

保护生态环境下的预测评价模型

摘 要:

随着我国经济社会的发展,工业化、城市化的快速进程,造成了生态环境严重破坏,如何实现可持续发展,解决环境问题是关键。本文应用经济与环境方面的数据对经济水平、工业水平、森林植被面积进行综合决策研究。最终决定对问题一构建基于中国森林面积的对比预测模型,对问题二构建基于DEMATEL—相关性分析模型,对问题三构建基于TOPSIS—RSR的综合评价模型。

针对问题一,选取中国1990-2020年中国森林面积数据,对数据进行显著性检验,将通过显著性检验的数据代入线性回归模型、ARIMA模型和ARIMA+SVM模型中。再运用混淆矩阵,得到三个模型的F1—score值和Accuracy值。**发现:**ARIMA+SVM模型相比线性回归模型和ARIMA模型,在计算Accuracy值上,模型的测试效果分别提升了70.7%和27.2%,在计算F1—score值上,模型测试效果分别提升了151.7%和30.4%。因此,我们选用ARIMA+SVM模型来预测中国在2025-2035年内的森林面积。

针对问题二,通过关系查找以及专家打分,得到人口、就业人数、财政收入、人均GDP、人均生活能源消费量、农业总产值指数、农业就业人数、城镇人口、纸板产量两两之间的影响力指标,并对指标进行两两评价,构建初始影响矩阵。使用DEMATEL方法求得各指标之间的权重,与原始数据的乘积即为加权后的数据。运用SPSS计算Pearson相关系数得到它们之间的相关性,并对它们进行**量化分析**。**发现:**退耕还林政策促进中国的工业发展,除了制约农业就业人数的提高,它对国民经济各个方面都呈现促进作用。

针对问题三,首先对国家数据进行预处理,剔除缺失值严重的列和国家名字缺失的列,再对其他数据进行预处理,并计算某一段时间内的增长率。将数据导入python中,以国家数据为参照运用pandas库合并数据,把合并完成的数据导入TOPSIS,求出相对距离 C_i ,利用RSR秩和比法对相对距离 C_i 进行分类评价,从而,得到在不同经济水平、工业水平、森林植被面积的国家的**评价结果**。**发现:**相对而言,大多数发达国家比一般国家、欠发达国家最终评分高,他们的经济水平、工业水平、森林植被面积相对较高。在评分前十的国家里高收入国家占到了50%,中高等收入国家占到了30%,中低等收入以及低等的国家占到了20%。

最后,我们对模型进行了评价,指出了模型的优点和缺点。

关 键 词: 基于 SVM 的 ARIMA 模型, 基于 DEMATEL 的相关性分析, 基于 TOPSIS 的 RSR 模型

一、问题重述

保护生态环境就是保护生产力,它在稳定国民生产生活中发挥着巨大的作用。随着工业发展,生态环境受到了巨大的破坏,同时,落后国家工业化发展就会排放巨量二氧化碳并与发达国家相比破坏更多森林面积,如何兼顾发达国家与发展中国家之间的矛盾,实现生态与社会的可持续发展,需要每个人思考并为此不断奋斗。本题提供了 GDP、二氧化碳排放量、能源使用量等九个数据。对比全球主要国家的森林面积,中国的植被面积逐年升高。

根据所提供的数据,你需要建立合适的数学模型来解决如下问题:

问题一:依据附件森林面积数据来预测中国在 2025-2035 十年内的森林面积。

问题二:查找相关数据通过构建合适的模型来具体分析退耕还林对于国家工业水平以及国民经济的影响。

问题三:依据经济、工业水平和森林植被面积等数据构建世界各国评价模型。

问题四:综合题目问题,向联合国递交一份报告书,提出若干建议和措施并具体描述新的发现。

二、问题分析

2.1 问题一的分析

针对问题一,求解 2025-2035 年中国森林面积的预测值,划分训练集和测试集之后,我们选用线性回归、ARIMA和ARIMA + SVM三种模型,代入处理后的数据得到预测值,综合使用混淆矩阵的方法求解 $F1 - score$ 以及计算Mae值评估三种模型的预测效果,选择效果最优的ARIME + SVM模型作为最终的中国森林面积预测模型。

2.2 问题二的分析

为了探究退耕还林政策对于国家工业水平以及国民经济水平的影响,拟额外选取十个二级指标,通过构建基于 DEMATEL-相关性分析的模型进行构建。先在网上查找相关数据,对指标之间两两评价,构建初始影响矩阵,利用 SPSSAU 求得权重后带入原始数据进行相关性分析,最终求得两两变量之间的相关程度。

2.3 问题三的分析

为了构建评价模型对于全国经济水平、工业水平、森林植被面积进行评价,先将原始数据进行预处理。衡量工业水平需要整合四个文档得到二氧化碳排放速率、工业增速、耗电增速、能源使用增速;衡量森林植被面积整合三个文档得到农业用地增速、森林面积增速、森林面积百分比增速;经济只需分析 GDP 增速,将整合后的数据代入基于 TOPSIS-RSR 模型中,得到不同经济水平、工业水平、森林植被面积的国家的的评价结果。

