中国研究生创新实践系列大赛

“华为杯”第二十届中国研究生

数学建模竞赛

题　目：　基于 XGBoost 与 LSTM 的碳排放指标与预测分析

摘　　要：

随着中国经济的飞速发展, 国家碳排放也呈现持续上升趋势, 这无疑加剧了全球气候变暖的问题. 为实现 2060 年碳中和的目标, 需深度研究能源、工业、交通、建筑及居民生活消费等部门的碳排放问题, 而在这些部门中, 化石能源是主要的二氧化碳来源. 因此, 发展绿色能源与提升能源利用效率显得尤为紧要.

针对问题一主要侧重于对区域碳排放及其相关指标的当前状态进行分析. 采用 SPSS 进行数据预处理, 并运用 **Anova 分析**和**基于梯度提升决策树（XGBoost）的指标贡献度模型**来探讨不同指标对碳排放的影响程度, 得到工业消费部门碳排放量显著高于其他部门的结论. 此外, 还利用基于**主成分分析（PCA）结合 Kaya 关联模型**对指标见的关联度进行了量化分析, 得到二氧化碳排放量 $C=\Omega\times \mathrm{GDP}_p\times\varepsilon\times C_r$.

针对问题二, 旨在建立一种**基于长短期记忆递归神经网络（LSTM）的多变量时间序列预测模型**, 预测 2021 至 2060 年间的能源消费量呈上升趋势, 反映了人口增长和经济发展对能源需求的影响. 并将各能源消费部门和各能源消费品种等指标加入碳排放量预测模型中, 融合了神经网络与时间序列预测的方法, 将碳排放量预测与各指标紧密联系，得到 2056-2060 年的能源消费量分别为 33255, 33266, 33276, 33285, 33294, 碳排放量分别为 76748, 76748, 76749, 76749, 76749, 实现碳达峰.

针对问题三, 围绕实现区域碳达峰和碳中和目标展开, 设计了自然情境、基准情景和雄心情景三种可能的未来场景. 在这些情境中, 重点探讨了政府政策、社会经济、市场潜力和可持续性等多个方面的影响, 并将这些因素与碳达峰和碳中和的时间点, 以及能效和非化石能源消费比例的提升紧密联系.

最后, 对模型进行了评估，分析了模型的优缺点.

关键词： 双碳目标 梯度提升决策树 主成分分析 长短期记忆递归神经网络 路径规划

目 录

一、问题重述

1.1 问题背景

随着我国经济的高速发展，中国温室气体排放量整体呈上升趋势，全球变暖的气候问题日益严重，中国作为世界碳排放量最大的国家，碳减排压力与责任都是前所未有的，所面临的风险与挑战也是未曾有过的[1]. 因此，要实现第二个百年奋斗目标与 2060 年碳中和目标，就必须将碳减排纳入亟待解决的难题中. 碳排放主要来自能源、工业、交通、建筑、居民生活消费五大部门.

其中能源为化石能源（燃料），主要包括天然资源有煤、油、天然气、油页岩等，化石燃料在燃烧过程中会生成大量的二氧化碳进入大气，增加温室气体的排放量，而这些化石能源被大量使用，是二氧化碳的主要来源. 降低碳排放，就要重视绿色技术创新，开展管理节能、技术节能和结构节能等能效工程来提高能源利用效率；提高非化石能源消费比重，加快发展风电、太阳能发电,因地制宜开发水电、生物质发电,积极安全有序发展核电[2].

1.2 问题提出

问题一：区域碳排放量以及经济、人口、能源消费量的现状分析

（1）建立指标与指标体系，指标要能够描述某区域经济、人口、能源消费量和碳排放量的状况，以及各部门的碳排放量状况，并研究各项指标之间的联系，计算部分指标的同比环比，为碳排放预测奠定基础.

（2）对（1）所得结果进行分析.

（3）建立碳排放量以及经济、人口、能源消费量各指标及其关联模型，分析环比同比的变化，确定碳排放预测模型的参数.

问题二：建立区域碳排放量以及经济、人口、能源消费量的预测模型

（1）建立基于人口和经济变化的能源消费量预测模型，从 2020 年开始，预测 2021 年到 2060 年间人口、经济和能源消费量的变化，将能源消费量分别和人口预测与经济预测进行关联.

（2）建立区域碳排放量预测模型，将其与人口、GDP、能源消费量预测进行关联，与各能源消费部门以及能源供应部门的能源消费量进行关联，并分别与他们的能源消费品种建立联系.

问题三：区域双碳（碳达峰与碳中和）目标与路径规划方法

（1）设计不少于三种情景，将其与碳达峰和碳中和时间节点进行关联，与能效提升和非化石能源消费比重提升关联.

（2）在多个情景下核算碳排放量，给出 3 个基本假设，要求区域碳排放量与多情景假设，各部门碳排放量总和一致，碳排放量核算模型与问题二中预测一致.

（3）确定双碳目标与路径，确定 GDP、人口、能源消费量的目标值，确定提高能源利用效率和提高非化石能源消费比重的目标值，完成能效提升等的定性定量分析.

二、问题分析

2.1 问题一分析

利用 SPSS 随数据进行预处理，补充缺失值，更改错误值. 问题一（1）需要建立指标分析区域和各部门的碳排放量以及经济、人口、能源消费量的现状， 考虑一级指标（影响

碳排放量指标）人口、经济、能源消费量，二级指标（评估碳排放量指标）年份、常驻人口数量、地区生产总值（GDP）、人均 GDP 等，利用 Anova 分析，比较不同部门之间的碳排放量差异，计算部分指标的同比环比.

问题一（2）要求分析该区域十二五、十三五期间的碳排放量的总量与变化趋势，建立**基于 XGBoost 的指标贡献度模型**，利用粒子群算法对模型参数优化，得到每个指标对碳排放量的影响程度，分析双碳目标面临的挑战.

问题一（3）参考双碳政策，建立**基于 PCA 的 Kaya 关联模型**，分析同比环比，将指标对碳排放量的影响进行量化分析，研究实现碳达峰与碳中和的路径选择，并确定碳排放量预测模型参数.

2.2 问题二分析

以问题一为基础，问题二（1）需要对区域的能源消费量与碳排放量进行预测，我们建立**基于 LSTM 的多变量时间序列预测模型**，利用改良的神经网络与时间序列预测模型相结合，将能源消费量的预测与人口预测、GDP 预测相关联，递归多步预测 2021 至 2060 年间的能源消费量.

由问题二（1）得到的能源消费量预测应用于问题二（2），实现碳排放量与人口、GDP 和能源消费量预测相关联. 为了实现碳排放量与各能源消费部门以及能源供应部门相关联，我们将部门进行精简，计算总碳排放量与各部门碳排放量以及能耗之间的 Pearson 系数，选择具有代表性的两个部门，将其加入预测模型的网络训练中，得到更完善的预测模型；并利用 **Kaya 模型**，将碳排放量的预测与各消费部门与能源供应部门的能源消费品种相关联，研究对碳排放因子产生的影响.

2.3 问题三分析

对于实现区域碳达峰碳中和目标设计三种情景，分别为自然情境、基准情景与率先碳达峰与碳中和的雄心情景，从三个情景做分析，各个情境的特点以及社会各部门技术发展速度如何，政府会采取哪些政策干预，在各个情景下下双碳目标何时达成，对社会经济、市场的潜在影响，是否可持续等，并与碳达峰碳中和的时间节点相关联，与能效提升和非化石能源消费比重提升相关联.

问题三（2）基于假设提出多情景下碳排放量核算方法. 并满足要求；问题三（3）通过查阅相关文献确定双碳目标与路径.

三、模型假设

假设 1：除了题目给出的影响指标之外，不会有其他影响指标，或者其他指标处于理想的状态下；

假设 2：煤炭、油品、其他能源包括的能源小类，如原煤、洗精煤、焦炭等碳排放因子为 0；

假设 3：损失的能源除电力和热力外不产生碳排放；

假设 4：数据完整性和准确性，假设提供的所有数据都是准确和完整的，不考虑由于数据收集、处理或传输中的误差所导致的不确定性.

假设 5：假设 GDP 和人口的增长是持续的，且能源消费量与人口和 GDP 的变化密切相关. 假定到 2035 年，GDP 将比 2020 年翻一番；到 2060 年，GDP 将比 2020 年翻两番.

假设 6: 假设到 2060 年, 生态碳汇的碳消纳量将达到基期碳排放量的 10%. 同样, 假设到 2060 年, 工程碳汇或碳交易的碳消纳量将达到基期碳排放量的 10%.

假设 7: 在多情景条件下, 我们假设区域与各部门能源消费量、能源消费品种及其碳排放量预测方法相一致, 将基于已经训练好的预测模型, 以确保预测的一致性和准确性.

假设 8: 假设政策出台和执行具有一定的连贯性和持续性, 且新政策的出台会在一定程度上推动技术的发展和应用.

四、符号说明

符号	说明
Φ_T	总能耗量
$\phi_i, i=1,2,3,4$	第一、二、三产业与生活能耗
$\Delta_i, i=1,2,3$	第一、二、三产业增加值
$\varphi_i, i=1,2,3$	第一、二、三产业单位增加值能耗
σ_f	生活能耗比重
ε	单位 GDP 能耗
Ω	总人口
M_p	人均能源消费量
C_r	单位能耗二氧化碳排放量
C_a	总碳排放量
C_p	人均碳排放量
NF_e	非化石能源发电比重
nf_e	非化石能源发电占比
EC_p	电力消费比重
$y_i, i=1,2,3$	第一、二、三产业 GDP 随时间的变化曲线

五、模型建立与求解

5.1 问题一的模型建立与求解

5.1.1 区域碳减排指标体系构建原则

（1）系统性原则

影响区域碳排放量的重要因素之一为经济, 国民经济各行业可分为第一、第二、第三产业和居民生活消费. 第一产业为农林消费部门; 第二产业分为能源供应部门和工业消费部门; 第三产业分为建筑消费部门和交通消费部门. 探究经济对于碳排放量的影响, 既要把握总体, 又要看到局部, 有整体性也有侧重点.

（2）逻辑关联性原则

建立系统的区域碳排放的指标体系, 要考虑各个指标之间的相互逻辑关系, 通过计算可以指出人口、经济、能源消耗量与碳排放量之间的关系, 各指标之间相互独立, 又彼此联系, 共同构成一个有机统一体. 指标体系的构建可以反映出具体的正负相关关系, 有助于总结各因素对于碳排放量的贡献, 并提出有针对性的措施与方法.

（3）可比与可量化性原则

指标体系的构建是为区域政策制定和科学管理服务的, 指标选取的计算量度需统一, 各指标应便于收集, 直观明了, 应具有较强的现实可操作性和可比性. 可以全面地、客观地评价过去、预测未来, 从而做出决策.

（4）动态性原则

人口、经济、能源消耗量与碳排放量之间的互动发展需要一定的时间尺度的指标才能够反映出来. 因此, 指标的选择要充分考虑到动态变化.

5.1.2 指标体系的确定与解释

基于指标体系的构建原则与碳减排相关理论方法, 同时查阅相关文献, 建立一个区域碳排放量预测指标体系如图 1, 下面分别从一级二级指标进行解释:

（1）一级指标为影响区域碳排放量的各类因素: 人口、经济、能源消费量.

（2）二级指标为评估区域碳排放量整体水平和变化趋势的基本评估指标: 年份、常驻人口数量、地区生产总值（GDP）、人均 GDP、总能源消费量、单位 GDP 能耗、人均能源消费量、单位能耗二氧化碳排放量、人均碳排放量、总碳排放量、农林消费部门碳排放量、工业消费部门碳排放量、交通消费部门碳排放量、建筑消费部门碳排放量、居民生活消费碳排放量. 以下是二级指标说明:

①年份反应了人口、经济、能源消耗量与碳排放量之间在一定时间尺度上的动态变化.

②人口的快速发展会促进社会经济发展水平, 而经济发展的快速发展消耗了大量的资源和能源, 经调查研究发现, 我国 70%的碳是企业排放的, 30%的碳是居民排放的, 这种发展模式使我国碳排放量快速上升. 所以人口是碳排放量的一大因素.

③地区生产总值与人均 GDP 直接代表了某地区的经济发展状况, GDP 水平越高代表该区域经济发展状况越好.

④总能源消费量指的是各个产业部门的能耗量与生活能耗的总和, 其中包括化石能源消费与非化石能源消费, 从加工转换来看, 包括一次能源（未加工转换的能源, 如煤炭、石油、天然气、太阳能、风能、水能、核能、生物质能、地热能等）和二次能源（指经过加工转换的能源, 如电能、热能、冷能、光伏、风电、水电、核电等）. 能源消耗会产生碳排放, 进而在探究碳排放量时不容忽视.

⑤单位 GDP 能耗又称为能源消费强度, 是区域能源利用效率的重要标志, 单位 GDP 能耗低, 则能源利用率高. 能源利用率的提高, 可实现经济增长与能源消费量增长的负相关变化, 进而破解发展与减排的矛盾. 单位 GDP 能耗为总能耗与 GDP 的比值, 总能耗Φ_T的计算公式见式（1）（2）:

$$\Phi_T = \phi_1 + \phi_2 + \phi_3 + \phi_4, \tag{1}$$

$$\phi_1 = \Delta_1 \times \varphi_1,\ \ \phi_2 = \Delta_2 \times \varphi_2,\ \ \phi_3 = \Delta_3 \times \varphi_3,\ \ \phi_4 = \Phi_T \times \sigma_f, \tag{2}$$

其中ϕ_1，ϕ_2，ϕ_3，ϕ_4分别为第一产业能耗、第二产业能耗、第三产业能耗以及生活能耗，Δ_1，Δ_2，Δ_3分别为第一产业增加值，第二产业增加值，第三产业增加值，φ_1，φ_2，φ_3分别为第一产业单位增加值能耗，第二产业增加值能耗，第三产业增加值能耗，σ_f为生活能耗比重.

