姓名和学号：李胜志 2210180232

章节名称：波士顿房价线性回归预测

知识目标：掌握线性回归方程的参数求解方法；掌握岭回归与套索回归的基本原理与参数调节方法。

能力目标：能够使用最小二乘法、岭回归和套索回归来构建线性回归模型；能够理解并应用Lasso、Ridge等正则化技术来防止过拟合。

素质目标：学习基础知识，提高选择合适方法解决不同问题的能力；养成分析问题、事先做好准本的良好习惯。

知识重点：基于最小二乘法、岭回归和套索回归创建模型、调整alpha参数值并计算模型预测准确率。

知识难点：正则化技术：L1正则化、L2正则化;均方误差、均方根误差、预测准确率（R²分数）等。

项目名称：波士顿房价线性回归预测

项目目标：使用线性回归、岭回归和套索回归三种算法训练模型，对波士顿数据集进行预测，比较三种算法的效果

数据集简介：

在scikit-learn库中，波士顿房价数据集（Boston Housing Dataset）是一个常用的回归分析数据集。这个数据集包含了波士顿地区房价的信息，以及与房价相关的多个特征。

波士顿房价数据集的特征包括：

1. CRIM：城镇人均犯罪率。
2. ZN：住宅用地超过25000平方英尺的比例。
3. INDUS：城镇非零售商用土地的比例。
4. CHAS：查尔斯河虚拟变量（如果边界是河流，则为1；否则为0）。
5. NOX：一氧化氮浓度。
6. RM：住宅平均房间数。
7. AGE：1940年之前建造的自用房屋比例。
8. DIS：到波士顿五个就业中心的加权距离。
9. RAD：辐射性公路的接近指数。
10. TAX：每10000美元的全额财产税税率。
11. PTRATIO：城镇师生比例。
12. B：城镇黑人比例。
13. LSTAT：人口中地位较低人群的比例。

实验步骤：

1. 数据处理：

# 导入相关的库

from sklearn.linear\_model import LinearRegression, Ridge, Lasso

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

import pandas as pd

import numpy as np

#读取数据并将数据集进行分离，拆分为特征变量（data）与标签（target）

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

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]

#将特征变量（data）与标签（target）分别赋值给x和y

x , y = data, target

# 划分训练集和测试集,将30%划分为测试集

x\_train, x\_test, y\_train, y\_test = train\_test\_split(x, y, test\_size=0.3)

1. 创建模型、训练模型和评估：

# 创建模型

lr = LinearRegression()

rd = Ridge()

ls = Lasso()

models = [lr, rd, ls]

names = ["LinearRegression", "Ridge", "Lasso"]

# 分别训练模型并进行回归，计算准确率

for model,name in zip(models,names):

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

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

print("%s模型的准确率是: %.5f" % (name, score))

输出结果：

LinearRegression模型的准确率是: 0.66347

Ridge模型的准确率是: 0.65936

Lasso模型的准确率是: 0.64705

1. 显示回归效果
2. 将alpha的值分别设置为0.0001，0.0005，0.001，0.005，0.01，0.05，0.1，0.5，1，5，10，50
3. 分别测试每个模型在每个alpha值下的准确率

# 定义一个列表用与设置alpha的值

alphas = [0.00001,0.00005,0.0001,0.0005,0.001,0.005,0.01,0.05,0.1,0.5,1,5,10,50]

scores = [] # 定义一个列表用于接收模型的准确率

for index,model in enumerate(models): # 使用enumerate枚举遍历models，得到模型对应的下标和模型

scores.append([])

for alpha in alphas:

model.alpha = alpha # 将模型的alpha值设置成alphas列表里的值

model.fit(x\_train, y\_train) # 训练模型

scores[index].append(model.score(x\_test, y\_test)) # 将准确率存到对应模型里

1. 绘制结果图并输出模型的最大准确率

import matplotlib.pyplot as plt

fig = plt.figure(figsize=(10,10)) # 绘制画布1000px \* 1000px

for i,name in enumerate(names):

plt.subplot(2, 2, i+1) # 在画布fig中绘制子图，2行2列

plt.plot(scores[i])

plt.rcParams["font.sans-serif"] = "Microsoft YaHei" # 设置字体

plt.title(name) # 设置标题

plt.xlabel("alpha值")

plt.ylabel("准确率")

print(name, "模型的最大准确率是：", max(scores[i]))

输出结果：

LinearRegression 模型的最大准确率是： 0.6129683469594822

Ridge 模型的最大准确率是： 0.612968291029508

Lasso 模型的最大准确率是： 0.6129649583709487