**kmeans聚类算法**

分别基于python和sklearn实现kmeans聚类算法对鸢尾花数据进行聚类（提示：鸢尾花数据集加载 from sklearn.datasets import load\_iris）

**源代码：**

#导入所需模块

import matplotlib.pyplot as plt

import numpy as np

from sklearn.cluster import KMeans

from sklearn.datasets import load\_iris

#导入鸢尾花数据集

iris = load\_iris()

X = iris.data[:]

# print(X)

print(X.shape)

#肘方法看k值

d=[]

for i in range(1,11): #k取值1~11，做kmeans聚类，看不同k值对应的簇内误差平方和

km=KMeans(n\_clusters=i,init='k-means++',n\_init=10,max\_iter=300,random\_state=0)

km.fit(X)

d.append(km.inertia\_) #inertia簇内误差平方和

plt.plot(range(1,11),d,marker='o')

plt.xlabel('number of clusters')

plt.ylabel('distortions')

plt.show()

#建立簇为3的聚类器

estimator = KMeans(n\_clusters=3,random\_state=0)

estimator.fit(X)

#获取标签

label\_pred = estimator.labels\_

print(label\_pred)

x0 = X[label\_pred==0]

x1 = X[label\_pred==1]

x2 = X[label\_pred==2]

#绘制聚类结果

plt.scatter(x0[:,0],x0[:,2],c='red',marker='o',label='label0')

plt.scatter(x1[:,0],x1[:,2],c='green',marker='\*',label='label1')

plt.scatter(x2[:,0],x2[:,2],c='blue',marker='v',label='label2')

centers = estimator.cluster\_centers\_

plt.scatter(centers[:,0],centers[:,2],marker='x',c='black',alpha=1,s=300)

plt.xlabel('petal length')

plt.ylabel('petal width')

plt.legend(loc=2)

plt.show()

**运行（测试）过程及结果：**

