# 任务

根据鸢尾花的特征训练集构建以knn算法为基础的分类模型，将鸢尾花分为三类，

# 功能

输入鸢尾花的四个特征属性，判断输出其所属类别

# 算法

## 一、基本思路

1. 对所有样本进行标准化处理
2. 指定参数K，确定距离度量方式
3. 对于每个测试样本，根据度量方式找到其k个近邻，根据等权投票规则对测试集进行类别划分

## 二、导入数据与数据集预处理

使用sklearn对iris数据进行导入：

# 导入iris数据集

iris = load\_iris() # 加载数据集

features = iris.data # 获取属性数据(各种特征值)

labels = iris.target # 获取类别数据（花的分类）

target\_names = iris.target\_names # 获取数据的分类名称（花的分类名称）

按照8：2的比例随机划分训练集和测试集:

X\_train,X\_test,y\_train,y\_test = train\_test\_split(features,labels,test\_size=0.2, random\_state=1,shuffle=True)

对数据进行标准化预处理：

std = StandardScaler()

X\_train = std.fit\_transform(X\_train)

X\_test = std.fit\_transform(X\_test)

## 三、 建立预测结果集

pred\_list=[]

acc=0

accuracy=0

## 四、knn构建

for i in range(n):

#计算当前 样本 与 训练集 的距离

near\_dis=sess.run(distance,feed\_dict={Xtr:X\_train,Xte:X\_test[i]})

#排序后，按距离远近取前k个值

knn=np.argsort(near\_dis)[:k]

#分三类别计数器，取数量最多的类别为样本类别

label=[0,0,0]

#根据k近邻内的样本标签对测试样本进行分类

for j in knn:

if(Y\_train[j]==0):

label[0]+=1

elif(Y\_train[j]==1):

label[1]+=1

else:

label[2]+=1

#取最大值

final\_label=np.argmax(label)

pred\_list.append(final\_label)

## 五、测试集运行结果

##### 程序运行结果

预测运行结果为

Test 0 Predict\_result: 0 True\_class: 0

Test 1 Predict\_result: 1 True\_class: 1

Test 2 Predict\_result: 1 True\_class: 1

Test 3 Predict\_result: 0 True\_class: 0

Test 4 Predict\_result: 2 True\_class: 2

Test 5 Predict\_result: 1 True\_class: 1

Test 6 Predict\_result: 2 True\_class: 2

Test 7 Predict\_result: 0 True\_class: 0

Test 8 Predict\_result: 0 True\_class: 0

Test 9 Predict\_result: 2 True\_class: 2

Test 10 Predict\_result: 1 True\_class: 1

Test 11 Predict\_result: 0 True\_class: 0

Test 12 Predict\_result: 2 True\_class: 2

Test 13 Predict\_result: 1 True\_class: 1

Test 14 Predict\_result: 1 True\_class: 1

Test 15 Predict\_result: 0 True\_class: 0

Test 16 Predict\_result: 1 True\_class: 1

Test 17 Predict\_result: 1 True\_class: 1

Test 18 Predict\_result: 0 True\_class: 0

Test 19 Predict\_result: 0 True\_class: 0

Test 20 Predict\_result: 1 True\_class: 1

Test 21 Predict\_result: 1 True\_class: 1

Test 22 Predict\_result: 1 True\_class: 1

Test 23 Predict\_result: 0 True\_class: 0

Test 24 Predict\_result: 2 True\_class: 2

Test 25 Predict\_result: 1 True\_class: 1

Test 26 Predict\_result: 0 True\_class: 0

Test 27 Predict\_result: 0 True\_class: 0

Test 28 Predict\_result: 1 True\_class: 1

Test 29 Predict\_result: 2 True\_class: 2

finsh

预测正确率为：

accuracy: 1.0