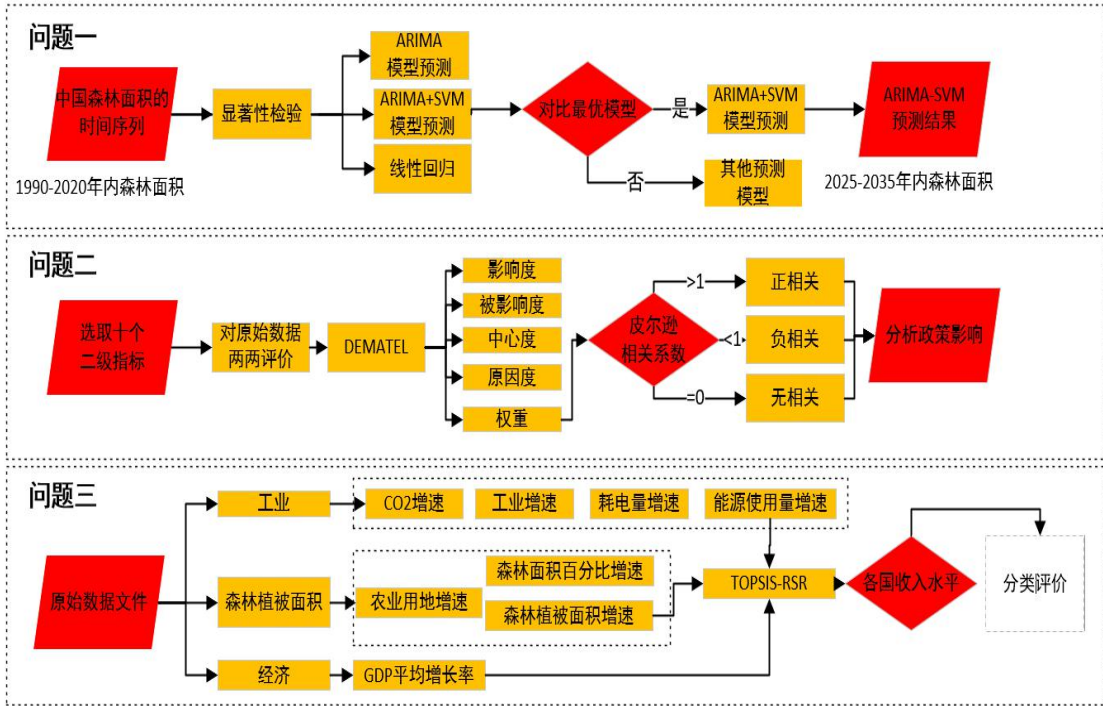


图 1 全文流程图

三、模型假设

- 1、假设中国 2025-2035 年国内森林面积平稳发展，不受突发条件影响
- 2、假设原始数据具有平稳性，不会受到异常/缺失值的影响
- 3、假设相关性水平相对于其他变量而言较小的变量属于无关变量可以忽略
- 4、假设世界各国经济发展、工业水平提高不受到突发条件的影响，比如战争，自然灾害等，没有外部以及国内的发展风险

四、主要符号说明

符号	符号说明
\hat{y}	样本预测值
N_t	平稳时间序列
e_i	白噪声序列
K_i	核函数
p, q	ARIMA 模型的阶数
f	影响度
Z_{ij}, X_{ij}, A_{ij}	矩阵

^{*}注：其余符号详见文中说明

五、模型的建立与求解

5.1 基于中国森林面积的对比预测模型

5.1.1 数据描述

我们首先从附件中提取出中国近 30 年的森林面积的数值，画出相应的直方图和曲线图（如图），从图中可以看出，中国的森林面积呈现出以一定的速率增长的趋势，累计面积逐年增加。我们计算出图表中的相关数据指标，发现森林面积的数据变异系数小，可以说明其离散程度小。

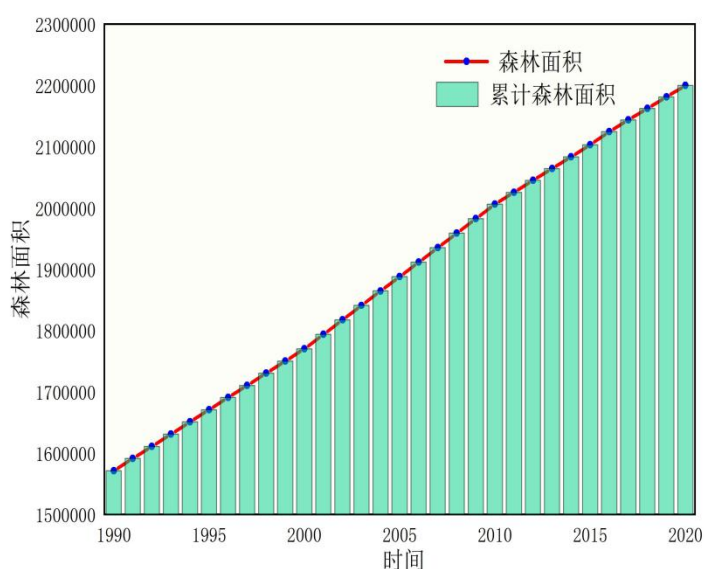


图 2 近 30 年中国累计森林面积变化图

表 1 数据描述

方差	38286063515
极差	628376
变异系数	0.1036
中位数	1888055
1/4 分位数	1720356
2/4 分位数	1888055

5.1.2 预测模型的建立

(1) 线性回归预测模型

设回归直线为：

$$y = b * x + a$$

y 到直线的距离可以用 $\sum_{i=1}^n [y_i - (a + bx_i)]^2$ 定量描述，将其看成 $Q(a, b)$ ：

$$Q(a, b) = \sum_{i=1}^n [y_i - (a + bx_i)]^2$$

找两个数 a, b 使二元函数 $Q(a, b)$ 在 $a = \bar{a}, b = \bar{b}$ 处达到最小，最后可以得到 a, b ：

$$b = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2}$$

$$a = \bar{y} - b\bar{x}$$

(2) ARIMA预测模型

N_t 为平稳时间序列, e_i 为白噪声序列, $\varphi_1 \cdots \varphi_p$ 为AR模型的系数, $\theta_1 \cdots \theta_q$ 为MA的系数, p 和 q 为模型的阶数, 具体公式如下所示:

$$N_t = \theta_1 N_{t-1} + \cdots + \theta_p N_{t-p} + e_t - \theta_1 e_{t-1} - \cdots - \theta_q e_{t-q}$$

(3) SVM预测模型

在训练集中, X_i 为 m 个包含 N 个特征值的 N 维向量, y_i 为 x_i 在高维空间 F 的内积运算。训练集中的自变量 X_i 满足如下不等式:

$$y_i(x_i w + b) - 1 + \tau \geq 0 (\tau \geq 0; i = 1, 2, \dots, m)$$

将线性问题转化为如下最优化问题:

$$\begin{aligned} \min \quad & \frac{1}{2} \|w\|^2 + c \sum_{i=1}^m \tau_i \\ \text{s.t.} \quad & y_i(x_i w + b) - 1 + \tau_i \geq 0 \end{aligned}$$

最终最优判别函数满足以下等式:

$$f(x_i) = \text{sign} \sum_{i=1}^m y_i K_i(x_i w + b)$$

核函数 K_i 为:

$$K(x_i, y_i) = \frac{1 - r^2}{2(1 - 2rcos(x - x_i) + r^2)}$$

(4) ARIMA + SVM组合模型

ARIMA模型没办法探究非线性函数, 这会降低它的预测精度. 而支持向量机(SVM)在处理小型数据集、探究非线性函数方面属于优势, 它可规避其他机器学习算法中易于出现的局部极小以及过拟合现象。为了更直观的展示ARIMA + SVM原理, 绘制流程图如下:

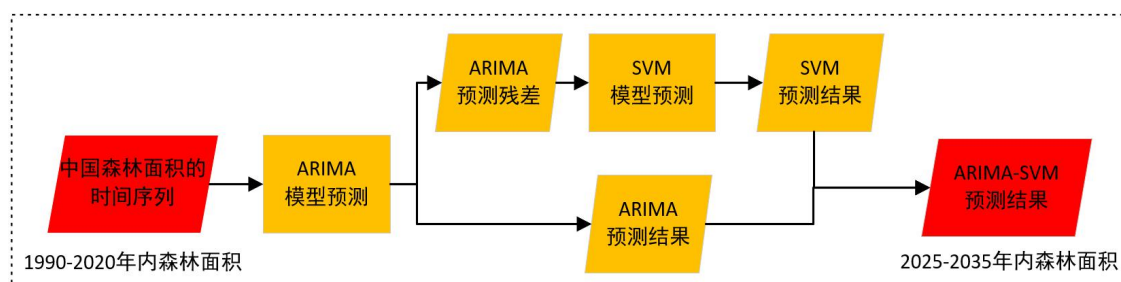


图 3 ARIMA+SVM 原理流程图

5.1.3 数据显著性检验

选取 1990-2019 森林面积数据利用SPSS进行置信区间为 0.95 的单样本 T 检验, 得到 $T = 54$, $p(T \leq t)$ 单尾 $= 5.8E - 31 < 0.05$, 说明数据不成正态分布, 数据存在显著性的差异, 再对数据进行 F 检验, F 为 $4.66E + 08$, $P(F \leq f) = 8E - 119 < 0.05$ 显著性水平较高, 通过显著性检验, 符合线性回归的条件。

