

## code\6-2\_cart\_model.py

```
1 #-*- coding: utf-8 -*-
2 #构建并测试CART决策树模型
3
4 import pandas as pd #导入数据分析库
5 from random import shuffle #导入随机函数shuffle, 用来打乱数据
6 import matplotlib.pyplot as plt
7
8 datafile = './data/model.xls' #数据名
9 data = pd.read_excel(datafile) #读取数据, 数据的前三列是特征, 第四列是标签
10 data = data.to_numpy()
11
12 shuffle(data) #随机打乱数据
13
14 p = 0.8 #设置训练数据比例
15 train = data[:int(len(data)*p),:] #前80%为训练集
16 test = data[int(len(data)*p):,:] #后20%为测试集
17
18
19 #构建CART决策树模型
20 from sklearn.tree import DecisionTreeClassifier #导入决策树模型
21
22 treefile = './tmp/tree.pkl' #模型输出名字
23 tree = DecisionTreeClassifier() #建立决策树模型
24 tree.fit(train[:, :3], train[:, 3]) #训练
25
26
27
28 from cm_plot import * #导入自行编写的混淆矩阵可视化函数
29 cm_plot(test[:, 3], tree.predict(test[:, :3])).show() #显示混淆矩阵可视化结果
30 #注意到Scikit-Learn使用predict方法直接给出预测结果。
31
32
33 from sklearn.metrics import roc_curve #导入ROC曲线函数
34
35 fpr, tpr, thresholds = roc_curve(test[:, 3], tree.predict_proba(test[:, :3])[:, 1], pos_label=1)
36 plt.plot(fpr, tpr, linewidth=2, label = 'ROC of CART', color = 'green') #作出ROC曲线
37 plt.xlabel('False Positive Rate') #坐标轴标签
38 plt.ylabel('True Positive Rate') #坐标轴标签
39 plt.ylim(0,1.05) #边界范围
40 plt.xlim(0,1.05) #边界范围
41 plt.legend(loc=4) #图例
42 plt.show() #显示作图结果
```