实验 9_任务 3_异常检测

任务 3 异常检测

任务描述：针对上述两个任务中的某一个任务的数据，尝试使用某种异常检测方法分析数据的情况，判断数据集合中是否存在异常数据点。

实验过程记录：

LAB9 TASK3

异常检测

调库

```
import pandas as pd
import numpy as np
import math
from sklearn.decomposition import PCA
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
```

导入文件

```
filename = 'E:/大学课程/AI程序设计/实验部分/实验9 聚类-关联-异常/实验课聚
data_origin = pd.read_csv(filename, engine = 'python')
k = 2
threshold = 2.5
iteration = 500
```

```
data_origin_matrix = data_origin.values
label_lst = []
for item in data_origin_matrix:
    if item[0]=='republican':
        label_lst.append(0)#共和党为0
    else:
        label_lst.append(1)#民主党为1
```

把投票结果'n'，'y'，'？'，分别用−1，1，0代替

```
data_product=data_origin.drop('A',axis=1)
data_product_matrix = data_product.values
pri_X = data_product_matrix

lst_extern = []

for item in pri_X:
    lst_essen = []
    for member in item:
        if member == 'n':
            lst_essen.append(-1)
        elif member == 'y':
            lst_essen.append(1)
        else:
            lst_essen.append(0)
    lst_extern.append(lst_essen)

X = np.mat(lst_extern)
```

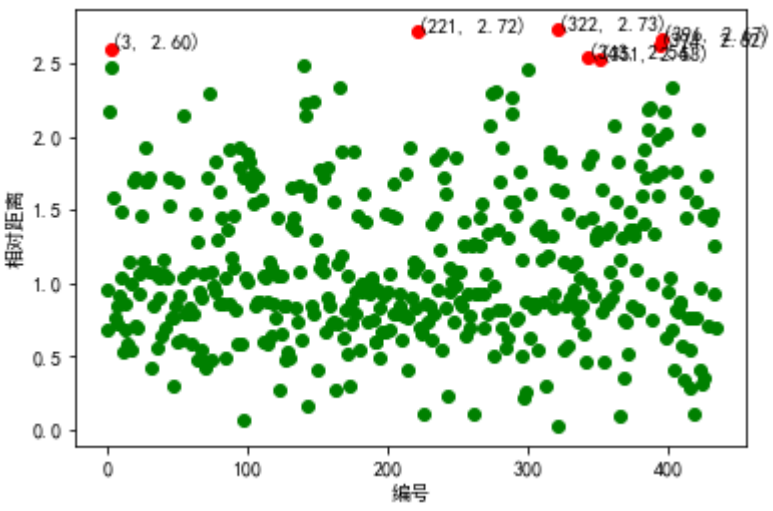
降维为2维

```
pca = PCA(n_components = 2)
reduced_X = pca.fit_transform(X)

X_lst = []
Y_lst = []

for item in reduced_X:
    X_lst.append(item[0])
    Y_lst.append(item[1])
```

做离群点检测————整体检测

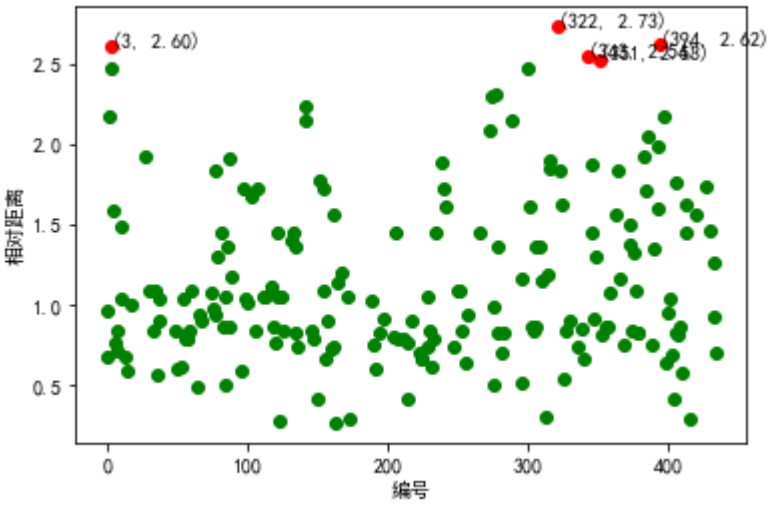
```
lst1 = []
lst2 = []
for item in reduced_X:
    lst1.append(item[0])
    lst2.append(item[1])

data = np.array(reduced_X)
DF = pd.DataFrame(data,index = range(435))

clf = KMeans(n_clusters=2)
model = KMeans(n_clusters = k, max_iter  = iteration)
data_fit = model.fit(DF)
y_pred = clf.fit_predict(reduced_X)

DF_result = pd.concat([DF, pd.Series(model.labels_, index = DF.index)
DF_result.columns = list(DF.columns) + [u'聚类类别'] #重命名表头
```

```
DF_norm = []
for i in range(2):
    norm_tmp = DF_result[[0,1]][DF_result['聚类类别']==i]-model.cluste
    norm_tmp = norm_tmp.apply(np.linalg.norm,axis = 1)
    DF_norm.append(norm_tmp/norm_tmp.median())

DF_norm = pd.concat(DF_norm)

import matplotlib.pyplot as plt
plt.rcParams['font.sans-serif'] = ['SimHei'] #用来正常显示中文标签
plt.rcParams['axes.unicode_minus'] = False #用来正常显示负号
DF_norm[DF_norm <= threshold].plot(style = 'go') #正常点

discrete_points = DF_norm[DF_norm > threshold] #离群点
discrete_points.plot(style = 'ro')


for i in range(len(discrete_points)): #离群点做标记
    id = discrete_points.index[i]
    n = discrete_points.iloc[i]
    plt.annotate('(%s, %0.2f)'%(id, n), xy = (id, n), xytext = (id, n


plt.xlabel('编号')
plt.ylabel('相对距离')
plt.show()
```