5.1.4 模型的评估标准

我们为模型设置了以下的评估标准:

我们选取 1990-2010 的森林面积作为训练集, 利用 2010-2020 森林面积数据作为测试集, 综合考量一下A、B两种方法得出最优预测方法

A. 我们采用混淆矩阵的方法计算精确度、召回率、准确度、 AUC 、 $F1 - score$ 选择几个指标对于三个预测模型进行评价。

真实结果与预测结果存在 TP、FN、FP、TN 四种关系, 如下表所示:

真实结果	预测结果	
	正例	反例
正例	TP	FN
反例	FP	FN

精确度:

$$precision = \frac{TP}{TP + FP}$$

召回率:

$$recall = \frac{TP}{TP + FN} = \frac{TP}{P}$$

准确度:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

B. 计算真实值与模型预测值之间的差异进行综合评价, 其中, 真实值 $y = (y_1, y_2, \dots, y_m)$, 模型预测值为 $\hat{y} = (\hat{y}_1, \hat{y}_2, \dots, \hat{y}_m)$ 。

$$Mae = \frac{1}{m} \sum_{i=1}^m |\hat{y}_i - y_i|$$

5.1.5 模型预测效果的比较

我们将附件中给出的 $AG, LND, FRST, K2$ 作为数据指标带入多元线性回归模型、 $ARIMA$ 模型以及 $ARIMA + SVM$ 模型中,并对三个模型的预测效果进行评估,训练集与测试集的预测效果对比表如下:

表 3 三种模型的评估对比表

评估效果	线性回归模型		ARIMA 模型		ARIMA+SVM 模型	
	训练集	测试集	训练集	测试集	训练集	测试集
平均绝对误差	5.3	5.67	3.0	2.98	1.21	1.3
评估结果	0.4871	0.2938	0.6764	0.5984	0.8736	0.8937
Accuracy	0.69	0.41	0.73	0.55	0.82	0.7
$F1 - score$	0.78	0.29	0.75	0.56	0.81	0.73

从上表可以看出, $ARIMA + SVM$ 模型对训练样本的预测结果在测试集上的表现领先 $ARIMA$ 模型 49.3%。同时, $ARIMA + SVM$ 模型在Accuracy和 $F1 - score$ 上也分别领先线性回归 70.7%, 151.7%, 领先 $ARIMA$ 模型 27.2%, 30.4%。

此外, 我们绘制了三个模型的预测结果与真实值的对比曲线图, 如下所示:

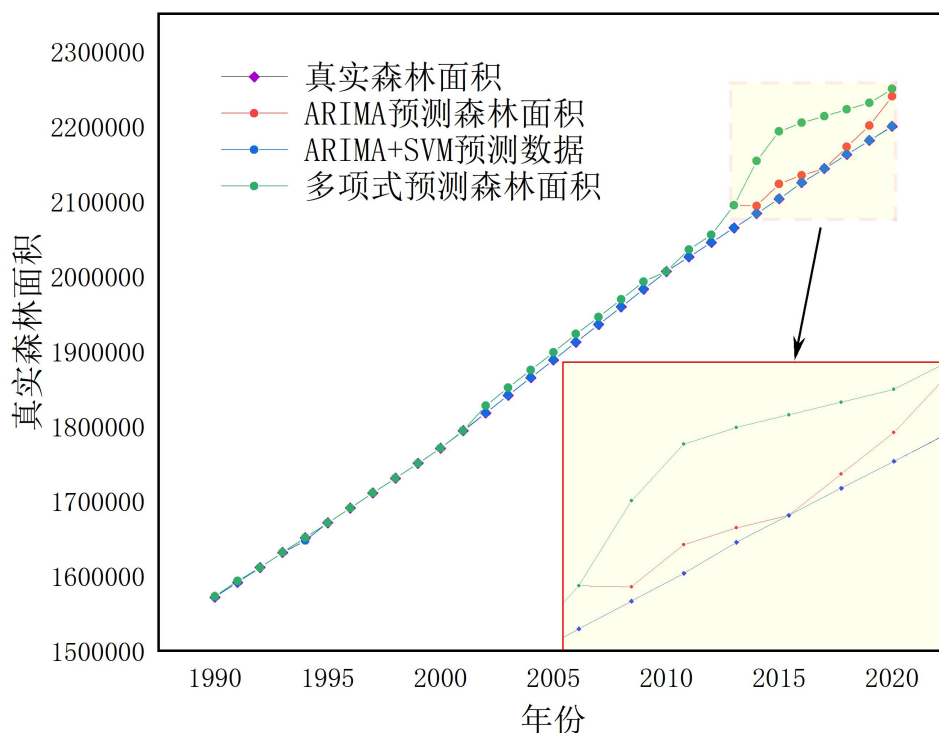


图 4 模型效果对比图

由上图可以明显看出, $ARIMA + SVM$ 模型在训练集和测试集上的预测值与真实数据极为贴近, 而线性回归和 $ARIMA$ 模型的预测值与真实值存在较大差异。这说明 $ARIMA + SVM$ 在样本内的学习方面与提前设定好函数形式的线性模型以及 $ARIMA$ 模型相比确实具有一定的优势, 它能够更加充分的学习样本内部的规律, 处理小样本、线性时具有明显优势, 建立出符合样本规律的最佳模型。对于函数选择假定所带来的人为训练误差进行有效的避免。因此我们选择 $ARIMA +$

*SVM*模型作为问题一的 2025-2035 年中国森林面积预测模型。

5.1.6 求解基于 *ARIMA* + *SVM* 中国森林面积预测模型

我们先将一组训练集时间序列的数据 y_i 看成两个部分: 线性自相关的结构 L_i 和非线性结构 N_i , 具体流程如下所示:

Step 1: 数据预处理。我们先将时间段内中国的森林面积数据样本归一化到 $[0,1]$ 区间。

Step 2: 用 *ARIMA* 模型对 y_i 进行预测。将预测结果设为 Q_i , E_i 为原序列与 *ARIMA* 模型预测结果的残差, 即残差 $E_i = y_i - Q_i$ 。序列 $\{E_i\}$ 隐含了原序列中的非线性关系。

Step 3: 根据 **Step 1** 得到的残差序列, 通过计算 **Step 1** 中的 *ARIMA* 模型的阶数最终确定输入残差阶数, 来去重建时间序列, 并利用 *SVM* 预测残差, 设预测结果为 E_i 。

