线性回归

线性回归

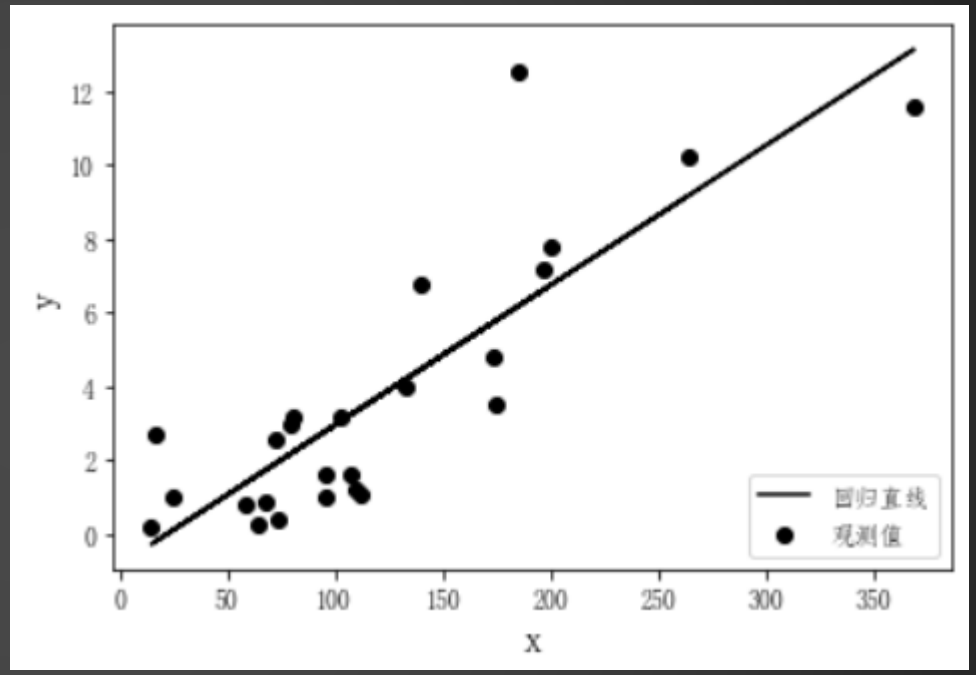
- 线性回归是研究自变量X和因变量y之间的线性关系。
- 自变量：x1，x2，x3，...，均为数值型。
- 因变量：y，为数值型。