GDP 的计算公式见（3）

$$\mathrm{GDP} = \Delta_1 + \Delta_2 + \Delta_3, \tag{3}$$

单位 GDP 见式（4）

$$\begin{aligned}
\varepsilon &= \frac{\Phi_T}{\mathrm{GDP}} = \frac{\phi_1 + \phi_2 + \phi_3 + \phi_4}{\Delta_1 + \Delta_2 + \Delta_3} \\
&= \frac{\Delta_1 \times \varphi_1 + \Delta_2 \times \varphi_2 + \Delta_3 \times \varphi_3 + \Phi_T \times \sigma_f}{\Delta_1 + \Delta_2 + \Delta_3} \\
&= \frac{\Delta_1}{\Delta_1 + \Delta_2 + \Delta_3}\varphi_1 + \frac{\Delta_2}{\Delta_1 + \Delta_2 + \Delta_3}\varphi_2 + \frac{\Delta_3}{\Delta_1 + \Delta_2 + \Delta_3}\varphi_3 + \frac{\Phi_T \times \sigma_f}{\Delta_1 + \Delta_2 + \Delta_3} \\
&= \delta_1\varphi_1 + \delta_2\varphi_2 + \delta_3\varphi_3 + \varepsilon\sigma_f,
\end{aligned} \tag{4}$$

其中δ_1，δ_2，δ_3分别为第一产业增加值权重，第二产业增加值权重，第三产业增加值权重，将上式（4）移项整理后得到式（5）：

$$\varepsilon = \frac{\delta_1\varphi_1 + \delta_2\varphi_2 + \delta_3\varphi_3}{1 - \sigma_f}, \tag{5}$$

⑥人均能源消费量=总能源消费量/总人数，人均能源消费量用来观察人口数量对于能源消费的影响，公式见（6），其中M_p为人均能源消费量，Ω为总人数，Φ_T为总能耗量:

$$M_p = \frac{\Phi_T}{\Omega}, \tag{6}$$

⑦单位能耗二氧化碳排放量=总碳排放量/总能耗量，是区域能源消费低碳排放的重要标志，单位能耗二氧化碳排放量低，表示能源消费中非化石能源消费比重大，能源消费产生的温室气体排放低，公式见（7），其中C_r为单位能耗二氧化碳排放量，C_a为总碳排放量，Φ_T为总能耗量:

$$C_r = \frac{C_a}{\Phi_T}, \tag{7}$$

⑧人均碳排放量=总碳排放量/总人数，反应人口数量对碳排放量的直接关系，人均碳排放量越大，说明人口对碳排放量的贡献越大，见式（8），其中C_p为人均碳排放量:

$$C_p = \frac{C_a}{\Omega}, \tag{8}$$

⑨总碳排放量C_a为国民经济各行业，第一、第二、第三产业和居民生活伴随着能源消费而产生的二氧化碳排放量，主要与化石能源消费量相关，既包含一次能源也包含二次能源.

⑩农林消费部门碳排放量、工业消费部门碳排放量、交通消费部门碳排放量、建筑消费部门碳排放量、居民生活消费碳排放量，是不同部门的碳排放量.

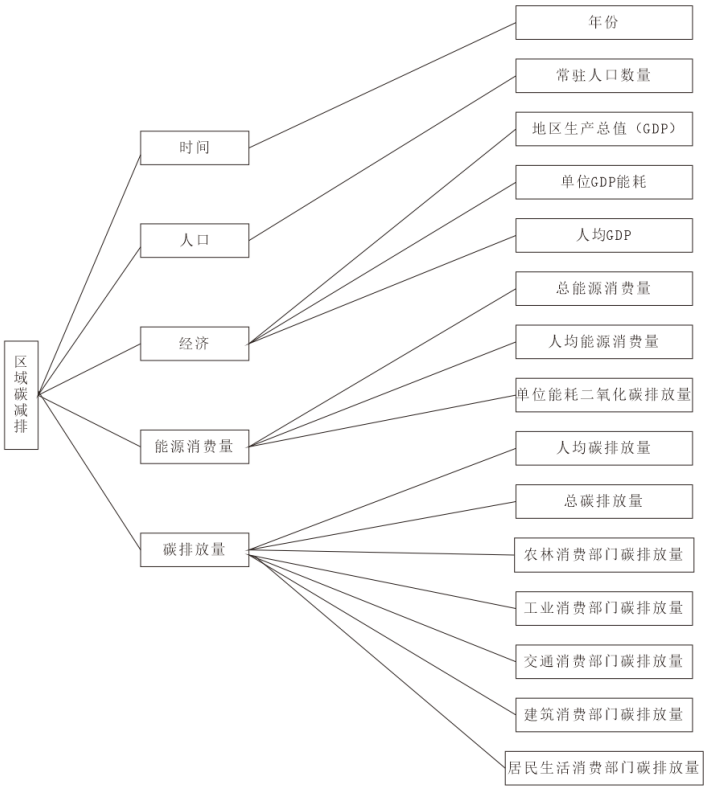

图 1 区域碳排放量评价指标体系

5.1.3 区域碳排放量以及经济、人口、能源消费量现状分析及指标变化

观察附件数据, 发现表中数据有缺失值与错误值, 故先利用 SPSS 进行数据处理, 将缺失值补充完整, 报错值进行修改.

根据 5.1.2 提出的指标, 对不同产业中生产总值随时间的变化进行初步分析.

参考经济与能源表中的生产总值（GDP）, 做出不同产业的生产总值随时间的变化图 2, 可以得出结论:

（1）第三产业的生产总值 GDP 呈非线性增长, 可以给出非线性回归曲线, 见式（9）

$$y_3 = 68180\sin(0.337x - 42.8) + 59740\sin(0.593x + 73.22) + 25200\sin(0.751x + 1020), \quad (9)$$

第二产业前期呈现增长趋势, 2018 年后走势趋于平缓, 给出其 GDP 变化曲线, 见式(10):

$$y_2 = -287.9\sin(x - \pi) + 0.424(x - 10)^2 - 1674000, \quad (10)$$

预测未来第二产业将继续趋于稳定, 第三产业 GDP 预计会持续增长后趋于平缓.

（2）2010 至 2014 年间, 第二产业生产总值 GDP 高于第三产业, 在 2014 到 2015 年, 二、三产业的生产总值交汇, 随后从 2015 年开始到 2020 年, 第三产业超过第二产业, 位于三个产业的 GDP 第一.

（3）第一产业（农林消费部门）的生产总值比较稳定, 预计将持续稳定发展, 给出 GDP 变化走势曲线, 见式（11）:

$$y_1 = 133.9\sin(x - \pi) + 0.7582(x - 10)^2 - 3017000. \quad (11)$$

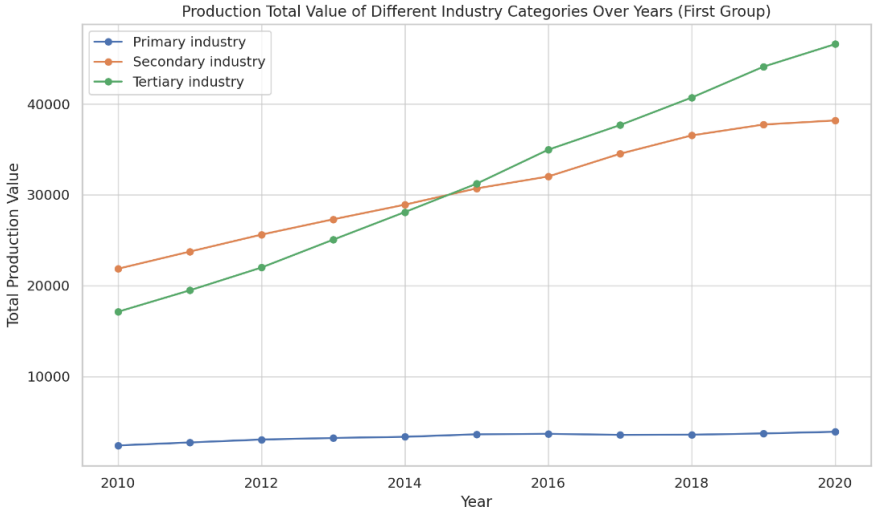

图 2 不同产业中 GDP 随时间的变化

根据经济与能源表中的生产总值，给出不同产业中各部门生产总值随时间的变化图 3，结合上图 2 不同产业总的生产总值，进一步得出结论：在第二产业中，生产总值集中分布于工业消费部分，在第三产业中，生产总值集中分布于建筑消费部门，其他部门的生产总值在总的生产总值中占比较低.

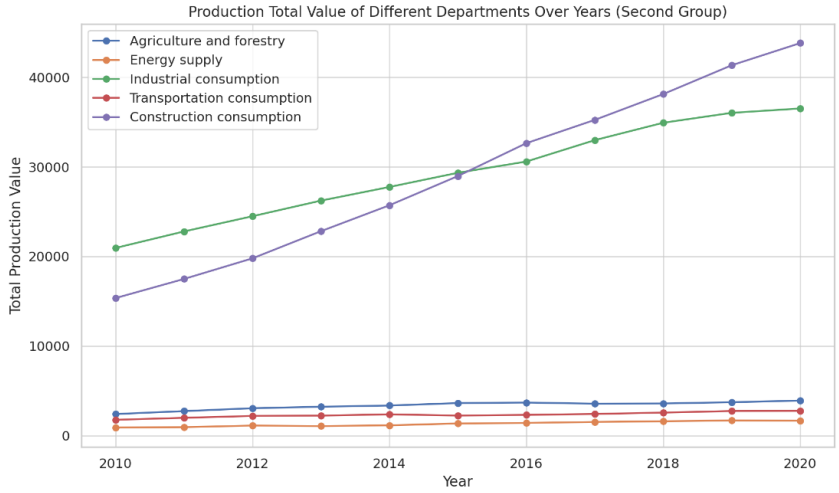

图 3 不同产业中各部门生产总值随时间的变化

下面给出不同产业能源消费量的分析.

根据经济与能源表中的能源消费量，给出各产业能源消费量随时间的变化图 4 以及不同产业中各部门能源消费量随时间的变化图 5，得出结论:

（1）根据图 4，第二产业能源消费量位居第一，且在 2020 年呈现下降趋势，第三产业消费量次之，第一产业消费量居于末位，与我国的经济发展模式相适应.

（2）根据图 5，第二产业工业消费部门的能源消费量最高，能源供应部门发电次之，能源产出呈现不断增长的趋势.

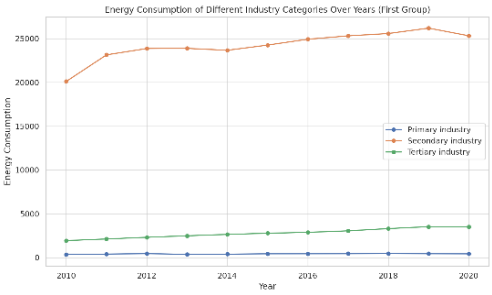

图 4 各产业能源消费量随时间的变化

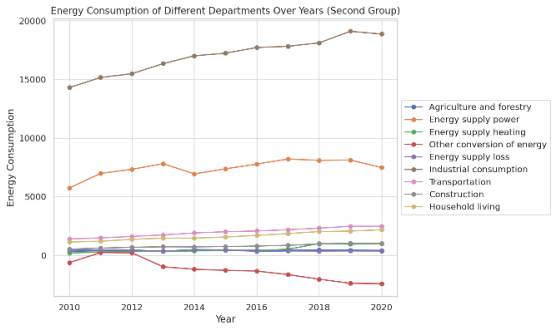

图 5 不同产业中各部门能源消费量随时间的变化

下面分析一次能源在 2010 到 2020 年间的消费量与产出量.

根据经济与能源表中的产业能耗结构，做出一、二、三产业对不同能源的消耗量均值图 6，得出结论:

（1）从一次能源类型来看，煤炭的消耗量主要集中于第二产业（即能源供应部门与工业消费部门），且煤炭消费量显著高于其他能源消费量，居于第一位；油品消耗量在第二产业与第三产业（即交通消费部门与建筑消费部门）的消耗中基本持平，还有部分在居民生活中消费，少量消耗于第一产业；天然气消耗量在第二产业中消耗居多，并有部分在居民生活中消费，还有一些在其他产业中.

（2）热力电力在产业能耗结构的第二产业能源供应部门中，大量产出用于第二能源的消费同时在工业消费部门中大量消费，保持经济的稳序发展，除此之外在其他产业中有少量消费.

（3）从产业类型来看，第二产业在一次能源消耗中占据主要地位，居高不下，同时在第二产业中二次能源的产出也是维持国家经济发展的重要环节，因此解决第二产业的能源消费量高的问题对实现碳达峰和碳中和有举足轻重的作用.

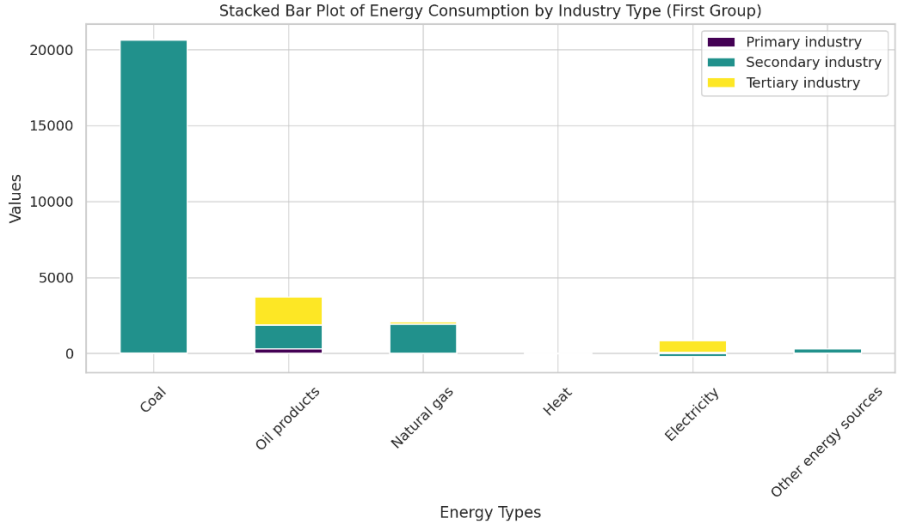

图 6 第一、二、三产业对不同能源的消耗量均值

根据经济与能源表中的产业能耗结构，做出一、二、三产业各个部门的能源消费量与产出量均值图 7，结合上图 6 对各产业的一次能源消费整体分析，进一步得出结论： 能源供应部门的消费量位居第一，同时其产出了大量的热力与电力，工业消费部门的能源消费量居于第二位，能源消费主要集中于第二产业的各部门.