Step 4: 将两个模型的结果通过简单的累加, 就能得到 2025-2035 年中国森林面积预测结果 $\{y_i\}$ 。训练完成后, 对面积验证数据采用同样的方法进行数据处理工作, 将对应的时间序列带入训练好的 *ARIMA* + *SVM* 模型中, 得到面积验证数据的预测结果。

我们通过选取 1990-2019 年的三十个数据, 将其带入 *ARIMA*+*SVM* 中国森林面积预测模型, 得到最终答案如下表所示:

表 4 2025-2035 中国森林的面积预测结果

年份	森林面积 (km^2)
2025	2317576.49
2026	2339087.12
2027	2360597.75
2028	2382108.38
2029	2403619.01
2030	2425129.64
2031	2446640.27
2032	2468150.9
2033	2489661.53
2034	2511172.16
2035	2532682.79

5.2 基于 DEMATEL-相关性分析的模型构建

5.1.1 指标选取

退耕还林政策会对国民经济以及工业水平产生影响，将其设置为一级指标，选取森林面积为退耕还林下的二级指标，选取人口、就业人数、财政收入、人均GDP、人均生活能源消费量为国民经济下的二级指标，选取农业总产值指数、农业就业人数、城镇人口、纸板产量作为工业化水平的二级指标。



图 5 二级指标图

表 5 模糊评价表

语义变量	模糊尺度
无影响	0
影响很小	2
影响较小	4
影响较大	6
影响很大	8

5.2.2 建立基于DEMATEL-相关性分析模型

利用模糊DEMATEL矩阵求指标权重，并结合相关性分析完成指标的筛选，具体流程如下：

Step 1: 通过网上查找相关数据，对两两指标之间按照上表模糊评价表进行评价，得到初始直接影响矩阵：

$$D = \begin{bmatrix} 0 & d_{12} & \dots & d_{1m} \\ d_{21} & 0 & \dots & d_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ d_{m1} & d_{m2} & \dots & 0 \end{bmatrix}$$

Step 2: 数据归一化处理，即当前值 d_{ij} 除以关系矩阵中的最大值，得到最终影响矩阵 Z ：

$$Z_{ij} = \frac{d_{ij}}{r} = \frac{d_{ij}}{\max \sum_{j=1}^m d_{ij}} = (Z_{ij})_{m \times m}$$

Step 3: 将规范化矩阵 Z 计算出最终矩阵 X :

$$X = Z_1 + Z_2 + \cdots + Z_n = Z(1 - Z)^{-1} = (Z_{ij})_{m \times m}$$

Step 4: 计算影响度 f 和被影响度 e , 并利用它们的数值来计算 p 以及 q 。中心度 $p = f + e$ 和原因度 $q = f - e$:

$$f_i = \sum_{j=1}^m X_{ij} (1 \leq i \leq m, 1 \leq j \leq m)$$

$$e_i = \sum_{j=1}^m X_{ji} (1 \leq i \leq m, 1 \leq j \leq m)$$

STEP5: 将权重 ω 与原始数据 A 相乘, 利用Pearson相关系数法求出 $|r|$ 值高的变量, 如果 $|r| \approx 0$ 说明两变量之间没有任何线性关系, 直接省略, 反之同理。

$$A_{ij} = \omega^T * A_{ij}, r = \frac{\sum_{i=1}^m (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=0}^m (X_i - \bar{X})^2 + \sum_{i=0}^m (Y_i - \bar{Y})^2}}$$

5.2.3 求解基于DEMATEL-相关性分析模型

1、求得初始影响矩阵 D :

表 6 初始影响矩阵_{11×11}

	森林面积	人口	就业人数	...	城镇人口	纸板产量
森林面积	0	0	0	...	6	8
人口	6	0	6	...	6	6
就业人数	0	0	0	...	7	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮
城镇人口	6	1	7	...	0	6
纸板产量	8	0	0	...	1	0

2、 基于 DEMATEL 求解结果的综合分析

表 7 综合影响矩阵

	影响度 D	被影响度 C	中心度 D+C	原因度 D-C
森林面积	1.338	1.657	2.994	-0.319
人口	2.152	0.05	2.202	2.102
就业人数	1.558	0.862	2.421	0.696
财政收入	0	2.05	2.05	-2.05
人均 GDP	0.842	1.183	2.025	-0.34
人均能源消耗	0.679	1.235	1.915	-0.556
农业总产值指数	1.491	1.263	2.754	0.228
农业就业人数	1.968	1.523	3.491	0.445
城镇人口	2.079	1.622	3.7	0.457
纸板产量	0.525	1.188	1.713	-0.663

由上表可以看出人口对于其他要素的影响较大，其次是城镇人口，农业就业人数，财政收入不会对其他因素产生影响；财政收入、森林面积、城镇人口主要受到其他多种因素的影响，人口几乎不会受到其他因素的影响；城镇人口、农业就业人数、森林面积在十个因素中重要性程度较高。