- 根据自变量X的个数分为**一元线性回归（一个自变量）**和**多元线性回归（多个自变量）**。

一元线性回归

- 特点：只涉及一个自变量x
- 回归方程：y= $wx + b$
- 一次函数：y=kx+b

- 任务目标：估计参数w, b



一元线性回归示意图

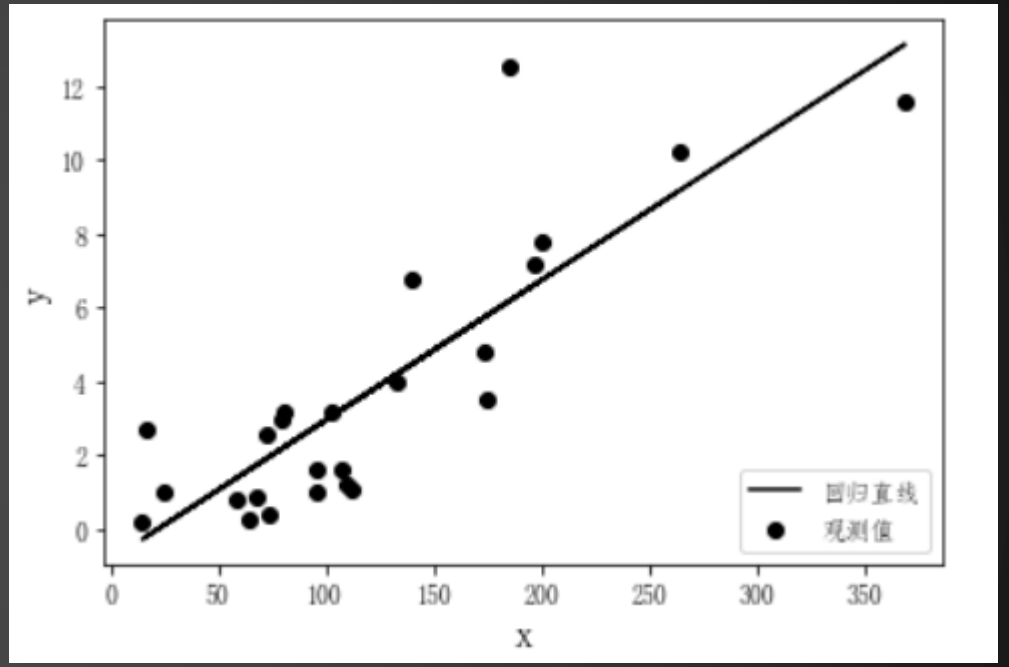
最小二乘法

假定有m个样本，$(x_1, y_1), (x_2, y_2), \cdots, (x_m, y_m)$

最小二乘法的基本思想是通过最小化这些点到直线的总误差来估计参数w和b。

根据最小二乘法，使以下这个式子最小。

$$\sum_{i=1}^{m}(y_i - \hat{y}_i)^2 = \sum_{i=1}^{m}(y_i - wx_i - b)^2$$



这个式子采用的是误差的平方和，加上平方是为了避免正负相抵。

在机器学习中，一般将最小化的目标函数称为损失函数（loss function）或者代价函数（cost function）。

最小二乘法

损失函数（loss function）：

$$l(w,b)=\sum_{i=1}^{m}(y_i-wx_i-b)^2$$

在给定样本数据后，l是w和b的函数，且最小值总是存在。

根据微积分的极值定理，对l求相应于w和b的偏导数，并令其等于0，便可求出w和b,即

$$\begin{cases}\frac{\partial l}{\partial w}=-2\sum_{i=1}^{m}x_i(y_i-wx_i-b)=0\\ \frac{\partial l}{\partial b}=-2\sum_{i=1}^{m}(y_i-wx_i-b)=0\end{cases}$$

解上述方程组得

$$\begin{cases}w=\frac{m\sum_{i=1}^{m}x_iy_i-\sum_{i=1}^{m}x_i\sum_{i=1}^{m}y_i}{m\sum_{i=1}^{m}x_i^2-(\sum_{i=1}^{m}x_i)^2}\\ b=\frac{1}{m}\sum_{i=1}^{m}(y_i-wx_i)\end{cases}$$

大家只要了解原理，一般借助于计算机来计算！！！

评估回归效果

将由回归方程计算出来的y值记为$\hat{y}$，对于每一个实际观测值y_i，其误差的大小用$y_i - \hat{y}_i$来表示，所有观测值的总误差可以通过每个观测值的误差的平方和来表示，即

$$\sum(y_i - \hat{y}_i)^2$$

以上式子可以称为误差平方和，或者残差平方和。

接着对误差做一下变形。

$$y_i - \hat{y}_i = y_i - \overline{y} - (\hat{y}_i - \overline{y})$$

移项得

$$y_i - \overline{y} = (y_i - \hat{y}_i) + (\hat{y}_i - \overline{y})$$

上式本质上是误差分解，不妨将其称为**误差分解式**。

接着，将误差分解式两边平方，并求和，得

$$\sum(y_i-\overline{y})^2=\sum[(y_i-\hat{y}_i)+(\hat{y}_i-\overline{y})]^2$$

将右边平方展开，得

$$\sum(y_i-\overline{y})^2=\sum(y_i-\hat{y}_i)^2+\sum(\hat{y}_i-\overline{y})^2+2\sum(y_i-\hat{y}_i)(\hat{y}_i-\overline{y})$$

上式中，可以证明$\sum(y_i-\hat{y}_i)(\hat{y}_i-\overline{y})=0$

说明：证明时，需要将$\hat{y}_i=wx+b$代入上式中，前面已经求得参数w和b的值。

于是，有

$$\sum(y_i-\overline{y})^2=\sum(y_i-\hat{y}_i)^2+\sum(\hat{y}_i-\overline{y})^2$$

评估回归效果

$$\sum(y_i-\overline{y})^2=\sum(y_i-\hat{y}_i)^2+\sum(\hat{y}_i-\overline{y})^2$$

