Multiple Linear Regression

When you have more than one input variables $x_1 | x_2 | x_3 | x_4 | y$

Everything we learned in simple linear regression is utilized in multiple linear regression.

CGPA | IQ | LPA

Lets say x_1 | x_2 are input y is o/p

As you know for 2-D we draw a "line", in case of 3D we will draw a plane.

In 4D and above we call it hyperplane.

In 2D we solved $y = mx + b$

$y = mx_1 + mx_2 + b$ → 3D

or in maths

3D → $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2$ → Eq of plane

4D → $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3$

Similarly this can be expanded to n dimensions

$$y = \beta_0 + \sum_{i=1}^{n} \beta_i x_i$$ → General Eq

For a 3-D data → CGPA, IQ, LPA

$$lpa = \beta_0 + \underbrace{\beta_1 \times CGPA + \beta_2 IQ}$$

↓

Represents weights of individual input variable. CGPA, IQ's importance of we can which input variable is valueable.

is 3D

- The plane drawn here tries to cut through all points in data

- Matrematical formulation

$$\hat{y} = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3$$

predicted

Suppose I have 100 students

$$\begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \hat{y}_3 \\ \vdots \\ \hat{y}_{100} \end{bmatrix} = \begin{bmatrix} \beta_0 + \beta_1 X_{11} + \beta_2 X_{12} + \beta_3 X_{13} \\ \beta_0 + \beta_1 X_{21} + \beta_2 X_{22} + \beta_3 X_{23} \\ \vdots \\ \beta_0 + \beta_1 X_{1001} + \beta_2 X_{1002} + \beta_3 X_{1003} \end{bmatrix}$$

$\hat{y}$: $y_1 = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2}$ → i^{th} row, 1^{st} column, 2^{nd} column

Big 'Y Hat' → predicted LPA of all 100 students

Now lets say we have n rows & m columns ⇒ General Case

$$\hat{Y} = \begin{bmatrix} \hat{y}_1 \\ \vdots \\ \hat{y}_n \end{bmatrix} = \begin{bmatrix} \beta_0 + \beta_1 x_{11} + \beta_2 x_{12} + \dots \beta_m x_{1m} \\ \beta_0 + \beta_1 x_{21} + \beta_2 x_{22} + \dots \beta_m x_{2m} \\ \\ \beta_0 + \beta_1 x_{n1} + \beta_2 x_{n2} + \dots \beta_m x_{nm} \end{bmatrix}$$

n^{th} row; m^{th} column

This matrix can be decomposed into product of 2 matrix

$$\begin{bmatrix} 1 & x_{11} & x_{12} & x_{13} & \dots & x_{1m} \\ 1 & x_{21} & x_{22} & x_{23} & \dots & x_{2m} \\ \vdots & & & & & \\ 1 & x_{n1} & x_{n2} & x_{n3} & \dots & x_{nm} \end{bmatrix} \quad A \qquad \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_m \end{bmatrix} \quad B$$

$\beta_0 + \beta_1 x_{11} + \beta_2 x_{12} + \dots \beta_m x_{1m}$ ← One 1st row in 1st matrix

$$\hat{y} = X\beta$$

$$\hat{Y}_{(matrix)} = X_{matrix}\, \beta_{matrix}$$

Now our task has become simple to calculate values of β matrix

$\hat{Y}$ = X β → matrix of All our coefficients

↓ ↓

prediction of All Input rows — whole input data

. . . — — — — — — — — — — . . .

$Y = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}$ → All actual LPA o/p of our data

$e = Y - \hat{Y}$ ⇒ Actual - predicted

$e = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} - \begin{bmatrix} \hat{y}_1 \\ \vdots \\ \hat{y}_n \end{bmatrix}$

$e = \begin{bmatrix} y_1 - \hat{y}_1 \\ y_2 - \hat{y}_2 \\ \vdots \\ y_n - \hat{y}_n \end{bmatrix}$

Now in Simple Linear Regression (Loss Funct)

$E = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$ → for the values of diff pts

Now here our Multiple LR loss function in terms of matrix representation

$E = e^T e$ → $\begin{bmatrix} y_1 - \hat{y}_1 \\ y_2 - \hat{y}_2 \\ \vdots \\ y_n - \hat{y}_n \end{bmatrix}$

So for squaring and getting +ve

$$E = \underbrace{\begin{bmatrix}(y_1 - \hat{y}_1) & (y_2 - \hat{y}_2) & \cdots & (y_n - \hat{y}_n)\end{bmatrix}}_{e^T \quad 1 \times n} \begin{bmatrix}(y_1 - \hat{y}_1)\\ (y_2 - \hat{y}_2)\\ \vdots \\ (y_n - \hat{y}_n)\end{bmatrix}_{n \times 1}$$

$$E = (y_1 - \hat{y}_1)\cdot(y_1 - \hat{y}_1) + (y_2 - \hat{y}_2)(y_2 - \hat{y}_2) + \cdots + (y_n - \hat{y}_n)(y_n - \hat{y}_n)$$

$$E = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

$$E = e^T e = (y - \hat{y})^T (y - \hat{y})$$

$$(A - B)^T = A^T - B^T$$

$$= (y^T - \hat{y}^T)(y - \hat{y})$$

$\hat{y} = X\beta$ ← Actual o/p

$$E = (y^T - (X\beta)^T)(y - X\beta)$$

$$E = y^T y - y^T X\beta - (X\beta)^T y + (X\beta)(X\beta)^T$$

These two through some matrix operation are equal

$$E = y^T y - 2y^T X\beta + \beta^T X^T X\beta$$

Now we try to get minimum value of loss fuction by putting slope as 0 once and intercept as 0 once.

$$\frac{dE}{d\beta} = \frac{d}{d\beta}\left[\underbrace{y^T y}_{=0} - 2y^T X\beta + \beta^T X^T X\beta\right] = 0$$

$$= 0 - 2y^T X + \frac{d}{d\beta}\left[\beta^T X^T X\beta\right] = 0$$

problem ↑ This is done through matrix differentiation

$$= -2y^T X + 2X^T X\beta^T = 0$$

$$y^T X = X^T X\beta^T$$

$$\beta^T = y^T X\left(X^T X\right)^{-1}$$

$$\left(\beta^T\right)^T = \left(\left(X^T X\right)^{-1}\right)^T X^T y$$

$$\beta = \left(X^T X\right)^{-1} X^T y$$

β = XTrain yTrain

β

β_0

β_1

β_n

- Why Gradient Descent ?

Sklearn To solve Linear Regression

- Ordinary least squares
- Gradient Descent

Vimp

Due to inverse operation the time complexity of algo become $O(n^3)$.

This in general will make our Algo slow

That's why we use Gradient Descent.