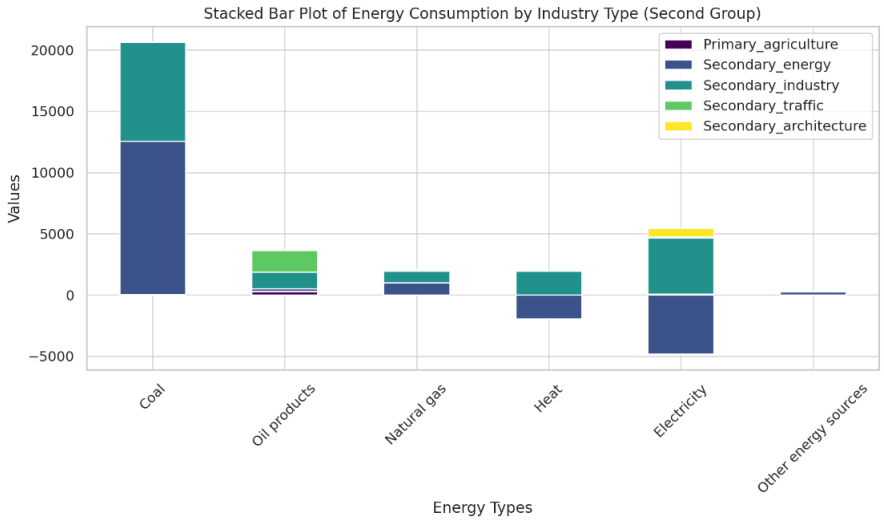

图 7 第一、二、三产业各个部门的能源消费量与产出量均值

下面分析二次能源在 2010 到 2020 年间的消费量.

根据附件中的经济与能源表, 参考能耗品种结构, 分别做出发电、供热与其他消费（二次能源）所产生的煤炭消费量、油品消费量以及天然气消费量, 做出如下不同能源类型在不同类别中的分布和占比图 8, 可以得出结论:

（1）发电产生的能源消费量最高, 供热产生的能源消耗量次之;

（2）煤炭消费量最高, 且煤炭消费量在发电中占比最高, 天然气消费量在发热中占比最高.

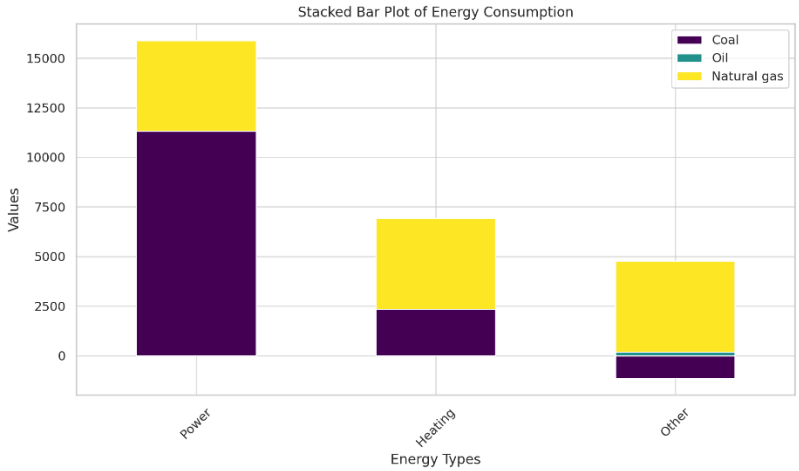

图 8 不同能源类型在发电、供热、其他中的分布和占比

根据附件中的经济与能源表, 做出能耗品种结构（即二次能源）中的煤炭消费量、油品消费量以及天然气消费量随着年份的增加, 其消费总量变化趋势图 9, 可以得出结论:

（1）发现煤炭消费量明显高于其他能源的消费量, 在 2011 年达到 20000 万 tce 之上后呈现稳定波动的趋势, 后在 2017 年后呈现波动下降的趋势, 预计 2020 年后其消耗量会不断降低, 减少煤炭燃烧所带来的碳排放.

（2）与此同时, 油品和天然气的消费总量呈现缓慢上升的趋势, 并逐渐在 2017 年后达到稳定, 且油品的消费量高于天然气的消费量.

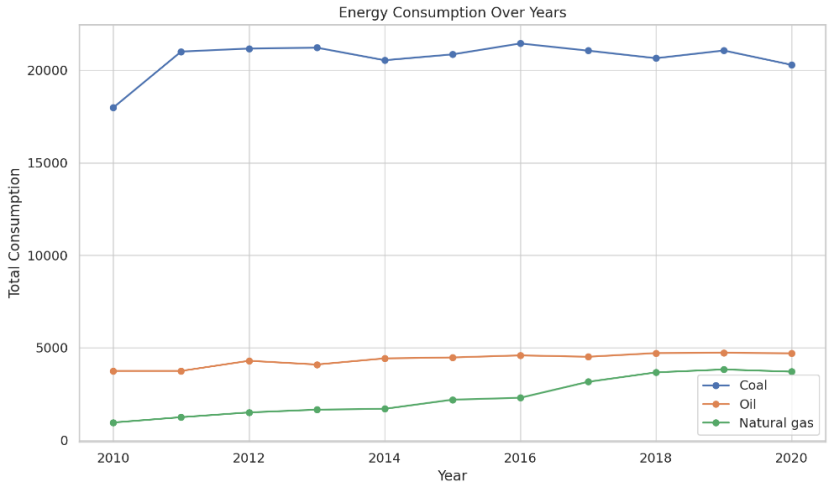

图 9 二次能源消费总量随时间变化图

下面进行碳排放量的分析.

根据数据碳排放表进行初步分析，做出图 10 与图 11:

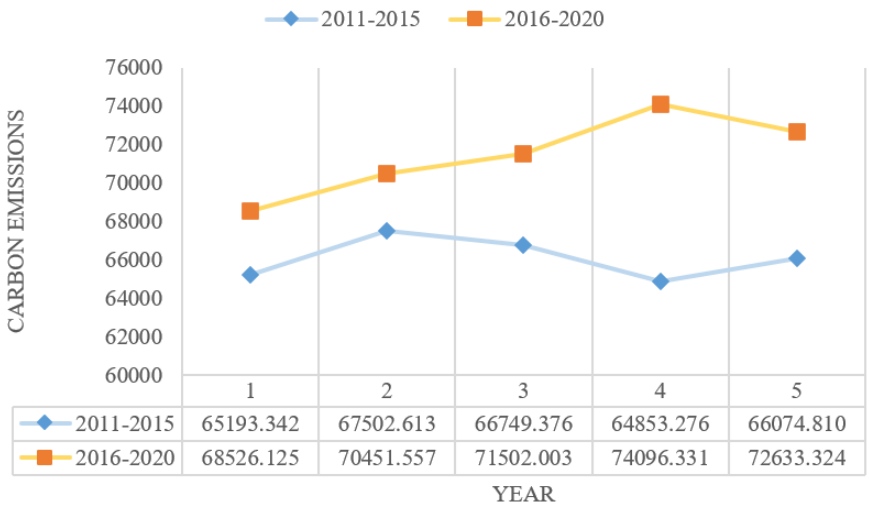

图 10 碳排放量随年份的变化图

由图 11 可以观察，碳排放总量在 2011 年到 2015 年期间呈现不断上升趋势，在 2014 年出现下降，在 2016 年到 2020 年，碳排放量继续上升，可以大致给出 2011-2020 年的碳排放量随时间的变化曲线，见式（12），其中 x 为年份:

$$z = -1795\sin(x-\pi) + 809.4\cos^3 x + 0.000083(x-10)^3 - 604600. \qquad (12)$$

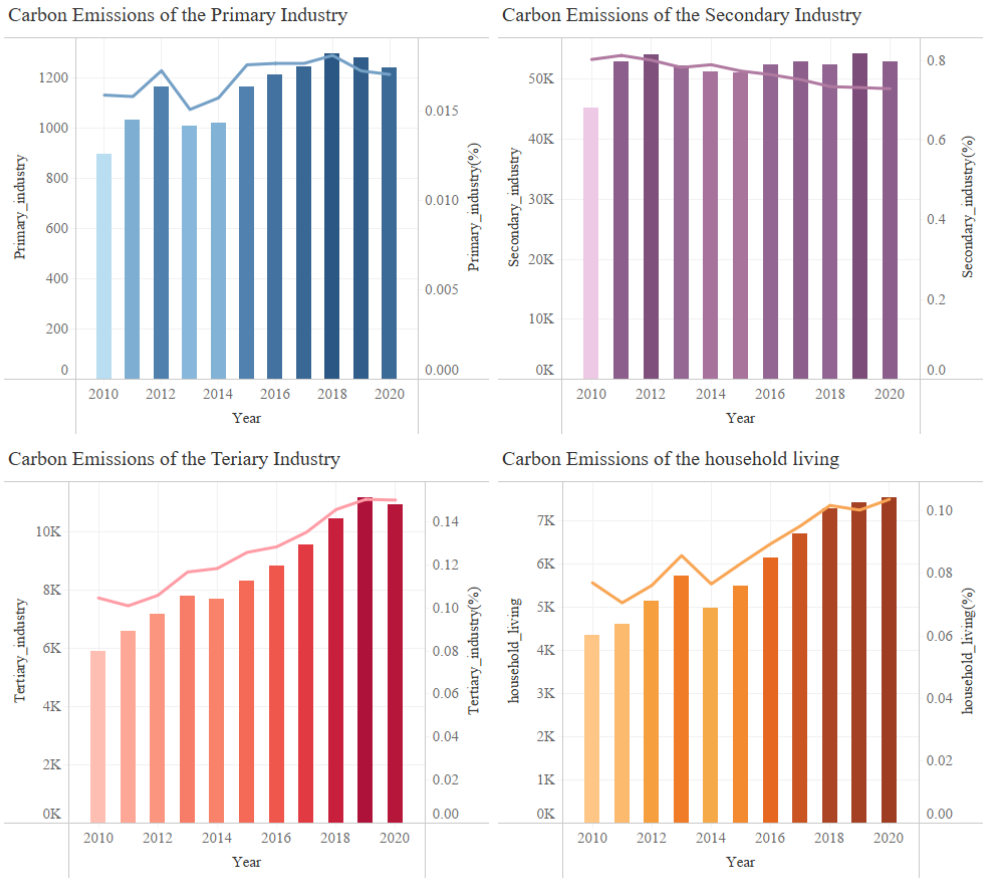

图 11 各产业的碳排放量与占比变化趋势

其中图 11 左上图为第一产业（农林消费）的碳排放量以及占比，右上图为第二产业（工业消费）碳排放量及其占比，左下图为第三产业（交通消费、建筑消费）碳排放量及其占比，右下图为居民生活消费的碳排放量图及其占比，得出结论:

（1）农林消费部门碳排放量呈现出平稳波动的趋势，工业消费部门碳排放量呈现稳定波动趋势，但在 2010 年和 2011 年之间出现了一次大幅上升；交通消费部门、建筑消费部门和居民生活消费部门的碳排放量都呈现出逐年上升的趋势.

（2）由占比可以发现，工业消费部门的碳排放量是所有部门中最高的，对总碳排放量的影响最大. 农林消费部门的碳排放量相对较低，其他部门的碳排放量都处于中等水平.

5.1.4 Anova 分析不同部门的碳排放状况

为了得到更为准确的结果，从实际问题出发，决定采用 Anova 分析来判断不同部门的碳排放量均值是否存在显著差异. Anova[4]基本原理如下:

可以证明，当若干样本都来自均值相同的正态总体时，则有（13）

$$\frac{S_A}{\sigma^2} \sim \chi^2(r-1),\ \frac{S_E}{\sigma^2} \sim \chi^2(n-r), \tag{13}$$

且 S_A 与 S_E 相互独立，于是有式（14）

$$F = \frac{MS_A}{MS_E} = \frac{S_A/(r-1)}{S_E/(n-r)} \sim F(r-1, n-r). \tag{14}$$

若由实验数据算得结果有 $F > F_\alpha(r-1, n-r)$，则拒绝 H_0，即认为因素 A 对试验结果有显著影响；若 $F < F_\alpha(r-1, n-r)$，则接受 H_0，即认为因素 A 对试验结果没有显著影响.

当然, 也可以通过检验的 p 值来决定是接受还是拒绝原假设.

如果取 $\alpha=0.01$ 时拒绝 H_0, 即 $F>F_{0.01}(r-1,n-r)$, 则称因素 A 的影响高度显著.

如果取 $\alpha=0.05$ 时拒绝 H_0, 但取 $\alpha=0.01$ 时不拒绝 H_0, 即式（15）

$$F_{0.01}(r-1,n-r)\geq F\geq F_{0.05}(r-1,n-r), \tag{15}$$

则称因素 A 的影响显著.

