# 实验7 回归问题

## 实验目的

①掌握一元线性回归模型的实现方法

②掌握多元线性回归模型的实现方法

③掌握三维数据可视化方法

## 2. 实验内容

①使用TensorFlow建立一元线性回归模型，使用商品房销售数据训练模型，并使用训练好的模型预测房价

②使用TensorFlow建立多元线性回归模型，使用商品房销售数据预测房价，并实现三维数据可视化

## 实验过程

题目一：

使用9.5小节中的“商品房销售记录表”作为样本数据，训练一元线性回归模型，根据商品房面积预测房价。

提示用户输入商品房面积，并进行输入校验。合理的输入如下：

面积：20-500之间的实数

如果输入正确，根据模型估计房价，并显示。

如果输入数据类型错误，或者输入数据范围不合理，根据错误类型提示，并等待用户重新输入，输错3次，则程序结束。

要求：

(1)编写代码，实现程序功能；

(2)记录实验过程和结果：尝试调试超参数，使模型达到最优的性能，记录实验过程和结果。

① 代码

import numpy as np

import tensorflow as tf

house\_area = np.array([137.97, 104.50, 100.00, 124.32, 79.20, 99.00, 124.00, 114.00,

                      106.69, 138.05, 53.75,  46.91,   68.00, 63.02, 81.26,  86.21])

# house\_room = np.array([3, 2, 2, 3, 1, 2, 3, 2, 2, 3, 1, 1, 1, 1, 2, 2])

house\_price = np.array([145.00, 110.00, 93.00, 116.00, 65.32, 104.00, 118.00, 91.00,

                       62.00, 133.00, 51.00, 45.00, 78.50, 69.65, 75.69, 95.30])

x\_mean = tf.reduce\_mean(house\_area)

y\_mean = tf.reduce\_mean(house\_price)

# 公式求w和b

w = tf.reduce\_sum((house\_area - x\_mean)\*(house\_price - y\_mean))/tf.reduce\_sum((house\_area - x\_mean)\*\*2)

b = y\_mean - w\*x\_mean

# 面积输入

def area\_input\_detector():

    chance = 3

    while chance:

        chance -= 1

        try:

            data = input('请输入面积（一个20-500之间的实数）：')

            if data == '' or data.isspace():

                print('对不起，你的输入为空！')

                print('您还有%d次机会，请重新输入。' % chance)

                print('-'\*30)

                continue

            data = *float*(data)

        except *ValueError* as ve:

            print('对不起，你的输入类型有误，并非实数数据！')

        else:

            if 20 <= data <= 500:

                return data

            else:

                print('对不起，您输入的数字范围不正确!')

        print('您还有%d次机会，请重新输入。' % chance)

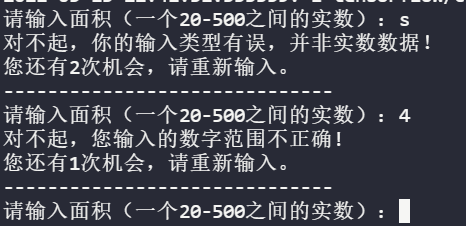
        print('-'\*30)

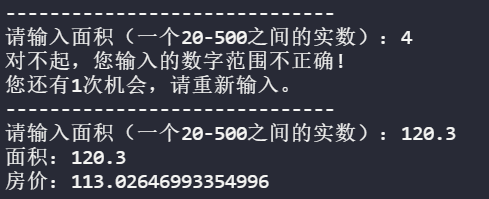
    print('对不起，您已经3次输入错误，程序退出。')

area\_input\_data = area\_input\_detector()

print(f'面积：{area\_input\_data}\n房价：{float(w\*area\_input\_data+b)}')

② 实验结果





题目二：

使用9.5小节中的“商品房销售记录表”作为样本数据，训练多元线性回归模型，实现一个房价预测系统。

要求：

(1)尝试调试超参数，使模型达到最优的性能，记录实验过程和结果；

(2)创建3D绘图对象来绘制空间点集。x轴表示房屋面积，y轴表示房间数，z轴表示样本的销售价格；

(3)提示用户输入商品房面积和房间数，并进行输入校验。如果输入正确，根据模型预测房价。

合理的输入如下：

面积：20-500之间的实数

房间数：1-10之间的整数

如果输入数据类型错误，或者输入数据范围不合理，根据错误类型给出提示，并等待用户重新输入，输错3次，则程序结束。

（输入时，请注意房间面积对应房间数的合理性）

提示：TensorFlow中矩阵求逆函数tf.linalg.inv()

