姓名和学号：李胜志 2210180232

章节名称：回归决策树的基本原理

知识目标：掌握决策树用于回归任务的基本原理；掌握构建回归决策树的流程。

能力目标：能够使用决策算法训练模型，构造决策树。

素质目标：关注国家资讯，增强民族意识，培养爱国主义精神；理解决策树算法的基本原理，培养勇为人先的精神。

知识重点：使用决策算法训练模型，构造决策树。

知识难点：选择最佳特征和分割点。

回归决策树的基本原理：

回归决策树是一种有监督的机器学习算法，主要用于预测连续数值型的目标变量。其与分类决策树类似，回归决策树也是由节点组成，包括根节点、内部节点和叶节点。每个内部节点代表一个特征的测试，叶节点则代表预测的输出值，通常是某个连续变量的值。对于新的待测样本，只要按照特征值将其归到某个单元，即可得到对应的输出值。

构建回归决策树的流程：

1. 选择最优分割点：在构建回归树时，找到一个特征和该特征的某个值，使得分割后的子节点的数据方差最小。
2. 分割数据集：根据选定的特征和分割点，将数据集分成两部分。
3. 递归构建：对分割后的每个子集重复上述过程，直到满足停止条件（例如，达到最大深度、节点的数据量小于某个阈值或方差减少量小于某个阈值）。

例如：回归决策树测试数据集（表 1），使用决策树算法训练模型，可构造如图 1所示的决策树。

表 1 回归决策树测试数据集

|  |  |  |  |
| --- | --- | --- | --- |
| x | y | x | y |
| 1 | 4 | 2 | 8 |
| 3 | 9 | 5 | 10 |
| 7 | 19 |  |  |

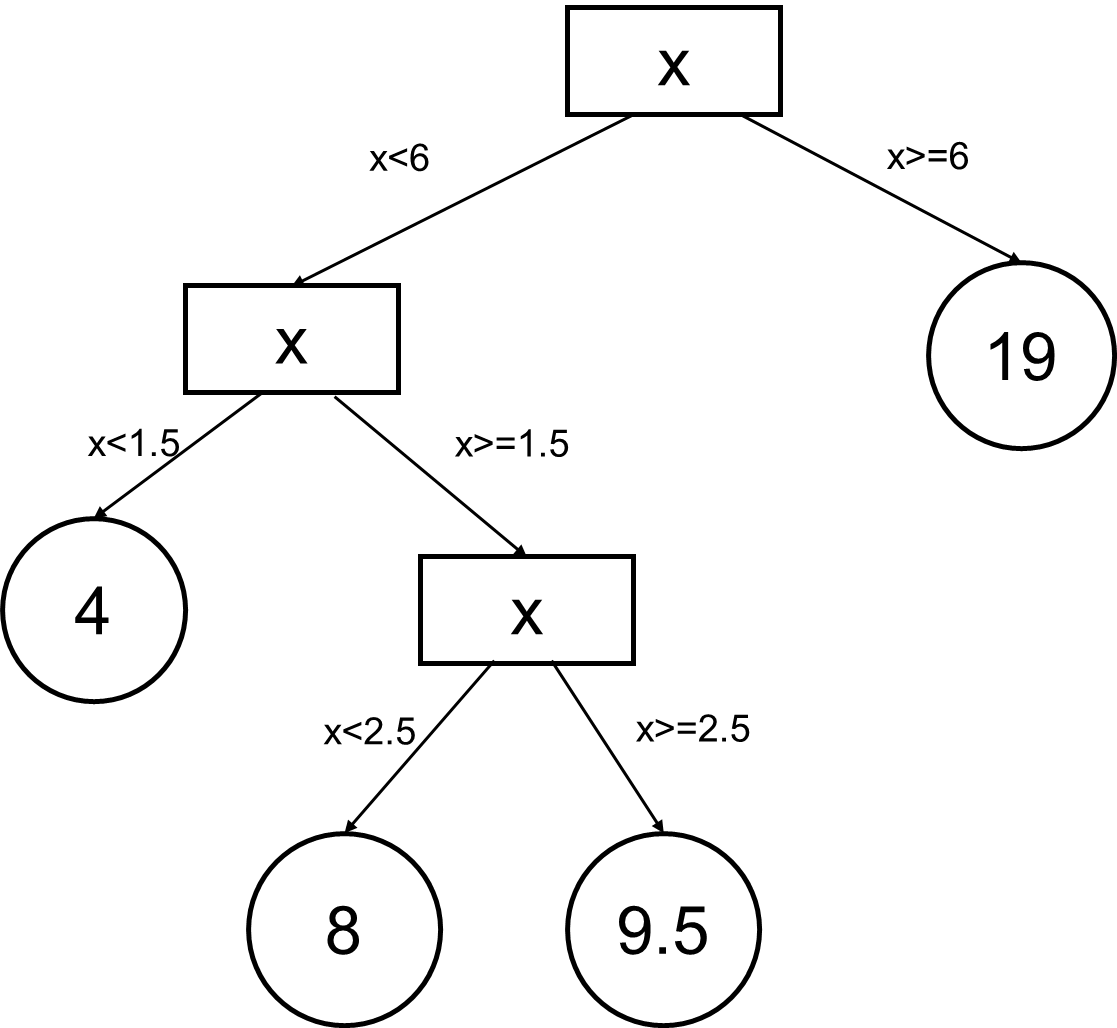


图 1 回归决策树

回归任务中，数据集的特征变量一般是通过统计学中的某种方法计算出来的，这些特征变量的值通常被表示为某个特定点的平均值或中位数。在决策树中，每个节点都代表一个特征变量的值，这个值被称为节点的“特征属性值”。将特征属性值进行排序，然后取相邻两点的中间值作为该节点的输出值，这就是决策树的分裂过程。根据图 1中的回归决策树可以将特征空间划分为多个单元，每个单元都有一个特点的输出值，如图 3所示。

