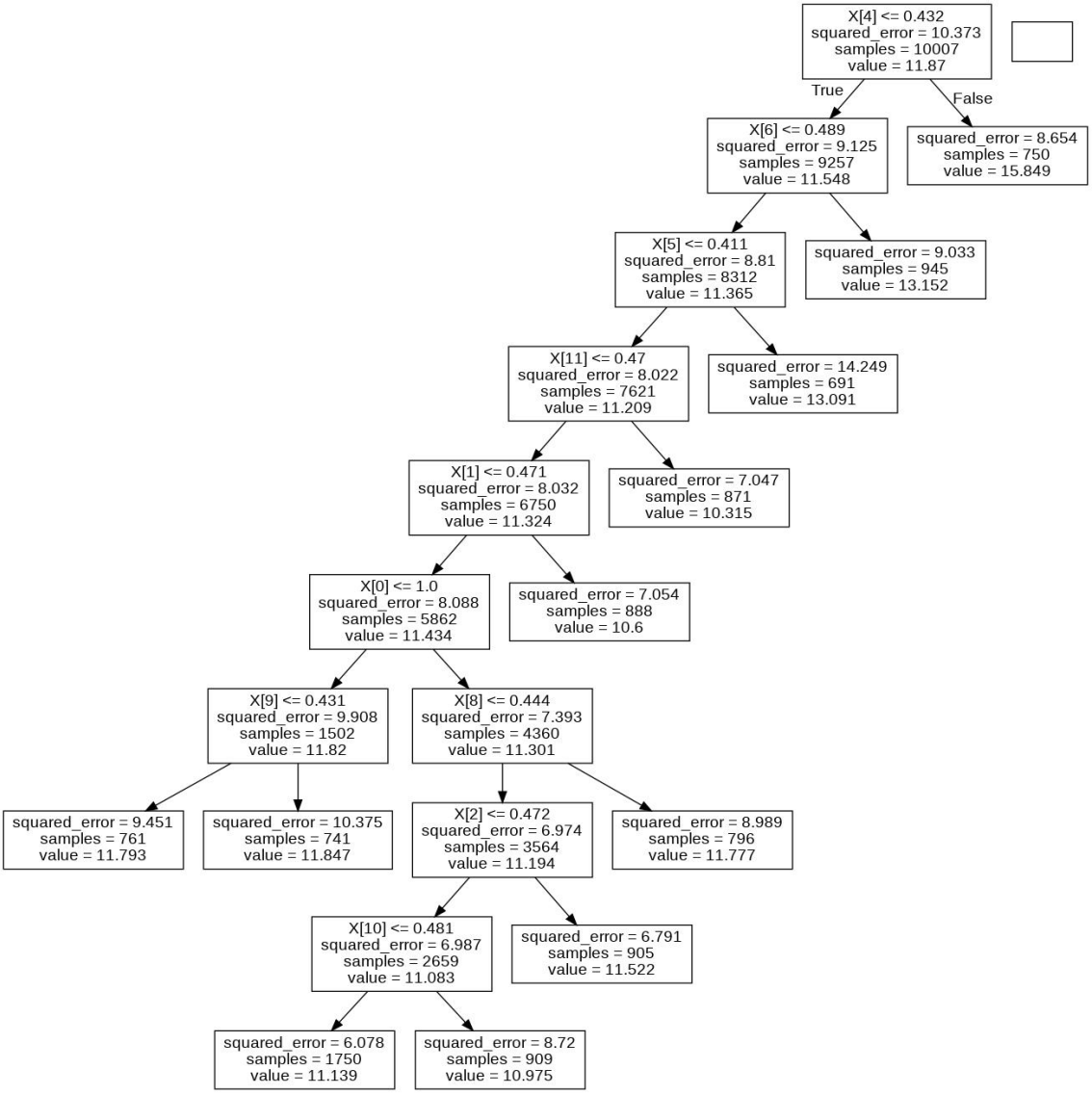


图 7.6 预测土壤 SOC 比的回归树模型

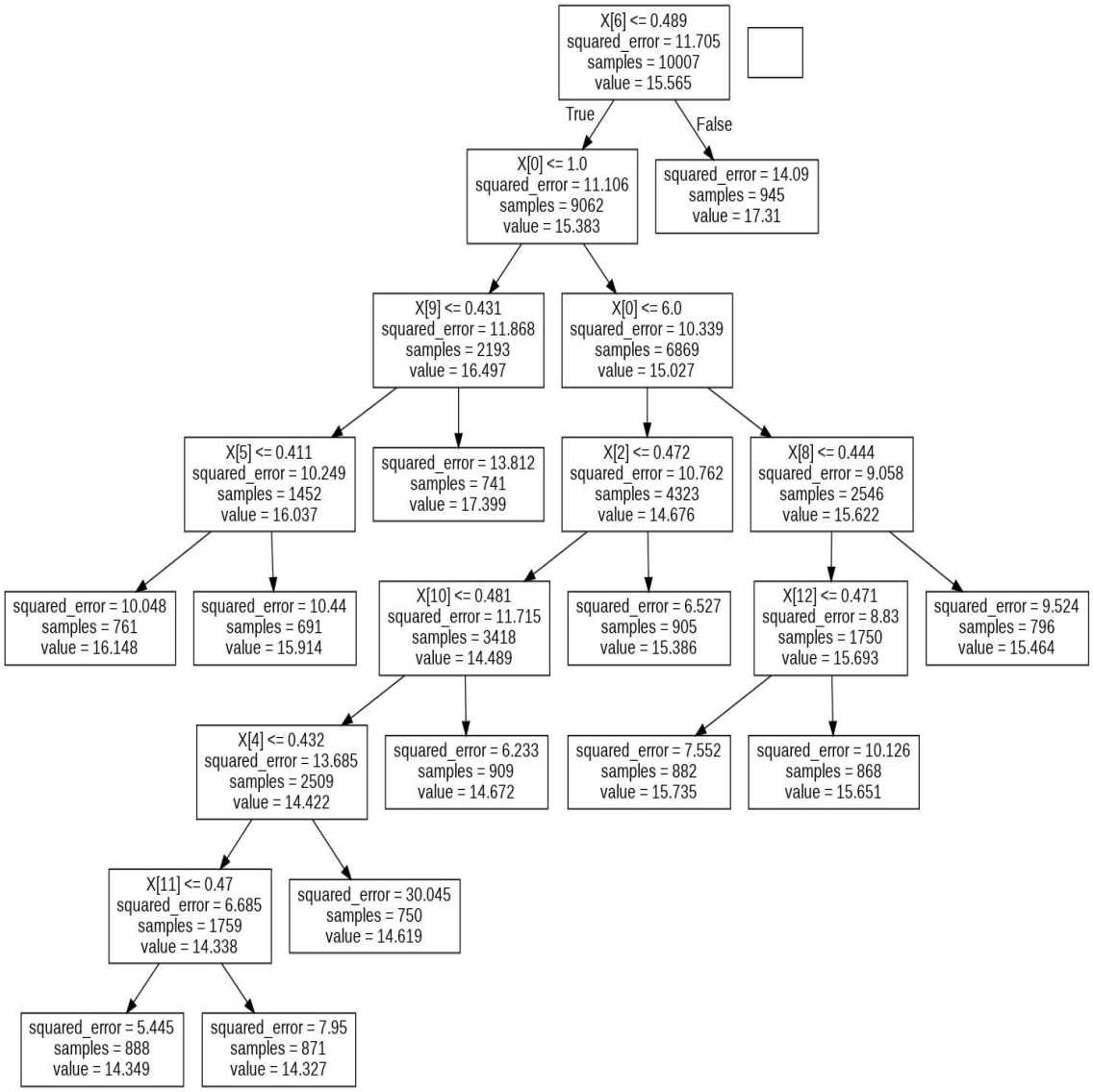


图 7.7 预测 C/N 土壤有机碳的回归树模型

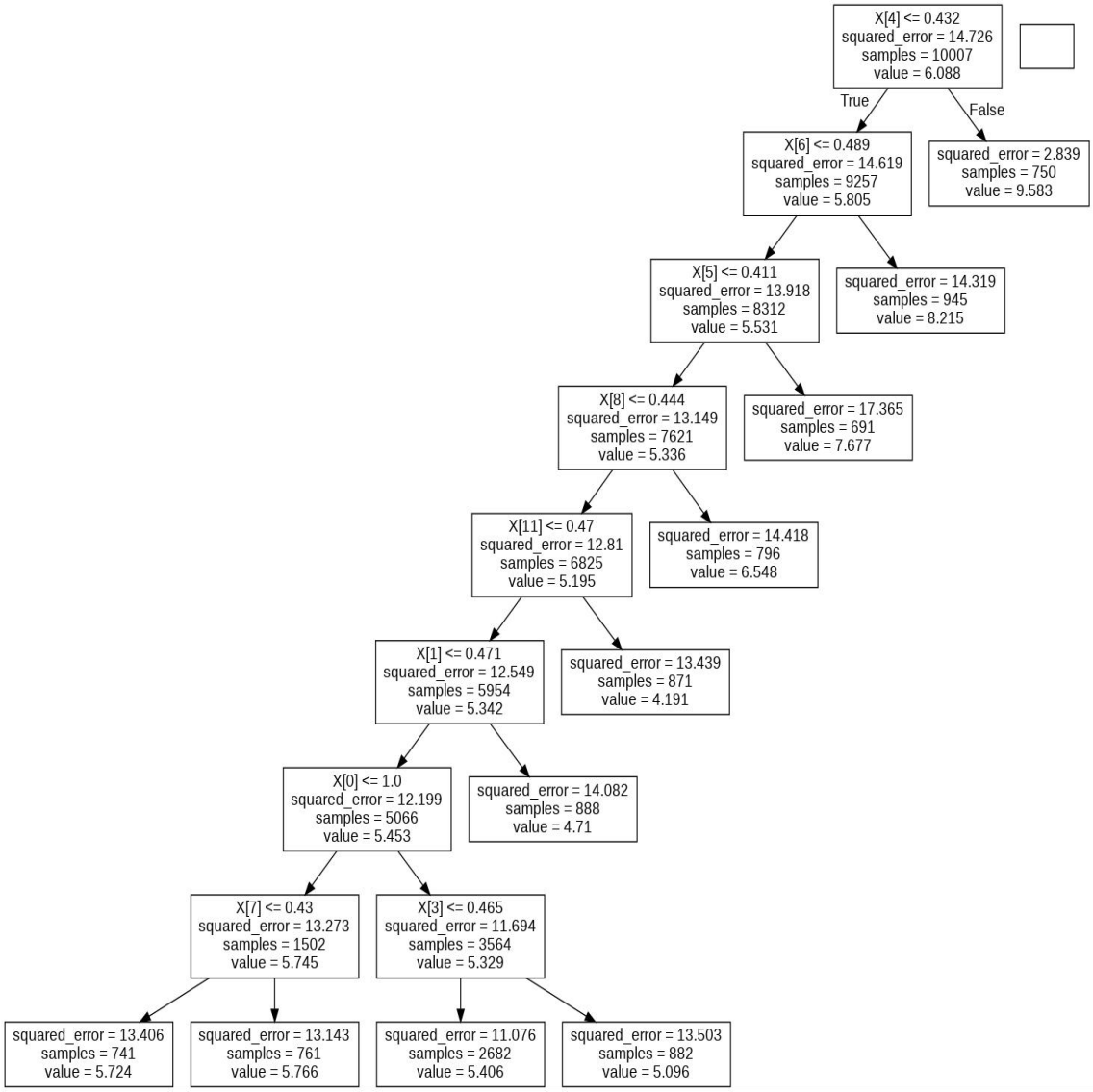


图 7.8 预测 SIC 土壤无机碳的回归树模型

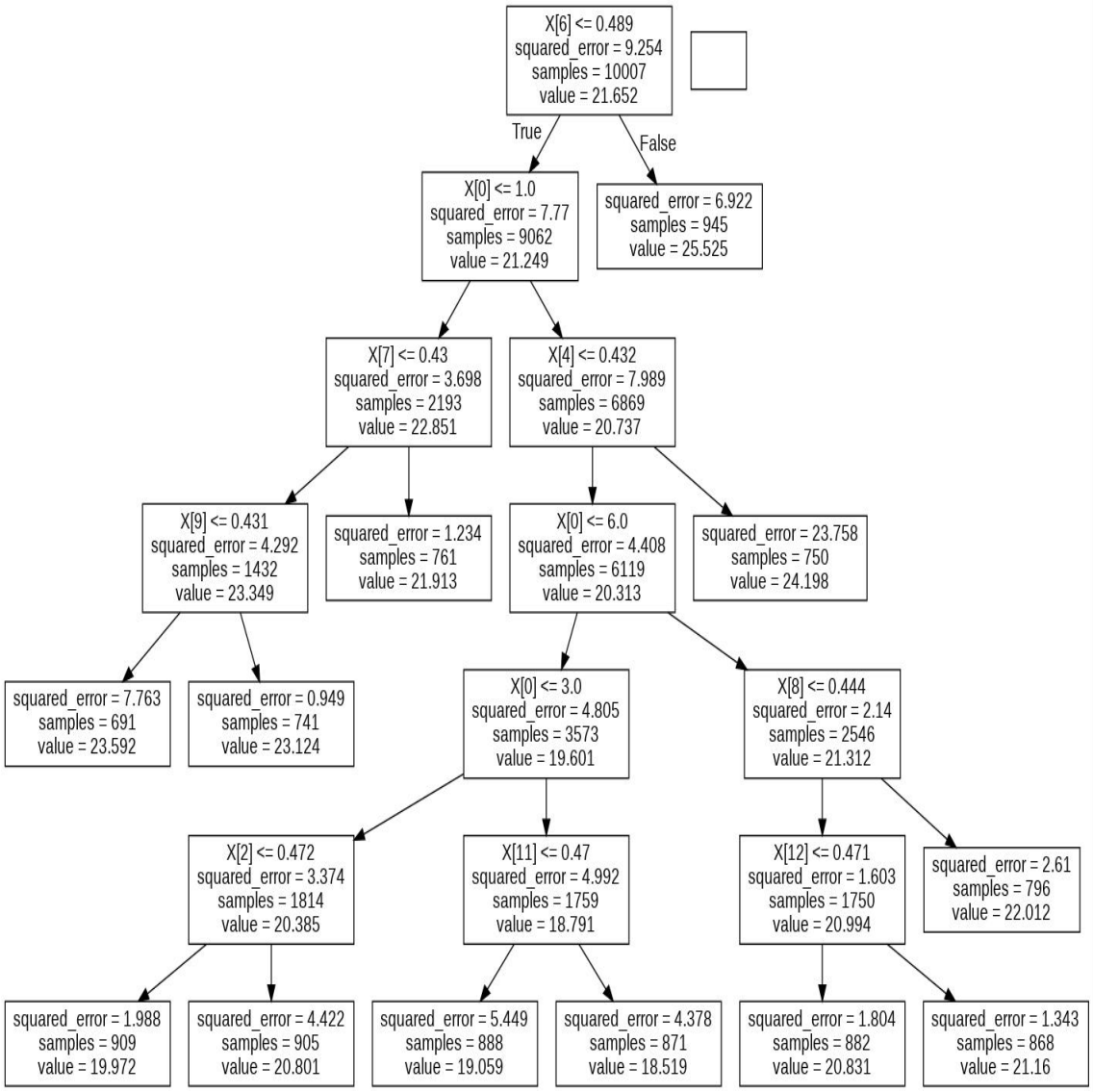


图 7.9 预测 STC 土壤全碳的回归树模型

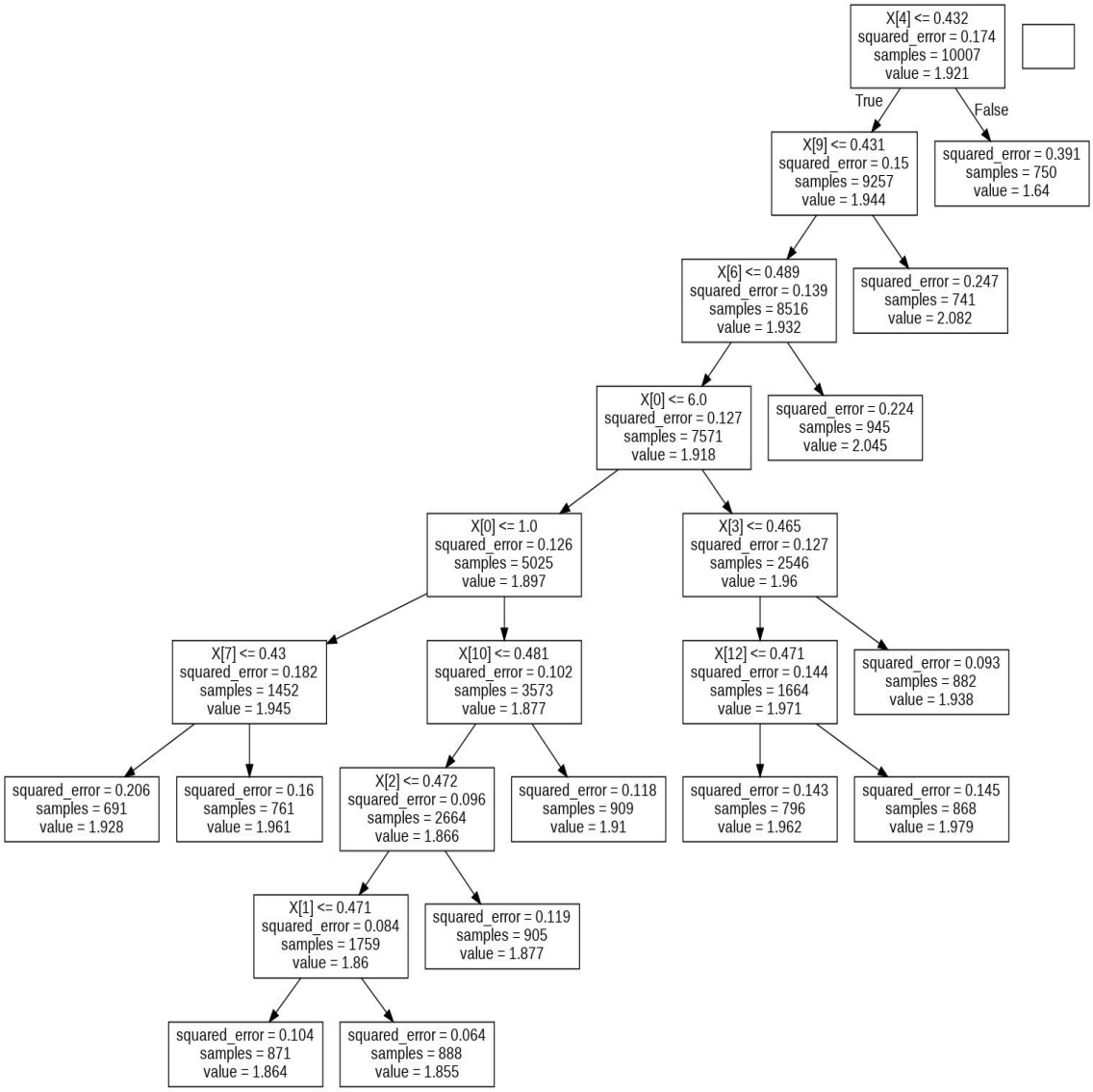


图 7.10 预测全氮 N 的回归树模型

可见即使加上超参数优化，也没有对回归决策树的泛化能力得到显著的增强。以上模型大部分 r2_score 接近 0，说明模型拟合能力不是特别好。但是已满足题干所需。

回归决策树预测题目结果如图 7. 10 所示：

表 7.11 问题 3 结果

放牧强度	Plot 放牧小区	SOC 土壤有机碳	SIC 土壤无机碳	STC 土壤全碳	全 N	土壤 C/N 比
NG	G17	15.91	7.68	23.59	1.93	13.09
	G19	16.15	5.77	21.91	1.96	11.79
	G21	17.40	5.72	23.12	2.08	11.85
LGI	G6	14.67	5.41	19.97	1.91	10.97
	G12	15.39	5.41	20.80	1.88	11.52
	G18	17.31	8.22	25.53	2.05	13.15

MGI	G8	14.33	4.19	18.52	1.86	10.32
	G11	14.35	4.71	19.06	1.86	10.60
	G16	14.62	9.58	24.20	1.64	15.85
HGI	G9	15.65	5.41	21.16	1.98	11.14
	G13	15.74	5.10	20.83	1.94	11.14
	G20	15.46	6.55	22.01	0.96	11.78

八、问题四模型的建立与求解

8.1 问题思路与分析

根据题目要求根据沙漠化程度指数预测模型表达式和板结化指数模型，以及沙漠化指数的影响因素（包括风速、降水、气温（三个气象因素）；植被盖度、地表水资源、地下水位（三个地表因素）；人口数量、牲畜数量、社会经济水平（三个人文因素））和板结化指数的影响因素（土壤湿度w越少，容重c越大，有机物含量o）进行分析。建立沙漠化程度指数预测模型和板结化预测模型，并对监测点不同放牧强度下的沙漠化程度指数，板结化指数进行预测。为了了解沙漠化和板结化相关知识，本文通过查阅相关文献了解沙漠化的机制，板结化的形成并调查了预测模型用于生成，测试的可行方式，结合文献中一些建议，用于预测实际过程中的指数。本题的解决思路是基于附件中沙漠化指数和板结化指数的影响因素数据，分析其特征和变化规律，收集并整理影响因素的数据，进行数据预处理，分析这些因素的影响程度。然后建立预测模型，设定好参数将模型进行实践，检验模型额正确性。最后通过预测不同放牧强度下的沙漠化指数和板结化指数，选取两者之和最小的放牧强度作为结果。

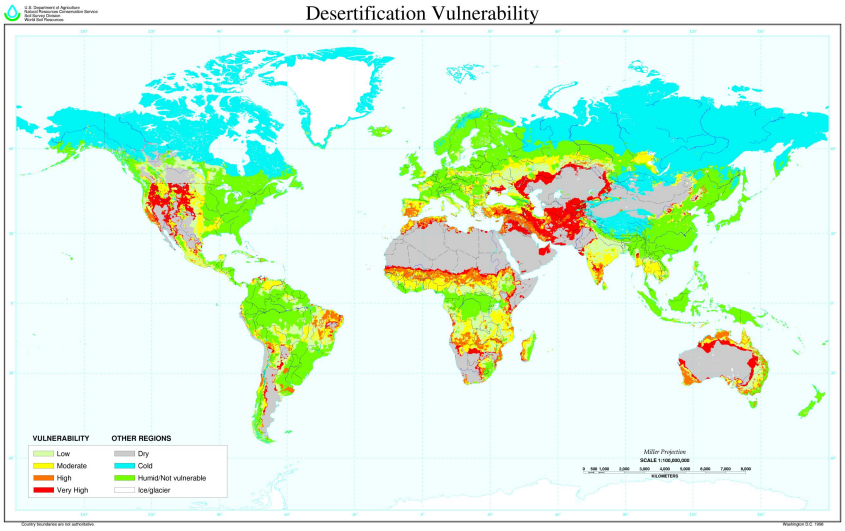


图 8.1 全球沙漠化威胁地图

8.2 沙漠化指数和板结化预测模型的构建

8.2.1 模型的构建思路

根据上述分析和相关研究，多重因素影响着沙漠化指数和板结化指数。为了能够等量的分析各种因素对于沙漠化指数和板结化指数的影响程度，并能够根据这些因素的情况对宏观上的草原沙漠化，草原土地板结化程度进行预测，本文尝试构建沙漠化程度指数预测模型，和板结化指数模型，从而可以寻找和求证影响土地沙漠化和板结化的因素。

为了能够对不同放牧强度下的沙漠化指数和板结化指数进行预测，将沙漠化指数和板结化指数的各影响因素间进行主成分分析，获取各因素的权值比重，构建不同强度下的沙漠化指数和板结化指数的预测。

8.2.2 数据集的手机及变量含义

为了分析各因素与全球气候变化的影响关系，需收集相关数据集。下表 8.2 给出本章所采用的数据集的变量的含义。

表 8.2 不同变量的含义

含义	符号
植被指数	NDVI
分别为入和出径流量	R
平均气温(℃)	T
降水量(mm)	PL
平均风速(knots)	KN
畜牧量	V
人口	RW
经济收入	G
10cm 湿度(kg/m2)	Q1
40cm 湿度(kg/m2)	Q2
100cm 湿度(kg/m2)	Q3
200cm 湿度(kg/m2)	Q4
