### **1. AHP + 熵权法案例：社区养老服务中心供应商选择**

**问题背景**：某社区计划引入 3 家养老服务供应商，提供助餐、医疗护理、文化娱乐 3 类核心服务。社区居民委员会需综合评价 5 家候选供应商，既要体现老年居民代表的主观偏好（如 “医疗护理比助餐更重要”），又要依据供应商的历史服务数据，避免决策片面性。

**数据**：

* 主观数据：10 位老年居民代表通过 1-9 标度法对 3 项服务的相对重要性打分，形成 AHP 判断矩阵；
* 客观数据：5 家供应商近 1 年的运营数据（助餐满意度 92%/88%/95%/85%/90%、医疗响应时效 15/20/10/25/18 分钟、文化活动频次 12/8/15/6/10 次 / 月）。

**要求**：综合主客观权重计算得分并排序，选择前 3 家供应商，需说明权重融合的合理性。

### 1. AHP + 熵权法代码：社区养老服务中心供应商选择

|  |
| --- |
| import numpy as np  import pandas as pd  # 数据准备  # 客观数据：助餐满意度(%)、医疗响应时效(分钟)、文化活动频次(次/月)  data = np.array([  [92, 15, 12],  [88, 20, 8],  [95, 10, 15],  [85, 25, 6],  [90, 18, 10]  ])  suppliers = [f'供应商{i+1}' for i in range(5)]  indicators = ['助餐满意度', '医疗响应时效', '文化活动频次']  # 医疗响应时效为成本型指标，进行正向化处理  data[:, 1] = max(data[:, 1]) - data[:, 1]  # AHP权重计算  def ahp\_weight(matrix):  eigenvalues, eigenvectors = np.linalg.eig(matrix)  max\_idx = np.argmax(eigenvalues)  weights = eigenvectors[:, max\_idx].real  weights /= np.sum(weights)  n = matrix.shape[0]  lambda\_max = eigenvalues[max\_idx].real  ci = (lambda\_max - n) / (n - 1)  ri\_table = {1:0,2:0,3:0.58}  cr = ci / ri\_table[n] if ri\_table[n] !=0 else 0  return weights, cr  # 构造判断矩阵（根据10位老年居民代表打分综合）  judgment\_matrix = np.array([  [1, 1/3, 2], # 助餐满意度相对重要性  [3, 1, 4], # 医疗响应时效相对重要性  [1/2, 1/4, 1] # 文化活动频次相对重要性  ])  ahp\_weights, cr = ahp\_weight(judgment\_matrix)  print(f"AHP一致性检验CR={cr:.4f}，{'通过' if cr<0.1 else '未通过'}")  # 熵权法权重计算  def entropy\_weight(data):  data\_norm = data / np.sum(data, axis=0) + 1e-10  n, m = data\_norm.shape  entropy = -np.sum(data\_norm \* np.log(data\_norm) / np.log(n), axis=0)  return (1 - entropy) / np.sum(1 - entropy)  entropy\_weights = entropy\_weight(data)  # 综合评价  alpha = 0.5  combined\_weights = alpha \* ahp\_weights + (1-alpha) \* entropy\_weights  normalized\_data = data / np.sum(data, axis=0)  scores = np.dot(normalized\_data, combined\_weights)  # 结果展示  result = pd.DataFrame({  '供应商': suppliers,  '综合得分': scores.round(4),  '排名': (scores.argsort()[::-1].argsort() + 1)  }).sort\_values('综合得分', ascending=False)  print(result) |