在上式中，左边称为**总平方和**，右边第一项称为**误差平方和**，右边第二项称为**回归平方和**，我们的目标是希望**误差平方和**尽可能小。

在总平方和一定的情况下，回归平方和越大，误差平方和就越小，所以可以借助于回归平方和占总平方和的比例来评估回归方程的好坏，我们将这个比例称之为**判定系数**，记为R^2，其表达式为：

$$R^2=\frac{\sum(y_i-\hat{y}_i)^2}{\sum(y_i-\overline{y})^2}=1-\frac{\sum(\hat{y}_i-\overline{y})^2}{\sum(y_i-\overline{y})^2}$$

可以看出，判定系数是一个介于0到1之间的数，判定系数越接近于1，说明回归方程拟合效果越好。

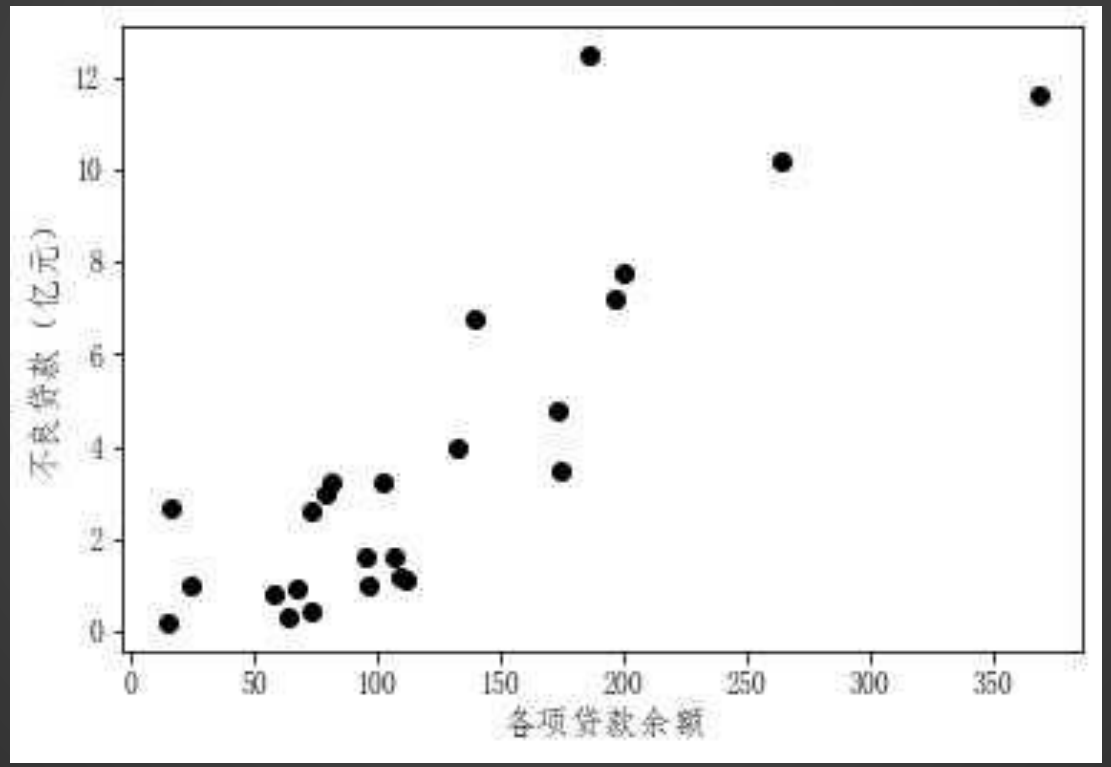
案例：分析不良贷款形成的原因

- 一家大型商业银行在多个地区有25家分行，其业务主要是进行基础设施建设、国家重点项目建设、固定资产投资等项目的贷款。
- 下面是该银行所属的25家分行2002年的有关业务数据（前10条）。