土壤蒸发量(mm)	E
叶面积指数	LAI
径流量	G
有机物含量	O

8.2.3 数据缺失值处理及其标准化

（1）数据缺失值处理

由于数据记录部门及周期不同，不同数据集的数据量不尽相同，为了使各变量的数据集具有可用性，对数据集中的缺失值做相应处理。本文对这些数据集中的缺失值处理方法主要包括以下两种:

a）均值插补:采用序列的均值来替换缺失值。

b)基于序列趋势的插值:对于序列中间的缺失值，如一段历史数据中某几个年份缺乏记录，首先分析该序列的趋势，用最佳的回归模型进行拟合或者采用时间序列模型进行拟合，从而计算出相应的估计值，替换缺失值。对于序列末尾的缺失值，可以通过对序列趋势及特性的分析，采用回归模型或时间序列模型来拟合预测，得到对应的预测值以替换缺失值。对于序列首段的缺失值，可将序列先倒序后再采用前述进行逆时间预测，得到对应的预测值从而替换缺失值。

（2）各变量的数据标准化

由于各变量的单位和数量级并不相同，为了便于构建多变量回归模型，可将各变量的数据进行标准化。本文采用 z-score 标准化对数据进行处理。标准分数(standard score)也叫 z 分数(z-score），是将变量值与平均数的差再除以标准差的过程。可采用下列公式进行表示:

$$z=\frac{(x-\mu)}{\sigma} \tag{20}$$

其中 x 为变量的某一具体指，μ为数据序列的平均数，σ 为数据序列的标准差。

标准化得到的 Z 值的量代表着变量值和数据序列的平均值之间的距离，是以标准差为单位计算。在变量的原始值低于平均值时 Z 值为负数，反之则为正数。

8.2.4 因子的关联性分析

为分析各变量间的关联性，可用 spss 和 matlab 软件对各个因子进行关联性分析，分析后可以的到显影结果，沙漠化指数影响因素如表 8.3 所示。板结化指数影响因素关联性如表 8.4 所示

表 8.3 沙漠化指数影响因素相关性矩阵

相关性矩阵

		Zscore(植被指数)	Zscore(径流量)	Zscore(平均气温)	Zscore(降水量)	Zscore(平均风速)	Zscore(畜牧量)	Zscore(人口)	Zscore(经济收入)
相关性	Zscore(植被指数)	1.000	0.780	0.708	-0.168	0.686	0.663	-0.170	-0.171
	Zscore(径流量)	0.780	1.000	0.764	-0.029	0.729	0.702	-0.206	-0.212
	Zscore(平均气温)	0.708	0.764	1.000	-0.197	0.820	0.950	0.160	0.147
	Zscore(降水量)	-0.168	-0.029	-0.197	1.000	-0.359	-0.166	-0.063	-0.041
	Zscore(平均风速)	0.686	0.729	0.820	-0.359	1.000	0.683	0.044	0.003
	Zscore(畜牧量)	0.663	0.702	0.950	-0.166	0.683	1.000	0.100	0.102
	Zscore(人口)	-0.170	-0.206	0.160	-0.063	0.044	0.100	1.000	0.994
	Zscore(经济收入)	-0.171	-0.212	0.147	-0.041	0.003	0.102	0.994	1.000

表 8.4 板结化指数影响因素关联性

相关性矩阵			
		Zscore(土壤湿度)	Zscore(有机物含量)
相关性	Zscore(土壤湿度)	1.000	0.437
	Zscore(有机物含量)	0.437	1.000

8.2.5 模型的构建与求解

（1）沙漠化指数预测模型的构建

运用 spss 软件构建得到主成分分析结果，通过计算获得各个因素的权重系数。各因素的分数如表 8.5 所示

表 8.5 主成分分析成分矩阵

成分矩阵 a			
	成分		
	1	2	3
Zscore(平均气温)	0.948	0.196	0.089
Zscore(畜牧量)	0.893	0.151	0.120
Zscore(平均风速)	0.887	0.061	-0.171
Zscore(径流量)	0.878	-0.225	0.207
Zscore(植被指数)	0.854	-0.185	0.044
Zscore(人口)	-0.021	0.991	0.066
Zscore(经济收入)	-0.035	0.988	0.092
Zscore(降水量)	-0.268	-0.134	0.945
提取方法：主成分分析法。			
a. 提取了 3 个成分。			