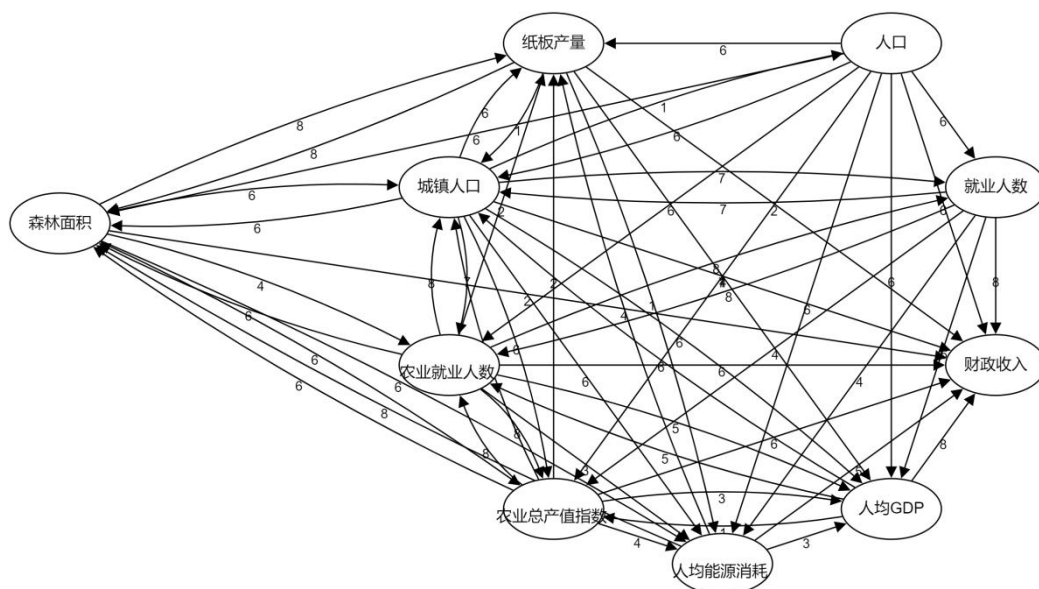


图 6 模型图

3、 权重提取

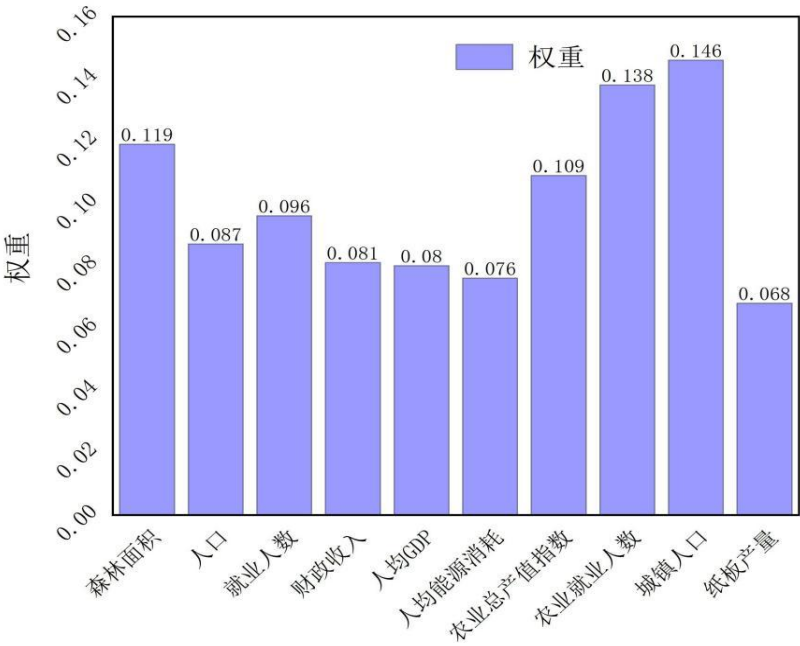


图 7 权重图

本题求得每一个变量的权重，具体如下图所示，由下图可以看出城镇人口、农业就业人数权重最高，纸板产量权重最低，几乎可以忽略。每一个权重和它对应的变量相乘得到新的数据集，将新的数据集导入到 SPSS 中，进行相关性分析。

4、 相关性分析

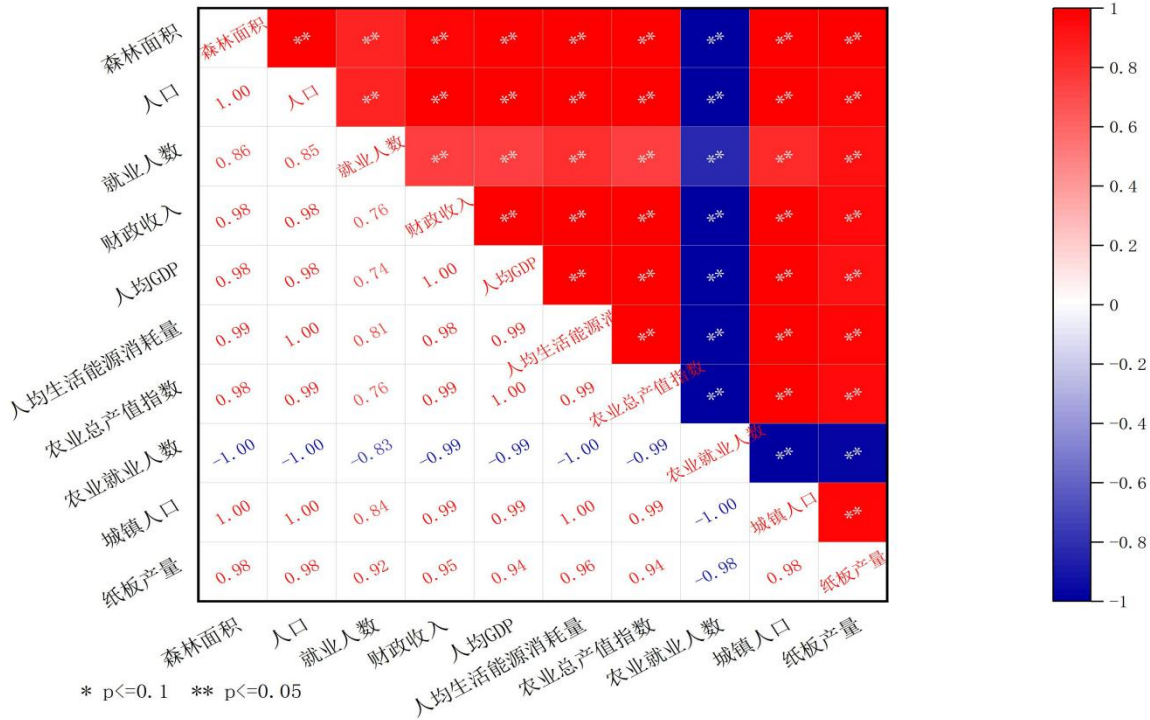


图 8 各要素之间的热图

由上图可以看出，退耕还林（森林面积变化）对于人口、城镇人口、生活能源消耗量、农业产值指数、人均 GDP、财政收入呈现正相关，相比之下对于就业人数相关性不是那么的强，对于农业就业人数呈现负相关的关系。综上所述，退耕还林政策促进中国的工业发展，除了不利于农业就业人数的提高，它对于国民经济的各个方面也呈现促进作用。