分行编号	不良贷款（亿元）	各项贷款余额	本年累计应收贷款（亿元）	贷款项目个数	本年固定资产投资额（亿元）
1	0.9	67.3	6.8	5	51.9
2	1.1	111.3	19.8	16	90.9
3	4.8	173.0	7.7	17	73.7
4	3.2	80.8	7.2	10	14.5
5	7.8	199.7	16.5	19	63.2
6	2.7	16.2	2.2	1	2.2
7	1.6	107.4	10.7	17	20.2
8	12.5	185.4	27.1	18	43.8
9	1.0	96.1	1.7	10	55.9
10	2.6	72.8	9.1	14	64.3

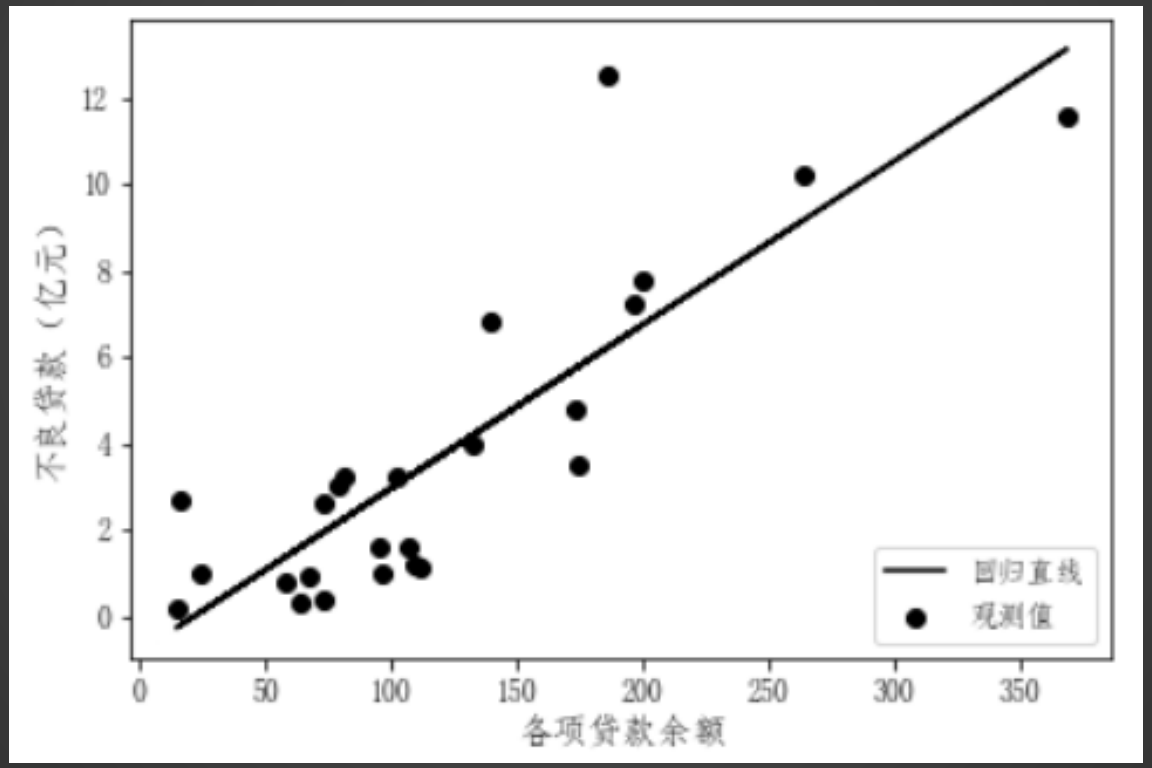
案例：分析不良贷款形成的原因