from sklearn.tree import DecisionTreeRegressor

import matplotlib.pyplot as plt

import pandas as pd

import numpy as np

# 输入测试集

mydata = {'x': [1,3,7,2,5], 'y': [4,9,19,8,10]}

mydata\_frame = pd.DataFrame(mydata)

x = mydata\_frame['x'].values.reshape(-1, 1)

y = mydata\_frame['y']

depth = np.arange(1, 15)

scores = []

for i in depth:

model = DecisionTreeRegressor(max\_depth=i)

model.fit(x, y)

score = model.score(x, y)

scores.append(score)

plt.plot(depth, scores, 'o-')

plt.rcParams['font.sans-serif'] = 'Microsoft Yahei'

plt.xlabel('决策树的深度')

plt.ylabel('预测准确率')

plt.show()

运行结果：

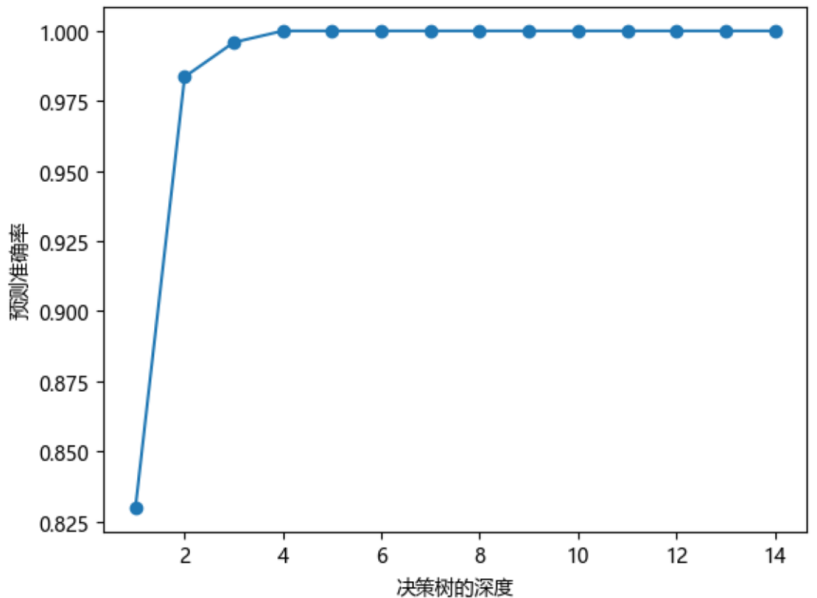


图 2决策树深度与模型准确率关系

# 构建决策树深度为4的模型

model = DecisionTreeRegressor(max\_depth=3)

model.fit(x, y)

plt.scatter(x, y) # 画出散点图

x\_test = np.arange(0, 10.0, 0.01).reshape(-1, 1)

plt.plot(x\_test, model.predict(x\_test))

plt.rcParams['font.sans-serif'] = 'Microsoft Yahei'

plt.xlabel('x')

plt.ylabel('y')

# print(model.score(x, y))

运行结果：

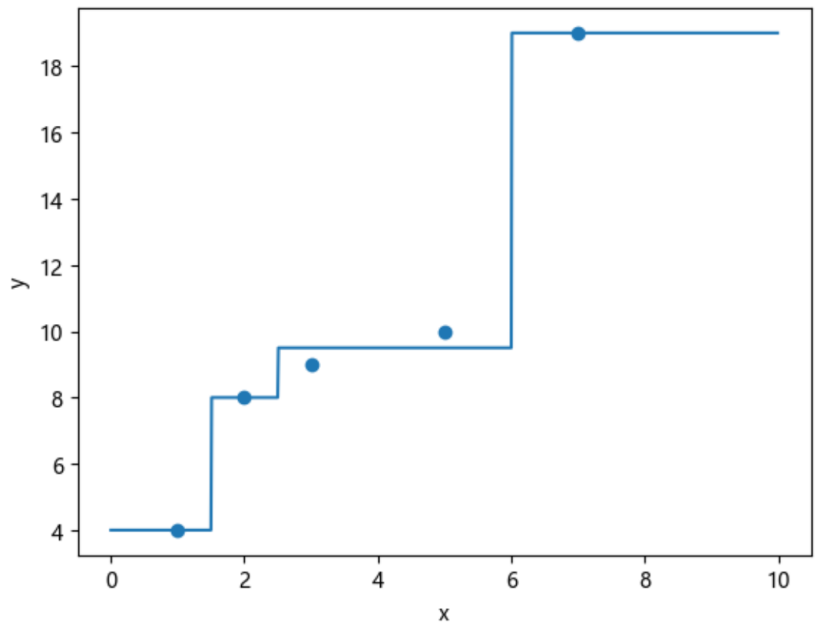


图 3回归决策树划分特征空间

当新的待测样本到来时，只需要将其特征值与这些单元进行比较，就可以得到相应的输出值。例如，新样本（x=4）的输出值为9.5。