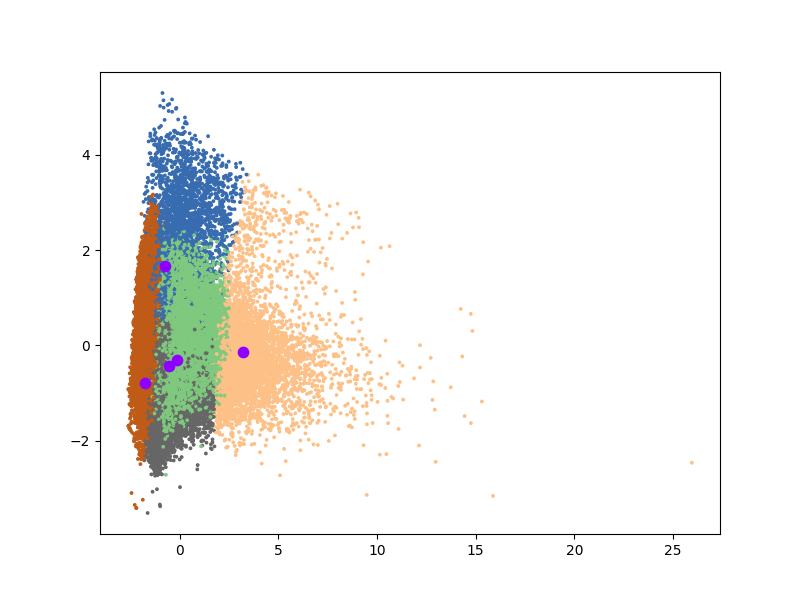
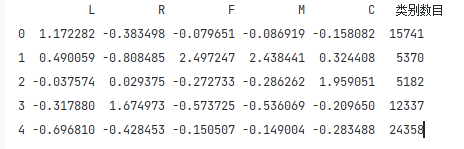


结合这张图，可以看到Cluster1 2 3 4在五个特征上各有突出，说明 K-Means 成功地捕捉到了数据集中的特征分布差异。



聚类可视化PCA



数据汇总结果

import pandas as pd  
from sklearn.cluster import KMeans  
from sklearn.decomposition import PCA  
from sklearn.manifold import TSNE  
import matplotlib.pyplot as plt  
import numpy as np  
from radar\_map import plot  
  
# 导入数据并标准化  
features = pd.read\_excel('air\_features.xlsx', index\_col='ID')  
features\_scaler = (features - features.mean()) / features.std()  
  
# 开始K-Means聚类  
model = KMeans(n\_clusters=5, random\_state=3)  
model.fit(features\_scaler)  
labs = model.labels\_  
centers = model.cluster\_centers\_  
  
# 简单打印结果  
r1 = pd.Series(model.labels\_).value\_counts()  
r2 = pd.DataFrame(model.cluster\_centers\_)  
r = pd.concat([r2, r1], axis=1)  
r.columns = list(features.columns) + ['类别数目']  
print(r)  
  
# 详细输出原始数据及对应的类别  
r = pd.concat([features, pd.Series(model.labels\_, index=features.index)], axis=1)  
r.columns = list(features.columns) + ['聚类类别']  
r.to\_excel('features\_type.xlsx')  
  
  
# 可视化雷达图  
def plot\_kmeans\_radar(kmeans\_model, columns):  
 # 计算雷达图的角度  
 angles = np.linspace(0, 2 \* np.pi, len(columns), endpoint=False).tolist()  
 angles += angles[:1] # 闭合雷达图  
  
 # 特征标签  
 feature = columns.tolist()  
 feature += feature[:1] # 闭合雷达图  
  
 # 绘图  
 fig, ax = plt.subplots(figsize=(6, 6), subplot\_kw=dict(polar=True))  
  
 # 获取聚类中心数据  
 cluster\_centers = kmeans\_model.cluster\_centers\_  
  
 for i, center in enumerate(cluster\_centers):  
 ax.plot(angles, center.tolist() + [center[0]], label=f'Cluster {i + 1}')  
  
 ax.set\_thetagrids(np.degrees(angles), feature)  
 ax.set\_title("K-Means Clustering Radar Plot")  
 ax.legend()  
 plt.show()  
  
  
# 调用雷达图函数  
plot\_kmeans\_radar(kmeans\_model=model, columns=features.columns)  
  
# PCA 降维并绘制聚类散点图  
pca = PCA(n\_components=2)  
pca.fit(features\_scaler)  
data\_pca = pca.transform(features\_scaler)  
data\_pca = pd.DataFrame(data\_pca, columns=['PC1', 'PC2'])  
data\_pca['labels'] = labs  
  
# 聚类中心降维  
pca.fit(centers)  
data\_pca\_centers = pca.transform(centers)  
data\_pca\_centers = pd.DataFrame(data\_pca\_centers, columns=['PC1', 'PC2'])  
  
# 可视化聚类结果  
plt.figure(figsize=(8, 6))  
plt.scatter(data\_pca['PC1'], data\_pca['PC2'], s=3, c=data\_pca['labels'], cmap='Accent')  
plt.scatter(data\_pca\_centers['PC1'], data\_pca\_centers['PC2'], marker='o', s=55, c='#8E00FF')  
plt.show()