**利用线性回归预测波士顿房价**

1. **引言：**
2. **技术：**

线性回归是利用称为线性回归方程的最小平方函数对一个或多个自变量和因变量之间关系进行建模的一种回归分析。这种函数是一个或多个称为回归系数的模型参数的线性组合。只有一个自变量的情况称为简单回归，大于一个自变量情况的叫做多元回归。

在线性回归中，数据使用线性预测函数来建模，并且未知的模型参数也是通过数据来估计。这些模型被叫做线性模型。最常用的线性回归建模是给定X值的y的条件均值是X的仿射函数。不太一般的情况，线性回归模型可以是一个中位数或一些其他的给定X的条件下y的条件分布的分位数作为X的线性函数表示。像所有形式的回归分析一样，线性回归也把焦点放在给定X值的y的条件概率分布，而不是X和y的联合概率分布。

1. **数据描述**

这是从sklearn.datasets中拿到的波士顿房价的只有506个案例。其中有13个特征值，最后一个为房价。

这里是13个特征量：

图形用户界面, 应用程序, 表格, Excel

描述已自动生成

这是目标量：

图形用户界面, 应用程序, 表格, Excel

描述已自动生成

1. **实践过程**
2. **环境**

利用sklearn进行线性拟合

利用pandas进行数据的输出

利用matplotlib进行画图

1. **编写程序**

**数据进入**

boston=load\_boston()

X=boston.data

y=boston.target

**划定测试集和训练集**

x\_train,x\_test,y\_train,y\_test=train\_test\_split(X,y,test\_size=0.2,random\_state=888)

**进行线性拟合**

lin=LinearRegression()

lin.fit(x\_train,y\_train)#利用训练集进行拟合

lin.score(x\_test,y\_test)#利用测试集进行评估

**对每一列斜率进行评估**

c=plt.plot(lin.coef\_)

plt.show()#每一列数据的斜率

lin.predict(x\_test)#对房价进行预测

dev=lin.predict(x\_test)-y\_test

图表, 折线图

描述已自动生成

根据得到的数据进行y值的预测并于y值进行对比：

dev=lin.predict(x\_test)-y\_test

画出图像：

result={"label":y\_test,"prediction":lin.predict(x\_test)}

result=pd.DataFrame(result)

result["label"].plot(style='b.',figsize=(15,5))

result["prediction"].plot(style='r.')

plt.legend(fontsize=15,markerscale=3)

plt.tick\_params(labelsize=25)

plt.grid()

plt.show()

图表, 散点图

描述已自动生成

数据输出

result={'prediction':lin.predict(x\_test)}

result=pd.DataFrame(result)

result.to\_csv('./data/结果.csv')

图形用户界面, 应用程序, 表格, Excel

描述已自动生成

总代码：

from sklearn.datasets import load\_boston

from sklearn.model\_selection import train\_test\_split  
import pandas as pd  
from sklearn.linear\_model import LinearRegression  
boston=load\_boston()  
X=boston.data  
y=boston.target  
put=pd.DataFrame(y)  
put.to\_csv('./data/房价.csv')#输出房价集  
put=pd.DataFrame(X)  
put.to\_csv('./data/特征.csv')#输出特征集  
x\_train,x\_test,y\_train,y\_test=train\_test\_split(X,y,test\_size=0.2,random\_state=888)#将数据分为训练集和测试集  
lin=LinearRegression()  
lin.fit(x\_train,y\_train)#利用训练集进行拟合  
lin.score(x\_test,y\_test)#利用测试集进行评估  
import matplotlib.pyplot as plt  
c=plt.plot(lin.coef\_)  
plt.show()#每一列数据的斜率  
lin.predict(x\_test)#对房价进行预测  
dev=lin.predict(x\_test)-y\_test  
#画出测试集以及预测的关系  
result={"label":y\_test,"prediction":lin.predict(x\_test)}  
result=pd.DataFrame(result)  
result["label"].plot(style='b.',figsize=(15,5))  
result["prediction"].plot(style='r.')  
plt.legend(fontsize=15,markerscale=3)  
plt.tick\_params(labelsize=25)  
plt.grid()  
plt.show()  
import numpy as np  
rmse=np.sum(np.sqrt(dev\*dev))/102 #算出均方根误差  
result={'prediction':lin.predict(x\_test)}  
result=pd.DataFrame(result)  
result.to\_csv('./data/结果.csv')