5.3 基于TOPSIS – RSR的综合评价模型

5.3.1 数据预处理

通过分析每年每个国家缺失数据的多少，我们选定缺失值数量相对较少的1990-2000年数据进行分析具体流程如下所示：

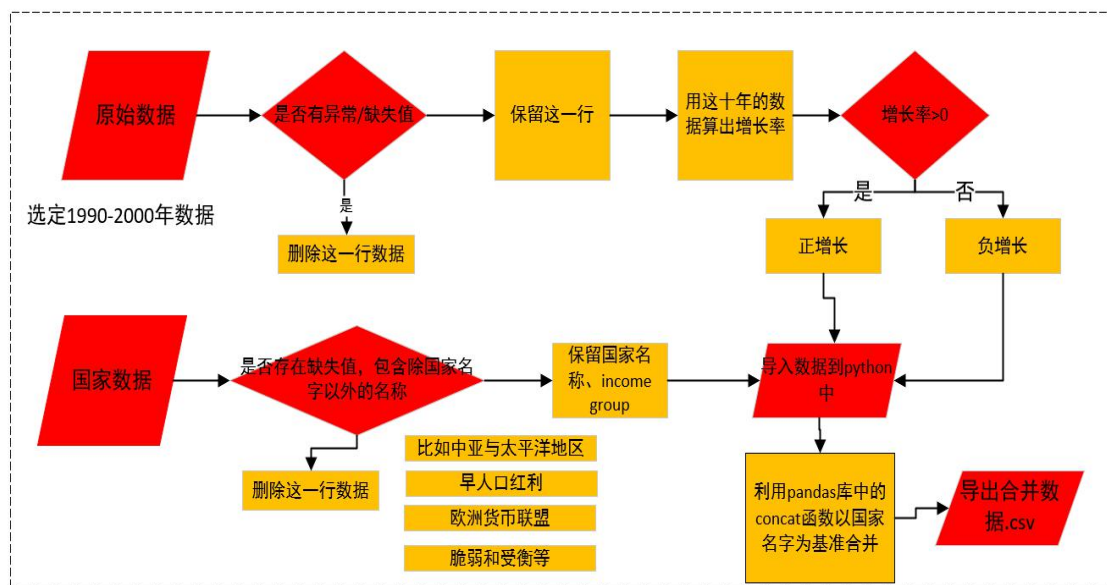


图 9 数据预处理操作

5.3.2 建立基于TOPSIS – RSR的评价模型

我们处理经过数据预处理后得到的数据，来建立基于 TOPSIS – RSR 的综合评价模型，具体步骤如下所示：

Step 1: 建立数据结构

假设一共有 n 行需要进行处理，每一行要处理有 m 个评价指标，得到的原始数据以矩阵的形式表示：

$$A = \begin{bmatrix} A_{11} & A_{12} & \dots & A_{1m} \\ A_{21} & A_{22} & \dots & A_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ A_{n1} & A_{n2} & \dots & A_{nm} \end{bmatrix}$$

Step 2: 数据归一化

先对数据进行归一化处理操作: $B_{2j} = \frac{A'_{ij}}{\sqrt{\sum_{i=1}^n (A'_{ij})^2}}$, 其中 $i = 1, 2, \dots, n; j = 1, 2, \dots, m$ 。归一化后的数据矩阵记为:

$$B = \begin{bmatrix} B_{11} & B_{12} & \dots & B_{1m} \\ B_{21} & B_{22} & \dots & B_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ B_{n1} & B_{n2} & \dots & B_{nm} \end{bmatrix}$$

Step 3: 确定指标最优值和最劣值

由决策矩阵得到最优值向量(正理想解) B^+ 和最劣值向量(负理想解) B^- , 在本方案中最优方案和最劣方案分别为:

$$\begin{aligned} B^+ &= (B_1^+, B_2^+, \dots, B_m^+) \\ B^- &= (B_1^-, B_2^-, \dots, B_m^-) \end{aligned}$$

式中: $B_j^+ = \max \{B_{1j}, B_{2j}, \dots, B_{nj}\}$, $B_j^- = \min \{B_{1j}, B_{2j}, \dots, B_{nj}\}$ ($j = 1, 2, \dots, m$),

B_j^+ 和 B_j^- 分别为第 j 个评价指标上的最大值和最小值。

Step 4: 分别计算每行每列的各个数据与最优值 D_i^+ 和最劣值 D_i^- 的距离

$$\begin{aligned} D_i^+ &= \sqrt{\sum_{j=1}^n (B_{ij} - B_j^+)^2} \quad (i = 1, 2, \dots, n) \\ D_i^- &= \sqrt{\sum_{j=1}^n (B_{ij} - B_j^-)^2} \quad (i = 1, 2, \dots, n) \end{aligned}$$

式中: D_i^+ 表示处理第 i 个数据与最佳结果的相对距离, D_i^- 表示处理第 i 个数据与最劣方案的相对距离, z_{ij} 表示某个处理 i 在第 j 个指标的取值。

Step 5: 计算相对距离 C_i

构建每个处理下的 (D_i^+, D_i^-) 二维数据空间, 并设点 $[\min(D_i^+), \max(D_i^-)]$ 为最优理想参照点 A , 计算每个点到该点之间的相对距离:

$$C_i = \sqrt{[D_i^+ - \min(D_i^+)]^2 + [D_i^- - \max(D_i^-)]^2} \quad (i = 1, 2, \dots, n)$$

根据 C_i 的值对各处理由小到大排序, 较优方案为 C_i 值较小的情况, 即最佳解是一个点到参照点 A 相对距离最近的那个点

STEP6: 根据 C_i 大小分组

提取已经经过TOPSIS排序的前 30 个数据, 根据 C_i 大小分组, 计算出 P, 然后通过查表得到概率的Probit值, 依据Probit值估算出回归方程, 根据回归方程对 C_i 进行排序:

$$p = \bar{R}/n$$

$$\hat{C}_i = a + b \times Probit$$

5.3.3 求解基于TOPSIS – RSR的评价模型

1、经过数据预处理后得到的数据（整合数据）

表 8 原始数据

国家	国家收入	二氧化碳 增速	工业增 速	能源使 用增速	耗电量 增速	...	森林 增速
安哥拉	低收入	0.95	-0.17	-0.11	0.41	:	-0.02
阿尔巴 尼亚	中低收入	-0.47	-0.27	-0.28	1.62	:	-0.02
安道尔	高收入	0.268	0	0.187	0.41	:	0
阿联酋	高收入	0.525	-0.04	0.182	0.60	:	0.26
...

2、基于 TOPSIS – RSR 模型下的工业数据结果展示

表 9 工业数据

国家	国家收入	相对接近度 C	TOPSIS 排序结果	相对 接近度	RSR 排名
:	:	:	:	:	:
安道尔共和国	中高收入	0.335	25	26	25
阿联酋	高收入	0.386	16	35	16
:	:	:	:	:	:
安提瓜巴布达	高收入	0.376	19	32	19
:	:	:	:	:	:
吉布提	中低收入	0.275	37	13	37


提取经过 TOPSIS 评分较高的前四十个国家及其它们的相对接近度 C ，进行 RSR 秩和比运算，最终得到的结果如上表所示：

排名前十的国家为：奥地利、德国、捷克共和国、西班牙、法国、巴西、中国、白俄罗斯、墨西哥、孟加拉国

3、结果分析

相对而言，大多数发达国家比一般国家、欠发达国家最终评分高，他们的经济水平、工业水平、森林植被面积相对较高。在评分前十的国家里高等国家占到了 50%，中高等国家占到了 40%，中低等以及低等的国家占到了 10%

