实验二：线性回归

姓名：　　　　　　　　　　　　　　　　　　　　学号：

实验目的

理解和掌握线性回归模型基本原理和方法，学会使用线性回归模型对分析问题进行建模和预测，掌握线性问题上模型评估方法。

实验内容

(1) 假设线性模型为$y = w_1x + w_2$，在给定数据集上训练模型，得到模型参数，计算模型在测试集上均方误差，并将训练数据、测试数据、训练模型绘制在一张图中。

(2) 假设二次线性模型为$y = w_1x^2 + w_2x + w_3$，在给定数据集上训练模型，得到模型参数，计算模型在测试集上均方误差，并将训练数据、测试数据、训练模型绘制在一张图中。

实验环境

python

numpy

matplotlib

实验代码

(1) 代码

```
import numpy as np
import matplotlib.pyplot as plt

load_data_testing = np.loadtxt('experiment_02_testing_set.csv', delimiter=',')
load_data_training = np.loadtxt('experiment_02_training_set.csv', delimiter=',')

plt.rcParams['font.sans-serif'] = ['SimHei']
plt.rcParams['axes.unicode_minus'] = False

X = np.ones((np.size(load_data_training, 0), 2))
X[:, 0] = load_data_training[:, 0]
w = np.linalg.inv(X.T @ X) @ X.T @ load_data_training[:, 1]
```

```
print(w)


plt.figure()
load_data_training = load_data_training[np.argsort(load_data_training[:, 0])]
load_data_testing = load_data_testing[np.argsort(load_data_testing[:, 0])]
plt.plot(load_data_training[:, 0], load_data_training[:, 1], label='training')
plt.plot(load_data_testing[:, 0], load_data_testing[:, 1], label='testing')
x = np.linspace(np.min(load_data_training[:, 0]), np.max(load_data_training[:, 0]), 50)
plt.plot(x, w[0] * x + w[1], label='拟合曲线')
plt.xlabel('x')
plt.ylabel('y')
plt.legend()
plt.show()
MSE = np.sum(np.square(load_data_testing[:, 1] - (load_data_testing[:, 0] * w[0] + w[1]))) / np.size(load_data_training,0)
print('均方误差:',MSE)
```

(2) 代码

```
import numpy as np
import matplotlib.pyplot as plt


load_data_testing = np.loadtxt('experiment_02_testing_set.csv', delimiter=',')
load_data_training = np.loadtxt('experiment_02_training_set.csv', delimiter=',')


plt.rcParams['font.sans-serif'] = ['SimHei']
plt.rcParams['axes.unicode_minus'] = False


X = np.ones((np.size(load_data_training, 0), 3))
X[:, 0] = load_data_training[:, 0]
X[:, 1] = np.square(load_data_training[:, 0])
w = np.linalg.inv(X.T @ X) @ X.T @ load_data_training[:, 1]
```

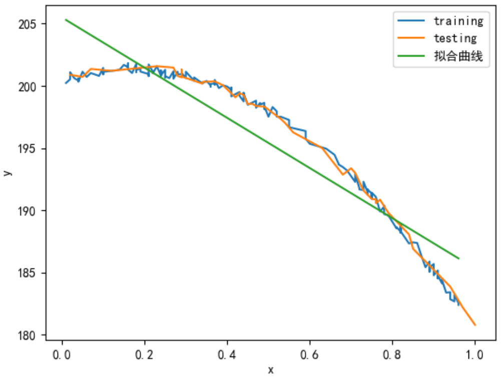
```
print(w)


plt.figure()
load_data_training = load_data_training[np.argsort(load_data_training[:, 0])]
load_data_testing = load_data_testing[np.argsort(load_data_testing[:, 0])]
plt.plot(load_data_training[:, 0], load_data_training[:, 1], label='training')
plt.plot(load_data_testing[:, 0], load_data_testing[:, 1], label='testing')
x = np.linspace(np.min(load_data_training[:, 0]), np.max(load_data_training[:, 0]), 50)
plt.plot(x, w[0] * x + w[1] * x ** 2 + w[2], label='拟合曲线')
plt.xlabel('x')
plt.ylabel('y')
plt.legend()
plt.show()


MSE = np.sum(np.square(
    load_data_testing[:, 1] - (load_data_testing[:, 0] * w[0] + load_data_testing[:, 0] ** 2 * w[1] + w[2])))
/ np.size(
    load_data_training,
    0)
print('均方误差:', MSE)
```

结果分析

(1) 模型参数为：[-20.16559945 205.49808198]　　y=x*(-20.16559945)+ 205.49808198

测试集均方误差为：1.1564084436265119

绘图结果为：

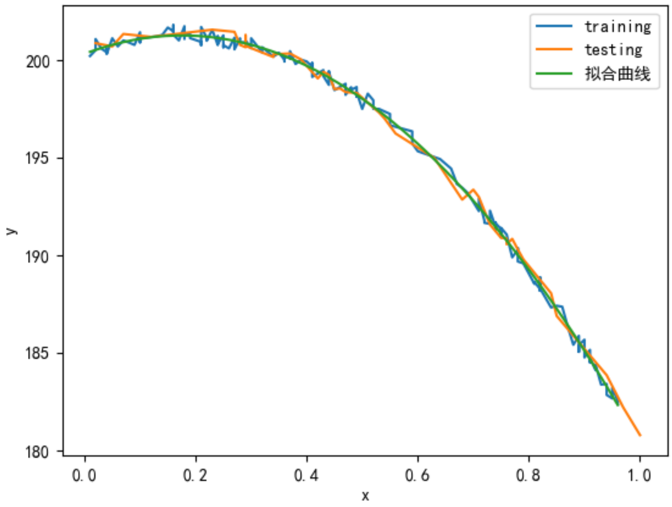


(2) 模型参数为：[10.77907734 -30.75765867 200.34082655]

y=x*10.77907734+x**2*-30.75765867+ 200.34082655

测试集均方误差为：0.025785367191348895

绘图结果为：


training
testing
拟合曲线
y
x