- 选择各项贷款余额为自变量x，不良贷款为因变量y，绘制x与y的散点图。



拟合直线

- 接下来需要找到一条直线（回归直线）来拟合这些数据点。



假设回归直线方程为：y= $wx + b$，估计参数w, b。

一元线性回归：Python实现

1. 数据读取
2. 绘制散点图
3. 利用sklearn建立回归模型
4. 得到回归方程，并进行预测
5. 模型评估：判定系数

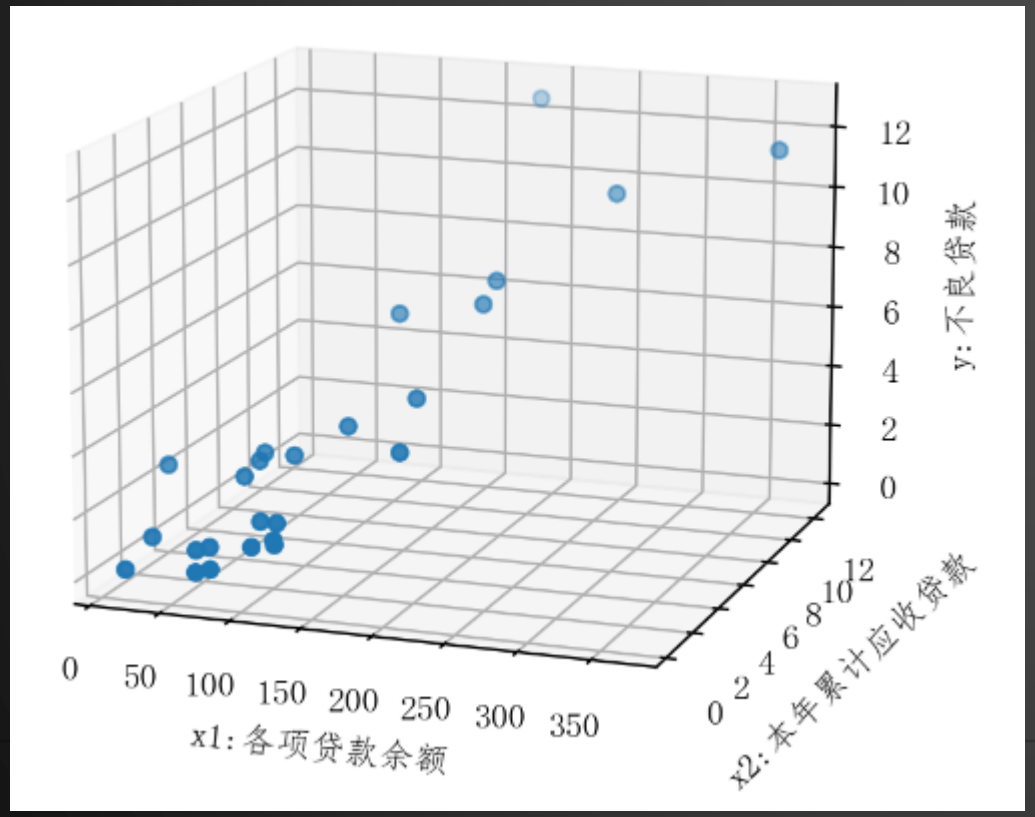
多元线性回归

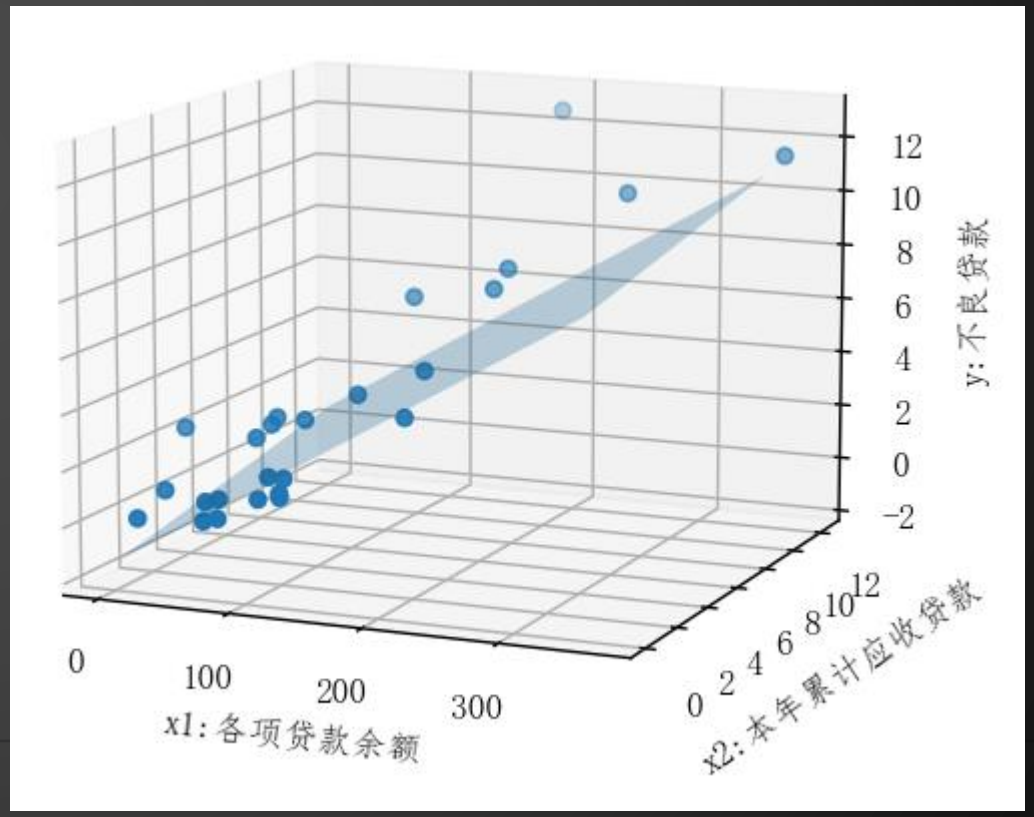
- 特点：涉及多个自变量，x1，x2，x3，.....
- 回归方程：$y = w_1x_1 + w_2x_2$+...+ w_nx_n+b