5.4 报告书

<h1 style="color: white;">报告书</h1>	
<h3>介绍</h3>	
<p>保护生态环境就是保护生产力，它在稳定国民生产生活中发挥着巨大的作用。但是随着各国工业的发展，世界环境受到了严重的破坏，根据已经探明的石油天然气储量来算，石油天然气等能源将在 100 年内枯竭，目前世界上的科技还不能完全做到替代石油等化石能源。同时，不发达的国家想要发展必将消耗更多的能源来支撑其工业的进步与发展，一些发达国家由于屡屡破坏自然环境而受到了大自然的惩罚，如何平衡发达国家与发展中国家之间的发展矛盾，可能成为保护环境的一个重点以及难点。</p>	
	<h3>目前世界各国发展现状</h3> <p>世界各国经济水平的提高有利于更好的保护环境，大多数发达国家（中高收入、高收入）经济水平相当高，国内森林面积也相对比较大，少数欠发达国家（中低收入、低收入）虽然经济水平低，但是国内森林面积相当的大，大多数欠发达国家经济水平和森林面积都很低下。工业水平与森林面积的相关度也相当的高，工业水平高的发达国家森林面积平均高于工业水平低的国家，但是也存在例外，工业水平高的国家森林面积比较小。</p>
<h3>建议</h3>	<ul style="list-style-type: none"> ● 每个国家的经济条件不同，分为低收入、中低收入、中高收入、高收入国家，但是如果这个国家不仅收入低，生态环境也破坏的相当严重，得到国际组织援助的优先级应该为最高。 ● 如果这个国家是高收入国家，经济水平和工业水平相当的高，但是森林面积较小，生态环境破坏的很严重，这个国家就应该被重视，其次也应该得到国际组织（联合国）的援助，确保能尽快将生态环境水平提高到中等水平。 ● 保护落实大国责任制，每一个地区的大国应该承担更多的环境保护义务，督促其他小国（经济不发达或者欠发达的国家）进行环境保护，如果这个小国在一段时间以内还是在不停的破坏环境，造成生态环境的巨大破坏，环境保护组织就对于大国进行处罚，让大国缴纳一定数量的罚金来弥补效果效果造成的环境损失。 ● 对于及时整改的小国（经济不发达或者欠发达的国家），很快建成生态友好型社会，或者他们把森林面积（森林覆盖率）提高到了中等水平，就应该给予这些小国一些奖励措施，将大国缴纳的罚金支付一部分给这些小国，作为它们保护环境的奖励。
<p>虽然全世界各国现在可能还做不到环境与社会可持续发展，但是，只要国与国之间齐心协力，国际组织尽力分配各种任务，大国发挥应该具有的社会责任，小国尽力地去扭转自身发展的困境，配合国际组织整改完善，就能找到正确的方法去化解社会与环境之间的矛盾，发达国家与发展中国家之间的矛盾，世界各国都能实现经济与社会环境健康、可持续性发展，让我们用实际行动，守护地球，为了我们自己也是为了我们的未来！</p>	
<h3>展望</h3>	

六、 结果分析

第三问不同数据预处理操作最终求得的结果之间可能存在差异,比如算每一个指标的年平均增长率或者算一段时间内的增长率。在进行数据预处理的过程中,没有考虑减少数据的删除量,较多的保留更多的国家,没有采取插值的方式导致最终求得的评价结果存在着误差,但误差在允许的范围,结果可被接受。

七、 模型评价与推广

7.1 模型的优点

(1)问题一采用的 SVM 模型可以针对小样本、有限样本可以获得较好的处理的效果,特别是处理非线性回归的问题

(2)模型的稳定性比较好,泛化能力强,问题二采用的 DEMATEL 模型已经广泛应用于经济生产、军事技术、事故评价、以及简单规划等场景,并且能和多种方法叠加使用,比如灰色关联度、相关性分析、TOPSIS 法、层次分析法。相比较层次分析法,它的计算更加简单,结果展示的更加直观,能以图的方式来展示最终的结果,专家打分更加合理考虑到影响力以及被影响力,相比层次分析法增加了一定的科学性。

(3)问题三采用的 TOPSIS-RSR 模型能巧妙的结合已知标准,能同时对于多个对象进行评价

(4)模型参数的灵敏度较高,改变一个参数或者同时改变几个参数不会对于结果产生较大的影响,模型的鲁棒性较高

7.2 模型的缺点

(1)问题二采用的 DEMATEL 模型有一定的主观成分,不同专家的出来的最终结果可能存在偏差

(2)问题三采用的 TOPSIS 法只能反映各评价指标之间内部的相对接近度,不能反映与期望的最佳方案的相对接近度

7.3 模型的改进

(1)问题二拟采用 DEMATEL 与层次分析法结合的方法,先利用层次分析法求得两个判别矩阵(A 对 B 的影响以及 B 对 A 的影响),将通过一致性检验后的两个矩阵结合而成的矩阵作为 DEMATEL 法的初始矩阵在进行操作。这个方法的优点是结合了层次分析法以及 DEMATEL 法的优点,最大程度上地避开了 DEMATEL 法专家评价主观性稍高的缺点。

八、参考文献

- [1]张青. 基于 ARIMA-SVM 组合模型的创业板股票价格预测分析[J]. 广西质量监督导报, 2019(12):131-132.
- [2]吴虹, 尹华. ARIMA 与 SVM 组合模型的石油价格预测[J]. 计算机仿真, 2010, 27(05):264-266+326.
- [3]王一任, 任力锋, 陈丽文, 孙振球. 一种新的改良 TOPSIS 法及其医学应用[J]. 中南大学学报(医学版), 2013, 38(02):196-201.
- [4]张金姬, 张英. 卫生管理系统中的 TOPSIS 法[J]. 江西中医学院学报, 2004(05):73-74.
- [5]张焱, 赵鸭桥, 周铝, 王奇, 冯璐. 基于改进 TOPSIS 法的乡村振兴评价及地区比较[J]. 中国农业资源与区划, 2021, 42(02):207-217.
- [6]王一任. 综合评价方法若干问题研究及其医学应用[D]. 中南大学, 2012.

九、附录

```
# 1 用 Pandas 加载数据集
import pandas as pd
from keras.losses import mean_absolute_error, mean_squared_error
from matplotlib import pyplot as plt
from sklearn.linear_model import LinearRegression

df = pd.read_excel("train.xlsx", header=0)
#df
# 2 绘制趋势图像
# 从原数据集中分离出需要的数据集 (DataFrame)
x = df['年份']
y = df['森林面积']
# 绘图
plt.plot(x, y, 'r')
plt.scatter(x, y)
plt.title('forest ')

# 3 训练集和测试集划分
# 将 1990-2010 年的数据定义为训练集, 2010-2020 年的数据定义为测试集
train_df = df[:int(len(df) * 0.7)]
test_df = df[int(len(df) * 0.3):]

# 定义训练和测试使用的自变量和因变量
train_x = train_df['年份'].values
train_y = train_df['森林面积'].values

test_x = test_df['年份'].values
test_y = test_df['森林面积'].values
# 4. 二次多项式预测
from sklearn.preprocessing import PolynomialFeatures

# 二次多项式回归特征矩阵
poly_features_2 = PolynomialFeatures(degree=2, include_bias=False)
poly_train_x_2 = poly_features_2.fit_transform(train_x.reshape(len(train_x), 1))
poly_test_x_2 = poly_features_2.fit_transform(test_x.reshape(len(test_x), 1))

# 二次多项式回归模型训练与预测
model = LinearRegression()
model.fit(poly_train_x_2, train_y.reshape(len(train_x), 1)) # 训练模型

results_2 = model.predict(poly_test_x_2) # 预测结果

results_2.flatten() # 打印扁平化后的预测结果
```



```

print("二次多项式 2021 年预测值为: ", results_2[-1])
print("二次多项式回归平均绝对误差: ", mean_absolute_error(test_y, results_2.flatten()))
print("二次多项式均方根误差: ", mean_squared_error(test_y, results_2.flatten()))

# 绘出已知数据散点图
# plt.scatter(x,y,color = 'blue')
plt.plot(x, y, color='blue', linewidth=2)
# 绘出预测线
plt.plot(test_x, results_2, color='red', linewidth=2)
plt.title('Predict the tmall total')
plt.xlabel('year')
plt.ylabel('total')
plt.show()

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import pylab as pl

file = '森林面积百分比.csv'
data = pd.DataFrame(pd.read_csv(file))
# 统计缺失值数量
missing=data.isnull().sum().reset_index().rename(columns={0:'missNum'})
# 计算缺失比例
missing['missRate']=missing['missNum']/data.shape[0]
# 按照缺失率排序显示
miss_analy=missing[missing.missRate>0].sort_values(by='missRate',ascending=False)
# miss_analy 存储的是每个变量缺失情况的数据框

fig = plt.figure(figsize=(18,6))
plt.bar(np.arange(miss_analy.shape[0]), list(miss_analy.missRate.values), align =
'center',color=['red','green','yellow','steelblue'])

plt.title('Histogram of missing value of variables')
plt.xlabel('variables names')
plt.ylabel('missing rate')
# 添加 x 轴标签, 并旋转 90 度
plt.xticks(np.arange(miss_analy.shape[0]),list(miss_analy['index']))