① 代码

import numpy as np

import tensorflow as tf

from mpl\_toolkits.mplot3d import Axes3D

import matplotlib.pyplot as plt

# config = tf.ConfigProto()

# config.gpu\_options.allow\_growth = True

# Check failed: cusolverDnCreate(&cusolver\_dn\_handle) == CUSOLVER\_STATUS\_SUCCESS Failed to create cuSolverDN instance.

house\_area = np.array([137.97, 104.50, 100.00, 124.32, 79.20, 99.00, 124.00, 114.00,

                      106.69, 138.05, 53.75,  46.91,   68.00, 63.02, 81.26,  86.21])

house\_room = np.array([3, 2, 2, 3, 1, 2, 3, 2, 2, 3, 1, 1, 1, 1, 2, 2])

house\_price = np.array([145.00, 110.00, 93.00, 116.00, 65.32, 104.00, 118.00, 91.00,

                       62.00, 133.00, 51.00, 45.00, 78.50, 69.65, 75.69, 95.30])

X = tf.stack((np.ones(len(house\_area)), house\_area, house\_room),*axis*=1)

Y = tf.constant(house\_price.reshape(-1,1))

W = tf.matmul(tf.matmul(tf.linalg.inv(np.matmul(tf.transpose(X), X)), tf.transpose(X)), Y)

def area\_input\_detector():

    chance = 3

    while chance:

        chance -= 1

        try:

            data = input('请输入面积(一个20-500之间的实数)：')

            if data == '' or data.isspace():

                print('对不起，你的输入为空！')

                print('您还有%d次机会，请重新输入。' % chance)

                print('-'\*30)

                continue

            data = *float*(data)

        except *ValueError* as ve:

            print('对不起，你的输入类型有误，并非实数数据！')

        else:

            if 20 <= data <= 500:

                return data

            else:

                print('对不起，您输入的数字范围不正确!')

        print('您还有%d次机会，请重新输入。' % chance)

        print('-'\*30)

    print('对不起，您已经3次输入错误，程序退出。')

    return 0

def room\_input\_detector():

    chance = 3

    while chance:

        chance -= 1

        try:

            data = input('请输入房间数(一个1-10之间的整数)：')

            if data == '' or data.isspace():

                print('对不起，你的输入为空！')

                print('您还有%d次机会，请重新输入。' % chance)

                print('-'\*30)

                continue

            data = *int*(data)

        except *ValueError* as ve:

            print('对不起，你的输入类型有误，并非实数数据！')

        else:

            if 1 <= data <= 10:

                return data

            else:

                print('对不起，您输入的数字范围不正确!')

        print('您还有%d次机会，请重新输入。' % chance)

        print('-'\*30)

    print('对不起，您已经3次输入错误，程序退出。')

    return 0

area\_input\_data = area\_input\_detector()

if area\_input\_data != 0:

    room\_input\_data = room\_input\_detector()

    print(f'面积：{area\_input\_data}\n房间数：{room\_input\_data}\n房价：{float(W[1]\*area\_input\_data+W[2]\*room\_input\_data+W[0])}')

    fig =plt.figure(*figsize*=(8,6))

    ax3d=Axes3D(fig)

    ax3d.scatter(house\_area, house\_room, house\_price, *c*='b',*marker*="\*")

    ax3d.scatter(area\_input\_data, room\_input\_data, *float*(W[1]\*area\_input\_data+W[2]\*room\_input\_data+W[0]), *c*='r',*marker*=".")

    ax3d.set\_xlabel("Area", *color*="r", *fontsize*=16)

    ax3d.set\_ylabel("Room", *color*="r", *fontsize*=16)

    ax3d.set\_zlabel("Price", *color*="r", *fontsize*=16)

    ax3d.set\_yticks([1,2,3])

    ax3d.set\_zlim3d(30,160)

    plt.show()

    X1,X2 = np.meshgrid(house\_area, house\_room)

    Y\_PRED = W[1]\*X1+W[2]\*X2+W[0]

    fig = plt.figure(*figsize* = (8,6))

    ax3d = Axes3D(fig)

    ax3d.plot\_surface(X1,X2,Y\_PRED,*cmap*="coolwarm")

    ax3d.set\_xlabel('Area',*color*='r',*fontsize*=16)

    ax3d.set\_ylabel('Room',*color*='r',*fontsize*=16)