基于以上理论, 通过 python 程序运行, 可以得到图 12：

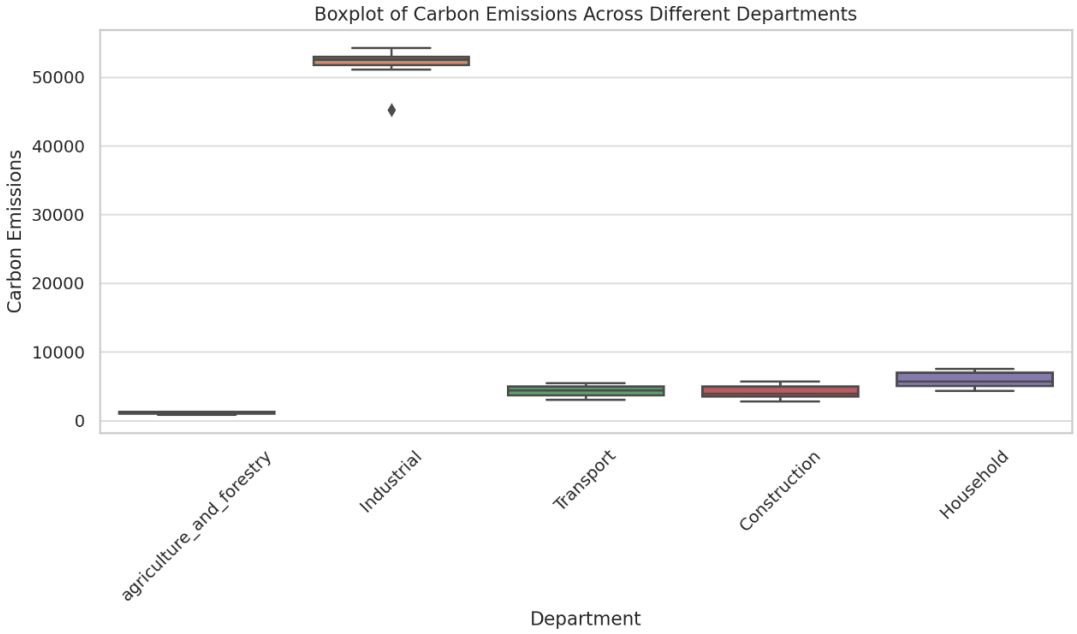

图 12 不同部门的碳排放量的显著性差异

得到运算结果 RESULT 为(2865.8826116422147, $2.2761104719487248\times10^{-58}$), 计算得出的 F 值为 2865.88. 这是一个相对较大的值, 表明组间差异显著大于组内差异, 计算得出的 p 值远小于 0.05（2.28×10^{-58}）, 可以拒绝零假设. 原假设是所有组的均值都相等.

由此得出结论: 不同部门的碳排放均值之间存在明显差异, 由箱线图 12 可知, 工业部门的碳排放量明显高于其他部门, 农林部门的排放量最低, 而交通、建筑、居民生活部门的碳排放量相对居中.

5.1.5 基于 XGBoost 的指标贡献度模型

为了研究该区碳排放量产生影响的各因素及其贡献, 考虑应用 XGBoost 算法, 并将特征变量作为输入, 使用 GridSearchCV 进行交叉验证和参数调优, 使用粒子群优化算法来进一步优化模型参数.

XGBoost（Extreme Gradient Boosting）即极致梯度提升算法, 陈天奇于 2016 年正式提出 [3], 是基于 GBDT（Gradient Boosting Decision Tree）的一种算法, 其思想和 GBDT 相同, 是 GBDT 的优化. XGBoost 基学习器包含树（gbtree）和分类器（gblinear）, 其目标函数由损失函数和正则化两部分组成.

通过二阶泰勒展开, 去除常数项, 对损失函数进行优化:

$$L(\varphi)=\sum_i l(y_i,\ y_i)+\sum_k \Omega(f_k), \tag{16}$$

再进行正则化项展开, 去除常数项, 优化正则化项;

$$L^{(t)} = \sum_{i=1}^{n}\left[g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega(f_t), \tag{17}$$

合并一次项, 二次项系数, 得到最终目标函数.

$$L^{(t)} = \sum_{j=1}^{T}\left[G_j w_j + \frac{1}{2}(H_j + \lambda) w_j^2 \right] + \gamma^T, \tag{18}$$

此过程使得准确度得到很好的提高并且不易过拟合, 可扩展性强. XGBoost 在实际训练 XGBoost 树中, 最佳分裂点的分裂方法有: Greedy Algorithm 算法、Approximate Algorithm 近似算法、Weighted Quantile Sketch 加权分位数草图法、Sparsity-aware Split Finding 稀疏感知法. XGBoost 的优点为提升模型的预测能力, 以“正则化提升”, 正则化降低了模型的方差, 使模型更加简单, 防止过拟合, XGBoost 在特征粒度使支持并行的, 每一个树都是由前一颗树迭代得来的.

1995 年, 受到鸟群觅食行为的规律性启发, James Kennedy 和 Russell Eberhart 建立了一个简化算法模型, 经过多年改进最终形成了粒子群优化算法（Particle Swarm Optimization, PSO） , 也可称为粒子群算法[5]. 其基本思想为鸟群通过共享信息找到食物最多的目的地即最优的目的地, 所有鸟在森林中随机搜索食物, 它们不知道食物的具体方向, 只能感知大概方向去依次寻找, 每只鸟都会记录曾找到食物最多的地方, 并向鸟群共享信息, 经过一段时间搜索, 找到食物最多的地方, 即得最优解（全局最优解）.

其中算法的核心要素为粒子的两个属性即位置（所求问题的一个解）与速度（下一步迭代时移动的方向和距离）.

$$v_i^d = w v_i^{d-1} + c_1 r_1 (pbest_i^d - x_i^d) + c_2 r_2 (gbest^d - x_i^d), \tag{19}$$

$$x_i^{d+1} = x_i^d + v_i^d, \tag{20}$$

其中, n 为粒子个数, c_1 粒子的个体学习因子, 也称个体加速因子, c_2 粒子的社会学习因子, 也称社会加速因子, w 速度的惯性权重, v_i^d 第 d 次迭代时, 第 i 个粒子的速度, x_i^d 第 d 次迭代时, 第 i 个粒子所在的位置, $f(x)$ 在位置 x 时的适应度值（一般取目标函数值）, $pbest_i^d$ 到第 d 次迭代为止, 第 i 个粒子经过的最好位置, $gbest^d$ 到第 d 次迭代为止, 所有粒子经过的最好的位置.

运用 XGBoost 模型, 将特征变量作为输入, 通过 GridSearchCV 方法进行交叉验证和参数调优, 并使用粒子群优化算法进一步优化模型参数, 每个特征的 SHAP 值, 代表了每个特征对模型预测的贡献, 通过观察 SHAP 图 13, 可知, 每个特征对模型输出的影响, 其中, y 轴上的位置由特征确定, x 轴上的位置由每个 Shapley value 确定. 颜色表示特征值（红色高, 蓝色低）, 颜色能使我们直观看到特征值的变化如何影响风险的变化, 即影响碳排放量的各因素贡献度变化如何影响碳排放量的变化. 可以了解到红点代表正值, 蓝点代表负值, 点的横坐标表示特征的贡献度.

由此可得, 位于图表顶部的特征对模型输出的贡献最大. Population（人口）特征对碳排放量的影响最大, 其次是 GDP（地区生产总值）、Per carbon emissions（人均碳排放量）、Total energy consumption（总能源消费量）, 其中 Agriculture and forestry（农林业）、Construction（建筑业）和 Household（居民生活）等特征对模型都有一定的影响. 每个特征的贡献都有正有负, 这表明这些特征都有增加和减少碳排放量的潜力.

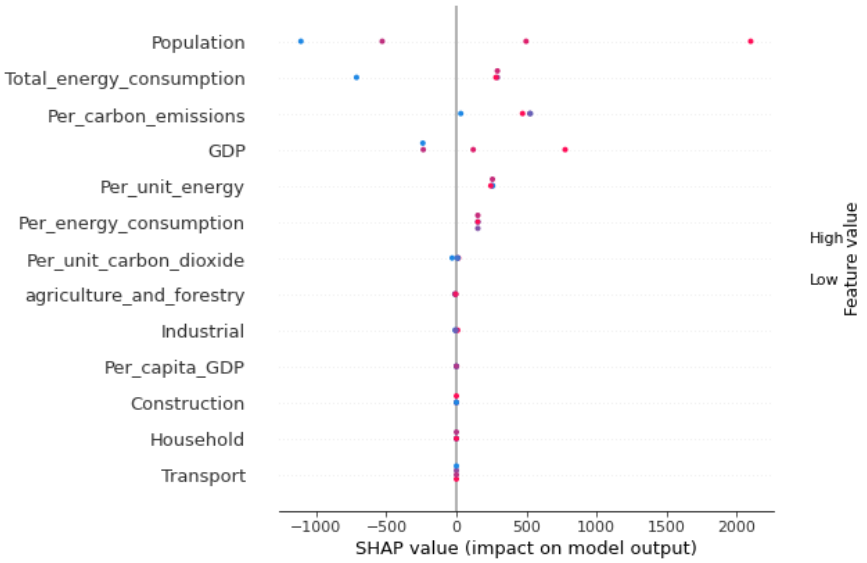

图 13 各特征的 SHAP 值

实现碳达峰与碳中和需要面对的主要挑战:

根据上述分析，得知该区域实现碳达峰与碳中和目标所面临的主要挑战需要全社会成员及各个部门做到积极响应与号召，有学者将碳减排主体大致分为三类[6]：政府类碳减排主体，包含农林、能源、交通、工业、建筑等职能管理部门；市场类碳减排主体，包括直接承担碳减排任务的企业，也包括间接助力碳减排的金融机构等；社会类碳减排主体，包括广大社会成员及社会组织. 实现碳达峰、碳中和目标是政府、市场和社会多元主体共同努力的结果. 学者齐利枝[7]指出碳达峰碳中和背景下环境管理的发展路径，应优化和调整产业结构，促进绿色产业发展、减少高碳行业的占比以实现经济可持续发展，同时政府应加大监管力度，推进技术改造和转型升级；加强排污检测监管能力建设，有效的排污有利于提高环境质量，加强监督企业和机构遵守环境法规和减排目标，确保监管工作的全面性、高效性和协调性；完善低碳规划建设体系建立，旨在减少温室气体排放，促进社会可持续发展；推动城市绿色低碳转型，鼓励公民绿色出行，如步行、骑行、乘坐公共交通工具出行，推广共享交通，减少交通尾气排放；加大碳达峰碳中和宣传力度，呼吁社会成员及各个部门自主参与环保行动，激发公众环保意识.

要想将碳排放量实现“净零”，需要更多的技术，尤其在重工业等难以脱碳的部门，大规模应用的低碳方案还未被解决. 我国还需加大对技术创新，产业创新的力度，提高能源利用率和非化石能源消费比重.

5.1.6 基于 PCA 的 Kaya 关联模型

考虑总碳排放量为因变量，其他与总碳排放量相关性较高的变量为自变量. 可以应用多元线性回归的 Kaya 模型来建立这些变量之间的关系.

在建立模型之前，需要考虑到变量之间可能存在的多重共线性问题. 我们可以使用 VIF（Variance Inflation Factor）来检查多重共线性. 我们将计算所有变量的方差膨胀因子（Variance Inflation Factor, VIF）来评估多重共线性. VIF 值大于 10 通常表示存在严重的多重共线性问题，我们可能需要考虑删除一些具有高 VIF 值的变量.

表 1 VIF 值

	Variable	IF
0	常驻人口	inf
1	生产总值 (GDP)	inf
2	总碳排放量	inf

3	人均能源消费量	inf
4	单位 GDP 能耗	inf
5	农林消费部门碳排放量	inf
6	工业消费部门碳排放量	inf
7	交通消费部门碳排放量	inf
8	建筑消费部门碳排放量	inf
9	居民生活消费碳排放量	inf
10	人均碳排放量	2.388915e+09
11	总能源消费量	2.034837e+09
12	单位能耗二氧化碳排放量	5.017091e+06
13	人均 GDP	8.192159e+05

观察表 1，许多变量的 VIF 值为无穷大，这明显表明存在严重的多重共线性问题. 此外，一些其他变量的 VIF 值也远远大于 10，表明这些变量之间也存在较高的共线性. 在这种情况下，我们需要采取一些策略来处理多重共线性，这里采用主成分分析（PCA）：使用主成分分析来减少变量的数量，同时保留大部分变量的信息. 其优点在于: PCA 将原始的、可能相关的变量转换为新的、彼此正交的变量，称为主成分. 这些主成分可以捕捉原始变量中的大部分信息，而且由于它们是正交的，因此不存在多重共线性问题.

我们通过计算相关系数矩阵来查看各个变量之间的相关性,有助于选择合适的变量进行路径分析. 可视化指标间的相关系数矩阵如图 14:

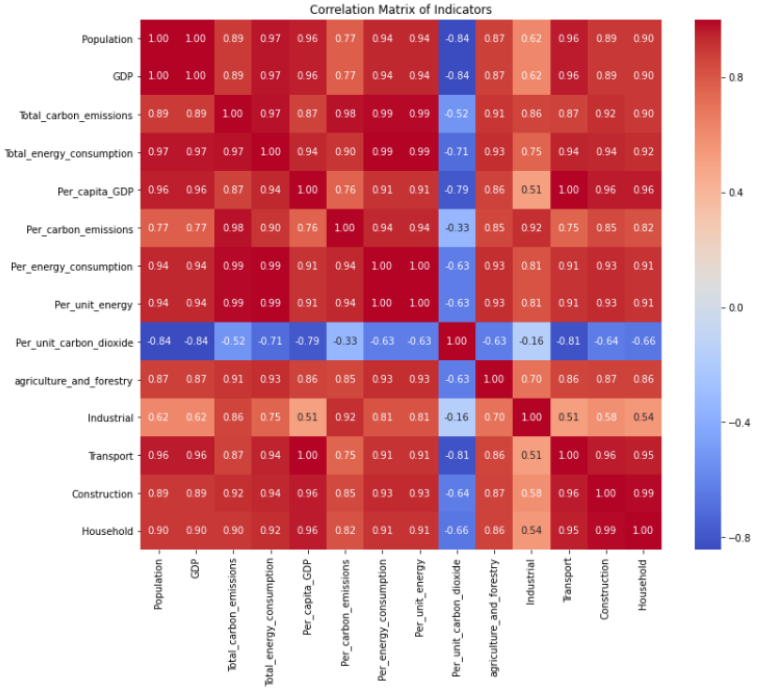

图 14 PCA 相关系数矩阵

可以观察到多个变量与总碳排放量具有较高的相关性，方便在进行路径分析时选择影响总碳排放量的因素. 由此根据这些相关性来选择一组合适的变量，建立基于主成分分析的多指标关联模型，具体方法如下:

（1）数据预处理: 我们需要对数据进行缩放，以便每个变量具有相同的尺度.