```



```

plt.xticks(rotation=90)
# 添加数值显示
for x,y in enumerate(list(miss_analy.missRate.values)):
    plt.text(x,y+0.12,'{:.2%}'.format(y),ha='center',rotation=90)
plt.ylim([0,1.2])

data=data.dropna(thresh=len(data)*0.2, axis=1)

# 统计缺失值数量
missing=data.isnull().sum().reset_index().rename(columns={0:'missNum'})
# 计算缺失比例
missing['missRate']=missing['missNum']/data.shape[0]
# 按照缺失率排序显示
miss_analy=missing[missing.missRate>0].sort_values(by='missRate',ascending=False)
# miss_analy 存储的是每个变量缺失情况的数据框

plt.show()

fig = plt.figure(figsize=(18,6))
plt.bar(np.arange(miss_analy.shape[0]), list(miss_analy.missRate.values), align =
'center',color=['red','green','yellow','steelblue'])

plt.title('Histogram of missing value of variables')
plt.xlabel('variables names')
plt.ylabel('missing rate')
# 添加 x 轴标签, 并旋转 90 度
plt.xticks(np.arange(miss_analy.shape[0]),list(miss_analy['index']))
plt.xticks(rotation=90)
# 添加数值显示
for x,y in enumerate(list(miss_analy.missRate.values)):
    plt.text(x,y+0.12,'{:.2%}'.format(y),ha='center',rotation=90)
plt.ylim([0,1.2])

plt.show()

from sklearn.neighbors import KNeighborsClassifier, KNeighborsRegressor

print(data)
data1 = data

```

```

data.to_csv('city.csv')

# def knn_filled_func(x_train, y_train, test, k = 3, dispersed = True):
#     # params: x_train 为目标列不含缺失值的数据（不包括目标列）
#     # params: y_train 为不含缺失值的目标列
#     # params: test 为目标列为缺失值的数据（不包括目标列）
#     if dispersed:
#         knn= KNeighborsClassifier(n_neighbors = k, weights = "distance")
#     else:
#         knn= KNeighborsRegressor(n_neighbors = k, weights = "distance")
#
#     knn.fit(x_train, y_train)
#     return test.index, knn.predict(test)
#
# knn = knn_filled_func(x_train=data1,y_train=data1,test=data1)

import numpy as np
import pandas as pd
data = pd.read_csv('city.csv',encoding='GBK')
# 将空值形式的缺失值转换成可识别的类型
data = data.replace(' ', np.NaN)
print(data.columns)#['id', 'label', 'a', 'b', 'c', 'd']
#将每列中缺失值的个数统计出来
null_all = data.isnull().sum()
#id      0
#label    0
#a        7
#b        3
#c        3
#d        8
print(null_all)
## #查看 a 列有缺失值的数据
## a_null = data[pd.isnull(data['ABW'])]
## #a 列缺失占比
## a_ratio = len(data[pd.isnull(data['ABW'])])/len(data) #0.45901639344262296
# print(a_null)
# print(a_ratio)

from fancyimpute import KNN
fill_knn = KNN(k=3).fit_transform(data)
data = pd.DataFrame(fill_knn)
print(data.head())

```

```
data.to_csv('森林面积 百分比.csv')
# #out
#      0      1      2      3      4      5
# 0  111.0  0.0  2.0   360.0  4.000000  1.0
# 1  112.0  1.0  9.0  1080.0  3.000000  1.0
# 2  113.0  1.0  9.0  1080.0  2.000000  1.0
# 3  114.0  0.0  1.0   360.0 *3.862873 *1.0
# 4  115.0  0.0  1.0   270.0  5.000000  1.0

import datetime

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

from matplotlib.pylab import rcParams
rcParams['figure.figsize']=15,6

#数据导入和查看
data=pd.read_csv('12.csv')
print(data.head())
print("\nData Types:")
print(data.dtypes)

#格式转换
dateparse=lambda dates: pd.datetime.strptime(dates,'%Y')
data=pd.read_csv('12.csv',parse_dates=['年份'],index_col='年份',date_parser=dateparse)
data.head()
# parse_dates: 指定包含日期时间信息的列。例子中的列名是'Month'
# index_col: 在 TS 数据中使用 pandas 的关键是索引必须是日期等时间变量。所以这个参数告诉 pandas 使用'Month'列作为索引
# date_parser: 它指定了一个将输入字符串转换为 datetime 可变的函数。pandas 默认读取格式为'YYYY-MM-DD HH:MM:SS'的数据。如果这里的时间格式不一样,就要重新定义时间格式,dataparse 函数可以用于此目的。

#查看数据的索引
print(data.index)
```

```
def test_stationarity(timeseries):
    # Determining rolling statistics
    rolmean = timeseries.rolling(12).mean()
    rolstd = timeseries.rolling(12).std()
    # Plot rolling statistics:
    orig = plt.plot(timeseries, color='blue', label='Original')
    mean = plt.plot(rolmean, color='red', label='Rolling Mean') # 均值
    std = plt.plot(rolstd, color='black', label='Rolling Std') # 标准差
    plt.legend(loc='best')
    plt.title('Rolling Mean & Standard Deviation')
    plt.show(block=False)

    # Perform Dickey-Fuller Test:
    print('Results of Dickey-Fuller Test:')
    dftest = adfuller(timeseries, autolag='AIC')
    dfoutput = pd.Series(dftest[0:4], index=['Test Statistic', 'p-value', '#Lags Used', 'Number
of Observations Used'])
    for key, value in dftest[4].items():
        dfoutput['Critical Value (%s)' % key] = value
    print(dfoutput)

# 检验结果
test_stationarity(data)
plt.show()
```