姓名和学号：李胜志 2210180232

章节名称：P159项目实训

知识目标：掌握机器学习常用库；

能力目标：能够使用决策算法训练模型，构建决策树模型。

素质目标：养成分析问题、事前规划的良好习惯。

知识重点：使用决策算法构建决策树模型并对模型进行评估；计算决策树的最佳深度。

知识难点：计算决策树的最佳深度。

==========================================================

实训目的：掌握波士顿房价数据集的导入方式；掌握回归决策树模型的训练方法。

实训内容：使用波士顿房价数据集训练回归决策树模型，并对模型进行评估。

实训步骤：

1. 启动Jupyter Notebook，以Python3工作方式创建Jupyter Notebook文档，并重命名为“item6-sx.inpynb”。
2. 数据准备。
3. 导入Pandas和NumPy库。

import pandas as pd

import numpy as np

1. 使用Pandas读取波士顿房价数据集，数据集网址为http://lib.stat.cmu.edu/datasets/boston，也可使用本书提供的配套素材“item6/item6-sx-data.txt”。

data\_url="http://lib.stat.cmu.edu/datasets/boston"

1. 分别获取数据集的特征变量与标签值。

raw\_df=pd.read\_csv(data\_url,sep="\s+",skiprows=22,header=None)

data=np.hstack([raw\_df.values[::2,:],raw\_df.values[1::2,:2]])

target=raw\_df.values[1::2,2]

1. 将处理完成的特征变量和标签分别存放于数组x和y中。

x,y=data,target

1. 输出x和y的值，查看特征变量与标签值。

print('数据集特征变量：\n', x)

print('数据集标签值：\n', y)

运行结果：

数据集特征变量：

[[6.3200e-03 1.8000e+01 2.3100e+00 ... 1.5300e+01 3.9690e+02 4.9800e+00]

[2.7310e-02 0.0000e+00 7.0700e+00 ... 1.7800e+01 3.9690e+02 9.1400e+00]

[2.7290e-02 0.0000e+00 7.0700e+00 ... 1.7800e+01 3.9283e+02 4.0300e+00]

...

[6.0760e-02 0.0000e+00 1.1930e+01 ... 2.1000e+01 3.9690e+02 5.6400e+00]

[1.0959e-01 0.0000e+00 1.1930e+01 ... 2.1000e+01 3.9345e+02 6.4800e+00]

[4.7410e-02 0.0000e+00 1.1930e+01 ... 2.1000e+01 3.9690e+02 7.8800e+00]]

数据集标签值：

[24. 21.6 34.7 33.4 36.2 28.7 22.9 27.1 16.5 18.9 15. 18.9 21.7 20.4

18.2 19.9 23.1 17.5 20.2 18.2 13.6 19.6 15.2 14.5 15.6 13.9 16.6 14.8

18.4 21. 12.7 14.5 13.2 13.1 13.5 18.9 20. 21. 24.7 30.8 34.9 26.6

25.3 24.7 21.2 19.3 20. 16.6 14.4 19.4 19.7 20.5 25. 23.4 18.9 35.4

24.7 31.6 23.3 19.6 18.7 16. 22.2 25. 33. 23.5 19.4 22. 17.4 20.9

24.2 21.7 22.8 23.4 24.1 21.4 20. 20.8 21.2 20.3 28. 23.9 24.8 22.9

23.9 26.6 22.5 22.2 23.6 28.7 22.6 22. 22.9 25. 20.6 28.4 21.4 38.7

43.8 33.2 27.5 26.5 18.6 19.3 20.1 19.5 19.5 20.4 19.8 19.4 21.7 22.8

18.8 18.7 18.5 18.3 21.2 19.2 20.4 19.3 22. 20.3 20.5 17.3 18.8 21.4

15.7 16.2 18. 14.3 19.2 19.6 23. 18.4 15.6 18.1 17.4 17.1 13.3 17.8

14. 14.4 13.4 15.6 11.8 13.8 15.6 14.6 17.8 15.4 21.5 19.6 15.3 19.4

17. 15.6 13.1 41.3 24.3 23.3 27. 50. 50. 50. 22.7 25. 50. 23.8

23.8 22.3 17.4 19.1 23.1 23.6 22.6 29.4 23.2 24.6 29.9 37.2 39.8 36.2

37.9 32.5 26.4 29.6 50. 32. 29.8 34.9 37. 30.5 36.4 31.1 29.1 50.

33.3 30.3 34.6 34.9 32.9 24.1 42.3 48.5 50. 22.6 24.4 22.5 24.4 20.

