决策树算法

1.介绍

决策树是常用的机器学习算法之一，决策树模型的决策过程非常类似人类做判断的过程，比较好理解。

sklearn 中的决策树包括

- `entropy`：表示使用 ID3 算法（信息增益）构造决策树。
- `gini`：表示使用CART 算法（基尼系数）构造决策树，为默认值。

sklearn 库的tree模块实现了两种决策树：

- `sklearn.tree.DecisionTreeClassifier` 类：分类树的实现。
- `sklearn.tree.DecisionTreeRegressor` 类：回归树的实现。

分类树用于预测**离散型**数值，回归树用于预测**连续性**数值。（sklearn 只实现了**预剪枝**，没有实现**后剪枝**。）

2.实战之鸢尾花数据集构造分类树

2.1鸢尾花数据集

鸢尾花数据集目的是通过花瓣的长度和宽度，及花萼的长度和宽度，预测出花的品种。包含150条数据，将鸢尾花分成了三类(每类是50条数据)，分别是：

- `setosa`，用数字`0`表示。
- `versicolor`，用数字`1`表示。
- `virginica`，用数字`2`表示。

2.2代码

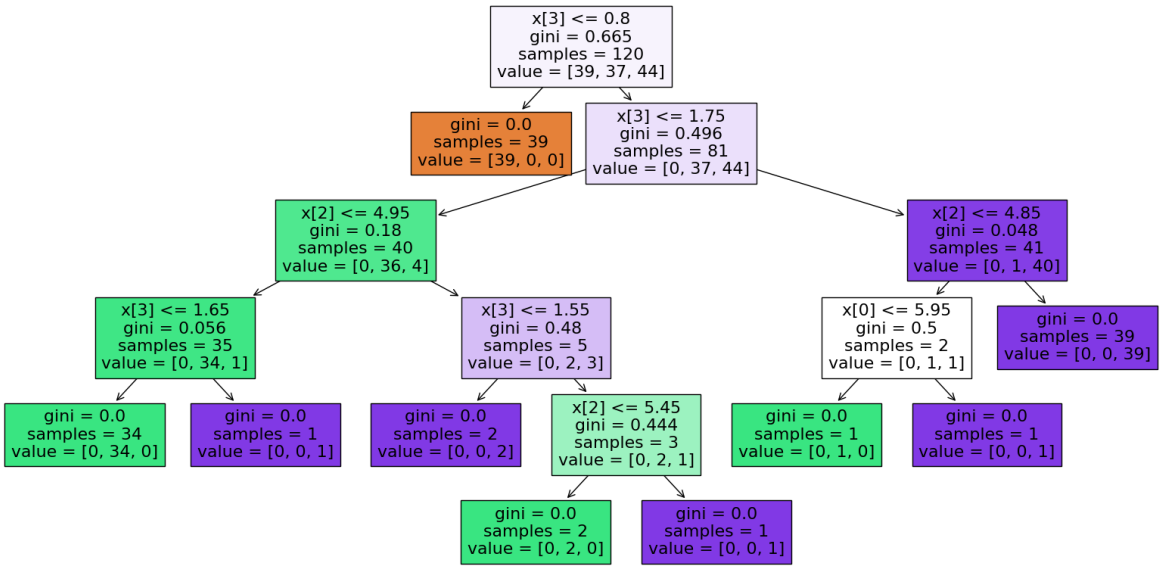
```
from sklearn.tree import DecisionTreeClassifier
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
iris = load_iris()       # 准备数据集
features = iris.data     # 获取特征集
labels = iris.target     # 获取目标集
train_features, test_features, train_labels, test_labels =
train_test_split(features, labels, test_size=0.2, random_state=0)
# 用CART 算法构建分类树（你也可以使用ID3 算法构建）
clf = DecisionTreeClassifier(criterion='gini')

# 用训练集拟合构造CART分类树
clf = clf.fit(train_features, train_labels)
test_predict = clf.predict(test_features)
score = accuracy_score(test_labels, test_predict)
score2 = clf.score(test_features, test_labels)
print(score, score2)

# 结果是：1.0 1.0 说明效果很好
```

```
# 绘制决策树
import matplotlib.pyplot as plt
from sklearn.tree import plot_tree

# 假设 clf 是您的决策树模型
# 绘制决策树
plt.figure(figsize=(20,10))  # 设置图形的大小
plot_tree(clf, filled=True)
```

