```
# 超随机树演示
from sklearn.ensemble import ExtraTreesClassifier

from sklearn.grid_search import RandomizedSearchCV

from sklearn.datasets import make_classification

from sklearn.metrics import classification_report
from sklearn.metrics import accuracy_score

from sklearn.cross_validation import train_test_split
from sklearn.cross_validation import cross_val_score

from operator import itemgetter
import numpy as np

# 生成分类数据集
n_f = 30
inf_f = int(0.6 * n_f)
red_f = int(0.1 * n_f)
rep_f = int(0.1 * n_f)

X,y = make_classification(
    n_samples=500,
    flip_y=0.03,
    n_features=n_f,
    n_informative=inf_f,
    n_redundant=red_f,
    n_repeated=rep_f)

X_train,X_test,y_train,y_test                                    =
train_test_split(X,y,test_size=0.3,random_state=9)

# 构造超随机树模型
et = ExtraTreesClassifier(n_estimators=100)
et.fit(X_train,y_train)

# 分别在训练集测试集上预测
y_train_pred = et.predict(X_train)
train_score = accuracy_score(y_train,y_train_pred)

y_test_pred = et.predict(X_test)
test_score = accuracy_score(y_test,y_test_pred)
```

```
print "Train Accuracy = %0.2f" % train_score
print "Test Accuracy = %0.2f" % test_score
Train Accuracy = 1.00
Test Accuracy = 0.82

# 在测试集上进行 5 折交叉验证
print "Cross validatioan score"
print cross_val_score(et,X_test,y_test,cv=5)

Cross validatioan score
[0.77419355 0.9        0.9        0.83333333 0.72413793]

# 平均值 0.83,比上面不做交叉验证 0.82 略好
np.mean([0.77419355, 0.9, 0.9, 0.83333333, 0.72413793])
0.8263329620000001

# 对超随机树进行随机搜索交叉验证
et = ExtraTreesClassifier()

n_f = X.shape[1]
sqr_nf = int(np.sqrt(n_f))

# 构造 20 个超随机树
n_iter = 20

# 每个超随机树的模型数从 75-200 中随机抽取,纯度度量和最大特征数也随机选取
param = {"n_estimators":np.random.randint(75,200,n_iter),
         "criterion":["gini","entropy"],
         "max_features":[sqr_nf,sqr_nf*2,sqr_nf*3,sqr_nf+10]}

# 构造随机搜索交叉验证:20 个随机森林各进行 5 折交叉验证
grid = RandomizedSearchCV(estimator=et,
                          param_distributions = param,
                          n_iter=n_iter,
                          cv=5,
                          verbose=1,
                          n_jobs=-1,
                          random_state=77)
grid.fit(X_train,y_train)
Fitting 5 folds for each of 20 candidates, totalling 100 fits
[Parallel(n_jobs=-1)]: Done  46 tasks    | elapsed:   22.0s
[Parallel(n_jobs=-1)]: Done 100 out of 100 | elapsed:  39.8s finished

# 按评分排序取前 5
```

```
scores                                                                =
sorted(grid.grid_scores_,key=itemgetter(1),reverse=True)[:5]

for m,score in enumerate(scores):
    print "M%d,Score = %0.3f" % (m+1,score.mean_validation_score)
    print "Param = {0}".format(score.parameters)
```

```
M1,Score = 0.869
Param = {'n_estimators': 103, 'max_features': 15, 'criterion': 'entropy'}
M2,Score = 0.866
Param = {'n_estimators': 82, 'max_features': 15, 'criterion': 'gini'}
M3,Score = 0.863
Param = {'n_estimators': 124, 'max_features': 10, 'criterion': 'gini'}
M4,Score = 0.863
Param = {'n_estimators': 82, 'max_features': 15, 'criterion': 'entropy'}
M5,Score = 0.860
Param = {'n_estimators': 167, 'max_features': 10, 'criterion': 'gini'}
```

```
# 对测试集预测
y_pred = grid.predict(X_test)
print classification_report(y_test,y_pred)
```

```
             precision    recall  f1-score   support

          0       0.77      0.91      0.83        69
          1       0.91      0.77      0.83        81

avg / total       0.85      0.83      0.83       150
```