实验一 线性回归

**附录：源码1**

**import numpy as np**

**import matplotlib.pyplot as plt**

***# 画图参数设置，可以去掉，此时为默认值，或自己修改数值看效果***

**plt.rcParams['axes.labelsize'] = 14**

**plt.rcParams['xtick.labelsize'] = 12**

**plt.rcParams['ytick.labelsize'] = 12**

***# 通过rand函数可以返回一个或一组服从“0~1”均匀分布的随机样本值。随机样本取值范围是[0,1)，不包括1***

**X = 2 \* np.random.rand(100, 1)**

***# 构造线性方程，加入随机抖动***

***# numpy.random.randn()是从标准正态分布中返回一个或多个样本值***

***# 1.当函数括号内没有参数时，返回一个浮点数；***

***# 2.当函数括号内有一个参数时，返回秩为1的数组，不能表示向量和矩阵***

***# 3.当函数括号内有两个及以上参数时，返回对应维度的数组，能表示向量或矩阵。np.random.randn(行,列)***

***# 4.np.random.standard\_normal()函数与np.random.randn类似，但是输入参数为元组（tuple）***

**y = 3\*X + 4 + np.random.randn(100, 1)**

**plt.plot(X, y, 'b.') *# b指定为蓝色,.指定线条格式***

**plt.xlabel('X\_1')**

**plt.ylabel('y')**

***# 设置x轴为0-2，y轴为0-15***

**plt.axis([0, 2, 0, 15])**

**plt.show()**

**"""线性回归方程实现"""**

***# numpy.c\_:按行连接两个矩阵，就是把两矩阵左右相加，要求行数相等。***

***# numpy.r\_:按列连接两个矩阵，就是把两矩阵上下相加，要求列数相等。***

***# ones()返回一个全1的n维数组，同样也有三个参数：shape（用来指定返回数组的大小）、dtype（数组元素的类型）、order（是否以内存中的C或Fortran连续（行或列）顺序存储多维数据）。后两个参数都是可选的，一般只需设定第一个参数。***

**X\_b = np.c\_[(np.ones((100, 1)), X)]**

***# np.linalg.inv:矩阵求逆***

**theta\_best = np.linalg.inv(X\_b.T.dot(X\_b)).dot(X\_b.T).dot(y)**

**print(theta\_best)**

**'''**

**[[3.99844262]**

**[3.09461187]]**

**'''**

***# 测试数据***

**X\_new = np.array([[0], [2]])**

**X\_new\_b = np.c\_[np.ones((2, 1)), X\_new]**

***# 预测结果***

**y\_predict = X\_new\_b.dot(theta\_best)**

**print(y\_predict)**

**plt.plot(X\_new, y\_predict, 'r--o') *# 指定红色和线条***

**plt.plot(X, y, 'b.') *# 指定蓝色和点***

**plt.axis([0, 2, 0, 15])**

**plt.show()**

**附录2：源代码，鸢尾花可视化**

**import matplotlib.pyplot as plt**

**import numpy as np**

**from sklearn.datasets import load\_iris**

**# 导入鸢尾花数据集**

**# 以二维数据为例 假设k=2，X为鸢尾花数据集前两维**

**iris = load\_iris()**

**X = iris.data[0:150, 0:2] ##鸢尾花数据集,只取特征空间中的前两个维度**

**print(len(X)) #长度150**

**print(X)**

**# 绘制数据分布图 显示前两维数据**

**x1=X[0:150, 0] #第一列数据**

**print(x1.shape) # 第一列大小为150行,1列**

**x2=X[0:150, 1] #第二列数据**

**print(x2.shape) # 第二列大小为150行,1列**

**plt.scatter(x1, x2, c="red", marker='o', label='class1')**

**plt.xlabel('petal length')**

**plt.ylabel('petal width')**

**plt.legend(loc=1) #图例放置位置,右上为1,逆时针排序**

**plt.title('myname')**

**plt.show() #输出图形**