（2）应用 PCA[8]: 在缩放的数据上应用 PCA.

（3）选择主成分: 选择足够数量的主成分，以便捕捉原始数据中的大部分变异性.

（4）建立模型: 使用选择的主成分作为自变量建立多元线性回归模型.

程序运行结果见表 2:

表 2 PCA 运行结果

Principal Component		Explained Variance Ratio	Cumulative Explained Variance
0	1	8.655971e-01	0.865597
1	2	9.316533e-02	0.958762
2	3	2.759802e-02	0.986361
3	4	1.024147e-02	0.996602
4	5	1.968092e-03	0.998570
5	6	1.354738e-03	0.999925
6	7	6.063330e-05	0.999985
7	8	1.152392e-05	0.999997
8	9	2.993107e-06	1.000000
9	10	4.289879e-08	1.000000
10	11	2.407452e-33	1.000000

主成分分析（PCA）的结果显示, 第一个主成分能解释原始数据中大约 86.56%的方差, 前两个主成分能解释大约 95.88%的方差, 而前三个主成分能解释约 98.64%的方差.

通常, 我们会选择足够多的主成分, 以便捕获到原始数据中的大部分方差.

（1）提取主成分: 我们将使用前三个主成分作为自变量.

（2）建立回归模型: 我们将使用这些主成分来建立一个多元线性回归模型, 目标是预测总碳排放量.

（3）模型评估: 我们将评估模型的性能.

经过分析, 确定了保留两个主成分可以解释约 95.88% 的方差. 该线性回归模型的 R^2 值为 0.9985, 这表明模型能够非常好地拟合数据, 方法可行, 具体相关性见表 3:

表 3 各主成分对模型的解释度

	PC-1	PC-2
Population	-0.278080	-0.163736
GDP	-0.278080	-0.163736
Total_carbon_emissions	-0.278073	0.217118
Total_energy_consumption	-0.286557	0.026840
Per_capita_GDP	-0.274908	-0.210395
Per_carbon_emissions	-0.256712	0.387757
Per_energy_consumption	-0.284490	0.107799
Per_unit_energy	-0.284490	0.107799
Per_unit_carbon dioxide	0.206859	0.545735
agriculture and forestry	-0.267528	0.057960
Industrial	-0.207315	0.565865
Transport	-0.275468	-0.218701
Construction	-0.274598	-0.052255
Household		-0.271901

PC-1: 这一主成分主要与所有指标负相关, 特别是与 Total_energy_consumption、Per_energy_consumption 和 Per_unit_energy 更强烈地负相关, 可能意味着这一主成分主要代表着总能源消费量和单位能源消费量的变动.

PC-2: 这一主成分与 Per_unit_carbon_dioxide、Industrial 正相关, 与 Per_capita_GDP、Transport 负相关, 意味着这一主成分主要代表着单位能耗二氧化碳排放量和工业消费部门碳排放量的变动.

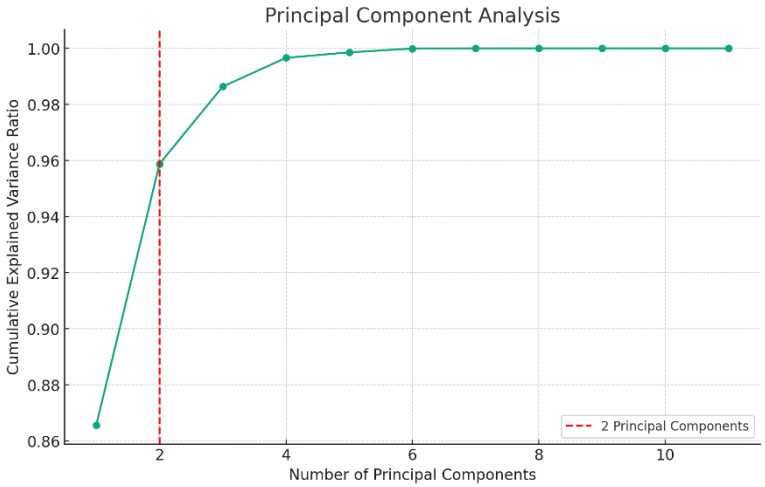

图 15 PCA 的主成分权重

图 15 中展现了保留的主成分捕获大数据的变动过程，同时列出了每个主成分在每个原始特征上的权重.

由此可以确定碳排放预测模型参数（能源利用效率提升和非化石能源消费比重）.

根据题目背景得知，要实现经济增长与能源消费量的负相关，以及能源消费量与碳排放量的负相关，就是要提高能源利用效率，提高非化石能源的消费比重，也就是降低单位 GDP 的能耗（总能耗/GDP），以及降低单位能耗碳排放量，从而实现碳达峰和碳中和.

非化石能源消费比重可以等同于非化石能源发电比重（非化石能源发电量与能源消费量的比值），等同于非化石能源发电比重 NF_e（非化石能源发电占比 nf_e *电力消费比重 EC_p），为了消费比重达到 80%，需要实现非化石能源发电比重和电力消费比重达 90%,见式（21）：

$$NF_e = nf_e \times EC_p \tag{21}$$

单位 GDP 能耗为总能耗与 GDP 的比值，见式（22）：

$$\varepsilon = \frac{\delta_1\varphi_1 + \delta_2\varphi_2 + \delta_3\varphi_3}{1-\sigma_f}. \tag{22}$$

Kaya 模型见式（23）：

$$\begin{aligned} C &= \Omega \times \mathrm{GDP}_p \times \varepsilon \times C_r = \Omega \times \frac{\mathrm{GDP}}{\Omega} \times \frac{\delta_1\varphi_1 + \delta_2\varphi_2 + \delta_3\varphi_3}{1-\sigma_f} \times \frac{C_a}{\Phi_T} \\ &= \mathrm{GDP} \times \frac{\delta_1\varphi_1 + \delta_2\varphi_2 + \delta_3\varphi_3}{1-\sigma_f} \times \frac{C_a}{\Phi_T}, \end{aligned} \tag{23}$$

其中 C 为二氧化碳排放量，Ω 为人口，GDP_p 为人均 GDP, GDP 为生产总值，

$$\varepsilon = \frac{\delta_1\varphi_1 + \delta_2\varphi_2 + \delta_3\varphi_3}{1-\sigma_f}$$

为单位 GDP 能耗，$C_r = \dfrac{C_a}{\Phi_T}$ 为单位能耗二氧化碳排放量.

《意见》中目标提升重点行业能源利用效率，单位 GDP 能耗比 2020 年下降 13.5%，到 2030 年重点能耗行业能源利用效率达到国际先进水平.

根据各指标间的正负相关性量化的数值，并结合非化石能源发电占比以及单位 GDP

能耗，将主成分分析得到的结果应用于 Kaya 模型，即用于预测二氧化碳排放量，可以确定预测模型参数非化石能源消费比重与能源利用效率取值分别为 0.4495 和 0.5289.

5.2 问题二的模型建立与求解

通过问题一的分析可知，人口、经济、能源消耗量的变化均对碳排放量产生一定的影响及贡献，而这些特征均具有时间上的动态性，因而可视为时间序列，可以采用基于 LSTM 的多变量多步时间序列预测模型来预测十四五至二十一五期间人口、经济（GDP）和能源消费量变化.

5.2.1 基于 LSTM 的多变量时间序列预测模型

在统计学研究中，按照时间顺序排列的一组随机变量来表示一个随机事件的时间序列，时间序列也称动态序列. 一组真实的数据，反应了某一现象发展变化状态，时间序列的背后是寻找该序列的某一现象的变化规律，寻求这一规律的过程即是时间序列分析.

LSTM(Long Short Term Memory network)长短期记忆神经网络，LSTM 是 RNN 模型的特殊类型，LSTM 可以记住长期信息，很好解决了长期依赖问题，RNN 模型的网格结构只有一个简单的 tanh 层，而 LSTM 却重复单元包含了 4 个交互的层，其包含细胞状态和门两个要素，细胞状态是不断向单元顶端输送的向量，做初级的线性变换，使其保持不变；门为可以让信息选择性通过的结构，其包含遗忘门、输入门和输出门三个门来保护和控制细胞状态.

下面利用时间序列 LSTM 模型对基于人口和经济变化的能源消费量进行预测，模型有两个 LSTM 层，每层有 50 个单元，并且有两个 Dense 层用于最终预测，总共有 33101 个可训练的参数. 训练数据集有 6 个样本，每个样本有 3 个时间步，每个时间步有 7 个特征，目标变量有 6 个值，每个值对应于一个样本. 使用递归多步预测策略预测 2021 年至 2060 年能源消耗量的值. 可得训练集 RMSE（Root Mean Square Error）为 286.7893，测试集的 RMSE 为 601.0229，这两个数值代表了模型在训练集和测试集上的性能，RMSE 是预测误差的平方根，通常情况下，值越小，模型的性能越好. 训练集上的 RMSE 较低，测试集上稍高，这是正常的，因为模型是在训练集的基础上训练的，因而所得结果符合预期.

如图 16 所示，蓝色曲线为 2010-2020 年能源消费量变化曲线，红色虚线代表基于 2010 年-2020 年数据的预测曲线，显然 2020 年到 2050 年呈缓慢上升趋势，2050 年到 2060 年上升趋势变弱，反映了 2060 年有望实现碳达峰目标. 同时该图也呈现了人口增长和经济发展对能源需求的影响.

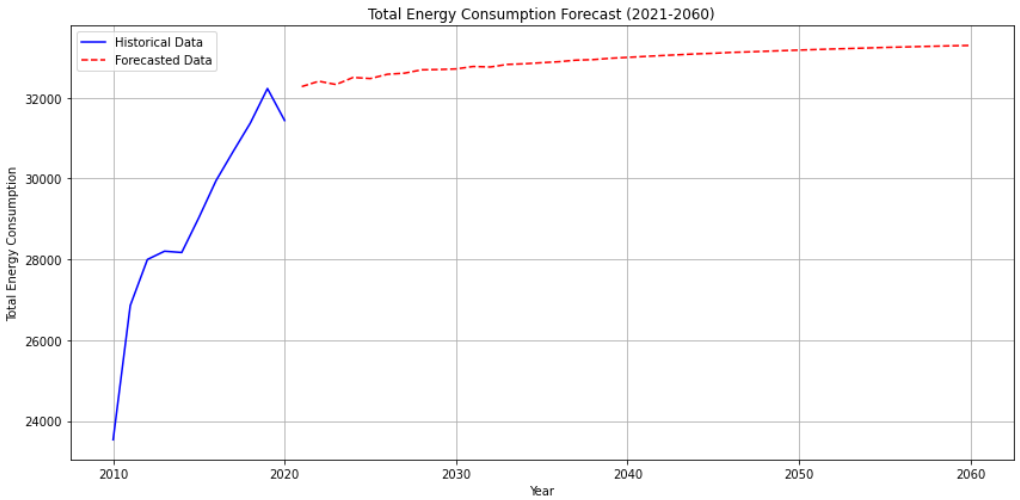

图 16 能源消费量变化及其预测曲线

部分能源消费量预测结果见表:

表 4 部分能源消费量预测结果

Year	2021	2022	2023	2024	2025	2026	2027	2028	…
Energy Consumption	32276	32408	32327	32502	32474	32581	32609	32691	…
Year	2052	2053	2054	2055	2056	2057	2058	2059	2060
Energy Consumption	33206	33218	33231	33244	33255	33266	33276	33285	33294

5.2.2 将碳排放量与人口、GDP、能源消费量预测相关联

利用 LSTM 神经网络时间序列预测算法, 将碳排放量与人口、GDP、能源消费量预测进行关联, 将 2010 到 2016 年的数据用于训练, 将 2017-2020 年的数据进行预测, 得到如下训练进度图 17:

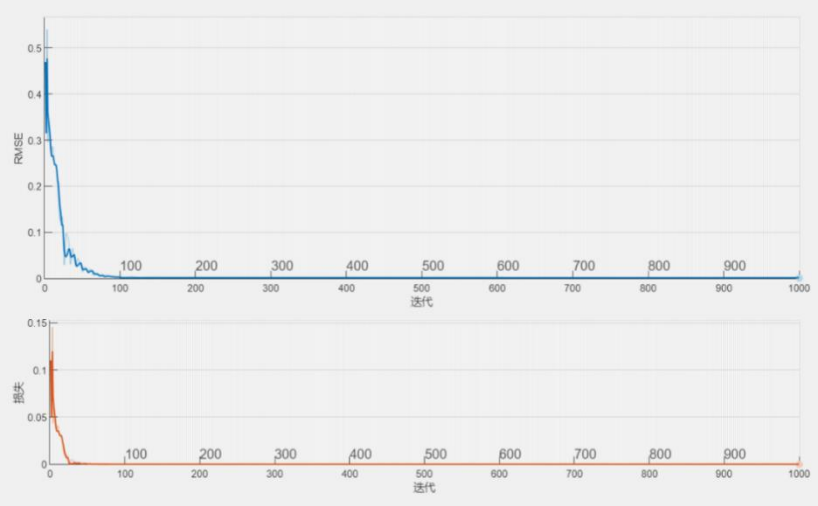

图 17 LSTM 时序预测的训练

碳排放量的部分预测结果见表 5（全部结果见附录）:

表 5 碳排放量部分预测结果

Year	2021	2022	2023	2024	2025	2026	2027	2028	2029	…
Carbon emissions	71704	71609	71839	72874	73744	74033	74237	74677	75134	…
Year	2051	2052	2053	2054	2055	2056	2057	2058	2059	2060
Carbon emissions	76882	76882	76882	76882	76882	76882	76883	76883	76883	76883

下图 18 为预测的 2017-2060 年的数据, 从表 5 与图 18 中可以发现, 碳排放量先逐渐增加, 在 2045 年开始后趋于平缓, 实现碳达峰, 碳排放量几乎保持稳定, 且数据训练较为良好, 结果具有参考性.