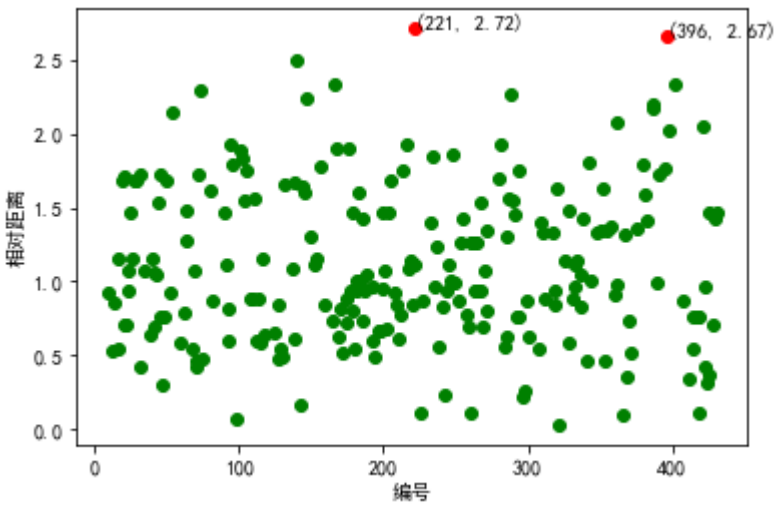
离群点检测————共和党检测

```
DF_norm_0 = []
for i in range(1):
    norm_tmp = DF_result[[0,1]][DF_result['聚类类别']==i]-model.cluster_centers_[i]
    norm_tmp = norm_tmp.apply(np.linalg.norm,axis = 1)
    DF_norm_0.append(norm_tmp/norm_tmp.median())

DF_norm_0 = pd.concat(DF_norm_0)

import matplotlib.pyplot as plt
plt.rcParams['font.sans-serif'] = ['SimHei'] #用来正常显示中文标签
plt.rcParams['axes.unicode_minus'] = False #用来正常显示负号
DF_norm_0[DF_norm_0 <= threshold].plot(style = 'go') #正常点

discrete_points = DF_norm_0[DF_norm_0 > threshold] #离群点
discrete_points.plot(style = 'ro')


for i in range(len(discrete_points)): #离群点做标记
    id = discrete_points.index[i]
    n = discrete_points.iloc[i]
    plt.annotate('(%s, %0.2f)'%(id, n), xy = (id, n), xytext = (id, n))


plt.xlabel('编号')
plt.ylabel('相对距离')
plt.show()
```



离群点检测————民主党检测

```
DF_norm_1 = []
for i in range(1,2):
    norm_tmp = DF_result[[0,1]][DF_result['聚类类别']==i]-model.cluster_centers_[i]
    norm_tmp = norm_tmp.apply(np.linalg.norm,axis = 1)
    DF_norm_1.append(norm_tmp/norm_tmp.median())

DF_norm_1 = pd.concat(DF_norm_1)

import matplotlib.pyplot as plt
plt.rcParams['font.sans-serif'] = ['SimHei'] #用来正常显示中文标签
plt.rcParams['axes.unicode_minus'] = False #用来正常显示负号
DF_norm_1[DF_norm_1 <= threshold].plot(style = 'go') #正常点

discrete_points = DF_norm_1[DF_norm_1 > threshold] #离群点
discrete_points.plot(style = 'ro')


for i in range(len(discrete_points)): #离群点做标记
    id = discrete_points.index[i]
    n = discrete_points.iloc[i]
    plt.annotate('(%s, %0.2f)'%(id, n), xy = (id, n), xytext = (id, n))


plt.xlabel('编号')
plt.ylabel('相对距离')
plt.show()
```



问题 1：

此离散的检测的思路是：

1.进行聚类。选择聚类算法（如 K-Means 算法），将样本集聚 K 簇，并找到各簇的质心。

2.计算各对象到它的最近质心的距离。

3.计算各对象到它的最近质心的相对距离。(相对距离是点到质心的距离与簇中所有点到质心的距离的中位数之比)

4. 如果某对象距离大于阈值，就认为该对象是离散点。【1】

取定阈值为 2.5，可以发现两个簇中均有离散点，但是，离散点数目较少，可以认为不是由于党派的差异，而是由于议员个人倾向的差异。

参考资料：

【1】：https://blog.csdn.net/sinat_25873421/article/details/80727352

提出问题：

1）在处理数据时，我采用了 PCA 降维，一定程度上，把离散的投票结果变成了连续的值，我认为这里缺乏科学性，但从聚类结果显示，似乎并没有产生很大影响。我对这里的原理不太理解，不知是不是因为对 PCA 的理解不到位产生的。