

# sklearn线性回归和正规方程 样例

## sklearn中线性方程的正规方程运算

安装：

代码块

```
1 pip install scikit-learn
```

导入包，计算

代码块

```
1 # linear线性, model模型、算法
2 #LinearRegression 线性回归
3 from sklearn.linear_model import LinearRegression
4
5 model = LinearRegression(fit_intercept=False) # 不计算拟合截距
6 X = ([[1,1],[2,-1]])
7 y = ([[14],[10]])
8 # X就是数据
9 # y是目标值
10 model.fit(X,y) # 拟合训练
11 var = model.coef_ # 结果返回值, 也就是最后的方程系数
12 print(var)
```

## 带截距（常数项）的线性方程计算

我这里解释一下为什么上边没有使用带截距的计算，我们来看之前的公式：

$$\hat{y} = w_1 X_1 + w_2 X_2 + \dots + w_n X_n + b \quad (1)$$

$$b \text{是截距, 也是使用 } w_0 \text{ 来表示} \quad (2)$$

(3)

$$\text{类似于这种公式, 你会发现在公式的右侧存在常数 } b \quad (4)$$

$$\text{这个 } b \text{ 在上述的例子中都不存在:} \quad (5)$$

$$f(x) = 8x_1 + 6x_2 \quad (6)$$

$$\text{他是没有 } b \text{ 的} \quad (7)$$

所以我们没有使用带截距计算，那我们就可以做一些改变么，因为我们知道，这个 b 实际上影响的是 y 目标值，我们可以在 y 的基础上加上一定值，就可以产生 b，我们再开启截距计算

但是我们运行发现：

代码块

```
1  from sklearn.linear_model import LinearRegression
2
3  model = LinearRegression(fit_intercept=True) # 计算截距
4  X = ([[1,1],[2,-1]])
5  # y = ([[14],[10]])
6  y = ([[18],[14]]) # +4
7  # X就是数据
8  # y是目标值
9  model.fit(X,y) # 计算
10 var = model.coef_ # 结果返回值，也就是最后的方程系数
11 print()
12 print(var)
13
14 ----
15
16 结果为：
17 [[-0.8  1.6]] # 并没有出现正常输出
```

不过我们通过计算，发现实际上程序预测的值（截距和斜率）计算出来的预测值 y 目标值 是一致的：

代码块

```
1  from numpy import *
2  from sklearn.linear_model import LinearRegression
3
4  model = LinearRegression(fit_intercept=True) # 计算截距
5  X = array([[1,1],[2,-1]])
6  # y = ([[14],[10]])
7  y = array([[18],[14]]) # +4
8  # X就是数据
9  # y是目标值
10 model.fit(X,y) # 计算
11 var = model.coef_ # 结果返回值，也就是最后的方程系数
12 print(model.intercept_) # 截距
13 print(var) # 方程系数
14 print(X.dot(var.T)+model.intercept_) # 预测值
15
16 ---
17
18 结果：
```

```
19  截距 [17.2]
20  系数 [[-0.8  1.6]]
21  预测值 [[18.]
22        [14.]]
```

$\hat{y}$ 与 $y$  是相同的，为什么？因为在 `LinearRegression()` 中指定了 `fit_intercept=True` 并使用内置的正规方程截距计算，而打上`False`，并手动加入一列作为数据1也可以正规方程的实现截距计算，两种方式都可以，但是更推荐使用内置的，也就是 `fit_intercept=True`：

那为什么会出现误差？因为此方程组无精确解（秩不足），实际输出为最小二乘解。如需精确解，需要增加样本量或使用线性代数方法。

**按照人话来就是就是，如果我们考虑到存在截距的话，也就多了一个未知参数，而这个方程组中方程的个数只有两个，此时未知数的参数有三个，程序没办法给出精确的未知数解**

也就是你给的样本太少了！