Sklearn-tree

输入的特征矩阵必须至少是一个二维矩阵
如果数据的确只有一个特征，那必须用reshape(-1,1)来给矩阵增维;
如果数据只有一个特征和一个样本，使用reshape(1,-1)来给你的数据增维

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

from sklearn.preprocessing import LabelEncoder
from sklearn import tree
```

```
filename = 'data中文.csv'
df = pd.DataFrame(pd.read_csv(filename))
le = LabelEncoder()
for col in df.columns:
    df[col] = le.fit_transform(df[col])
print(df)
attr_data=df[['天气','温度','湿度','风速']]
result_mat=df['活动']
attr_names=['天气','温度','湿度','风速']
```

构造决策树

```
clf = tree.DecisionTreeClassifier(criterion='entropy')
clf.fit(attr_data, result_mat)
print(clf)
```

```
DecisionTreeClassifier(ccp_alpha=0.0, class_weight=None, criterion='entropy',
                       max_depth=None, max_features=None, max_leaf_nodes=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, presort='deprecated',
                       random_state=None, splitter='best')
```

Sklearn决策树实例

```
from sklearn.tree import export_text
r = export_text(clf, feature_names=attr_names)
print(r)
```

```
|--- 湿度 <= 0.50
|   |--- 天气 <= 1.50
|   |   |--- class: 1
|   |--- 天气 >  1.50
|   |   |--- 风速 <= 0.50
|   |   |   |--- class: 1
|   |   |--- 风速 >  0.50
|   |   |   |--- class: 0
|--- 湿度 >  0.50
|   |--- 天气 <= 0.50
|   |   |--- class: 0
|   |--- 天气 >  0.50
|   |   |--- 天气 <= 1.50
|   |   |   |--- class: 1
|   |   |--- 天气 >  1.50
|   |   |   |--- 风速 <= 0.50
|   |   |   |   |--- class: 1
|   |   |   |--- 风速 >  0.50
|   |   |   |   |--- class: 0
```

```
clf = tree.DecisionTreeClassifier(max_depth=2)
clf.fit(attr_data[:10],result_mat[:10])
print(clf)
```

```
from sklearn.tree import export_text
r = export_text(clf, feature_names=attr_names)
print(r)
```

```
|--- 天气 <= 0.50
|   |--- 湿度 <= 0.50
|   |   |--- class: 1
|   |--- 湿度 >  0.50
|   |   |--- class: 0
|--- 天气 >  0.50
|   |--- 风速 <= 0.50
|   |   |--- class: 1
|   |--- 风速 >  0.50
|   |   |--- class: 0
```

Sklearn决策树实例

```
pre = clf.predict(attr_data[11:])
print(pre)
import sklearn.metrics as metrics
print(metrics.classification_report(result_mat[11:],pre))
```

```
[1 1 0]
              precision    recall  f1-score   support

           0       1.00      1.00      1.00         1
           1       1.00      1.00      1.00         2

    accuracy                           1.00         3
   macro avg       1.00      1.00      1.00         3
weighted avg       1.00      1.00      1.00         3
```

```
score=decision_tree.score(X_test,y_test)
score#返回的是平均准确度，即accuracy
```

```
0.9777777777777777
```

课堂练习

导入鸢尾花数据集
划分训练集和测试集
绘制决策树模型（CART）
评估模型好坏

```
|--- petal width (cm) <= 0.80
|    |--- class: 0
|--- petal width (cm) >  0.80
|    |--- petal width (cm) <= 1.75
|    |    |--- class: 1
|    |--- petal width (cm) >  1.75
|    |    |--- class: 2
```

【模型预测值】

```
[2 1 0 1 2 1 2 2 2 1 1 0 1 0 0 1 0 1 0 1 2 0 0 0 1 0 2 2 0 0 0 0 0 1 1 2 0
0 1 0 2 1 1 0 0]
```

	precision	recall	f1-score	support
0	1.00	1.00	1.00	20
1	0.80	1.00	0.89	12
2	1.00	0.77	0.87	13
accuracy			0.93	45
macro avg	0.93	0.92	0.92	45
weighted avg	0.95	0.93	0.93	45