# K近邻树进行iris数据集的类别预测

针对iris数据集，编写k近邻树进行类别预测

In [ ]:

```
from sklearn.datasets import load_iris
import numpy as np
from sklearn.neighbors import KDTree
```

读取数据

In [ ]:

```
iris = load_iris()
x = iris.data
y= iris.target

random_index=np.random.choice(x.shape[0],x.shape[0],replace=False)
x=x[random_index]
y=y[random_index]
print(x.shape)
print(y[:5])
```

```
(150, 4)
[2 1 2 1 0]
```

In [ ]:

```
#归一化
x_max=np.max(x,axis=0)
x_min=np.min(x,axis=0)
x=(x-x_min)/(x_max-x_min)
print(x[:5,:])
```

```
[[0.94444444 0.75       0.96610169 0.875     ]
 [0.5        0.375      0.62711864 0.54166667]
 [0.55555556 0.33333333 0.69491525 0.58333333]
 [0.41666667 0.25       0.50847458 0.45833333]
 [0.41666667 0.83333333 0.03389831 0.04166667]]
```

分为样本和预测

In [ ]:

```
x_sample=x[:120,:]
y_sample=y[:120]
x_test=x[120:,:]
y_test=y[120:]
```

In [ ]:

```
tree=KDTree(x_sample)
dist,index=tree.query(x_test,k=5)
y_pre=[]
for i in range(len(index)):
    ans=np.argmax(np.bincount(y_sample[index[i]]))
    y_pre.append(ans)
print('预测： ',y_pre)
print('实际： ',y_test)
print('准确率： ',(y_pre==y_test).sum()/len(y_pre))
```

```
预测：  [1, 2, 0, 1, 1, 0, 0, 2, 2, 0, 1, 2, 1, 0, 2, 0, 0, 0, 1, 0, 1, 1, 2, 1, 2,
0, 0, 1, 2, 2]
实际：  [1 2 0 1 1 0 0 2 2 0 1 2 1 0 2 0 0 0 1 0 1 1 2 1 2 0 0 1 2 2]
准确率：  1.0
```