- 任务目标：估计参数$w_1, w_2, \cdots, w_n$, b

二元线性回归示意图

- 假设有两个自变量，x1，x2，加上一个因变量y，可以绘制一个三维散点图。
- 此时，回归方程：$y = w_1 x_1 + w_2 x_2 + b$，是一个平面，用这个平面去拟合这些样本点。





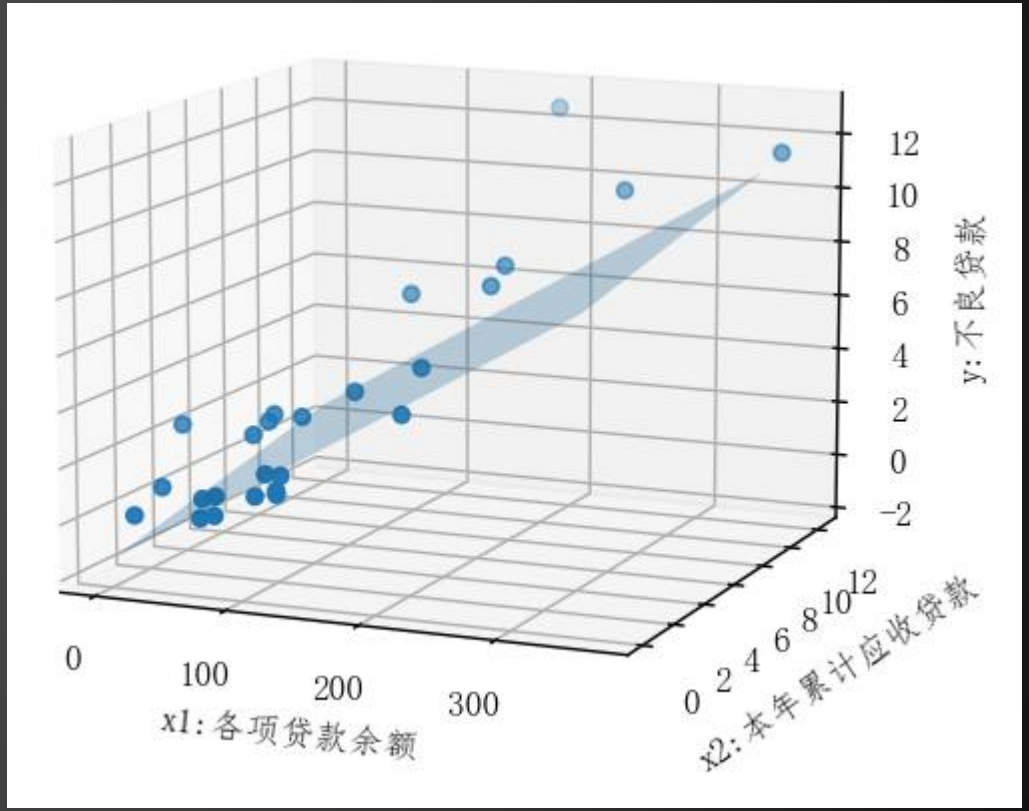
二元线性回归的最小二乘法

假定有m个样本，$\left(x_1^{(1)}, x_2^{(1)}, y_1\right), \left(x_1^{(2)}, x_2^{(2)}, y_2\right), \cdots, \left(x_1^{(m)}, x_2^{(m)}, y_m\right)$

