# 线性回归

## 导入实验所需包

import numpy as np  
import pandas as pd  
from sklearn.model\_selection import train\_test\_split  
from sklearn.linear\_model import LinearRegression #线性回归  
from sklearn.linear\_model import LogisticRegression #逻辑回归  
from sklearn.preprocessing import StandardScaler  
from sklearn.metrics import mean\_squared\_error

## 定义模型

# 定义模型  
def linear\_model(X\_train, X\_test, y\_train, y\_test):  
 # 线性回归  
 lr = LinearRegression()  
 lr.fit(X\_train, y\_train)  
 y\_pred = lr.predict(X\_test)  
 print('MSE：', mean\_squared\_error(y\_test, y\_pred))

## 数据处理

def dataProcess():  
 # 加载 dataset/housing-data.csv  
 df = pd.read\_csv('dataset/housing-data.csv')  
 # 数据处理  
 # 平均值填充  
 df = df.fillna(df.mean())  
 X = df.iloc[:, :-1]  
 y = df.iloc[:, -1]  
 X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=0)  
 # 标准化  
 sc = StandardScaler()  
 X\_train = sc.fit\_transform(X\_train)  
 X\_test = sc.transform(X\_test)  
 return X\_train, X\_test, y\_train, y\_test

## 主函数

X\_train, X\_test, y\_train, y\_test = dataProcess()  
linear\_model(X\_train, X\_test, y\_train, y\_test)

MSE： 33.448979997676496

# 逻辑回归

## 导入实验所需包

import numpy as np  
import pandas as pd  
from sklearn.model\_selection import train\_test\_split  
from sklearn.linear\_model import LinearRegression #线性回归  
from sklearn.linear\_model import LogisticRegression #逻辑回归  
from sklearn.preprocessing import StandardScaler  
from sklearn.metrics import mean\_squared\_error

## 定义模型

def logistic\_model(X\_train, X\_test, y\_train, y\_test):  
 # 逻辑回归  
 lr = LogisticRegression()  
 lr.fit(X\_train, y\_train)  
 y\_pred = lr.predict(X\_test)  
 # 混淆矩阵  
 import seaborn as sns  
 import matplotlib.pyplot as plt  
 from sklearn.metrics import confusion\_matrix  
 cm = confusion\_matrix(y\_test, y\_pred)  
 sns.heatmap(cm, annot=True, fmt='d')  
 plt.show()

## 数据处理

# 数据处理  
def dataProcess():  
 path = 'dataset/breast-cancer-wisconsin.data'  
 df = pd.read\_csv(path, header=None)  
 # 0填充'?'  
 df = df.replace('?', 0)  
 # 将所有数据转换为int类型  
 df = df.astype(int)  
 # 特征值  
 X = df.iloc[:, 1:-1]  
 # 目标值  
 y = df.iloc[:, -1]  
 # 划分数据集  
 X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=0)  
 # 标准化  
 sc = StandardScaler()  
 X\_train = sc.fit\_transform(X\_train)  
 X\_test = sc.transform(X\_test)  
 return X\_train, X\_test, y\_train, y\_test

## 主函数

X\_train, X\_test, y\_train, y\_test = dataProcess()  
logistic\_model(X\_train, X\_test, y\_train, y\_test)