表 8.6 方差解释

总方差解释

成分	初始特征值			提取载荷平方和			旋转载荷平方和		
	总计	方差百分比	累积 %	总计	方差百分比	累积 %	总计	方差百分比	累积 %
1	4.055	50.687	50.687	4.055	50.687	50.687	3.977	49.715	49.715
2	2.128	26.597	77.284	2.128	26.597	77.284	2.117	26.457	76.172
3	1.003	12.538	89.821	1.003	12.538	89.821	1.092	13.649	89.821
4	0.351	4.388	94.209						
5	0.277	3.460	97.669						
6	0.162	2.022	99.692						
7	0.021	0.261	99.953						
8	0.004	0.047	100.000						

提取方法：主成分分析法。

通过 spss 的主成分分析提取出来三个主成分，累计贡献达到 89.821%符合巨大部分变量的信息了。

主成分分析中提取主成分表和方差表中的信息，进行权重计算：

1）计算线性组合中的系数，标准化数/对应主成分特征根的平方根；

2）计算综合得分模型中的系数，公式为：（第一主成分方差 x100xC8+第二个主成分方差 x100xD8）/(第一主成分方差+第二主成分方差)；

3）最后一步，计算权重，也即标准化，将所有指标进行归一化处理，使其权重综合为 1。公式为：指标权重=标综合得分模型系数/指标综合得分模型之和。

将得到的各种影响因素的权重带入到沙漠化程度指数预测模型表达式中求得：

$$SM = 0.184030762\,\mathrm{T} + 0.172319967\,V + 0.137286029\,KN + 0.132717344\,G + 0.120370564 + 0.118386597\,NDVI + 0.117860661\,RW + 0.017028075\,PL \quad (21)$$

选用最新 1 年（2020 年）的数据，应用构建的沙漠化指数预测模型对不同放牧强度进

行预测，如表 8.7 所示。

表 8.7 放牧强度与沙漠化指数对应表

（2）板结化指数模型构建与求解

放牧强度	符号	沙漠化指数
对照	NG	0
轻度放牧	LGI	0.25
中度放牧强度	MGI	0.5
重度放牧强度	HGI	1

有板结化指数模型可知，相关因素为土壤湿度，容重，有机物含量。其中容重因为数据缺失，设为常数对模型没有影响。有机物含量可以通过主成分分析计算权重得到，并且与放牧强度有着对应关系。土壤湿度，可以用运用 spss 软件构建得到降水量，植被指数，放牧强度（牧畜量除以草原面积近似得到），蒸发量对于 10cm 土壤湿度的多元线性回归模型求借得到。最后在根据土壤湿度和有机含量间进行主成分分析计算权重，最终获得板结化指数。

首先对土壤湿度使用进行多元回归模型的构建模型，其中模型可信度如表 8.8 所示。

表 8.8 模型可信度

模型摘要 [b]

模型	R	R 方	调整后 R 方	标准估算的错误
1	.955[a]	0.913	0.884	0.97369

a. 预测变量：(常量)，土壤蒸发，降水，畜牧，植被指数

b. 因变量：湿度 10cm

其中 R 平方为 0.913，说明模型效果很好。

进一步得到多元回归的系数和常数。如表 8.9 所示

表 8.9 多元回归系数

系数 [a]

模型		未标准化系数		标准化系数			共线性统计	
		B	标准错误	Beta	t	显著性	容差	VIF
1	(常量)	13.487	1.146		11.774	0.000		
	植被指数	-7.857	3.611	-0.348	-2.176	0.050	0.283	3.528
	降水	0.003	0.002	0.118	1.362	0.198	0.966	1.035

畜牧	-0.001	0.001	-0.084	-0.571	0.578	0.339	2.947
土壤蒸发	0.276	0.043	1.309	6.403	0.000	0.173	5.766

a. 因变量：湿度 10cm

表 8.10 个因素间的共线性关系

共线性诊断 [a]

模型	特征值	条件指标	方差比例				
			(常量)	植被指数	降水	畜牧	土壤蒸发
1	3.957	1.000	0.00	0.00	0.02	0.00	0.00
2	0.740	2.313	0.00	0.00	0.62	0.00	0.02
3	0.233	4.119	0.07	0.00	0.35	0.01	0.15
4	0.053	8.624	0.00	0.80	0.00	0.14	0.17
5	0.017	15.285	0.92	0.20	0.02	0.85	0.66

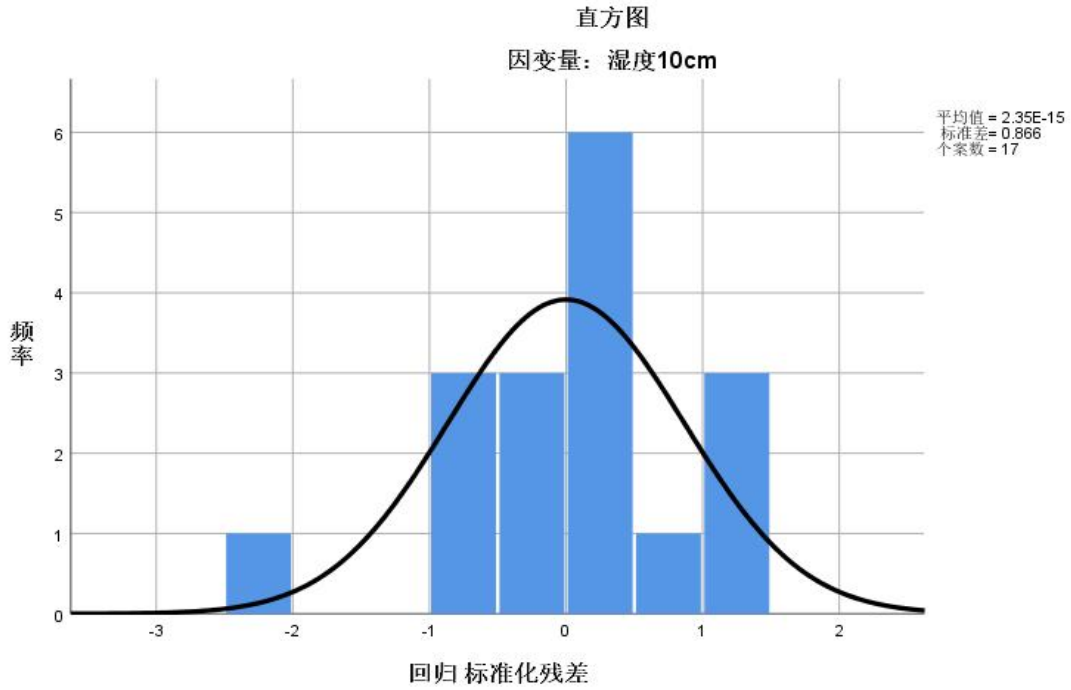


图 8.11 分布符合高斯分布

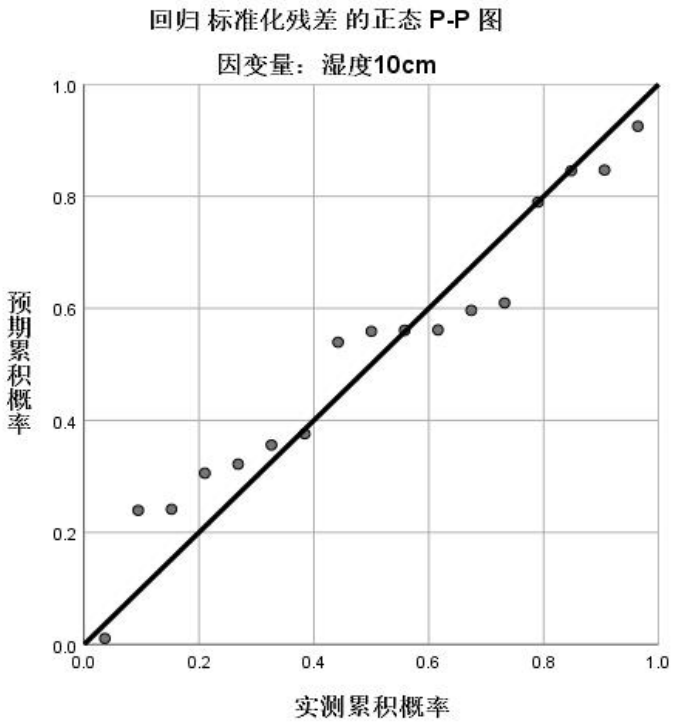


图 8.12　P-P 图

根据系数表格，提取相关的信息。用以构建土壤湿度的多元回归方程：

$$B = 0.276E - 0.001V + 0.003PL + -7.857 * DNVI + 13.487 \quad (22)$$

选用最新一年（2020）的数据代入，同时使用牧畜量除以草原面积近似放牧强度作为自变量。可以预测数不同放牧强度下土地湿度。如表 8-13 所示

表 8-13 不同放牧强度下土地湿度

放牧强度	符号	土地湿度
对照	NG	18.1617
轻度放牧	LGI	17.7777
中度放牧强度	MGI	17.3937
重度放牧强度	HGI	16.6257

构建板结化指数预测模型，还需要确定有机物和容重。容重因为数据短缺。设置为常数忽略不计，不影响总模型。有机物容量，应用 2020 年的各种因素的数据，对有机物进行主成分分析计算得到相应的权重。通过权重我们可以获得不同放牧强度下有机物量。如表 8.14 所示。

表 8.14 不同放牧强度下有机物含量

放牧强度	符号	有机物含量
对照	NG	18.09676667
轻度放牧	LGI	16.760483333
中度放牧强度	MGI	14.65
重度放牧强度	HGI	16.6205

有机物含量和土壤湿度进行主成分分析，计算权重并代入板结化指数模型：

$$B = 0.5*W + 0.5*O + 1.39 \tag{23}$$

根据板结化指数模型，代入不同放牧强度下的有机物含量和土壤湿度的值，可以求的不同放牧强度下的板结化指数，求出后进行归一化。 如表 8.15 所示。

表 8.15 不同放牧强度下板结化指数

放牧强度	符号	板结化指数
对照	NG	1
轻度放牧	LGI	0.5917
中度放牧强度	MGI	0
重度放牧强度	HGI	0.2853

沙漠化指数与板结化指数相加得到综合指数。如表 8-16 所示。

表 8-16 不同放牧强度下综合指数

放牧强度	符号	综合指数
对照	NG	1
轻度放牧	LGI	0.8417
中度放牧强度	MGI	0.5000
重度放牧强度	HGI	1.2853

由表 8-16 不同放牧强度下综合指数，可以知道在综合指数最小的放牧强度为中度放牧。

九、问题五模型的建立与求解

模型建立与求解

首先尝试最简单的枚举法进行求解，根据问题 4 所得到的公式：

$$Q = 0.276E - 0.001V + 0.003PL - 7.857 * DNVI + 13.487 \tag{24}$$

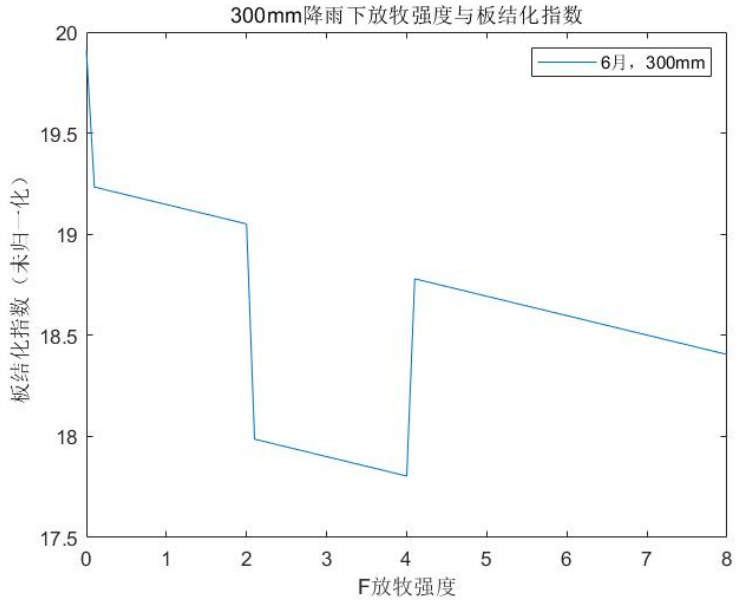
$$B = 0.5 * Q + 0.5 * O + 1.39$$

此时利用 2019 年的各项影响因素数值数据，其中降水量取值为[300mm 600mm 900mm 1200mm],放牧强度作为未知量 F。代入已知数据的板结化指数公式为：

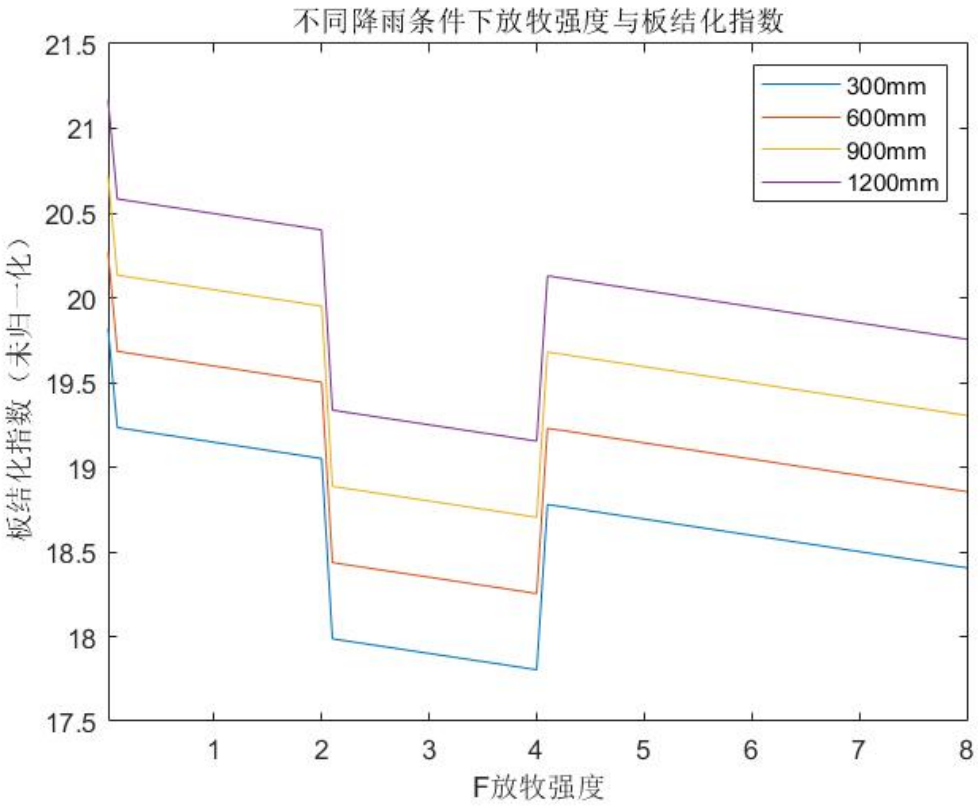
$$B = \begin{cases} 22.3698 - 0.0960\text{F} & \text{F=0} \\ 21.7014 - 0.0960\text{F} & \text{0<F<=2} \\ 20.6464 - 0.0960\text{F} & \text{2<F<=4} \\ 21.6317 - 0.0960\text{F} & \text{4<F} \end{cases}$$