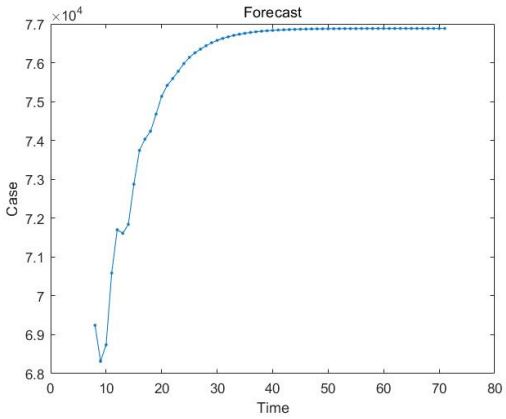

图 18　2021-2060 碳排放量预测

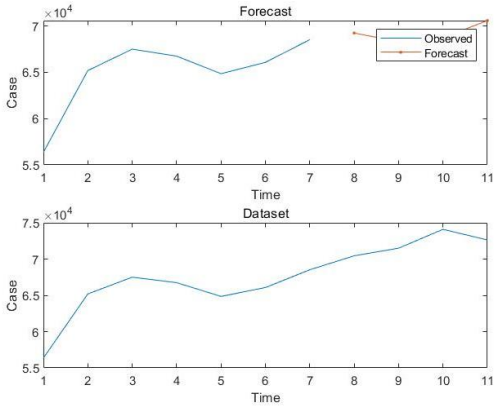

图 19 碳排放量预测与实际数据对比

图 19 为预测数据与实际数据的对比，具体数据对比见表 6，可以发现预测准确性高，算法可行性高.

表 6　碳排放量的实际值与预测值对比

Year	2017	2018	2019	2020
Dataset	70451.557	71502.003	74096.331	72633.324
Forecast	69244	70317	72739	73186

5.2.3　将碳排放量与各能源消费部门与能源供应部门的能源消费量关联

为了将碳排放量与各能源消费部门以及能源供应部门的消费量相关联，将指标进行精简，考虑利用皮尔逊相关系数，提取出具有关键作用的指标，运行程序得到的结果如下表 7 与表 8:

表 7　能源消费部门的 Pearson 系数

Index	Agricultural	Industry	Traffic	Construction	Household
Pearson	0.91728	0.87287	0.88462	0.92656	0.90808
Order	2	5	4	1	3

表 8　能源供应部门的 Pearson 系数

Index	Electricity	Heating	Other	Loss
Pearson	0.91071	0.83056	-0.67909	-0.61805
Order	1	2	3	4

其中 Index 为能源的各项指标，包括能源消费部门中的 Agricultural 农林消费部门、Industry 工业消费部门、Traffic 交通消费部门、Construction 建筑消费部门、Household 居民生活消费，能源供应部门的 Electricity 发电，Heating 供热，Other Loss 其他转换，损失，Pearson 为 Pearson 系数，Order 为系数排序，因此选取能源消费部门中的建筑消费部门，能源供应部门中的发电加入 LSTM 的时序模型中.

增加 Construction 与 Electricity 指标后的碳排放量的模型运行结果如下表 9:

表 9　碳排放量的部分预测值

Year	2021	2022	2023	2024	2025	2026	2027	2028	2029	…
Carbon emissions	71719	71621	71834	72852	73706	73963	74142	74581	75038	…
Year	2051	2052	2053	2054	2055	2056	2057	2058	2059	2060
Carbon emissions	76748	76748	76748	76748	76748	76748	76748	76749	76749	76749

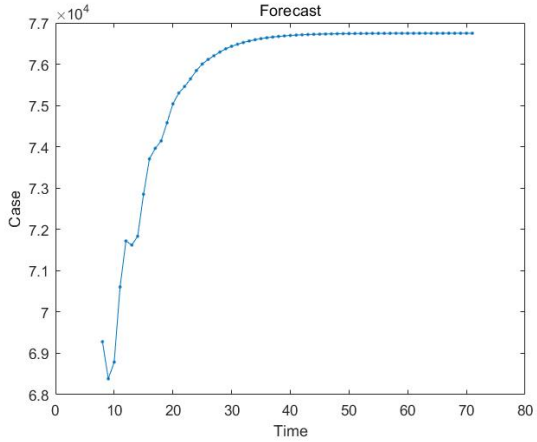

图 20　2021-2060 碳排放量二次预测

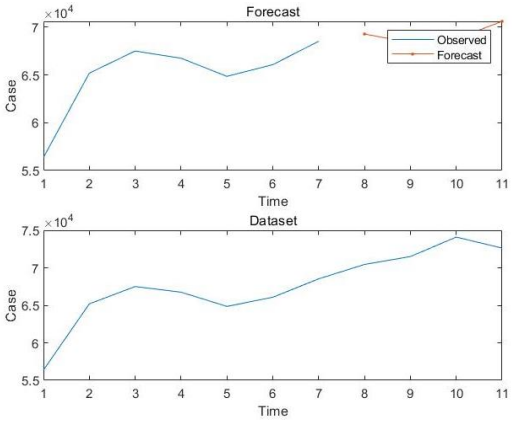

图 21 碳排放量预测与实际数据二次对比

表 10 碳排放量二次预测值与实际值的对比

Year	2017	2018	2019	2020
Dataset	70451.557	71502.003	74096.331	72633.324
Forecast	69282	71317	73401	71422

根据表 10 和图 20, 图 21 呈现的结果, 能效提升对能源消费部门的工业消费部门仍然消费高, 即占比高, 则新的模型更具有参考性.

5.2.4 将碳排放量与各能源消费部门、能源供应部门的能源消费品种关联

已知 Kaya 模型见式（23）:

$$C=\Omega\times \mathrm{GDP}_p\times\varepsilon\times C_r=\Omega\times\frac{\mathrm{GDP}}{\Omega}\times\frac{\delta_1\varphi_1+\delta_2\varphi_2+\delta_3\varphi_3}{1-\sigma_f}\times\frac{C_a}{\Phi_T},\qquad(23)$$

其中C为二氧化碳的排放量，Ω为人口，ε为单位 GDP 能耗，C_r为单位能耗二氧化碳排放量, 对上式取对数对时间求偏导数, 有式（24）:

$$\delta C=\delta\Omega+\delta\mathrm{GDP}_p+\delta\varepsilon+\delta C_r,\qquad(24)$$

其中, δX 是参数 X 对某基准年的相对变化率.

碳达峰是指碳排放量不再增长, 即$\delta C=0$, 碳中和是指碳排放量与碳汇消纳量平衡, 其要求$\delta C<<0$.

为了将碳排放量与各能源消费部门以及能源供应部门的能源消费品种相关联,由于建筑消费部门与碳排放量的 Pearson 系数最高, 因此以建筑消费部门为例, 做出碳排放量, 以及煤炭、油品、天然气、热力、电力的碳排放因子变化图 22 如下:

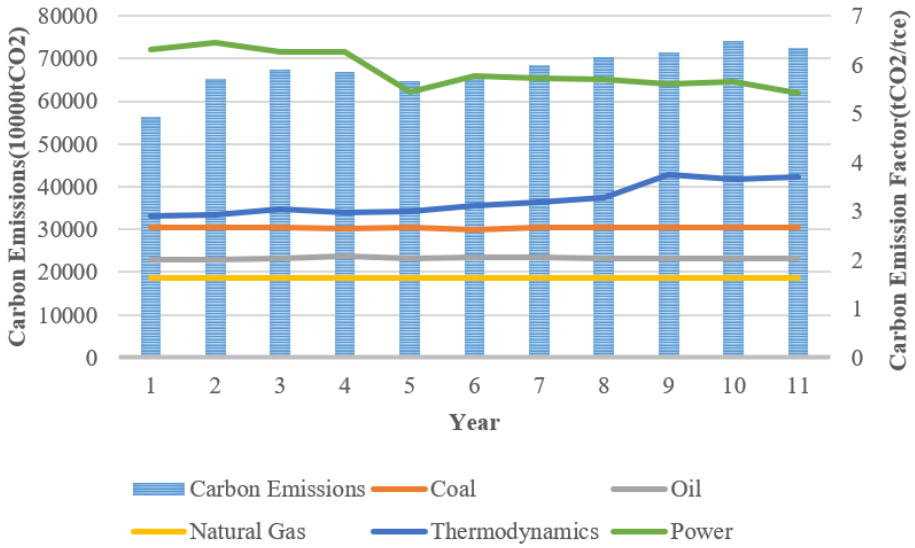

图 22 碳排放量与碳排放因子

利用 SPSS 对数据处理补充缺失值，错误值之后，由图可以看出，随着碳排放量的逐年变化，碳排放因子也随之变化，初步看出煤炭、油品、天然气等化石能源的碳排放因子呈现下降趋势，而电力的碳排放因子一直高于其他碳排放因子，且呈现降低的趋势，热力的碳排放因子呈现上升的趋势.

5.3 问题三的模型建立与求解

5.3.1 情景设计

（1）自然情景（Natural Scenario）

特点：体现社会公众及各部门的自主性

政策维持：这个情景假设政府不采取额外的政策干预，维持现有的环境保护和碳排放政策.

技术发展速度一致：技术进步和更新按照当前的速度和方向发展，没有额外的加速.

无新政策：不会出台新的碳排放减少政策或者推广可再生能源的政策.

潜在影响：可能会出现市场失灵，政府等各个部门得不到有效的监管.

碳排放持续：可能会看到碳排放的持续增加或在高水平维持，对环境和气候变化产生更多的负面影响.

可持续发展困难：长期来看，缺乏足够的可持续发展政策和措施，可能导致资源的过度开发和消耗，影响区域甚至全球的可持续发展.

（2）基准情景（Baseline Scenario）

特点：符合自然规律以及社会规律

政策目标明确：政府设定明确的碳达峰和碳中和目标，并采取一系列政策来实现这些目标.

技术和能源结构变革：推广能效高、碳排放低的技术，提高非化石能源的消费比重.

政策措施：政府加大监管力度，加强排污检测监管能力建设，完善低碳规划建设体系.

提高能效：推广能效高的产品和服务，提升能源利用效率.

非化石能源推广：通过政策引导和支持，推广太阳能、风能等可再生能源的应用.

潜在影响：社会环境质量变优，居民生活质量提高，社会经济可持续发展.

碳排放减少：可以预期碳排放会按照政府的目标逐渐减少，有助于减缓气候变化.

可再生能源发展：非化石能源的比重提高将推动相关产业的发展，带动经济转型.

（3）雄心情景（Ambitious Scenario）

特点：提前达到目标

政策目标更为严格：在基准情景的基础上，设定更为严格和远大的目标，力图在更短的时间内实现碳达峰和碳中和.

技术创新加速：需要更快的技术创新速度和更高的技术更新换代率，来满足更为严格的碳排放标准.

政策措施：建立健全早日实现碳达峰碳中和目标的计划，加强监管，对环保企业提出更高要求，积极号召各个部门全力配合.

更大力度的政策支持：提供更多的政策和财政支持，来推动低碳技术的研发和应用.

更快的能源结构调整：更快地减少化石能源的比重，更快地提高可再生能源的比重.

潜在影响：人民向往的美好生活水平

快速碳排放减少：在这一情景下，碳排放的减少速度将大大加快，有助于更早地实现全球碳中和目标.

产业结构快速调整：推动快速的产业结构和能源结构调整，可能带来巨大的经济和社会变革.

（4）这三个情景分别代表了不同程度的政策干预和社会变革. 通过比较这三个情景下的碳排放量，能源结构，以及其他相关指标，可以深入了解不同政策和措施对未来可持续发展的影响，为政府和社会提供更为全面和详尽的参考信息.

5.3.2 多情景下碳排放量核算方法

经济增长与能源需求：可知 2010 年-2020 年随着生产总值 GDP 的快速增长，该地区的能源需求也在稳速增长.

碳排放方程：根据能源需求和能源结构（化石能源和可再生能源比例）来估算碳排放

$$GHG=AD\times EF \tag{25}$$

其中 GHG 为温室气体排放量, AD 为活动数据即各种化石能源的消耗量, EF 为与活动水平数据相对应的系数，包括单位热值含碳量或元素碳含量、氧化率等.

常见化石能源为煤炭、油品、天然气和电力等，非化石能源为新能源热力、新能源电力、外地调入电以及其他能源，经调查研究可知 2010 年-2020 年中国东南沿海地区化石能源比例及非化石能源比例变化趋势如图所示，显然化石能源占比在 11 年间的社会发展中是呈递减趋势的，而非化石能源及可再生能源是呈递增趋势的，可侧面反映出我国对碳排放量问题的高度重视，并采取的有效碳减排措施，且达到一定的成效.

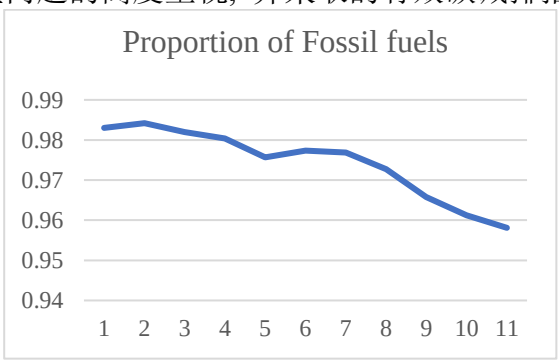

图 23 化石能源比例

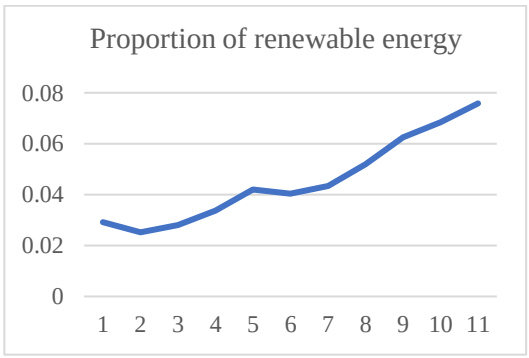

图 24 可再生能源占比

碳汇是指消纳二氧化碳的能力，碳汇包含生态碳汇和工程碳汇两种类型.