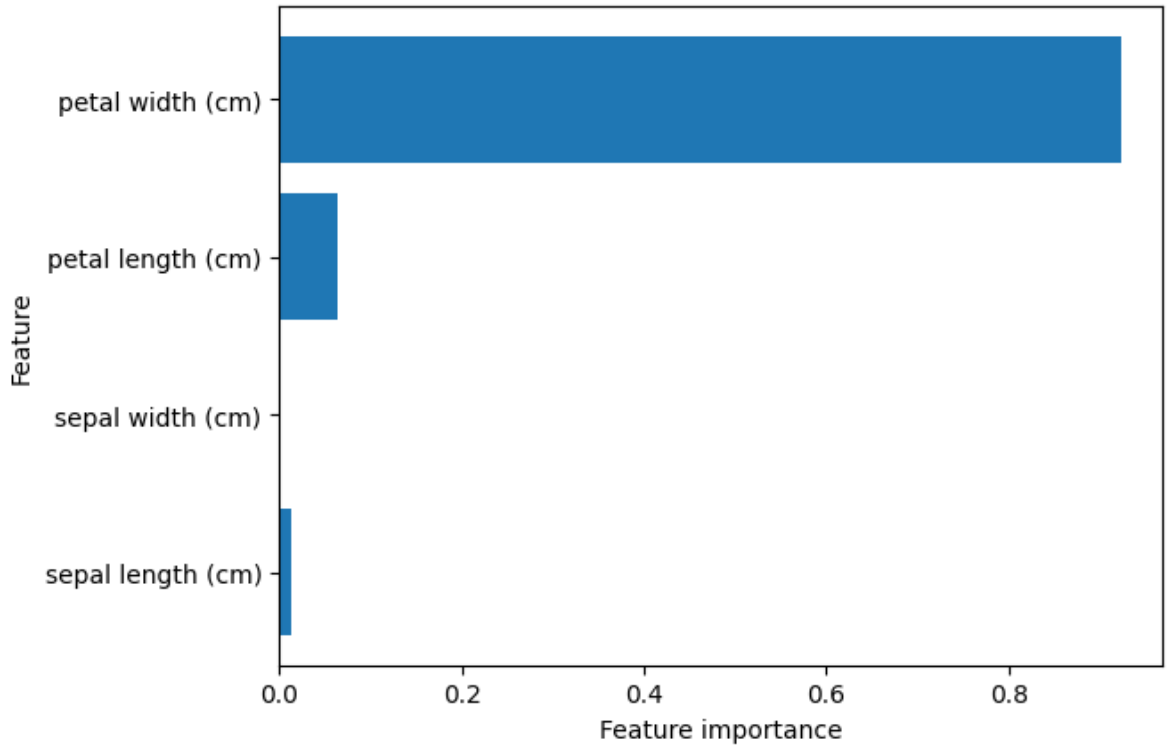



```
# 特征重要性排行
import matplotlib.pyplot as plt
import numpy as np

# mode 是我们训练出的模型，即决策树对象
# data 是原始数据集
def plot_feature_importances(model, data):
    n_features = data.data.shape[1]
    plt.barh(range(n_features), model.feature_importances_, align='center')
    plt.yticks(np.arange(n_features), data.feature_names)
    plt.xlabel("Feature importance")
    plt.ylabel("Feature")

    plt.show()

plot_feature_importances(clf, iris)
```



3.实战之波士顿房价预测回归模型

在波士顿房价数据集中，有很多特征：

首先，我们认为房价是有很多因素影响的，在这个数据集中，影响房价的因素有13 个：

1. **"CRIM"，人均犯罪率。**
2. **"ZN"，住宅用地占比。**
3. **"INDUS"，非商业用地占比。**
4. **"CHAS"，查尔斯河虚拟变量，用于回归分析。**
5. **"NOX"，环保指数。**
6. **"RM"，每个住宅的房间数。**
7. **"AGE"，1940 年之前建成的房屋比例。**
8. **"DIS"，距离五个波士顿就业中心的加权距离。**
9. **"RAD"，距离高速公路的便利指数。**
10. **"TAX"，每一万美元的不动产税率。**
11. **"PTRATIO"，城镇中教师学生比例。**
12. **"B"，城镇中黑人比例。**
13. **"LSTAT"，地区有多少百分比的房东属于是低收入阶层。**

数据中的最后一列的数据是房价：

1. "MEDV"，自住房屋房价的中位数。

因为房价是一个连续值，而不是离散值，所以需要构建一棵**回归树。**

代码：

```
from sklearn.tree import DecisionTreeRegressor
from sklearn.model_selection import train_test_split
from sklearn.metrics import r2_score, mean_absolute_error, mean_squared_error
import pandas as pd
from sklearn.datasets import fetch_openml
from sklearn.model_selection import train_test_split
```

```
data_x, data_y = fetch_openml(name="boston", version=1, as_frame=True,
return_X_y=True, parser="pandas")


train_x, test_x, train_y, test_y = train_test_split(data_x, data_y,
test_size=0.2, random_state=1001)

# 创建CART回归树
dtr = DecisionTreeRegressor()

# 拟合构造CART回归树
dtr.fit(train_x, train_y)

# 预测测试集中的房价
predict_price = dtr.predict(test_x)

# 测试集的结果评价
print('回归树准确率:', dtr.score(test_x, test_y))
print('回归树r2_score:', r2_score(test_y, predict_price))
print('回归树二乘偏差均值:', mean_squared_error(test_y, predict_price))
print('回归树绝对值偏差均值:', mean_absolute_error(test_y, predict_price))

# 回归树准确率: 0.7032769036733362
# 回归树r2_score: 0.7032769036733362
# 回归树二乘偏差均值: 29.91343137254902
# 回归树绝对值偏差均值: 3.2754901960784317
```

