

8.3.2 决策树的实现

2023 年 9 月 8 日

决策树的 Python 实现

决策树的 Python 实现主要使用 sklearn 机器学习库 tree 模块下的 DecisionTreeClassifier 类。

```
class sklearn.tree.DecisionTreeClassifier(*, criterion='gini', splitter='best', max_depth=None
```

类别	名称	含义
参数	criterion	默认“gini”，可选“entropy”，该参数用于划分的效果。“gini”表示基尼纯度，“熵”表示信息增益。
	splitter	默认“best”，可选“random”，“best”为最佳分割，“random”为最佳随机分割。
	max_dept	树的最大深度。如果为 None，则节点将被扩展，直到所有叶子都是纯的或直到所有叶子包含小于 min_samples_split 样本。
	min_samples_split	分隔内部节点所需的最小样本数
属性	class_	类别标签
	tree_	树对象
方法	fit(x,y)	根据训练集简历决策树分类器
	predict(X)	预测 X 的类别
	score(X,y)	返回给定 X 和 y 的准确率
	get_params()	获得模型的参数

模型输入 X 和 y

上述回归模型中使用 X 和 y 的数据结构如下， X 包含多个样本，以及每个样本的属性，也就是自变量，和 X 的每个样本对应的就是我们的预测目标 y ，也就是因变量。

在实际编程中，一般使用 `pd.DataFrame` 来表示 X 和 y 。

	属性1	属性2	属性3
样本1			
样本2			
样本3			
...			

	类别
样本1	
样本2	
样本3	
...	

实现流程

针对于决策树，其一般化的流程如下：



1) 创建学习器，也就是初始化决策树分类模型

```
from sklearn import tree
tree_clf = tree.DecisionTreeClassifier
```

也可以这样表述：

```
from sklearn.tree import DecisionTreeClassifier
tree_clf = DecisionTreeClassifier()
```

2) 训练模型

```
model.fit(X, y)
```

3) 生成预测结果

```
predicted_y = model.predict(X)
```

4) 计算模型预测精度

```
precision = model.score(X, y)
```

查准率、查全率和 F1 值:

```
from sklearn import metrics
precision = metrics.precision_score(y, predicted_y)
recall = metrics.recall_score(y, predicted_y)
f1_score = metrics.f1_score(y, predicted_y)
```

5) 绘制决策树

```
from sklearn.tree import plot_tree
plt.figure()
plot_tree(tree_clf)
plt.show()
```

参考

1. sklearn 官网: [链接](#)