陆地生态系统碳汇为陆地生态系统对二氧化碳的吸收量大于排放量，那么此时的陆地

生态系统为二氧化碳的汇，反之，若生态系统不加以干预，陆地生态系统碳汇等于净生态系统生产力，即植物光合作用固定的总碳量减去生态系统呼吸消耗后的剩余部分，此时称之为碳源. 碳汇是植物光合作用将二氧化碳吸收和生态系统呼吸的二氧化碳排放过程共同作用的结果. 因此，碳汇功能并不是陆地生态系统的固有属性. 要实现生态碳汇倍增，需要统筹国土空间，发挥森林、草原、湿地、滨海固碳作用. 工程碳汇是指通过二氧化碳捕集、利用与存储（CCUS）等工程手段形成的碳汇.

碳中和是指碳排放量与碳汇（生态碳汇+工程碳汇+碳交易）消纳量相平衡.

确定双碳目标与路径

实现碳达峰、碳中和的关键是能源低碳转型，关乎我国社会发展全局，刘晓龙等[7]综合分析我国现状，提出我国碳达峰碳中和目标的实现应从节能增效和“三废”协同治理加快能源转型和技术创新，提高能源利用率，以及一大隐形途径，即思想观念的创新. 气候变暖是全球共同面临的挑战，需要我们共同去迎接，去应对，坚持命运共同体，必须加强国际间的合作.

六、模型分析与评价

6.1 灵敏度分析

为了保证结果的可行性，选择了 matlab 工具箱中的四个参数 SSE,R–square, Adjusted R-square, RMSE.

SSE 这个统计量测量的是拟合值域实际值的总偏差和，它也被称为残差的平方求和，式子见（25）

$$SSE=\sum_{i=1}^{n}w_i(y_i-y_i)^2 \tag{26}$$

SSE 值越小，拟合程度越好.

R - square 由三个公式计算得来，衡量了拟合在解释数据变化方面的成功程度.

$$SSR=\sum_{i=1}^{n}w_i(y_i-y)^2 \tag{27}$$

$$SST=\sum_{i=1}^{n}w_i(y_i-\overline{y_i})^2 \tag{28}$$

$$R-square=\frac{SSR}{SST}=1-\frac{SSE}{SST} \tag{29}$$

越接近 1，表示模型在方差中所占的比例更大.

Adjusted R - square 调整后的 $R-$ 平方统计量可以接受任何小于或等于 1 的值，而接近 1 的值表示更好的拟合.当模型包含无助于预测响应的项时，可能会出现负值.

RMSE 由以下两个公式计算得出:

$$MSE=\frac{SSE}{n} \tag{30}$$

$$RMSE=s=\sqrt{MSE} \tag{31}$$

与 SSE 和 RMSE 一样, RMSE 值越小，拟合程度越小.

表 11　曲线拟合误差分析

函数	SSE	R-Square	Adjusted R-square	RMSE
y_1	23728	0.9989	0.9944	34.4444
y_2	168020	0.9948	0.9935	458.28

y_3	14608	0.9986	0.9982	427.321
z	42660	0.9550	0.9325	843.203

由参数 R-square 和 Adjusted R-square 以及 RMSE 的取值，可知第一、二、三产业的生产总值以及碳排放量曲线拟合误差在允许范围内，具有参考性

6.2 模型优缺点分析

6.2.1 模型优点

（1）Anova 可以考察多个因素之间的关联，不受统计组数的限制，可对样本较多统计数量进行多重分析，是一种定量分析方法，有较强的可比性.

（2）XGBoost 具有较高的准确性、鲁棒性、可解释性、可扩展性，其是基于决策树的集成学习方法，可用于分类、回归和排序等任务. 采用了泰勒展开近似优化以及正则化项来控制模型的复杂度，对于大规模数据集有较高的效率.

（3）LSTM 一定程度上缓解了 RNN 长期依赖问题，具备长时记忆的能力，可以消解决部分梯度消失的问题，实现简单.

6.2.2 模型缺点

（1）拟合数据由于基数大，有一定的拟合障碍.

（2）XGBoost 参数较多，需要花费一定的时间金鼎参数调优；内存消耗较快；容易过拟合.

（3）LSTM[9]模型结构相对复杂，训练模型时比较耗时，不利于并行化.

（4）Anova 对于组数较多的数据集，计算复杂，前提条件要求严格，需对数据进行方差齐性检验.

七、参考文献

[1]张水根. 绿色技术创新、经济增长对碳排放的影响研究[D].江西财经大学,2023.

[2]林楚. 2025 年我国非化石能源发电量比重将达到 39%左右[N]. 机电商报,2022-03-28(A06).DOI:10.28408/n.cnki.njdsb.2022.000113.

[3] Chen, T., Guestrin, C. XGBoost: A Scalable Tree Boosting System[A]//Ithaca: ACM, 2016: 785-794.

[4] Goran R, Miroslav D, Branislav D, et al. Research on the Effect of Load and Rotation Speed on Resistance to Combined Wear of Stainless Steels Using ANOVA Analysis.[J]. Materials (Basel, Switzerland),2023,16(12).

[5]Kennedy J. Particle swarm optimization[J]. Proc. of 1995 IEEE Int. Conf. Neural Networks, (Perth, Australia), Nov. 27-Dec. 2011, 4(8):1942-1948 vol.4.

[6]宋国恺.中国落实碳达峰、碳中和目标的行动主体及实现措施[J].城市与环境研究,2021(04):47-60.

[7]齐利枝.碳达峰碳中和背景下环境管理的现状及发展路径探究[J].黑龙江环境通报,2023,36(04):103-105.

[8] Rongge X,Qi Z,Shuaishuai J, et al. Evaluation of influencing factors of pipeline wax deposition strength based on principal component analysis[J]. Petroleum Science and Technology,2023,41(6).

[9] 刘天阳.基于注意力机制的 CNN-LSTM 模型股价趋势预测[J].科技资讯,2022,20(23):1-5.DOI:10.16661/j.cnki.1672-3791.2205-5042-2715.

附录

1.1	区域碳排放量以及经济、人口、能源消费的现状分析
1.2	Anova 分析不同部门之间碳排放量差异
1.3	基于 XGBoost 的指标贡献度模型
1.4	主成分分析, 基于 PCA 的 Kaya 关联模型
2.1	基于 LSTM 的多变量时间序列预测模型

问题一的附录

1.1 区域碳排放量以及经济、人口、能源消费的现状分析

```
# Provided data for stacked bar plot
data = {
    'Coal': [11319.33636, 2333.909091, -1141.933636],
    'Oil': [4.785454545, 5.773636364, 182.5763636],
    'Natural gas': [4564.517045, 4581.819176, 4595.046573]
}
rows = ['Power', 'Heating', 'Other']

# Creating DataFrame from the provided data
stacked_data_df = pd.DataFrame(data, index=rows)

# Plotting stacked bar plot
plt.figure(figsize=(10,6))
stacked_data_df.plot(kind='bar', stacked=True, figsize=(10,6), colormap='viridis')
plt.title('Stacked Bar Plot of Energy Consumption')
plt.xlabel('Energy Types')
plt.ylabel('Values')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

1.2 Anova 分析不同部门之间碳排放量差异

```
from scipy import stats
import seaborn as sns

# Performing ANOVA analysis
fvalue, pvalue = stats.f_oneway(
    carbon_emission_data_df['agriculture_and_forestry'],
    carbon_emission_data_df['Industrial'],
    carbon_emission_data_df['Transport'],
    carbon_emission_data_df['Construction'],
    carbon_emission_data_df['Household']
)

# Preparing data for boxplot
melted_df = carbon_emission_data_df.melt(id_vars='Year', value_vars=['agriculture_and_forestry', 'Industrial', 'Transport', 'Construction', 'Household'])
```

```
# Plotting boxplot for visualizing the distribution of carbon emissions across different departments
plt.figure(figsize=(10,6))
sns.boxplot(x='variable', y='value', data=melted_df)
plt.title('Boxplot of Carbon Emissions Across Different Departments')
plt.xlabel('Department')
plt.ylabel('Carbon Emissions')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()

fvalue, pvalue
```

1.3 基于 XGBoost 的指标贡献度模型

```
# 划分训练集和测试集
import pandas as pd
from sklearn.model_selection import train_test_split

# Re-load the data
data = {
    "Year": [2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020],
    "Population": [7869.34, 8022.99, 8119.81, 8192.44, 8281.09, 8315.11, 8381.47, 8423.50, 8446.19, 8469.09, 8477.26],
    "GDP": [7869.34, 8022.99, 8119.81, 8192.44, 8281.09, 8315.11, 8381.47, 8423.50, 8446.19, 8469.09, 8477.26],
    "Total_carbon_emissions": [56360.05, 65193.34, 67502.61337, 66749.3757, 64853.28, 66074.81, 68526.12, 70451.55739, 71502.00286, 74096.33, 72633.32],
    "Total_energy_consumption": [23539.3144313, 26860.025811662, 27999.218108463, 28203.104274887, 28170.505764677, 29033.608068382, 29947.976618214, 30669.886456736, 31373.126649144, 32227.505385374, 31437.997554448],
    "Per_capita_GDP": [5.258874315, 5.727621498, 6.239086875, 6.784317004, 7.288826712, 7.883479593, 8.431182934, 8.992960348, 9.569724566, 10.10216374, 10.46130644],
    "Per_carbon_emissions": [7.16, 8.13, 8.31, 8.15, 7.83, 7.95, 8.18, 8.36, 8.47, 8.75, 8.57],
    "Per_energy_consumption": [2.99, 3.35, 3.45, 3.44, 3.40, 3.49, 3.57, 3.64, 3.71, 3.81, 3.71],
    "Per_unit_energy": [2.99, 3.35, 3.45, 3.44, 3.40, 3.49, 3.57, 3.64, 3.71, 3.81, 3.71],
    "Per_unit_carbon_dioxide": [2.39, 2.43, 2.41, 2.37, 2.30, 2.28, 2.29, 2.30, 2.28, 2.30, 2.31],
    "agriculture_and_forestry": [896.07, 1031.18, 1165.275005, 1007.477719, 1020.85, 1162.51, 1211.03, 1245.019174, 1295.487242, 1278.38, 1238.76],
    "Industrial": [45225.70, 52975.79, 54048.28159, 52229.08417, 51187.98, 51101.87, 52382.22, 52975.84927, 52506.88057, 54235.44, 52954.05],
    "Transport": [3068.03, 3280.29, 3561.729529, 3847.155716, 4157.35, 4398.07, 4556.29, 4826.105744, 5125.16509, 5449.65, 5456.84],
    "Construction": [2830.25, 3304.26, 3586.044942, 3944.316621, 3518.72, 3916.46, 4244.72, 4697.909213, 5296.948286, 5701.31, 5449.19],
    "Household": [4340.00, 4601.82, 5141.282313, 5721.34147, 4968.37, 5495.89, 6131.85, 6706.673989, 7277.521668, 7431.55, 7534.49]
}
```

```
carbon_emission_data_df = pd.DataFrame(data)

# Define the feature variables and the target variable
X = carbon_emission_data_df.drop(columns=['Year', 'Total_carbon_emissions'])
y = carbon_emission_data_df['Total_carbon_emissions']

# Split the data into training and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Checking the shapes of training and test sets
X_train.shape, X_test.shape, y_train.shape, y

import xgboost as xgb
from sklearn.model_selection import GridSearchCV

# Define the XGBoost regressor model
xg_reg = xgb.XGBRegressor(objective ='reg:squarederror', seed=42)

# Define the parameter grid
param_grid = {
    'n_estimators': [10, 50, 100, 200],
    'learning_rate': [0.01, 0.1, 0.2, 0.3],
    'max_depth': [3, 4, 5, 6],
    'colsample_bytree': [0.7, 0.8, 0.9, 1],
    'subsample': [0.7, 0.8, 0.9, 1]
}

# Set up GridSearchCV
grid_clf = GridSearchCV(xg_reg, param_grid, scoring='neg_mean_squared_error', cv=3,
n_jobs=-1)

# Fit the model
grid_clf.fit(X_train, y_train)

# Get the best parameters
best_params = grid_clf.best_params_
best_params

# Re-defining the best parameters found and re-initializing the model
best_params_reduced = {
    'n_estimators': 200,
    'learning_rate': 0.2,
    'max_depth': 5,
    'colsample_bytree': 0.7,
    'subsample': 0.8
}

# Re-initializing the optimal XGBoost model with the best parameters found
optimal_xg_reg = xgb.XGBRegressor(
```

```
    objective ='reg:squarederror',
    seed=42,
    n_estimators=best_params_reduced['n_estimators'],
    learning_rate=best_params_reduced['learning_rate'],
    max_depth=best_params_reduced['max_depth'],
    colsample_bytree=best_params_reduced['colsample_bytree'],
    subsample=best_params_reduced['subsample']
)

# Fit the optimal model on the training data
optimal_xg_reg.fit(X_train, y_train)

# Using a subset of the training data to calculate SHAP values to avoid memory error
subset_size = int(0.5 * len(X_train))   # Using 50% of the training data
X_train_subset = X_train.sample(subset_size, random_state=42)

# Using TreeExplainer instead of Explainer to avoid AttributeError
explainer = shap.TreeExplainer(optimal_xg_reg)
shap_values_subset = explainer.shap_values(X_train_subset)

# Re-plot SHAP summary plot using TreeExplainer and subset of the training data
shap.summary_plot(shap_values_subset, X_train_subset)
```

1.4 主成分分析，基于 PCA 的 Kaya 关联模型

```
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler

# Standardize the data (important for PCA)
scaler = StandardScaler()
scaled_data = scaler.fit_transform(numeric_data)

# Apply PCA
pca = PCA()
principal_components = pca.fit_transform(scaled_data)

# Get the explained variance ratio of the principal components
explained_variance_ratio = pca.explained_variance_ratio_

# Calculate cumulative explained variance
cumulative_explained_variance = np.cumsum(explained_variance_ratio)

# Display the explained variance ratio and cumulative explained variance
explained_variance_df = pd.DataFrame({
    'Principal Component': range(1, len(explained_variance_ratio) + 1),
    'Explained Variance Ratio': explained_variance_ratio,
    'Cumulative Explained Variance': cumulative_explained_variance
})

explained_variance_df
# Select numeric data using the correct column name 'Year'
```