放牧强度 F 的取值为[0,8],并且可以是小数，因此以 0.1 步进对 F 进行取值进行遍历。板结化指数如图 9.1 所示：

图 9.1 300mm 降雨放牧强度与板结化指数



根据不同降雨条件下，进行计算，如图 9.2 所示



可以看出 300mm 降水量时，板结化指数较小。现在对板结化指数进行归一化，并且设定 0.5 为板结化指数的阈值。对不同降雨条件下的最大放牧强度进行求解。

编写 matlab 代码，计算出全部的板结化程度指数，使用 find 函数找到板结化指数小于等于 0.5 的值所在下标。如表 9.1 所示

表 9.1 不同降水量条件下放牧强

降水量	放牧强度阈值
300mm	7.9
600mm	7.9
900mm	7.9
1200mm	7.9

十、问题六模型的建立与求解

本题基于问题 4 放牧方案，即放牧强度为中度放牧(MGI)，放牧方式为划区轮牧(没有轮次，仅有放牧小区 G8 G11 G16)。并且假设附件 13 对问题的求解没有影响。土地状态用问题 4 得到的有机物含量和土壤湿度来表征。

依次选取问题四输出的在中度放牧情况下不同放牧小区 9 月份的所有数据并按年份进行合并。这样得到的数据即为该放牧小区在中度放牧的情况下历年来 9 月份的土地状态。土地状态也与年份序列相关。用 LSTM 多变量模型进行预测，同时输出评价指标 MAPE，RMSE，MAE 对模型进行打分。最后以二维图方式表示 2023 年九月份的土地状态见 10.3。

10.1 算法流程及实现

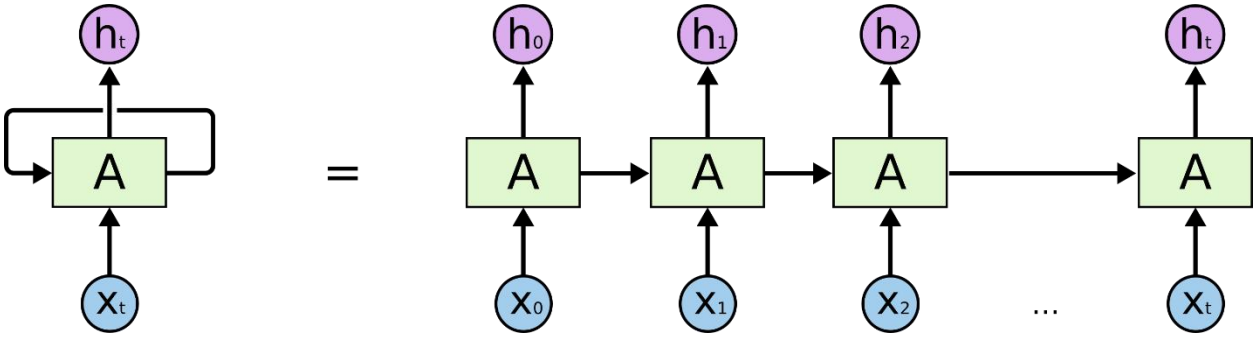
表 10.1 算法流程

过程： 1.在放牧强度为中度放牧 MGI 的前提下，依次选定放牧小区 G8 G11 G16 的 9 月份数；2.按年份进行分组，每组内部合并，各列数据由均值表征； 3.训练 LSTM 模型，输入为年份序列，输出为代表土地状态的列即土壤湿度和有机物含量； 4.以年份序列最后一项为输入再次输进 LSTM 模型中获得下一个年份序列元素也就是该小区 23 年 9 月份的土地状态信息； 5.显示该小区 2012,2014,2016,2018,2020 年的土地状态和 2012,2014,2016,2018,2020， 2023 年的土地状态对比图

10.2 模型说明

10.2.1LSTM 模型

原始的 RNN 基本结构图如下图所示

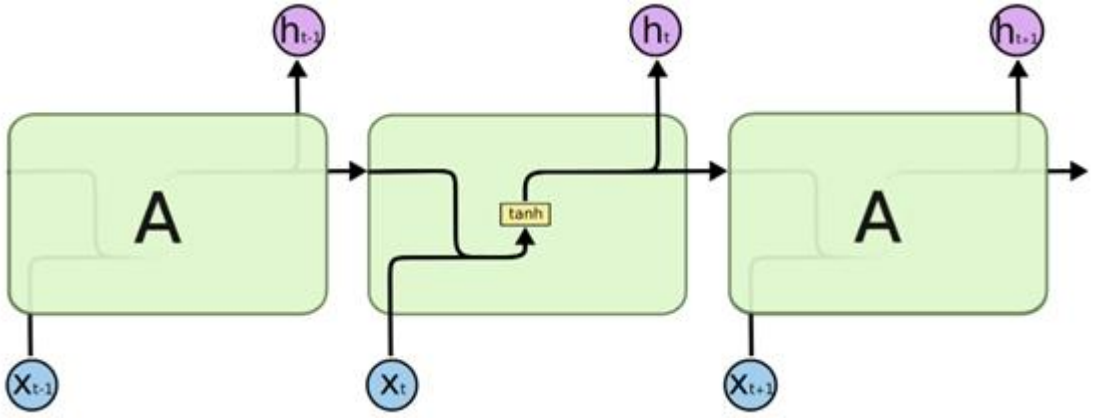


由上图可知，RNN 展开后由多个相同的单元连续连接。但是，RNN 的实际结构确和上图左边的结构所示，是一个自我不断循环的结构。即随着输入数据的不断增加，上述自我循环的结构把上一次的

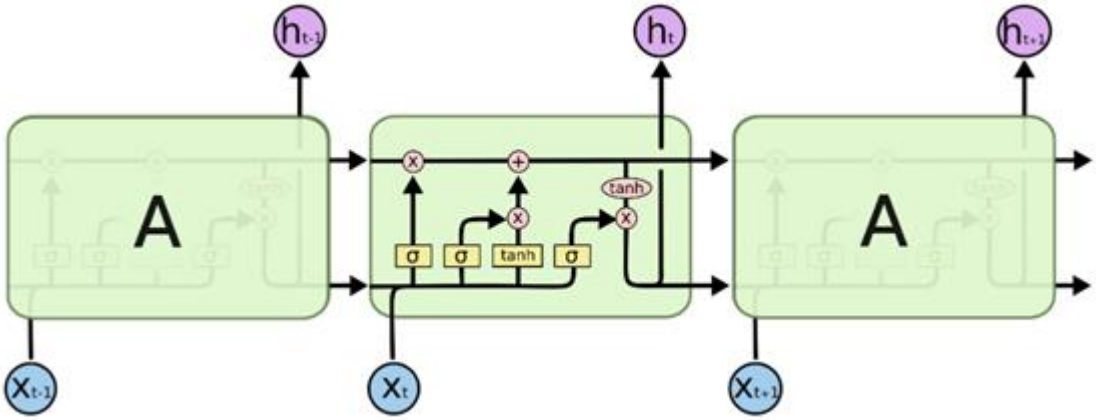
状态传递给当前输入，一起作为新的输入数据进行当前轮次的训练和学习，一直到输入或者训练结束，最终得到的输出即为最终的预测结果。

LSTM 是一种特殊的 RNN，两者的区别在于普通的 RNN 单个循环结构内部只有一个状态。而 LSTM 的单个循环结构(又称为细胞)内部有四个状态。相比于 RNN，LSTM 循环结构之间保持一个持久的单元状态不断传递下去，用于决定哪些信息要遗忘或者继续传递下去。

包含三个连续循环结构的 RNN 如下图，每个循环结构只有一个输出：



包含三个连续循环结构的 LSTM 如下图，每个循环结构有两个输出，其中一个即为单元状态



一层 LSTM 是由单个循环结构结构组成，既由输入数据的维度和循环次数决定单个循环结构需要自我更新几次，而不是多个单个循环结构连接组成，即当前层 LSTM 的参数总个数只需计算一个循环单元就行，而不是计算多个连续单元的总个数。

