### 4. PCA + SVM案例：信用卡欺诈交易识别

**问题背景**：某银行需构建模型识别信用卡欺诈交易，交易数据含 20 项特征（如交易金额、时段、地点、商户类型等），存在高维冗余（如 “交易时段” 与 “商户类型” 强相关），且欺诈样本占比低（仅 3%）。

**数据**：

* 10000 条交易记录，每条含 20 项特征及标签（1 = 欺诈，0 = 正常），其中欺诈样本 300 条。

**要求**：用 PCA 降维至保留 95% 信息的维度，再用 SVM 分类，对比降维前后的识别准确率（尤其是欺诈样本的召回率），说明降维对模型效率的提升。

### 4. PCA + SVM代码：信用卡欺诈交易识别

|  |
| --- |
| import numpy as np  import pandas as pd  from sklearn.decomposition import PCA  from sklearn.svm import SVC  from sklearn.model\_selection import train\_test\_split  from sklearn.preprocessing import StandardScaler  from sklearn.metrics import classification\_report  # 生成模拟数据（10000条交易，20项特征）  np.random.seed(42)  n\_samples = 10000  n\_features = 20  X = np.random.randn(n\_samples, n\_features)  # 生成欺诈标签（3%欺诈率）  weights = np.random.randn(n\_features)  y = np.where(np.dot(X, weights) + np.random.randn(n\_samples) > 2, 1, 0)  print(f"欺诈样本比例：{sum(y)/n\_samples:.2%}")  # 数据划分与标准化  X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=42)  scaler = StandardScaler()  X\_train\_scaled = scaler.fit\_transform(X\_train)  X\_test\_scaled = scaler.transform(X\_test)  # PCA降维  pca = PCA(n\_components=0.95, random\_state=42)  X\_train\_pca = pca.fit\_transform(X\_train\_scaled)  X\_test\_pca = pca.transform(X\_test\_scaled)  print(f"降维前特征数：{n\_features}，降维后：{pca.n\_components\_}")  # SVM分类与评估  svm = SVC(kernel='rbf', class\_weight='balanced', random\_state=42)  svm.fit(X\_train\_pca, y\_train)  y\_pred = svm.predict(X\_test\_pca)  print("\n降维后分类报告：")  print(classification\_report(y\_test, y\_pred))  # 对比不降维的情况  svm\_raw = SVC(kernel='rbf', class\_weight='balanced', random\_state=42)  svm\_raw.fit(X\_train\_scaled, y\_train)  y\_pred\_raw = svm\_raw.predict(X\_test\_scaled)  print("\n不降维分类报告：")  print(classification\_report(y\_test, y\_pred\_raw)) |