```
numeric_data_reloaded = indicator_data_reloaded.select_dtypes(include=[np.number]).drop(columns=['Year'])

# Standardize the data again
scaled_data_reloaded = scaler.fit_transform(numeric_data_reloaded)

# Apply PCA again
principal_components_reloaded = pca.fit_transform(scaled_data_reloaded)

# Extract the first three principal components again
X_pca_reloaded = principal_components_reloaded[:, :n_components]

# Define the target variable (Total Carbon Emissions) again
y_reloaded = numeric_data_reloaded['Total_carbon_emissions'].values

# Initialize the Linear Regression Model and fit the model again
linear_reg_model_reloaded = LinearRegression()
linear_reg_model_reloaded.fit(X_pca_reloaded, y_reloaded)

# Predict the target variable and Evaluate the model again
y_pred_reloaded = linear_reg_model_reloaded.predict(X_pca_reloaded)
mse_reloaded = mean_squared_error(y_reloaded, y_pred_reloaded)
r2_reloaded = r2_score(y_reloaded, y_pred_reloaded)

# Return the model evaluation metrics again
mse_reloaded, r2_reloaded

import matplotlib.pyplot as plt

# Plotting Cumulative Variance
plt.figure(figsize=(10,6))
plt.plot(range(1, len(cumulative_variance) + 1), cumulative_variance, marker='o')
plt.title('Principal Component Analysis')
plt.xlabel('Number of Principal Components')
plt.ylabel('Cumulative Explained Variance Ratio')
plt.grid(True)
plt.axvline(x=n_components, color='r', linestyle='--', label=f'{n_components} Principal Components')
plt.legend(loc='lower right')
plt.show()

# Display Principal Component weights for each feature
pc_weights = pd.DataFrame(pca.components_[:n_components], columns=numeric_data_reloaded.columns, index=[f'PC-{i}' for i in range(1, n_components + 1)])
pc_weights.T
```

问题二的附录

2.1 基于 LSTM 的多变量时间序列预测模型

```
from sklearn.preprocessing import MinMaxScaler
```

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, LSTM, Dropout
import numpy as np
import pandas as pd

# Create the DataFrame
data = {
    'Year': [2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020],
    'Resident population': [7869.3, 8023, 8119.8, 8192.4, 8281.1, 8315.1, 8381.5, 8423.5, 8446.2, 8469.1, 8477.3],
    'Agriculture and forestry': [2409.2, 2736.9, 3057.8, 3228.5, 3358.6, 3636.1, 3690.6, 3568.5, 3591.6, 3726.6, 3916.8],
    'Energy supply': [904.65, 947.43, 1121.2, 1065.5, 1149.8, 1357.6, 1417.9, 1527, 1604.6, 1692.7, 1660.7],
    'Industrial consumption': [20949, 22793, 24492, 26233, 27758, 29343, 30595, 32987, 34929, 36037, 36523],
    'Transportation consumption': [1767.2, 1988.4, 2199.5, 2233.9, 2378.9, 2240.4, 2316.4, 2420.2, 2570.7, 2749.1, 2761.5],
    'Construction consumption': [15354, 17487, 19790, 22820, 25714, 28975, 32646, 35249, 38132, 41350, 43822],
    'Total energy consumption': [23539, 26860, 27999, 28203, 28171, 29034, 29948, 30670, 31373, 32228, 31438]
}

df = pd.DataFrame(data)
df.set_index('Year', inplace=True)

# Scale the features
scaler = MinMaxScaler(feature_range=(0,1))
scaled_data = scaler.fit_transform(df)

# Define the lookback period
look_back = 3
X, y = [], []
for i in range(look_back, len(scaled_data)):
    X.append(scaled_data[i-look_back:i, :])
    y.append(scaled_data[i, -1])

X, y = np.array(X), np.array(y)

# Split the data into training and testing sets (80% - 20%)
train_size = int(0.8 * X.shape[0])
X_train, X_test = X[:train_size], X[train_size:]
y_train, y_test = y[:train_size], y[train_size:]

# Model architecture
model = Sequential()
model.add(LSTM(units=50, return_sequences=True, input_shape=(X_train.shape[1], X_train.shape[2])))
model.add(LSTM(units=50, return_sequences=False))
```

```
    model.add(Dense(units=25))
    model.add(Dense(units=1))

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

    # Summary of the model architecture
    model.summary(), X_train.shape, y_train.shape

    .array(Xs), np.array(ys)

    time_steps = 3
    X, y = create_dataset(features_scaled, target_scaled, time_steps)

    # Splitting the data into training and testing sets
    train_size = int(len(X) * 0.67)
    test_size = len(X) - train_size
    X_train, X_test = X[0:train_size], X[train_size:len(X)]
    y_train, y_test = y[0:train_size], y[train_size:len(y)]

    # Initialize the model
    model = Sequential()
    model.add(LSTM(50, input_shape=(X_train.shape[1], X_train.shape[2]),
return_sequences=True))
    model.add(LSTM(50))
    model.add(Dense(25))
    model.add(Dense(1))
    model.compile(loss='mean_squared_error', optimizer='adam')

    # Train the model
    history = model.fit(X_train, y_train, epochs=100, batch_size=1, validation_split=0.2,
verbose=0)

    # Predicting and evaluating the model
    y_train_pred = model.predict(X_train)
    y_test_pred = model.predict(X_test)

    # Invert the scale to compare
    y_train_pred = scaler_target.inverse_transform(y_train_pred)
    y_train = scaler_target.inverse_transform(y_train.reshape(-1, 1))
    y_test_pred = scaler_target.inverse_transform(y_test_pred)
    y_test = scaler_target.inverse_transform(y_test.reshape(-1, 1))

    # Calculate RMSE
    train_score = np.sqrt(mean_squared_error(y_train[:, 0], y_train_pred[:, 0]))
    test_score = np.sqrt(mean_squared_error(y_test[:, 0], y_test_pred[:, 0]))

    # Prepare for recursive multi-step forecast
    def forecast(model, features_scaled, time_steps, steps):
        input_seq = features_scaled[-time_steps:]
```

```
        forecasts = []
        for _ in range(steps):
            prediction = model.predict(input_seq.reshape(1, time_steps, -1))[0, 0]
            forecasts.append(prediction)
            new_seq = np.array([prediction])
            input_seq = np.append(input_seq, new_seq)
            input_seq = input_seq[1:]
        return np.array(forecasts)

    # Forecasting the future values
    steps = 40  # Predicting from 2021 to 2060
    forecasts_scaled = forecast(model, features_scaled, time_steps, steps)
    forecasts = scaler_target.inverse_transform(forecasts_scaled.reshape(-1, 1))

    # Extracting forecasted years
    years_forecasted = np.arange(2021, 2021 + steps).reshape(-1, 1)

    # Visualizing the results
    plt.figure(figsize=(12, 6))
    plt.plot(df.index, target, label="Historical Data", color='blue')
    plt.plot(years_forecasted, forecasts, label="Forecasted Data", linestyle="--", color='red')
    plt.xlabel('Year')
    plt.ylabel('Total Energy Consumption')
    plt.title('Total Energy Consumption Forecast (2021-2060)')
    plt.legend()
    plt.grid(True)
    plt.tight_layout()

    # Returning plot, RMSE, and forecasted values with years
    plt, train_score, test_score, np.concatenate([years_forecasted, forecasts], axis=1)

    import pandas as pd
    from sklearn.preprocessing import MinMaxScaler
    from sklearn.model_selection import train_test_split
    from sklearn.metrics import mean_squared_error
    import tensorflow as tf
    from tensorflow.keras.models import Sequential
    from tensorflow.keras.layers import Dense, LSTM, Dropout
    import numpy as np
    import matplotlib.pyplot as plt

    # Load the dataset
    data = {
        "Year": [2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020],
        "Resident population": [7869.3, 8023, 8119.8, 8192.4, 8281.1, 8315.1, 8381.5, 8423.5,
8446.2, 8469.1, 8477.3],
        "Agriculture and forestry": [2409.2, 2736.9, 3057.8, 3228.5, 3358.6, 3636.1, 3690.6,
3568.5, 3591.6, 3726.6, 3916.8],
        "Energy supply": [904.65, 947.43, 1121.2, 1065.5, 1149.8, 1357.6, 1417.9, 1527, 1604.6,
1692.7, 1660.7],
```

```
        "Industrial consumption": [20949, 22793, 24492, 26233, 27758, 29343, 30595, 32987, 34929, 36037, 36523],
        "Transportation consumption": [1767.2, 1988.4, 2199.5, 2233.9, 2378.9, 2240.4, 2316.4, 2420.2, 2570.7, 2749.1, 2761.5],
        "Construction consumption": [15354, 17487, 19790, 22820, 25714, 28975, 32646, 35249, 38132, 41350, 43822],
        "Generate electricity": [5752.1, 6992.8, 7340, 7820.6, 6951.1, 7380.5, 7786.5, 8219.7, 8104.9, 8131.6, 7491.1],
         "heating": [204.84, 286.73, 354.09, 335.18, 382.1, 440.86, 415.01, 523.98, 1010.8, 953.12, 1004],
         "Other conversion": [-610.67, 243.37, 223.05, -968.04, -1181.7, -1275.6, -1334.3, -1635.7, -2028.8, -2376.4, -2414.7],
        "Supply loss": [454.13, 450.76, 457.5, 353.57, 500.15, 474.01, 337.92, 377.27, 370.15, 378.33, 372.9],
        "industry": [14313, 15172, 15495, 16348, 17019, 17242, 17725, 17832, 18124, 19110, 18873],
        "traffic": [1398.3, 1494.7, 1618.2, 1743.7, 1915.9, 2019.3, 2083.6, 2188, 2324.7, 2482.9, 2484.5],
        "unit": [534.58, 620.83, 690.81, 738.21, 728.32, 756.83, 794.94, 861.24, 976.14, 1039.5, 1023.1],
        "Resident life": [1148, 1205.1, 1372.1, 1469.9, 1472.5, 1566.2, 1706.3, 1862.6, 2033.7, 2070.4, 2180.6],
            "Carbon emission": [56360.05184, 65193.34223, 67502.61337, 66749.3757, 64853.27604, 66074.80995, 68526.12467, 70451.55739, 71502.00286, 74096.33108, 72633.32425]
    }

    df = pd.DataFrame(data)

    # Extract features and target variable
    features = df.drop(columns=["Year", "Carbon emission"])
    target = df["Carbon emission"]

    # Normalize the features
    scaler_x = MinMaxScaler(feature_range=(0, 1))
    scaled_features = scaler_x.fit_transform(features)
    scaler_y = MinMaxScaler(feature_range=(0, 1))
    scaled_target = scaler_y.fit_transform(target.values.reshape(-1, 1))

    # Convert to time series format
    def create_dataset(X, y, time_steps=1):
        Xs, ys = [], []
        for i in range(len(X) - time_steps):
            v = X[i:(i + time_steps)]
            Xs.append(v)
            ys.append(y[i + time_steps])
        return np.array(Xs), np.array(ys)

    TIME_STEPS = 3
    X_time_series, y_time_series = create_dataset(scaled_features, scaled_target, TIME_STEPS)
```

```
# Split the data into train and test sets
X_train, X_test, y_train, y_test = train_test_split(X_time_series, y_time_series, test_size=0.2, random_state=42, shuffle=False)

# Model architecture
model = Sequential()
model.add(LSTM(units=50, return_sequences=True, input_shape=(X_train.shape[1], X_train.shape[2])))
model.add(LSTM(units=50))
model.add(Dropout(0.2))
model.add(Dense(units=1))

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

# Train the model
history = model.fit(X_train, y_train, epochs=100, batch_size=32, validation_split=0.1, shuffle=False)

# Predictions
y_pred = model.predict(X_test)
y_train_inv = scaler_y.inverse_transform(y_train.reshape(1, -1))
y_test_inv = scaler_y.inverse_transform(y_test.reshape(1, -1))
y_pred_inv = scaler_y.inverse_transform(y_pred)

# Plot training & validation loss values
plt.plot(history.history['loss'], label='Train Loss')
plt.plot(history.history['val_loss'], label='Validation Loss')
plt.title('Model loss')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(loc='upper right')
plt.show()

# Plotting the actual vs predicted values
plt.plot(y_test_inv.flatten(), marker='.', label='True')
plt.plot(y_pred_inv.flatten(), 'r', marker='.', label='Predicted')
plt.ylabel('Carbon Emission')
plt.xlabel('Time Step')
plt.legend()
plt.show()

# Evaluate the model
mse = mean_squared_error(y_test_inv.flatten(), y_pred_inv.flatten())
print('Mean Squared Error on Test Data: ', mse)

# Long-term Prediction
# Here we would ideally append the predicted values to the input features and continue predicting.
```

```
    # A loop can be constructed to predict the next value, append it to features and predict again.

    def predict_future(model, features, time_steps, future_steps):
        future_predictions = []
        for _ in range(future_steps):
            input_data = features[-time_steps:]
            input_data = input_data.reshape((1, time_steps, features.shape[1]))
            prediction = model.predict(input_data)
            features = np.vstack((features, prediction))
            future_predictions.append(prediction)
        return future_predictions

    future_steps = 40   # predicting for 2021-2060
    future_predictions_scaled = predict_future(model, scaled_features, TIME_STEPS, future_steps)
    future_predictions = scaler_y.inverse_transform(future_predictions_scaled).flatten()

    # Plotting future predictions
    plt.figure(figsize=(10,6))
    plt.plot(range(2021, 2061), future_predictions, 'r', marker='.', label='Predicted Carbon Emission')
    plt.title('Future Carbon Emission Predictions')
    plt.ylabel('Carbon Emission')
    plt.xlabel('Year')
    plt.legend()
    plt.show()
```