10.3 结果分析

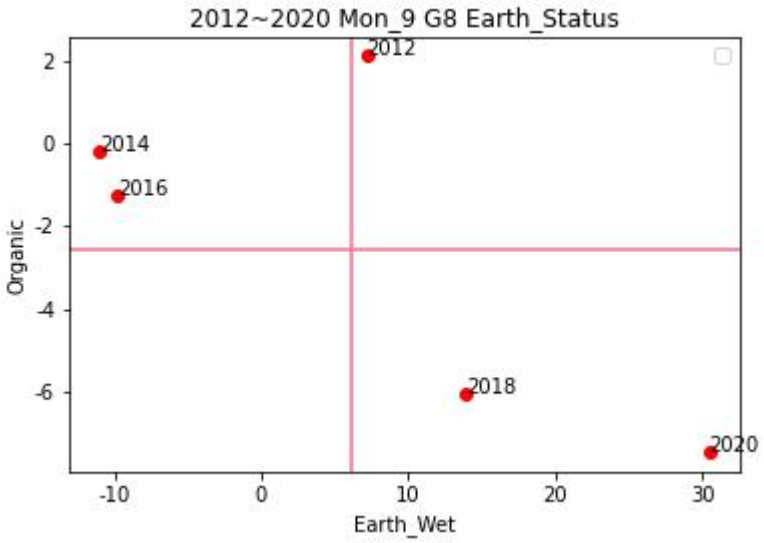


图 10.2 放牧小区 G8 2012-2020 年 9 月份土地状态预测

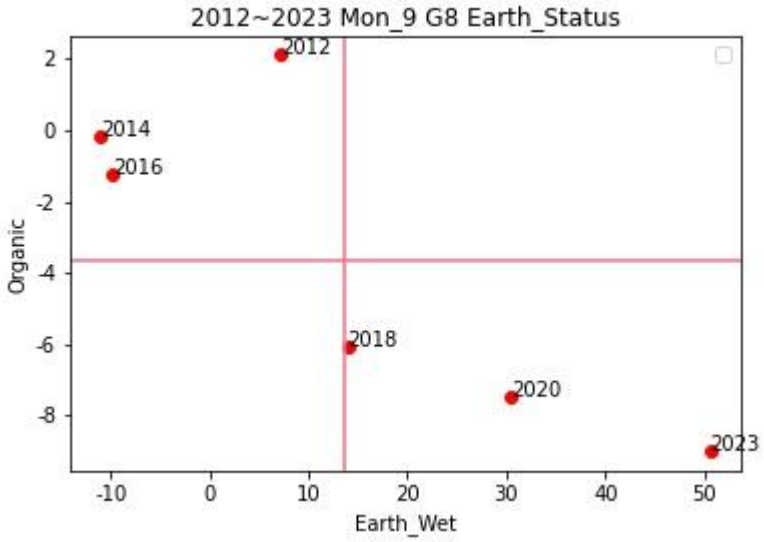


图 10.3 放牧小区 G8 2012-2023 年 9 月份土地状态预测

表 10.4 放牧小区 G8 历年 9 月份土壤湿度和有机物含量预测值与真实值对比评价指标

	MAPE	RMSE	MAE
土壤湿度	50.09	7.97	6.34
有机物含量	50.40	1.41	1.09

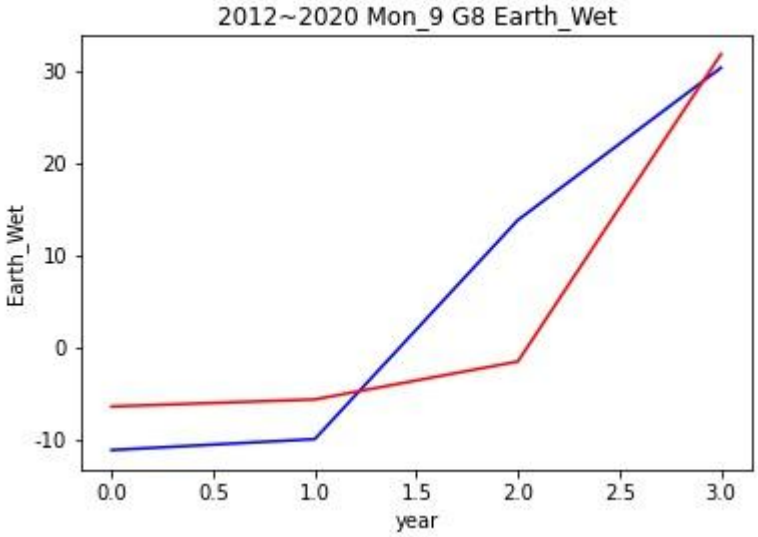


图 10.5 放牧小区 G8 历年 9 月份土壤湿度预测值（红色）与真实值（蓝色）对比

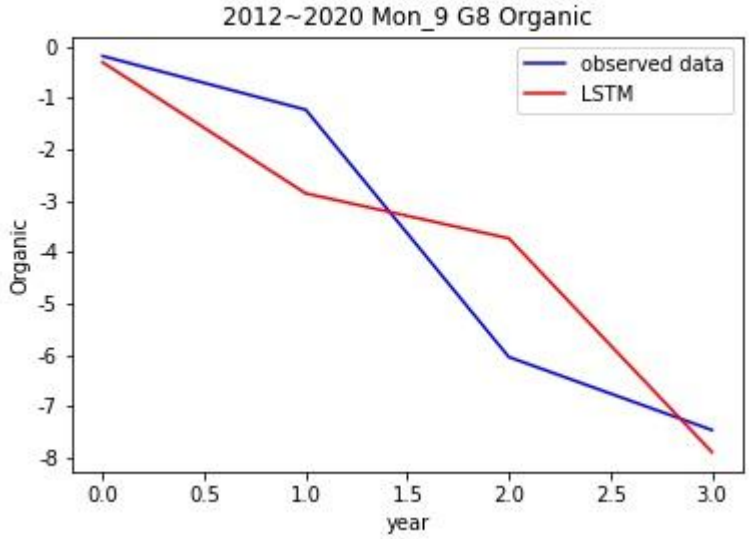


图 10.6 放牧小区 G8 历年 9 月份有机物含量预测值（红色）与真实值（蓝色）对比

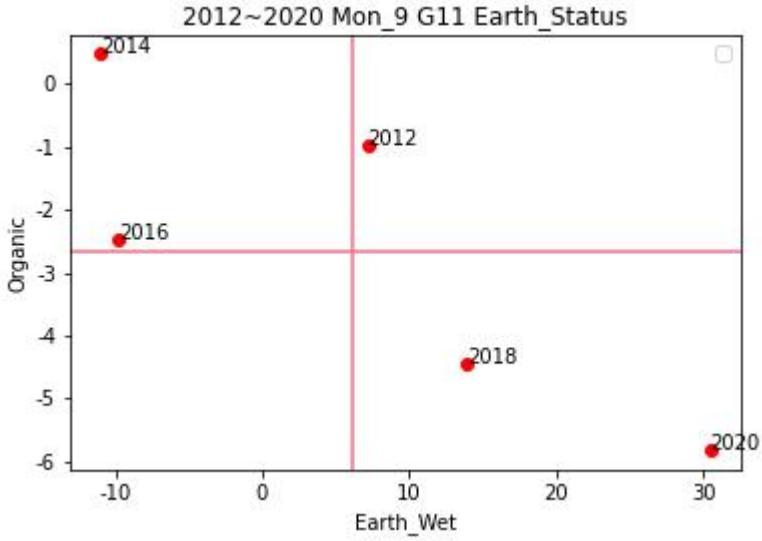


图 10.7 放牧小区 G11 2012-2020 年 9 月份土地状态预测

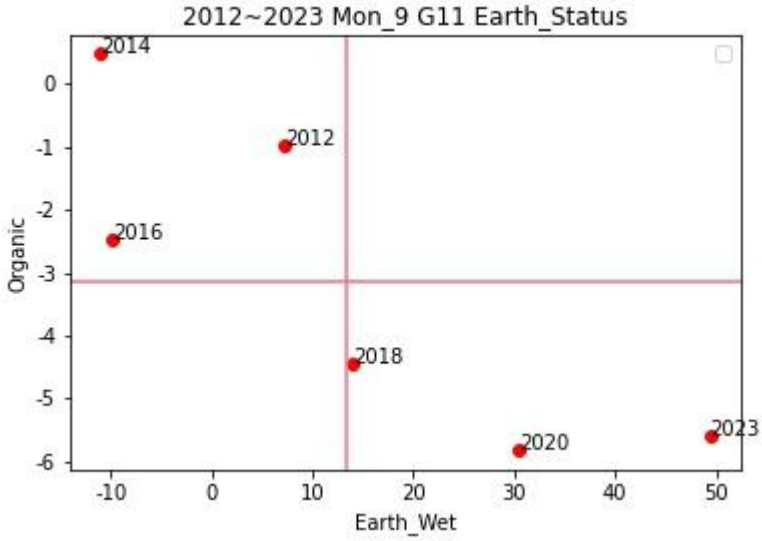


图 10.8 放牧小区 G11 2012-2023 年 9 月份土地状态预测

表 10.9 放牧小区 G11 历年 9 月份土壤湿度和有机物含量预测值与真实值对比评价指标

	MAPE	RMSE	MAE
土壤湿度	53.92	8.83	6.64
有机物含量	114.08	1.19	1.09

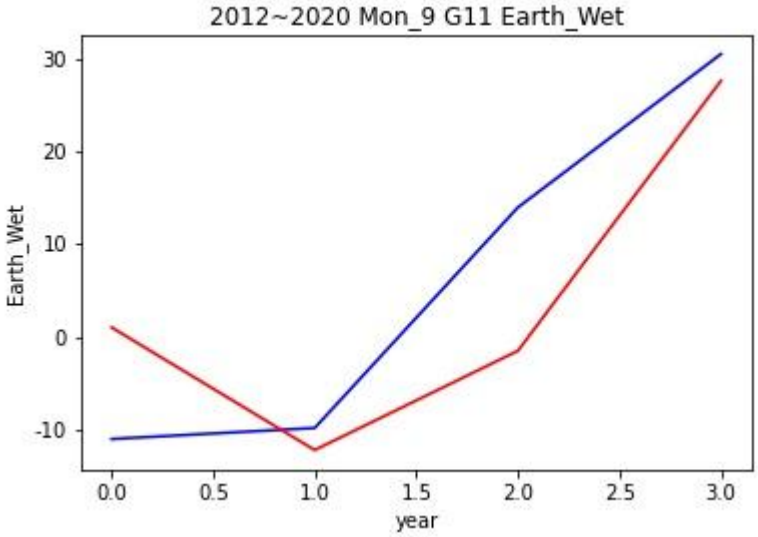


图 10.10 放牧小区 G11 历年 9 月份土壤湿度预测值（红色）与真实值（蓝色）对比

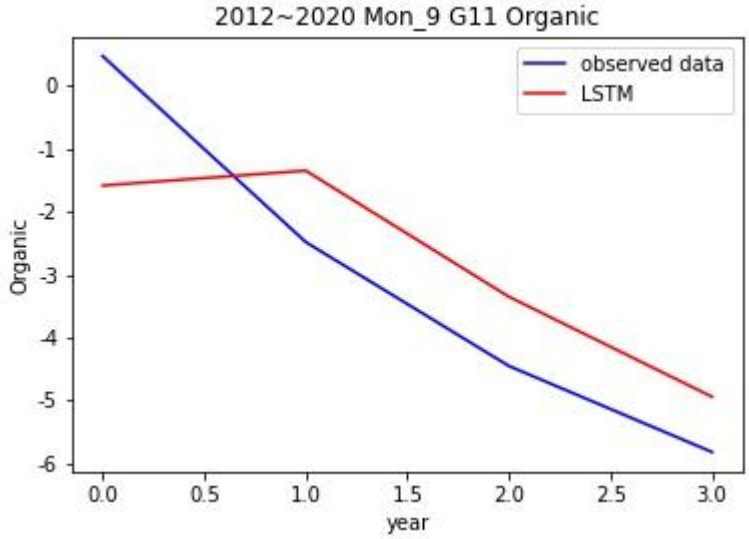


图 10.11 放牧小区 G11 历年 9 月份有机物含量预测值（红色）与真实值（蓝色）对比

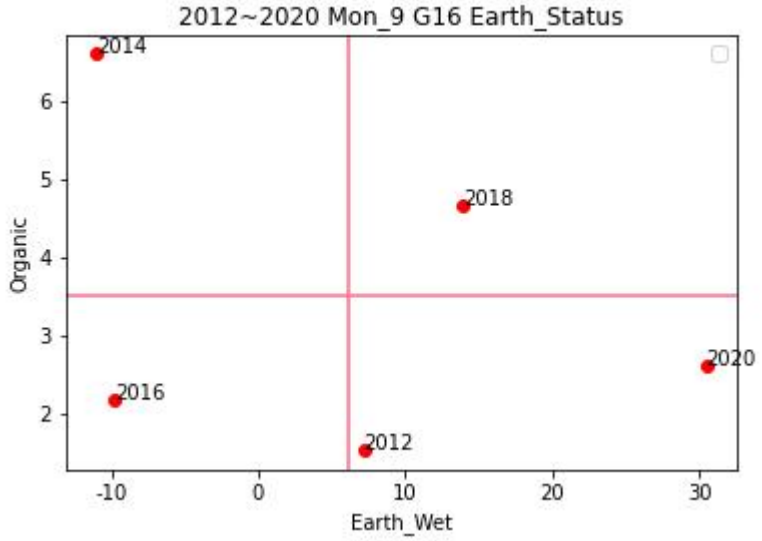


图 10.12 放牧小区 G16 2012-2020 年 9 月份土地状态预测

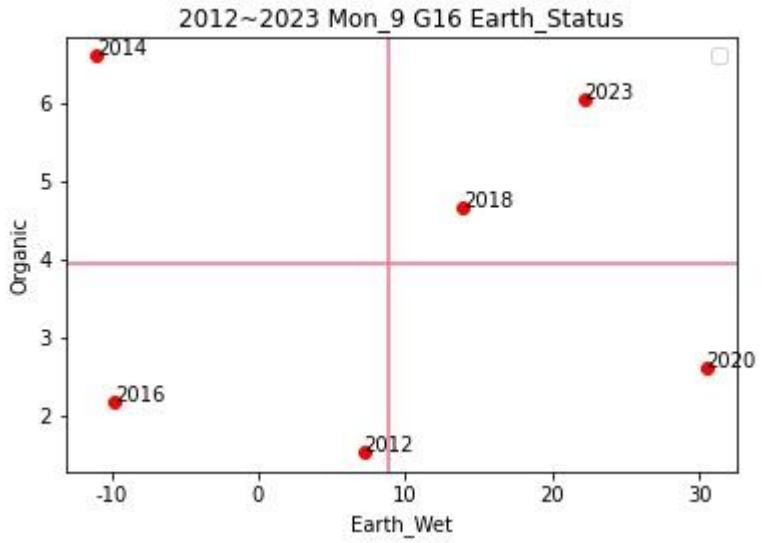


图 10.13 放牧小区 G16 2012-2023 年 9 月份土地状态预测

表 10.14 放牧小区 G16 历年 9 月份土壤湿度和有机物含量预测值与真实值对比评价指标

	MAPE	RMSE	MAE
土壤湿度	95.66	13.07	12.77
有机物含量	12.62	0.41	0.36

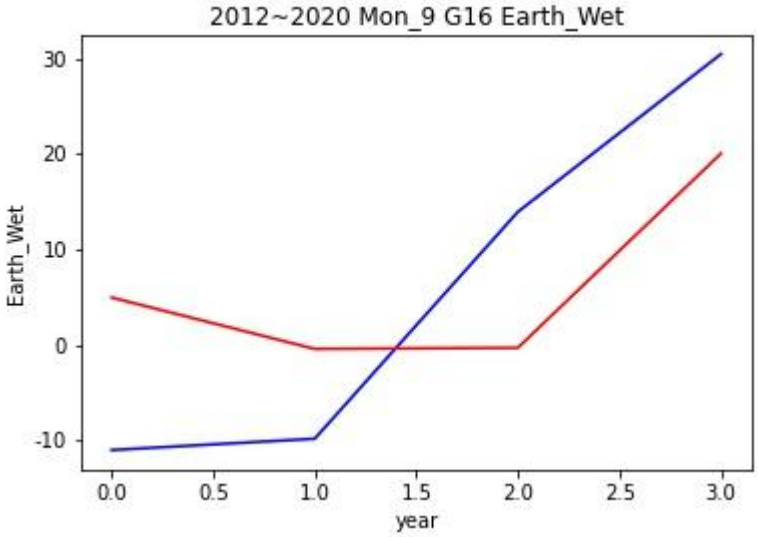


图 10.15 放牧小区 G16 历年 9 月份土壤湿度预测值（红色）与真实值（蓝色）对比

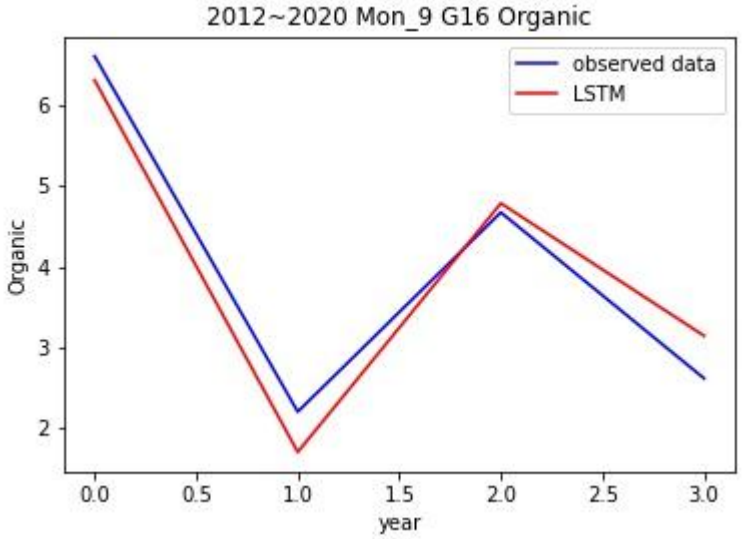


图 10.16 放牧小区 G16 历年 9 月份有机物含量预测值（红色）与真实值（蓝色）对比

如上所述，可见除了 G16 小区外，其余小区 MAE 偏大。MAE 越大则说明模型拟合的程度越不好，泛化性能越低。造成这种问题的原因可能是数据量过少。因为这种将某月不同年份的全部年限数据求均值并保存，会使得数据缩减到仅仅具有 6 行(因为只有 6 年)。数据过少很容易在模型训练中出现欠拟合。从而导出不完美的模型。改进的做法是用全体数据而不是用历年 9 月的数据进行模型训练。数据量增多更容易得到泛化性好的模型。

十一、模型的分析与检验

11.1 误差分析

11.1.1 问题二的误差分析

未对数据进行预处理，包括清洗，去极值，归一化等。对最终结果的准确度会造成一定影响。

11.1.2 问题三的误差分析

数据使用不当。虽然用附件 15 与附件 14 合并进行数据扩充，但附件 15 的大部分数据都未曾使用到。

11.1.3 问题六的误差分析

数据集做均值近似后数据量过少，拟合的效果不好。模型不够泛化；数据集都是偶数年，题干需要预测奇数 2023 年。年份不是偶数可能也会对预测数据造成影响。

11.2 模型的检验

本文要解决的六个问题中模型在建立的过程中通过了相应软件的检验，具有一定的合理性。

十二、模型的评价

12.1 模型优点

(1) 算法速度快，响应性好；
(2) 搜寻的结果满足所有约束要求，有较强的的实用性；
(3) 建立了问题之间的联系，使得整个问题之间具备整体性；
(4) 在筛选主要变量的过程中，综合了降维方法的优点，使得筛选效果更佳；

12.2 模型缺点

问题二：在与标准 LSTM 模型对比下，存在不接近的情况。说明模型还有一定的改进空间。推测这部分原因是来自代码实现中模型未经过调参。

问题三：模型特征的使用太少，仅有放牧强度和放牧小区，实质上只有两个特征。从而大部分模型 r2_score 接近于 0，模型性能不佳，没有很好的拟合效果。

问题六：模型训练时数据量过少，得到的模型不佳。可以尝试用所有数据进行训练，而不要先对年份进行分组后进行合并作为该模型训练的数据集；根据数据集，模型应该只适用于偶数年，对奇数年的预测可能会不准确。

总体而言本文所建的模型和使用的算法可行性高，具有有效性，探索空间大，值得进一步的研究。

十三、参考文献

[1]Westheimer FH. Why nature chose phosphates [J]. Science, 1987, 235 (4793): 1173-1178

[2]WU Y, TIAN Y, ZHOU J, et al. Ecological stoichiometric characteristics of soil carbon, nitrogen, and phosphorus under different grazing regimes[J]. China J Appl Environ Biol, 2019, 25: 801-807.