21.7 19.3 22.4 28.1 23.7 25. 23.3 28.7 21.5 23. 26.7 21.7 27.5 30.1

44.8 50. 37.6 31.6 46.7 31.5 24.3 31.7 41.7 48.3 29. 24. 25.1 31.5

23.7 23.3 22. 20.1 22.2 23.7 17.6 18.5 24.3 20.5 24.5 26.2 24.4 24.8

29.6 42.8 21.9 20.9 44. 50. 36. 30.1 33.8 43.1 48.8 31. 36.5 22.8

30.7 50. 43.5 20.7 21.1 25.2 24.4 35.2 32.4 32. 33.2 33.1 29.1 35.1

45.4 35.4 46. 50. 32.2 22. 20.1 23.2 22.3 24.8 28.5 37.3 27.9 23.9

21.7 28.6 27.1 20.3 22.5 29. 24.8 22. 26.4 33.1 36.1 28.4 33.4 28.2

22.8 20.3 16.1 22.1 19.4 21.6 23.8 16.2 17.8 19.8 23.1 21. 23.8 23.1

20.4 18.5 25. 24.6 23. 22.2 19.3 22.6 19.8 17.1 19.4 22.2 20.7 21.1

19.5 18.5 20.6 19. 18.7 32.7 16.5 23.9 31.2 17.5 17.2 23.1 24.5 26.6

22.9 24.1 18.6 30.1 18.2 20.6 17.8 21.7 22.7 22.6 25. 19.9 20.8 16.8

21.9 27.5 21.9 23.1 50. 50. 50. 50. 50. 13.8 13.8 15. 13.9 13.3

13.1 10.2 10.4 10.9 11.3 12.3 8.8 7.2 10.5 7.4 10.2 11.5 15.1 23.2

9.7 13.8 12.7 13.1 12.5 8.5 5. 6.3 5.6 7.2 12.1 8.3 8.5 5.

11.9 27.9 17.2 27.5 15. 17.2 17.9 16.3 7. 7.2 7.5 10.4 8.8 8.4

16.7 14.2 20.8 13.4 11.7 8.3 10.2 10.9 11. 9.5 14.5 14.1 16.1 14.3

11.7 13.4 9.6 8.7 8.4 12.8 10.5 17.1 18.4 15.4 10.8 11.8 14.9 12.6

14.1 13. 13.4 15.2 16.1 17.8 14.9 14.1 12.7 13.5 14.9 20. 16.4 17.7

19.5 20.2 21.4 19.9 19. 19.1 19.1 20.1 19.9 19.6 23.2 29.8 13.8 13.3

16.7 12. 14.6 21.4 23. 23.7 25. 21.8 20.6 21.2 19.1 20.6 15.2 7.

8.1 13.6 20.1 21.8 24.5 23.1 19.7 18.3 21.2 17.5 16.8 22.4 20.6 23.9

22. 11.9]

1. 确定最佳决策树深度值。
2. 使用train\_test\_split()函数分割数据集，要求测试集的样本数量为50个。

from sklearn.model\_selection import train\_test\_split

x\_train,x\_test,y\_train,y\_test=train\_test\_split(x,y,random\_state=1,test\_size=50)

1. 导入回归决策树模块。

from sklearn.tree import DecisionTreeRegressor

1. 设置决策树深度值为1~30（不包含30）。
2. 当决策树深度值为1~30时，分别构建决策树模型。
3. 计算每个决策树模型的预测误差率。

depth=np.arange(1,30)

err\_list=[]

for i in depth:

model=DecisionTreeRegressor(max\_depth=i)

model.fit(x\_train,y\_train)

r2=model.score(x\_test,y\_test)

err=1-r2

err\_list.append(err)

1. 导入Matplotlib库，绘制图像，确定决策树的最佳深度值。图像的横轴为决策树深度值，纵轴为预测误差率。

import matplotlib.pyplot as plt

plt.plot(depth,err\_list,'ro-')

plt.rcParams['font.sans-serif']='Simhei'

plt.xlabel('决策树深度')

plt.ylabel('预测误差率')

plt.show()

运行结果：

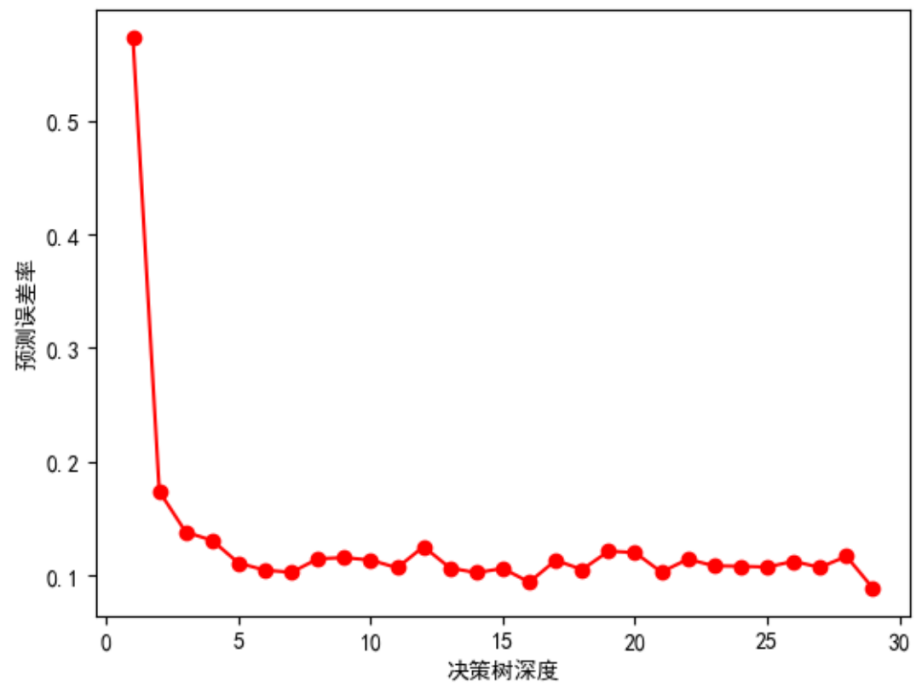


图 1决策树深度

1. 训练与评估模型。
2. 选择合适的决策树深度值，构建决策树模型。
3. 使用fit()函数训练模型。
4. 对训练完成的模型进行评估，输出预测准确率。

由图 1可知，当决策树深度为5时，模型预测

model=DecisionTreeRegressor(max\_depth=5)

model.fit(x\_train,y\_train)

scores=model.score(x\_test,y\_test)

print(scores)

运行结果：

0.8891463100747868