### 6. 粗糙集 + 模糊聚类案例：电商用户消费群体细分

**问题背景**：某电商平台需对 5000 名用户细分，用于精准营销。用户数据含 10 项特征（如年龄、性别、月消费额、购买频率等），部分特征冗余（如 “月消费额” 与 “购买频率” 高度相关），且用户边界模糊（如 “中等消费” 与 “高消费” 无明确阈值）。

**数据**：

* 5000 条用户记录，特征包括：年龄（18-60 岁）、性别（男 / 女）、月消费额（0-5000 元）、购买频率（1-30 次 / 月）等。

**要求**：用粗糙集约简特征（保留核心特征），再用模糊聚类分为 3-5 个群体，输出每个群体的特征（如 “25-35 岁女性，月消费 2000-3000 元”）及隶属度。

### 6. 粗糙集 + 模糊聚类代码：电商用户消费群体细分

|  |
| --- |
| import numpy as np  import pandas as pd  from sklearn.preprocessing import KBinsDiscretizer  from roughsets.roughsets import RoughSet # 需安装：pip install roughsets  from fcmeans import FCM # 需安装：pip install fcmeans  import matplotlib.pyplot as plt  # 1. 数据准备（模拟5000名用户数据）  np.random.seed(42)  n\_users = 5000  # 特征：年龄(18-60)、性别(0=男,1=女)、月消费额(0-5000)、购买频率(1-30)、浏览时长(小时/周)、收藏数量(个/月)  age = np.random.randint(18, 61, n\_users)  gender = np.random.randint(0, 2, n\_users)  monthly\_spend = np.random.randint(0, 5001, n\_users)  purchase\_freq = np.random.randint(1, 31, n\_users)  browse\_time = np.random.uniform(1, 20, n\_users)  favorite\_count = np.random.randint(0, 50, n\_users)  X = np.column\_stack((age, gender, monthly\_spend, purchase\_freq, browse\_time, favorite\_count))  features = ['年龄', '性别', '月消费额', '购买频率', '浏览时长', '收藏数量']  # 2. 粗糙集特征约简  # 数据离散化（粗糙集要求离散数据）  discretizer = KBinsDiscretizer(n\_bins=5, encode='ordinal', strategy='uniform')  X\_discrete = discretizer.fit\_transform(X).astype(int)  # 构造决策系统（用月消费额离散值作为伪决策属性）  decision = X\_discrete[:, 2] # 以月消费额作为决策属性  rs = RoughSet(X\_discrete, decision)  reduct = rs.reduct() # 获取约简特征索引  print(f"原始特征数：{X.shape[1]}，约简后特征数：{len(reduct)}")  print(f"保留的核心特征索引：{reduct}，对应特征：{[features[i] for i in reduct]}")  # 提取约简后的特征  X\_reduced = X[:, reduct]  # 3. 模糊聚类（分为4个群体）  fcm = FCM(n\_clusters=4, random\_state=42)  fcm.fit(X\_reduced)  membership = fcm.u # 隶属度矩阵  labels = np.argmax(membership, axis=1) # 最终聚类标签  # 4. 分析聚类结果  cluster\_centers = fcm.centers  result = pd.DataFrame(X\_reduced, columns=[features[i] for i in reduct])  result['聚类标签'] = labels  # 输出每个群体的特征描述  print("\n各消费群体特征：")  for i in range(4):  cluster\_data = result[result['聚类标签'] == i]  print(f"\n群体{i+1}（样本数：{len(cluster\_data)}）：")  print(cluster\_data.iloc[:, :-1].describe().round(2).loc[['mean']])  # 5. 可视化聚类结果（取前2个特征）  plt.figure(figsize=(10, 6))  plt.scatter(X\_reduced[:, 0], X\_reduced[:, 1], c=labels, cmap='viridis', alpha=0.6)  plt.scatter(cluster\_centers[:, 0], cluster\_centers[:, 1], marker='X', s=200, c='red', label='聚类中心')  plt.xlabel(features[reduct[0]])  plt.ylabel(features[reduct[1]])  plt.title('电商用户消费群体聚类')  plt.legend()  plt.show() |