最小二乘法的基本思想是通过最小化这些点到拟合平面的总误差来估计参数w_1, w_2, b

根据最小二乘法，使以下这个式子最小。

$$\sum_{i=1}^{m}(y_i - \hat{y}_i)^2 = \sum_{i=1}^{m}(y_i - w_1 x_1^{(i)} - w_2 x_2^{(i)} - b)^2$$



在机器学习中，一般将最小化的目标函数称为损失函数（loss function）或者代价函数（cost function）。

多元线性回归的最小二乘法

假定有m个样本，$(x_1^{(1)}, x_2^{(1)}, \cdots, x_n^{(1)}, y_1), (x_1^{(2)}, x_2^{(2)}, \cdots, x_n^{(2)}, y_2), \cdots, (x_1^{(m)}, x_2^{(m)}, \cdots, x_n^{(m)}, y_m)$。

损失函数：

$$l(w_1, w_2, ..., w_n, b) = \sum_{i=1}^{m} (y_i - w_1 x_1^{(i)} - w_2 x_2^{(i)} - ... - w_n x_n^{(i)} - b)^2$$

注意：n个参数，m个样本。

根据微积分的极值定理，对l求相应于$w_1, w_2, ..., w_n$和b的偏导数，并令其等于0，即

$$\begin{cases} \frac{\partial l}{\partial w_j} = -2 \sum_{i=1}^{m} x_j^{(i)} (y_i - w_1 x_1^{(i)} - w_2 x_2^{(i)} - \cdots - w_n x_n^{(i)} - b) = 0, j = 1, 2, ..., n \\ \frac{\partial l}{\partial b} = -2 \sum_{i=1}^{m} (y_i - w_1 x_1^{(i)} - w_2 x_2^{(i)} - \cdots - w_n x_n^{(i)} - b) = 0 \end{cases}$$

解上述方程组可以得到各参数的值。

大家只要了解原理，一般借助于计算机来计算！！！

评估回归效果

在一元线性回归中，我们用回归平方和占总平方和的比例来评估回归方程的好坏，我们将这个比例称之为**判定系数**，记为R^2，其表达式为：

$$R^2 = \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \overline{y})^2} = 1 - \frac{\sum(\hat{y}_i - \overline{y})^2}{\sum(y_i - \overline{y})^2}$$

在多元线性回归中，也可以推导出一个类似的公式，我们将其称为**多重判定系数。**

多重判定系数也是一个介于0到1之间的数，判定系数越接近于1，说明回归方程拟合效果越好。

案例：分析不良贷款形成的原因

- 一家大型商业银行在多个地区有25家分行，其业务主要是进行基础设施建设、国家重点项目建设、固定资产投资等项目的贷款。
- 下面是该银行所属的25家分行2002年的有关业务数据（前10条）。

分行编号	不良贷款（亿元）	各项贷款余额	本年累计应收贷款（亿元）	贷款项目个数	本年固定资产投资额（亿元）
1	0.9	67.3	6.8	5	51.9
2	1.1	111.3	19.8	16	90.9
3	4.8	173.0	7.7	17	73.7
4	3.2	80.8	7.2	10	14.5
5	7.8	199.7	16.5	19	63.2
6	2.7	16.2	2.2	1	2.2
7	1.6	107.4	10.7	17	20.2
8	12.5	185.4	27.1	18	43.8
9	1.0	96.1	1.7	10	55.9
10	2.6	72.8	9.1	14	64.3

案例：分析不良贷款形成的原因

- 自变量
  - x1：各项贷款余额
  - x2：本年累计应收贷款
  - x3：贷款项目个数
  - x4：本年固定资产投资额
- 因变量
  - y：不良贷款

多元线性回归：Python实现

1. 数据读取
2. 建立回归模型
3. 得到回归方程，并进行预测
4. 模型评估：多重判定系数