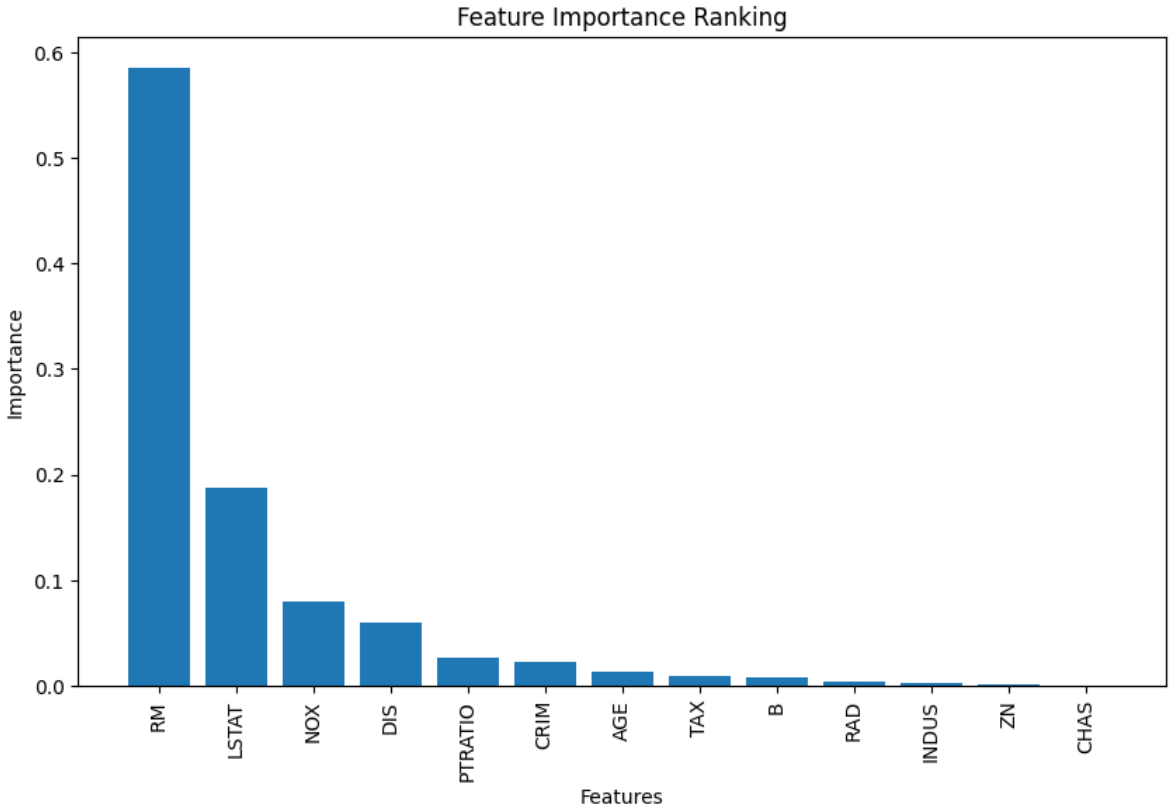
```
# 重要性排行
import matplotlib.pyplot as plt

# 提取特征重要性
importance = dtr.feature_importances_

# 将特征重要性与特征名称对应起来
feature_names = list(train_x.columns)

# 将特征重要性进行排序
sorted_indices = importance.argsort()[::-1]
sorted_importance = importance[sorted_indices]
sorted_feature_names = [feature_names[i] for i in sorted_indices]

# 绘制特征重要性排行图
plt.figure(figsize=(10, 6))
plt.bar(range(len(importance)), sorted_importance,
tick_label=sorted_feature_names)
plt.xticks(rotation=90)
plt.xlabel('Features')
plt.ylabel('Importance')
plt.title('Feature Importance Ranking')
```


Feature Importance Ranking
Importance
0.0
0.1
0.2
0.3
0.4
0.5
0.6
RM
LSTAT
NOX
DIS
PTRATIO
CRIM
AGE
TAX
B
RAD
INDUS
ZN
CHAS
Features