封装机器学习算法--KNN

In [1]:
```
from ML.decision_tree import DecisionTreeClassifier
import numpy as np
```

In [2]:
```
X = np.loadtxt('x.txt')
y = np.loadtxt('y.txt')
```

In [3]:
```
dt_clf = DecisionTreeClassifier()
```

In [4]:
```
dt_clf.fit(X, y)
```

Out[4]:
```
d=2, v=2.45, g=0.5, l=None
```

In [5]:
```
dt_clf.tree_
```

Out[5]:
```
d=2, v=2.45, g=0.5, l=None
```

In [6]:
```
dt_clf.predict(X[:3])
```

Out[6]:
```
array([0., 0., 0.])
```

In [7]:
```
y_predict = dt_clf.predict(X)
```

In [8]:
```
from ML.metrics import accuracy_score
```

In [9]:
```
accuracy_score(y, y_predict)
```

Out[9]:
```
0.98
```

In [10]:
```
dt_clf.accuracy_rate(X, y)    # 调用封装在决策树算法中的求准确率的函数
```

Out[10]:
```
0.98
```

In [11]:
```
dt_clf.show_tree()
```

In [12]:
```
from ML.knn import KNeighborsClassifier
```

In [13]:
```
knn_clf = KNeighborsClassifier()
```

In [14]:
```
knn_clf.fit(X, y)
```

Out[14]:
```
<ML.knn.KNeighborsClassifier at 0x114d72940>
```

In [15]:
```
knn_clf.predict(X[:3])
```

Out[15]:
```
array([0., 0., 0.])
```

In [16]:
```
knn_clf.accuracy_rate(X, y)
```

Out[16]: 0.9666666666666667