[3]李素英, 李晓兵, 莺歌, et al. 基于植被指数的典型草原区生物量模型——以内蒙古锡林浩特市为例[J]. 植物生态学报, 2007.

[4]魏卫东, 李希来. 放牧草地载畜量与放牧率研究方法分析[J]. 草业与畜牧, 2011, 000(008):1-4,28.

[5]杜春雨,范文义.叶面积指数与植被指数关系研究[J].林业勘查设计,2013(2):77-80

[6]何霜. 基于 MODIS 数据的植被指数与植被覆盖度关系研究——以比值植被指数和归一化植被指数为例[J]. 科技创新与应用, 2015.

[7]徐霞, 成亚薇, 江红蕾, 等. 风速变化对草原生态系统的影响研究进展[J]. 生态学报, 2017, 37(12): 4289-4298.

[8]张敬超, 金磊. 内蒙古鄂温克族自治旗温性草甸草原土壤含水量动态变化与气候因子的关系[J]. 畜牧与饲料科学, 2019, 40(1): 75-79.

[9]朱晨, 师春香, 席琳, 等. 中国区域不同深度土壤湿度模拟和评估[J]. 气象科技, 2013, 41(3): 529-536.

[10]封建民, 王涛. 呼伦贝尔草原沙漠化现状及历史演变研究[J]. 干旱区地理, 2004, 27(3): 356-360.

[11]李永宏, 莫文红, 杨持, 等. 内蒙古主要草原植物群落地上生物量和理论载畜量及其与气候的关系[J]. 干旱区资源与环境, 1994, 8(4): 43-50.

[12]董世魁, 江源, 黄晓霞. 草地放牧适宜度理论及牧场管理策略[J]. 资源科学, 2002, 24(6): 35-41.

[13]Woodward S J R. Wake G C. McCall D G. Optimal grazing of a multi—paddock system using a discrete time model[j]. Agri—cultural Systems. 1995，48：119—139.

附录源程序

###问题 2Python 源程序###

```
import xlrd
import numpy as np
from sklearn.ensemble import RandomForestRegressor
import pandas as pd
import warnings
warnings.filterwarnings('ignore')



f_name = '问题 2 数据.xlsx'
data = xlrd.open_workbook(f_name)
table = data.sheets()[0]
df = pd.DataFrame(columns=['年份', '月份', '10cm 湿度', '40cm 湿度', '100cm 湿度', '200cm 湿度'])  # 存储最终结果

for mon in [0, 1, 2]:  # 1~3 月
    # 蒸发量 气温 降水量 气压 风速
    zf = [table.cell_value(i, 2) for i in range(mon * 11 + 1, mon * 11 + 12)]
    qw = [table.cell_value(i, 3) for i in range(mon * 11 + 1, mon * 11 + 12)]
    js = [table.cell_value(i, 4) for i in range(mon * 11 + 1, mon * 11 + 12)]
    qy = [table.cell_value(i, 5) for i in range(mon * 11 + 1, mon * 11 + 12)]
    fs = [table.cell_value(i, 6) for i in range(mon * 11 + 1, mon * 11 + 12)]
    Factor = np.array([zf, qw, js, qy, fs]).transpose()# 统称为气候因素
    # 不同深度的湿度
    sd10 = np.array([table.cell_value(i, 7) for i in range(mon * 11 + 1, mon * 11 + 12)]).flatten()
    sd40 = np.array([table.cell_value(i, 8) for i in range(mon * 11 + 1, mon * 11 + 12)]).flatten()
    sd100 = np.array([table.cell_value(i, 9) for i in range(mon * 11 + 1, mon * 11 + 12)]).flatten()
```

```
    sd200 = np.array([table.cell_value(i, 10) for i in range(mon * 11 + 1, mon * 11 + 12)]).flatten()

    # 该月份第几次出现
    time = np.array(range(len(zf))).reshape(-1, 1)
    #t_test = np.array(range(12)).reshape(-1, 1)
    time_test = np.array(range(len(zf)+1)).reshape(-1, 1)

    # 随机森林构造月份出现的次数与气候因素之间的关系模型
    zf_model = RandomForestRegressor()
    zf_model.fit(time, zf)  # 月份出现的次数与蒸发量之间关系模型
    qw_model = RandomForestRegressor()
    qw_model.fit(time, qw)  # 月份出现的次数与气温之间关系模型
    js_model = RandomForestRegressor()
    js_model.fit(time, js)  # 月份出现的次数与降水量之间关系模型
    qy_model = RandomForestRegressor()
    qy_model.fit(time, qy)  # 月份出现的次数与气压之间关系模型
    fs_model = RandomForestRegressor()
    fs_model.fit(time, fs)  # 月份出现的次数与风速之间关系模型

    # 随机森林构造气候因素与不同深度土壤湿度之间的关系模型
    sd10_model = RandomForestRegressor()
    sd10_model.fit(Factor, sd10)  # 10cm 湿度预测模型
    sd40_model = RandomForestRegressor()
    sd40_model.fit(Factor, sd40)  # 40cm 湿度预测模型
    sd100_model = RandomForestRegressor()
    sd100_model.fit(Factor, sd100)  # 100cm 湿度预测模型
    sd200_model = RandomForestRegressor()
    sd200_model.fit(Factor, sd200)  # 200cm 湿度预测模型

    # 预测未来蒸发量 气温 降水量 气压 风速 time_test 是 0~11 指该月份在 2012-2023 共 12 年的时间序列 预测结果只取最后一年的 即 2023 年的 其他年的肯定和已知数据有偏差 很正常
    zf_test = zf_model.predict(time_test)
    qw_test = qw_model.predict(time_test)
    js_test = js_model.predict(time_test)
    qy_test = qy_model.predict(time_test)
    fs_test = fs_model.predict(time_test)
    Factor_test = np.array([zf_test, qw_test, js_test, qy_test, fs_test]).transpose()

    # 预测未来的湿度
    sd10_test = sd10_model.predict(Factor_test)
    sd40_test = sd40_model.predict(Factor_test)
    sd100_test = sd100_model.predict(Factor_test)
    sd200_test = sd200_model.predict(Factor_test)

    # 输出结果
    s='当前是 2023 年第'+repr(mon+1)+'月'
    print(s)
    print('输出不同深度的湿度值为', [2023,mon+1,sd10_test[-1], sd40_test[-1], sd100_test[-1], sd200_test[-1]] )
    df.loc[len(df)] =[2023,mon+1,sd10_test[-1], sd40_test[-1], sd100_test[-1], sd200_test[-1]]

Results = {}
for mon in [3, 4, 5, 6, 7, 8, 9, 10, 11]:  # 4~12 月
    # 蒸发量 气温 降水量 气压 风速
```

```
    zf = [table.cell_value(i, 2) for i in range(mon * 10 + 4, mon * 10 + 14)]
    qw = [table.cell_value(i, 3) for i in range(mon * 10 + 4, mon * 10 + 14)]
    js = [table.cell_value(i, 4) for i in range(mon * 10 + 4, mon * 10 + 14)]
    qy = [table.cell_value(i, 5) for i in range(mon * 10 + 4, mon * 10 + 14)]
    fs = [table.cell_value(i, 6) for i in range(mon * 10 + 4, mon * 10 + 14)]
    Factor = np.array([zf, qw, js, qy, fs]).transpose()# 统称为气候因素
    # 不同深度的湿度
    sd10 = np.array([table.cell_value(i, 7) for i in range(mon * 10 + 4, mon * 10 + 14)]).flatten()
    sd40 = np.array([table.cell_value(i, 8) for i in range(mon * 10 + 4, mon * 10 + 14)]).flatten()
    sd100 = np.array([table.cell_value(i, 9) for i in range(mon * 10 + 4, mon * 10 + 14)]).flatten()
    sd200 = np.array([table.cell_value(i, 10) for i in range(mon * 10 + 4, mon * 10 + 14)]).flatten()

    # 该月份第几次出现
    time = np.array(range(len(zf))).reshape(-1, 1)
    #t_test = np.array(range(11)).reshape(-1, 1)
    time_test = np.array(range(len(zf)+2)).reshape(-1, 1)#+2 是因为 4-12 月份只到 2021 年 2022 2023 的没有 所以需要输出 12 年的预测值 取最后两年

    # 随机森林构造月份出现的次数与气候因素之间的关系模型
    zf_model = RandomForestRegressor()
    zf_model.fit(time, zf)  # 月份出现的次数与蒸发量之间关系模型
    qw_model = RandomForestRegressor()
    qw_model.fit(time, qw)  # 月份出现的次数与气温之间关系模型
    js_model = RandomForestRegressor()
    js_model.fit(time, js)  # 月份出现的次数与降水量之间关系模型
    qy_model = RandomForestRegressor()
    qy_model.fit(time, qy)  # 月份出现的次数与气压之间关系模型
    fs_model = RandomForestRegressor()
    fs_model.fit(time, fs)  # 月份出现的次数与风速之间关系模型

    # 随机森林构造气候因素与不同深度土壤湿度之间的关系模型
    sd10_model = RandomForestRegressor()
    sd10_model.fit(Factor, sd10)  # 10cm 湿度预测模型
    sd40_model = RandomForestRegressor()
    sd40_model.fit(Factor, sd40)  # 40cm 湿度预测模型
    sd100_model = RandomForestRegressor()
    sd100_model.fit(Factor, sd100)  # 100cm 湿度预测模型
    sd200_model = RandomForestRegressor()
    sd200_model.fit(Factor, sd200)  # 200cm 湿度预测模型

    # 预测未来蒸发量 气温 降水量 气压 风速 time_test 是 0~11 指该月份在 2012-2023 共 12 年的时间序列 预测结果只取最后一年的 即 2023 年的 其他年的肯定和已知数据有偏差 很正常
    zf_test = zf_model.predict(time_test)
    qw_test = qw_model.predict(time_test)
    js_test = js_model.predict(time_test)
    qy_test = qy_model.predict(time_test)
    fs_test = fs_model.predict(time_test)
    Factor_test = np.array([zf_test, qw_test, js_test, qy_test, fs_test]).transpose()
```

```
    # 预测未来的湿度
    sd10_test = sd10_model.predict(Factor_test)
    sd40_test = sd40_model.predict(Factor_test)
    sd100_test = sd100_model.predict(Factor_test)
    sd200_test = sd200_model.predict(Factor_test)

    # 输出结果
    Results[str(mon)] = [[], []]
    for i in [-2, -1]:#倒数前两个结果是预测的 2022 2023 年的 4-12 月份数据
        if i == -2:
            Results[str(mon)][i + 2] =[2022,mon+1,sd10_test[i], sd40_test[i], sd100_test[i], sd200_test[i]]
        else:
            Results[str(mon)][i + 2] =[2023,mon+1,sd10_test[i], sd40_test[i], sd100_test[i], sd200_test[i]]

for mon in range(3, 12):
    s='当前是 2022 年第'+repr(mon+1)+'月'
    print(s)
    df.loc[len(df)] = Results[str(mon)][0]
    print('输出不同深度的湿度值为', Results[str(mon)][0] )
for mon in range(3, 12):
    s='当前是 2023 年第'+repr(mon+1)+'月'
    print(s)
    df.loc[len(df)] = Results[str(mon)][1]
    print('输出不同深度的湿度值为', Results[str(mon)][1] )

df.to_csv('第 2 题结果.csv', sep=',')
```

