### 5. DEA + TOPSIS案例：县域医院运营效率评估

**问题背景**：某省卫健委需评价 10 家县级医院的运营绩效，既要衡量资源利用效率（如 “医护人员人均诊疗量”），又要综合患者满意度等非效率指标，为资源调配提供依据。

**数据**：

* 投入指标：医护人员数（人：320/280/400/...）、床位数（张：500/450/600/...）、年度运营成本（万元：2800/2500/3200/...）；
* 产出指标：年诊疗人次（千：150/130/180/...）、治愈率（%：82/80/85/...）、患者满意度（%：85/83/88/...）。

**要求**：用 DEA 计算效率值，结合 TOPSIS 综合所有指标，排序后指出非效率医院的改进方向（如 “减少床位数” 或 “提高治愈率”）。

### 5. DEA + TOPSIS代码：县域医院运营效率评估

|  |
| --- |
| import numpy as np  import pandas as pd  from scipy.optimize import linprog  # 数据准备（10家医院）  # 投入：医护人员数、床位数、运营成本(万元)  inputs = np.array([  [320, 500, 2800], [280, 450, 2500], [400, 600, 3200],  [220, 350, 2000], [350, 550, 3000], [250, 400, 2300],  [380, 580, 3100], [200, 300, 1800], [300, 480, 2600],  [360, 520, 2900]  ])  # 产出：年诊疗人次(千)、治愈率(%)、患者满意度(%)  outputs = np.array([  [150, 82, 85], [130, 80, 83], [180, 85, 88],  [100, 78, 80], [160, 83, 86], [120, 79, 82],  [170, 84, 87], [90, 76, 79], [140, 81, 84],  [155, 82, 85]  ])  hospitals = [f'医院{i+1}' for i in range(10)]  # DEA效率计算（CCR模型）  def dea\_ccr(inputs, outputs):  n, m = inputs.shape  s = outputs.shape[1]  efficiencies = []  for j in range(n):  c = np.zeros(n + 1)  c[0] = -1  A\_eq = np.hstack((np.zeros((s, 1)), outputs.T))  b\_eq = outputs[j, :]  A\_ub = np.hstack((np.ones((m, 1)), -inputs.T))  b\_ub = -inputs[j, :]  bounds = [(None, None)] + [(0, None)] \* n  result = linprog(c, A\_ub=A\_ub, b\_ub=b\_ub, A\_eq=A\_eq, b\_eq=b\_eq, bounds=bounds, method='highs')  efficiencies.append(-result.fun)  return np.array(efficiencies)  dea\_efficiencies = dea\_ccr(inputs, outputs)  # TOPSIS综合评价  def topsis\_evaluation(inputs, outputs, dea\_scores):  inputs\_norm = inputs / np.sqrt(np.sum(inputs\*\*2, axis=0))  outputs\_norm = outputs / np.sqrt(np.sum(outputs\*\*2, axis=0))  inputs\_norm = 1 - inputs\_norm # 投入指标反向化  dea\_norm = dea\_scores / np.max(dea\_scores)  all\_data = np.column\_stack((inputs\_norm, outputs\_norm, dea\_norm.reshape(-1, 1)))  weights = np.ones(all\_data.shape[1]) / all\_data.shape[1]  weighted = all\_data \* weights  ideal\_best = weighted.max(axis=0)  ideal\_worst = weighted.min(axis=0)  d\_best = np.sqrt(np.sum((weighted - ideal\_best)\*\*2, axis=1))  d\_worst = np.sqrt(np.sum((weighted - ideal\_worst)\*\* 2, axis=1))  return d\_worst / (d\_best + d\_worst)  topsis\_scores = topsis\_evaluation(inputs, outputs, dea\_efficiencies)  combined\_scores = 0.5 \* dea\_efficiencies + 0.5 \* topsis\_scores  # 结果展示  result = pd.DataFrame({  '医院': hospitals,  'DEA效率值': dea\_efficiencies.round(4),  'TOPSIS得分': topsis\_scores.round(4),  '组合得分': combined\_scores.round(4),  '排名': (combined\_scores.argsort()[::-1].argsort() + 1)  }).sort\_values('组合得分', ascending=False)  print(result) |