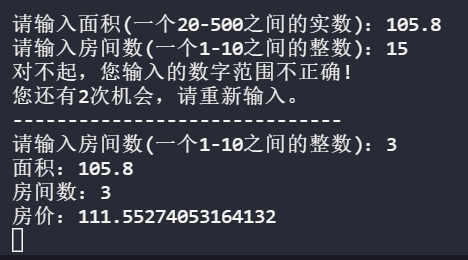
    ax3d.set\_zlabel('Price',*color*='r',*fontsize*=16)

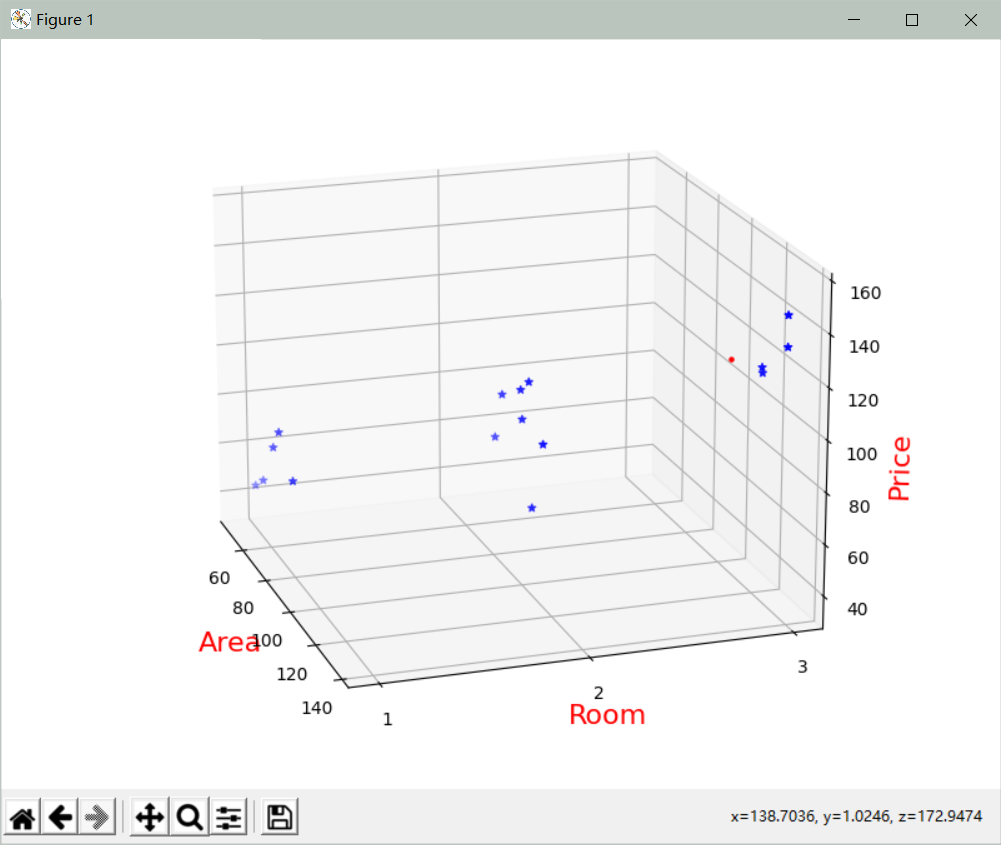
    ax3d.set\_yticks([1,2,3])  #y轴坐标轴刻度

    # ax3d.view\_init(10,-70)  #改变观察视角（水平视角，水平旋转的角度）

    plt.show()

② 实验结果





说明：以上程序矩阵运算部分采用TensorFlow实现，数据加载、输入、输出等可以根据需要采用Python/NumPy来实现。

## 实验小结

① 实验过程中遇到了哪些问题，你是如何解决的？

无

② 分别使用Numpy和TensorFlow进行数据的加载和数组的堆叠

③ 题目一和题目二在代码实现过程中，有重复的代码段，你知道该如何精简自己的代码吗？请进行简要的描述。

函数参数。

④ 在题目基本要求的基础上，你对每个题目做了那些扩展和提升？或者你觉得在编程实现过程中，还有哪些地方可以进行优化？

题目2在3d图里加了预测的点。