###问题 2Python 源程序###

```
import xlrd
import numpy as np
from sklearn.ensemble import RandomForestRegressor,GradientBoostingClassifier
import pandas as pd
import warnings
warnings.filterwarnings('ignore')


f_name = '问题 2 数据.xlsx'
data = xlrd.open_workbook(f_name)
table = data.sheets()[0]
df = pd.DataFrame(columns=['年份', '月份', '10cm 湿度', '40cm 湿度', '100cm 湿度', '200cm 湿度'])  # 存储最终结果

for mon in [0, 1, 2]:  # 1~3 月
    # 蒸发量 气温 降水量 气压 风速
    zf = [table.cell_value(i, 2) for i in range(mon * 11 + 1, mon * 11 + 12)]
    qw = [table.cell_value(i, 3) for i in range(mon * 11 + 1, mon * 11 + 12)]
    js = [table.cell_value(i, 4) for i in range(mon * 11 + 1, mon * 11 + 12)]
    qy = [table.cell_value(i, 5) for i in range(mon * 11 + 1, mon * 11 + 12)]
    fs = [table.cell_value(i, 6) for i in range(mon * 11 + 1, mon * 11 + 12)]
    Factor = np.array([zf, qw, js, qy, fs]).transpose()# 统称为气候因素
    # 不同深度的湿度
    sd10 = np.array([table.cell_value(i, 7) for i in range(mon * 11 + 1, mon * 11 + 12)]).flatten()
    sd40 = np.array([table.cell_value(i, 8) for i in range(mon * 11 + 1, mon * 11 + 12)]).flatten()
    sd100 = np.array([table.cell_value(i, 9) for i in range(mon * 11 + 1, mon * 11 + 12)]).flatten()
    sd200 = np.array([table.cell_value(i, 10) for i in range(mon * 11 + 1, mon * 11 + 12)]).flatten()

    # 该月份第几次出现
```

```
    time = np.array(range(len(zf))).reshape(-1, 1)
    #t_test = np.array(range(12)).reshape(-1, 1)
    time_test = np.array(range(len(zf)+1)).reshape(-1, 1)

    # 提升树构造月份出现的次数与气候因素之间的关系模型
    zf_model = GradientBoostingClassifier()
    zf_model.fit(time, zf)  # 月份出现的次数与蒸发量之间关系模型
    qw_model = GradientBoostingClassifier()
    qw_model.fit(time, qw)  # 月份出现的次数与气温之间关系模型
    js_model = GradientBoostingClassifier()
    js_model.fit(time, js)  # 月份出现的次数与降水量之间关系模型
    qy_model = GradientBoostingClassifier()
    qy_model.fit(time, qy)  # 月份出现的次数与气压之间关系模型
    fs_model = GradientBoostingClassifier()
    fs_model.fit(time, fs)  # 月份出现的次数与风速之间关系模型

    # 提升树构造气候因素与不同深度土壤湿度之间的关系模型
    sd10_model = GradientBoostingClassifier()
    sd10_model.fit(Factor, sd10)  # 10cm 湿度预测模型
    sd40_model = GradientBoostingClassifier()
    sd40_model.fit(Factor, sd40)  # 40cm 湿度预测模型
    sd100_model = GradientBoostingClassifier()
    sd100_model.fit(Factor, sd100)  # 100cm 湿度预测模型
    sd200_model = GradientBoostingClassifier()
    sd200_model.fit(Factor, sd200)  # 200cm 湿度预测模型

    # 预测未来蒸发量 气温 降水量 气压 风速 time_test 是 0~11 指该月份在 2012-2023 共 12 年的时间序列 预测结果只取最后一年的 即 2023 年的 其他年的肯定和已知数据有偏差 很正常
    zf_test = zf_model.predict(time_test)
    qw_test = qw_model.predict(time_test)
    js_test = js_model.predict(time_test)
    qy_test = qy_model.predict(time_test)
    fs_test = fs_model.predict(time_test)
    Factor_test = np.array([zf_test, qw_test, js_test, qy_test, fs_test]).transpose()

    # 预测未来的湿度
    sd10_test = sd10_model.predict(Factor_test)
    sd40_test = sd40_model.predict(Factor_test)
    sd100_test = sd100_model.predict(Factor_test)
    sd200_test = sd200_model.predict(Factor_test)

    # 输出结果
    s='当前是 2023 年第'+repr(mon+1)+'月'
    print(s)
    print('输出不同深度的湿度值为', [2023,mon+1,sd10_test[-1], sd40_test[-1], sd100_test[-1], sd200_test[-1]] )
    df.loc[len(df)] =[2023,mon+1,sd10_test[-1], sd40_test[-1], sd100_test[-1], sd200_test[-1]]

Results = {}
for mon in [3, 4, 5, 6, 7, 8, 9, 10, 11]:  # 4~12 月
    # 蒸发量 气温 降水量 气压 风速
    zf = [table.cell_value(i, 2) for i in
```

```
range(mon * 10 + 4, mon * 10 + 14)]
    qw = [table.cell_value(i, 3) for i in range(mon * 10 + 4, mon * 10 + 14)]
    js = [table.cell_value(i, 4) for i in range(mon * 10 + 4, mon * 10 + 14)]
    qy = [table.cell_value(i, 5) for i in range(mon * 10 + 4, mon * 10 + 14)]
    fs = [table.cell_value(i, 6) for i in range(mon * 10 + 4, mon * 10 + 14)]
    Factor = np.array([zf, qw, js, qy, fs]).transpose()# 统称为气候因素
    # 不同深度的湿度
    sd10 = np.array([table.cell_value(i, 7) for i in range(mon * 10 + 4, mon * 10 + 14)]).flatten()
    sd40 = np.array([table.cell_value(i, 8) for i in range(mon * 10 + 4, mon * 10 + 14)]).flatten()
    sd100 = np.array([table.cell_value(i, 9) for i in range(mon * 10 + 4, mon * 10 + 14)]).flatten()
    sd200 = np.array([table.cell_value(i, 10) for i in range(mon * 10 + 4, mon * 10 + 14)]).flatten()

    # 该月份第几次出现
    time = np.array(range(len(zf))).reshape(-1, 1)
    #t_test = np.array(range(11)).reshape(-1, 1)
    time_test = np.array(range(len(zf)+2)).reshape(-1, 1)#+2 是因为 4-12 月份只到 2021 年 2022 2023 的没有 所以需要输出 12 年的预测值 取最后两年

    # 提升树构造月份出现的次数与气候因素之间的关系模型
    zf_model = GradientBoostingClassifier()
    zf_model.fit(time, zf)  # 月份出现的次数与蒸发量之间关系模型
    qw_model = GradientBoostingClassifier()
    qw_model.fit(time, qw)  # 月份出现的次数与气温之间关系模型
    js_model = GradientBoostingClassifier()
    js_model.fit(time, js)  # 月份出现的次数与降水量之间关系模型
    qy_model = GradientBoostingClassifier()
    qy_model.fit(time, qy)  # 月份出现的次数与气压之间关系模型
    fs_model = GradientBoostingClassifier()
    fs_model.fit(time, fs)  # 月份出现的次数与风速之间关系模型

    # 提升树构造气候因素与不同深度土壤湿度之间的关系模型
    sd10_model = GradientBoostingClassifier()
    sd10_model.fit(Factor, sd10)  # 10cm 湿度预测模型
    sd40_model = GradientBoostingClassifier()
    sd40_model.fit(Factor, sd40)  # 40cm 湿度预测模型
    sd100_model = GradientBoostingClassifier()
    sd100_model.fit(Factor, sd100)  # 100cm 湿度预测模型
    sd200_model = GradientBoostingClassifier()
    sd200_model.fit(Factor, sd200)  # 200cm 湿度预测模型

    # 预测未来蒸发量 气温 降水量 气压 风速 time_test 是 0~11 指该月份在 2012-2023 共 12 年的时间序列 预测结果只取最后一年的 即 2023 年的 其他年的肯定和已知数据有偏差 很正常
    zf_test = zf_model.predict(time_test)
    qw_test = qw_model.predict(time_test)
    js_test = js_model.predict(time_test)
    qy_test = qy_model.predict(time_test)
    fs_test = fs_model.predict(time_test)
```

```
    Factor_test = np.array([zf_test, qw_test, js_test, qy_test, fs_test]).transpose()

    # 预测未来的湿度
    sd10_test = sd10_model.predict(Factor_test)
    sd40_test = sd40_model.predict(Factor_test)
    sd100_test = sd100_model.predict(Factor_test)
    sd200_test = sd200_model.predict(Factor_test)

    # 输出结果
    Results[str(mon)] = [[], []]
    for i in [-2, -1]:#倒数前两个结果是预测的 2022 2023 年的 4-12 月份数据
        if i == -2:
            Results[str(mon)][i + 2] =[2022,mon+1,sd10_test[i], sd40_test[i], sd100_test[i], sd200_test[i]]
        else:
            Results[str(mon)][i + 2] =[2023,mon+1,sd10_test[i], sd40_test[i], sd100_test[i], sd200_test[i]]

for mon in range(3, 12):
    s='当前是 2022 年第'+repr(mon+1)+'月'
    print(s)
    df.loc[len(df)] = Results[str(mon)][0]
    print('输出不同深度的湿度值为', Results[str(mon)][0] )
for mon in range(3, 12):
    s='当前是 2023 年第'+repr(mon+1)+'月'
    print(s)
    df.loc[len(df)] = Results[str(mon)][1]
    print('输出不同深度的湿度值为', Results[str(mon)][1] )

df.to_csv('问题 2 结果.csv', sep=',')
```

###问题三 Python 源程序###

```
# -*- coding: utf-8 -*-
"""问题 3 代码及流程.ipynb

Automatically generated by Colaboratory.

Original file is located at

https://colab.research.google.com/drive/1EWskMI89-601ru7eGClwZiL6LWiAk4rT

# **问题 3**
"""

! pip install --upgrade xlrd
! pip install pandas==1.2.0
! pip install xgboost
! pip install lightgbm
! pip install matplotlib
! pip install mplfonts
! pip install pybaobabdt
! apt install libgraphviz-dev
! pip install pygraphviz
! pip install pydotplus

import warnings
import chardet
import pybaobabdt
import pandas as pd
import xgboost as xgb
import lightgbm as lgb
import numpy as np
from sklearn.metrics import r2_score,classification_report,f1_score,make_scorer
from sklearn.model_selection import train_test_split,GridSearchCV,KFold
warnings.filterwarnings('ignore')
import matplotlib.pyplot as plt
from mplfonts.bin.cli import init
init()
from mplfonts import use_font
use_font('Noto Serif CJK SC')#指定中文字体
from sklearn.tree import
```

```
DecisionTreeRegressor

# 附件 14 是 12 14 16 18 20 年
# 附件 15 是 16 17 18 19 20 年
# 只合并相同的 16 18 20 年
attachment_14 = pd.read_excel('附件 14 不同放牧强度土壤碳氮监测数据.xlsx')
tmp1=attachment_14[attachment_14.year==2020]
tmp2=attachment_14[attachment_14.year==2018]
tmp3=attachment_14[attachment_14.year==2016]
attachment_14=pd.concat([tmp1,tmp2,tmp3],axis=0)
attachment_14

# 输出文件原格式
def  GetEncodingSheme(_filename):
    with open(_filename, 'rb') as file:
        buf = file.read()
    result = chardet.detect(buf)
    return result['encoding']

# 原格式转换为 utf-8
def ChangeEncoding(_infilename, _outfilname, _encodingsheme='utf-8'):
  ifEncodeSheme = GetEncodingSheme(_infilename)
  with open(_infilename, 'r', encoding=ifEncodeSheme) as fr:
    tempContent = fr.read()
  with open(_outfilname, 'w', encoding=_encodingsheme) as fw:
    fw.write(tempContent)

filename = '附件 15 群落结构监测数据集.csv'
print('原格式为：', GetEncodingSheme(filename)) #输出原格式
ChangeEncoding('附件 15 群落结构监测数据集.csv', '附件 15 群落结构监测数据集_new.csv', 'utf-8') #格式转化为 utf-8
print('文件已转换为 utf-8 编码')

f=open('附件 15 群落结构监测数据集_new.csv')
attachment_15=pd.read_csv(f,encoding='utf-8')
tmp1=attachment_15[attachment_15.年份==2020]
tmp2=attachment_15[attachment_15.年份==2018]
tmp3=attachment_15[attachment_15.年份==2016]
attachment_15=pd.concat([tmp1,tmp2,tmp3],axis=0)
attachment_15

# 附件 14 15 生成合并索引 bloar=小区-年份 如 G6-2012
attachment_14['bloar']=attachment_14.apply(lambda x:x['放牧小区（plot）']+'-'+str(x['year']),axis=1)
attachment_15['bloar']=attachment_15.apply(lambda x:x['放牧小区 Block']+'-'+str(x['年份']),axis=1)

# 以 bloar 列为索引 合并附件 14 15
attachment_1415=pd.merge(attachment_14,attachment_15,on='bloar')
attachment_1415

# 不同列缺失值数量 生殖苗过多 认为是无关项
attachment_1415.isnull().sum()

# 人为保留相关项
# '放牧小区（plot）','轮次'——放牧方式 只考虑选择划区轮牧
# '放牧强度（intensity）'——放牧强度 附件 14 的放牧强度(NG LGI MGI HGI)和附件 15 的处理(无牧（0 天） 轻牧（3 天） 中牧（6 天）重牧（12 天）)完全对应 仅保留一个就行
#  'SOC 土壤有机碳', 'SIC 土壤无机碳','STC 土壤全碳', '全氮 N', '土壤 C/N 比' ——化学
```

```
性质
attachment_1415=attachment_1415[['年份', '放牧小区（plot）', '轮次',
                                 '放牧强度（intensity）',
                                 'SOC 土壤有机碳', 'SIC 土壤无机碳','STC 土壤全碳', '全氮 N', '土壤 C/N 比']]

plt.hist(attachment_1415['放牧强度（intensity）'], bins=50, color='steelblue')
plt.xlabel('放牧强度（intensity）')
plt.ylabel('数量')

plt.hist(attachment_1415['轮次'], bins=50, color='steelblue')
plt.xlabel('轮次')
plt.ylabel('数量')

plt.hist(attachment_1415['放牧小区（plot）'], bins=50, color='steelblue')
plt.xlabel('放牧小区（plot）')
plt.ylabel('数量')

# 放牧方式量化
# 将小区号 轮次号进行 one-hot 编码 1 代表 attachment_1415 本行原本是该小区号/轮次号
attachment_1415=pd.concat([attachment_1415,pd.get_dummies(attachment_1415['放牧小区（plot）'],prefix='小区号_')],axis=1)#12 列
attachment_1415=pd.concat([attachment_1415,pd.get_dummies(attachment_1415['轮次'],prefix='轮次_')],axis=1)#5 列
attachment_1415

# 放牧强度量化
# 对照（NG， 0 羊/天/公顷 ）、轻度放牧强度（LGI， 2 羊/天/公顷 ）、中度放牧强度（MGI，4 羊/天/公顷 ）和重度放牧强度（HGI，8 羊/天/公顷 ）
attachment_1415['放牧强度（intensity）']=attachment_1415['放牧强度（intensity）'].map(
{
    'NG':0,
    'LGI':2,
    'MGI':4,
    'HGI':8
})
attachment_1415

# 重新选择
attachment_1415=attachment_1415[['年份', '放牧强度（intensity）',
        '小区号_G11', '小区号_G12', '小区号_G13','小区号_G16', '小区号_G17', '小区号_G18', '小区号_G19', '小区号_G20','小区号_G21', '小区号_G6', '小区号_G8', '小区号_G9',
        '轮次_牧前', '轮次_第一轮牧后', '轮次_第二轮牧后','轮次_第三轮牧后', '轮次_第四轮牧后',
        'SOC 土壤有机碳', 'SIC 土壤无机碳','STC 土壤全碳', '全氮 N', '土壤 C/N 比'
        ]]
attachment_1415

attachment_1415.isnull().sum()/attachment_1415.shape[0]#计算缺失比例

# 3sigma 去极值
def filter_extreme_3sigma(dataframe,n=3):
  for i in dataframe.columns:
    mean=dataframe[i].mean()
    std=dataframe[i].std()
    max_range=mean+n*std
    min_range=mean-n*std
    dataframe[i] = pd.DataFrame(np.clip(dataframe[i].values, min_range, max_range), columns=None)
  return dataframe
attachment_1415=filter_extreme_3sigma(attachment_1415)

# 需要预测的数据
data = pd.read_excel('需要预测的值.xlsx')
```

```
data = pd.concat([data,pd.get_dummies(data['放牧小区（plot）'],prefix='小区号_')],axis=1)#12 列
data['放牧强度（intensity）']=data['放牧强度（intensity）'].map(
{
    'NG':0,
    'LGI':2,
    'MGI':4,
    'HGI':8
})
data = data[['放牧强度（intensity）',
        '小区号__G11', '小区号__G12', '小区号__G13','小区号__G16', '小区号__G17', '小区号__G18', '小区号__G19', '小区号__G20','小区号__G21', '小区号__G6', '小区号__G8', '小区号__G9',
        'SOC 土壤有机碳', 'SIC 土壤无机碳','STC 土壤全碳', '全氮 N', '土壤 C/N 比'
        ]]
data

# 模型评估
def performance_metric(y_true, y_predict):
    """计算并返回预测值相比于预测值的分数"""
    score = r2_score(y_true,y_predict)
    return score

def fit_model(X, y):
    """ 基于输入数据 [X,y]，利于网格搜索找到最优的决策树模型"""
    cross_validator = KFold()
    regressor = DecisionTreeRegressor()
    params = {"max_depth":np.arange(1,11)}
    scoring_fnc = make_scorer(performance_metric)
    grid = GridSearchCV(regressor,params,scoring=scoring_fnc)
    # 基于输入数据 [X,y]，进行网格搜索
    grid = grid.fit(X, y)
    print(pd.DataFrame(grid.cv_results_))
    # 返回网格搜索后的最优模型
    return grid.best_estimator_

# 模型训练
X=attachment_1415[['放牧强度（intensity）',
  '小区号__G11', '小区号__G12', '小区号__G13','小区号__G16', '小区号__G17', '小区号__G18', '小区号__G19', '小区号__G20','小区号__G21', '小区号__G6', '小区号__G8', '小区号__G9']]
from IPython.display import Image
from sklearn import tree
import pydotplus

y_SOC=attachment_1415['SOC 土壤有机碳']
y_SIC=attachment_1415['SIC 土壤无机碳']
y_STC=attachment_1415['STC 土壤全碳']
y_N =attachment_1415['全氮 N']
y_CN=attachment_1415['土壤 C/N 比']

X_data=data[['放牧强度（intensity）',
  '小区号__G11', '小区号__G12', '小区号__G13','小区号__G16', '小区号__G17', '小区号__G18', '小区号__G19', '小区号__G20','小区号__G21', '小区号__G6', '小区号__G8', '小区号__G9']]

count=0
name=['y_SOC','y_SIC','y_STC','y_N','y_CN']
for y in [y_SOC,y_SIC,y_STC,y_N,y_CN] :
  x_train, x_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=12)

  print('当前预测指标为：',name[count])


  #训练拟合
  # 决策树 基于训练数据，获得最优模型
  optimal_model = fit_model(x_train,y_train)
  print('最优模型', optimal_model)
```

```
    dot_data = tree.export_graphviz(optimal_model, out_file=None)
    graph = pydotplus.graph_from_dot_data(dot_data)
    s=name[count]+'.png'
    graph.write_png(s)
    # 输出最优模型的 'max_depth' 参数
    print("Parameter 'max_depth' is {} for the optimal model.".format(optimal_model.get_params()['max_depth']))

    #验证
    y_pred=optimal_model.predict(x_test)
    #print('预测值：',y_pred)
    #print('真实值：',y_test)

#print(classification_report(y_pred,y_test.))

    #模型评估
    score = performance_metric(y_test,y_pred)
    print('score:', score)
    #print('coef_', optimal_model.coef_)
    #print('intercept_', optimal_model.intercept_)

    # 推理
    y_data=optimal_model.predict(X_data)
    print('需要预测的值：',y_data)

    count+=1

###问题 4 Matlab 源程序###
clear;clc;

%2020 年数据为例
az=readmatrix("指数.xlsx")
a1=az(1,:)
a2=az(2:5,:)

s=[0 192*2 192*4 192*8]
for j=1:4
    k1=0
    for i=1:length(a1)

        k2 =a1(1,i)*a2(j,i)
        k1 =k1+k2
    end
    p(j)=k1
end
l1=mapminmax(p,0,1)

%综合有机含量
bz=readmatrix("有机 1.xlsx")
a3=az(7,1:5)
youji1=0
for i=1:length(a3)
    youji2=a3(i)*bz(:,i)
    youji1=youji1+youji2
end


%2020 年数据为例
a=0.258%植被
b=20.42%降水
c=s%畜量
d=24.06%蒸发
for i=1:length(c)

y(i)=-7.857*a+0.003*b-0.001*c(i)+0.276*d+13.487
end

%2020 年数据为例
k1=0.5%权重
k2=0.5
shidu=[ 18.1617    17.7777    17.3937    16.6257]
youjiwu=[18.0967666666667  16.76  14.65  16.6205]
for i=1:4
    p1(i)=k1*shidu(i)+k2*youjiwu(i)+1.39
end
l2=mapminmax(p1,0,1)
```

```
l3=l1+l2

###问题 5 Matlab 源程序###

clear;clc;
%2020 年数据为例
az=readmatrix("指数.xlsx")
a1=az(1,:)
a=[0.042454545 0.082 0.177727273
0.179545455 0.2063 0.3129 0.4382 0.4626
0.3922 0.265 0.2111 0.1335]%植被
b=[300 600 900 1200]%降水
c=[0 192*2 192*4 192*8]%畜量
d=[0.599090909 0.701818182 4.01 9.604
17.158 25.425 26.983 15.919 11.444 7.955
2.078 0.741]%蒸发
youjiwu=[18.0967666666667 16.76 14.65
16.6205]
k1=0.5%权重
k2=0.5
%for j=6%月份
    a1=a(6)
    d1=d(6)

    for k=1:4 %降水
        b1=b(k)
        qw=0
        for i=0:0.1:8%放牧强度
            qw=qw+1
            if i==0
                youjiwu =
18.0967666666667
            elseif i <= 2
                youjiwu = 16.76
            elseif i <= 4
                 youjiwu = 14.65
            else
                 youjiwu = 16.6205
            end

y=-7.857*a1+0.003*b1-0.001*i*192+0.276
*d1+13.487
            p1=k1*y+k2*youjiwu+1.39
            kg(k,qw)=p1
            kg=mapminmax(kg,0,1)
            gg=kg.'

            %kg=kg.'
            %kg(j,qw)=p1
            %kg=mapminmax(kg,0,1)
            %gg=kg.'

        end
    end
find(a<0.5)
%end
s=0:0.1:8
w=[300 600 900 1200]
%kg1=kg(:,:,6)

###问题 6 Python 源程序###

# -*- coding: utf-8 -*-
"""问题 6.ipynb

Automatically generated by Colaboratory.

Original file is located at

https://colab.research.google.com/drive/1
hkSB3vWrrdi-2k_Har1wAV8zHwEOXzeC
"""

! pip install matplotlib
! pip install mplfonts

# Commented out IPython magic to ensure
Python compatibility.
import pandas as pd
import warnings
from sklearn.metrics import r2_score
warnings.filterwarnings('ignore')
# %matplotlib inline
import numpy as np
from sklearn.preprocessing import
MinMaxScaler
from keras.models import Sequential
from keras.layers import Dense
```

```
from keras.layers import LSTM
from keras.models import Sequential, load_model
np.set_printoptions(suppress=True)

import matplotlib.pyplot as plt
from mplfonts.bin.cli import init
init()
from mplfonts import use_font
use_font('Noto Serif CJK SC')# 指定中文字体

from sklearn.preprocessing import MinMaxScaler
from pylab import *
plt.rcParams['font.sans-serif'] = ['SimHei']
plt.rcParams['axes.unicode_minus'] = False
import seaborn as sns
sns.set_palette("husl") # 设置所有图的颜色，使用 hls 色彩空间
import numpy as np
from sklearn import metrics
from sklearn.metrics import mean_squared_error #均方误差
from sklearn.metrics import mean_absolute_error #平方绝对误差

q4_res=pd.read_excel('MGIG8.xlsx')# 需要更改为三个文件遍历
q4_res

#选择区域
Block='G8'#按文件名一致

# 选出某个小区(如 G21)月份是 9 月份 的所有数据
gBlock_9=q4_res[(q4_res['放牧小区（plot）']==Block ) & (q4_res['月份']==9 )]
gBlock_9.reset_index(inplace=True,drop=True)
gBlock_9=gBlock_9.groupby('年份').mean()# 按年份合并，每一年的数值为均值
gBlock_9.reset_index(inplace=True,drop=False)
gBlock_9

# G21 5 年的土壤湿度和有机物含量相对关系
wet=gBlock_9['土壤湿度'].values
org=gBlock_9['有机物含量'].values
year=[2012,2014,2016,2018,2020]

fig,ax=plt.subplots()
ax.scatter(wet,org,c='r')

for i,txt in enumerate(year):
    ax.annotate(txt,(wet[i],org[i]))

plt.axhline(y=org.mean(),ls="-")
plt.axvline(x=wet.mean(),ls="-")
plt.legend()
#plt.xlabel('土壤湿度')
#plt.ylabel('有机物含量')
plt.xlabel('Earth_Wet')
plt.ylabel('Organic')
s='2012~2020 Mon_9 '+Block+' Earth_Status'
plt.title (s)
plt.savefig(s+'.png')

dataset=gBlock_9[['土壤湿度', '有机物含量']]
dataset

#########LSTM 多变量模型#############
def split_sequences(sequences, n_steps):
    X, y = list(), list()
    for i in range(len(sequences)):
        end_ix = i + n_steps
        if end_ix > len(sequences)-1:
            break
        # 最关键的不一样在这一步
        seq_x, seq_y = sequences[i:end_ix, :], sequences[end_ix, :]
        X.append(seq_x)
        y.append(seq_y)
```

```
        return np.array(X), np.array(y)
def
mean_absolute_percentage_error(y_true,
y_pred):

        return      np.mean(np.abs((y_true      -
y_pred) / y_true)) * 100

def fitlstmmodel(dataset,n_steps=1):
        #dataset：数据标准化后的 dataset
        # n_steps：分片大小，默认为 1
        #依次为：'PM2.5','AQI',  'PM10', 'SO2',
'CO', 'NO2', 'O3_8h', '最高气温', '最低气温'

        in_seq1=
dataset[:,0].reshape((dataset.shape[0], 1))
        in_seq2=
dataset[:,1].reshape((dataset.shape[0], 1))

        dataset      =      np.hstack((in_seq1,
in_seq2))
        X,    y    =    split_sequences(dataset,
n_steps)
        n_features = X.shape[2]#2
        model = Sequential()
        model.add(LSTM(300,
activation='relu',    return_sequences=True,
input_shape=(n_steps, n_features)))
        model.add(LSTM(300,
activation='relu'))

        # 和多对一不同点在于，这里多对多
的 Dense 的神经元=features 数目
        model.add(Dense(n_features))
        model.compile(optimizer='adam',
loss='mse')
        model.fit(X,        y,        epochs=100,
verbose=2,shuffle=False)
        model.save('lstm_model.h5')
        last_input=np.array(dataset[-1:,:])
        return X,y,last_input,n_features,n_steps
#  将整型变为 float
dataset = dataset.astype('float32')
#对数据集合进行标准化
scaler  =  MinMaxScaler(feature_range=(0,
1))

dataset=scaler.fit_transform(dataset)
#输入为标准化后的 dataset      #输出：X
为 lstm 的输入，y 为 lstm 的输出，
x_input_last 为最后一行 dataset 的数据，用
于预测未来的输入,n_features 是特征维度，
n_steps 是切片分层
X,y,last_input,n_features,n_steps=fitlstmmo
del(dataset,n_steps=1)

###预测与评分
#输入 1 为 lstm 的输入 X，输入 2 为 lstm
的输出 y，用于训练模型,输入 3 为标准化模
型
#输出：testPredict 为预测 close 的训练数
据，testY 为 close 的真实数据
#该函数目标输出训练的 RMSE 以及预测与
训练数据的对比
def Predict_RMSE_BA(X,y,scaler):
        model=load_model('lstm_model.h5')
        trainPredict = model.predict(X)
        testPredict                          =
scaler.inverse_transform(trainPredict)
        testY = scaler.inverse_transform(y)
        score(testY[:,0], testPredict[:,0])

        #土壤湿度', '有机物含量'
        plt.plot(testY[:,0],color='blue',
label='observed data')
        plt.plot(testPredict[:,0],      color='red',
label='LSTM')
        plt.xlabel('year')
        plt.ylabel('Earth_Wet')
        s='2012~2020      Mon_9      '+Block+'
Earth_Wet'
        plt.title (s)
        plt.savefig(s+'.jpg')
        plt.show()


        score(testY[:,1], testPredict[:,1])
        plt.plot(testY[:,1],color='blue',
```

```
label='observed data')
    plt.plot(testPredict[:,1], color='red', label='LSTM')
    plt.xlabel('year')
    plt.ylabel('Organic')
    s='2012~2020 Mon_9 '+Block+' Organic'
    plt.title (s)
    plt.legend()
    plt.savefig(s+'.jpg')
    plt.show()
    return testPredict,testY
def score(y_true, y_pre):
    # MSE
    print("MAPE :")

print(mean_absolute_percentage_error(y_true, y_pre))
    # RMSE
    print("RMSE :")

print(np.sqrt(metrics.mean_squared_error(y_true, y_pre)))
    # MAE
    print("MAE :")

print(metrics.mean_absolute_error(y_true, y_pre))
    # # R2
    # print("R2 :")
    #
print(np.abs(r2_score(y_true,y_pre)))
testPredict,testY=Predict_RMSE_BA(X,y,scaler)

def Predict_future_plot(predict_forword_number,x_input,n_features,n_steps,scaler,testPredict,testY):
    model=load_model('lstm_model.h5')
    predict_list=[]
    predict_list.append(x_input)
    while len(predict_list) < predict_forword_number:
        x_input = predict_list[-1].reshape((-1, n_steps, n_features))
        yhat = model.predict(x_input, verbose=0)
        #预测新值
        predict_list.append(yhat)
    #取出

    Predict_forword = scaler.inverse_transform(np.array([ i.reshape(-1,1)[:,0].tolist() for i in predict_list]))
    return Predict_forword[1:,:].tolist()

y_pre=Predict_future_plot(2,last_input,n_features,n_steps,scaler,testPredict,testY)
y_pre

# G21 5 年的土壤湿度和有机物含量相对关系
wet=gBlock_9['土壤湿度'].values
org=gBlock_9['有机物含量'].values
wet=np.append(wet,y_pre[0][0])
org=np.append(org,y_pre[0][1])
year=[2012,2014,2016,2018,2020,2023]

fig,ax=plt.subplots()
ax.scatter(wet,org,c='r')

for i,txt in enumerate(year):
    ax.annotate(txt,(wet[i],org[i]))

plt.axhline(y=org.mean(),ls="-")
plt.axvline(x=wet.mean(),ls="-")
plt.legend()
#plt.xlabel('土壤湿度')
#plt.ylabel('有机物含量')
plt.xlabel('Earth_Wet')
plt.ylabel('Organic')
s='2012~2023 Mon_9 '+Block+' Earth_Status'
plt.title (s)
plt.savefig(s+'.jpg')
```