Multiple Linear Regression

LR → Simple, Multiple, Polynomial

Simple: x | y

Multiple: x_1 | x_2 | x_3 | y

cgpa | gender | iq | lpa

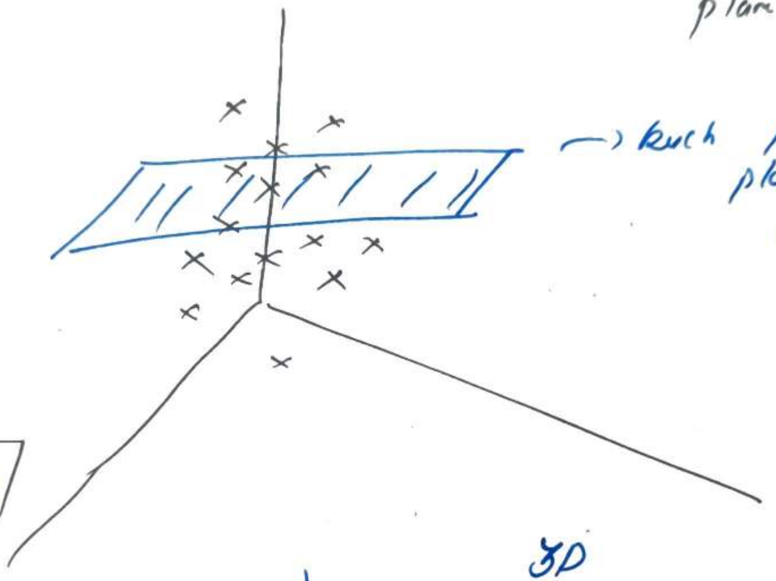
x_1 | x_2 | y → cgpa | iq | lpa

3D → plane

4D → Hyperplane

$y = mx + b$

3D



→ kuch points plane se upar or niche.

$y = mx_1 + nx_2 + b$

$$\boxed{y = \beta_0 + \beta_1 x_1 + \beta_2 x_2}$$

$\boxed{\beta_0, \beta_1, \beta_2}$ *

4D

$$\boxed{y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3}$$

nD

$$\boxed{y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_n x_n}$$

$$\boxed{y = \beta_0 + \sum_{i=1}^{n} \beta_i x_i}$$

x_1 → cgpa
x_2 → iq
y → lpa

3D

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2$$

$$\boxed{lpa = \beta_0 + \beta_1 \times cgpa + \beta_2 \times iq}$$

Coeff. are like weights.

$\beta_2 \to x_2$ bata raha hai ki lpa calculate karne mai cgpa kitna contribute karega.

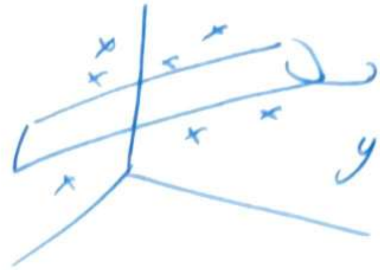
$\beta_0 \to$ offset.

~~Hyperplane~~

Mathematical Formulation

Multiple

$\boxed{\text{find } \beta_0, \beta_1, \dots, \beta_n}$



2D plan

$y = \beta_0 + \beta_1 x_1 + \dots + \beta_n x_n$

cgpa	iq	gender	lpa
x_1	x_2	x_3	y

4D Column

$y = mx + b \implies \hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3$

$y = \beta_0 + \beta x$

predicted

100 student

(100, 4)

Assume $\beta_0, \beta_1, \beta_2, \beta_3$ found

to find $\hat{y}$

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

$\hat{y}_1 = \beta_0 + \beta_1 x_{11}$

Assume 100 rows → n rows

3 cols → m cols

$$\hat{y} = \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_n \end{bmatrix} = \begin{bmatrix} \beta_0 + \beta_1 x_{11} + \beta_2 x_{12} \dots + \beta_m x_{1m} \\ \beta_0 + \beta_1 x_{21} + \beta_2 x_{22} \dots \beta_m x_{2m} \\ \vdots \\ \beta_0 + \beta_1 x_{n1} + \beta_2 x_{n2} \dots + \beta_m x_{nm} \end{bmatrix}$$

$$= \begin{bmatrix} 1 & x_{11} & x_{12} & \dots & x_{1m} \\ 1 & x_{21} & x_{22} & \dots & x_{2m} \\ \vdots \\ 1 & x_{n1} & x_{n2} & \dots & x_{nm} \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_m \end{bmatrix}$$

** $\boxed{\hat{y} = X\beta}$ — (1)

prediction of all row

Coeff. ka Matrix

$$\begin{array}{cccc} & cgpa & gradi & iq \\ 1 & x_{11} & x_{12} & x_{13} \\ 1 & x_{21} & x_{22} & x_{23} \\ 1 & x_{31} & x_{32} & x_{33} \\ \vdots & \vdots & & \end{array}$$

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$ actual output value

$$e = Y - \hat{Y} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{bmatrix} - \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_n \end{bmatrix}$$

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

Sing LR

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

Lum

** $\boxed{E = e^T e}$

$$(y_1 - \hat{y}_1)^2 + (y_2 - \hat{y}_2)^2 + \ldots + (y_n - \hat{y}_n)^2$$

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

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

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

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

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

$$= \left[Y^T - (X\beta)^T\right](Y - X\beta)$$

$$= Y^T Y - Y^T X\beta - X\beta^T Y + (X\beta)^T X\beta$$

$\mathcal{E}$ = ~~**~~ $y^T X\beta$ and $(X\beta)^T y$ equal

$y = A$
$X\beta = B$

$A^T B = B^T A$ -(1)

$(A^T B)^T = B^T A$ -(2)

$A^T B = C \Rightarrow C = C^T$

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

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

$y = \begin{bmatrix} - \\ - \\ - \\ - \\ - \end{bmatrix}_{n\times 1} \rightarrow 1\times n$

$1\times n \ (n\times(m+1))((m+1)\times 1)$

$= (1\times m+1)(m+1\times 1)$

$= (1\times 1)$

$X\beta \rightarrow n\times(m+1)$

$= T$

$[\,] = [\,]$

Hence equal.

*

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

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

$\Rightarrow -2y^T X + \frac{d}{d\beta}\left(\beta^T X^T X\beta\right)\Big] = 0$

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

$\Rightarrow \cancel{2}X^T X\beta^T = \cancel{2}y^T X$

$\Rightarrow \beta^T = \frac{y^T X}{X^T X} = y^T X (X^T X)^{-1}$

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

* matrix Differentiation

$y = A^T X A$

$\frac{dy}{dA} = 2XA^T$

$$\beta = \left[(X^T X)^{-1}\right]^T (Y^T X)^T$$

$$\beta = \left[(X^T X)^{-1}\right]^T (X^T Y)$$

Squar matri so Transc Sam.

$$\beta = (X^T X)^{-1} X^T Y$$

$$\boxed{\beta = (X^T X)^{-1} X^T Y}$$

$X =$ X_train

$Y =$ y_train

$$\begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_m \end{bmatrix}$$

$(m+1) \times 1$

$(m \times (m+1))^T = ((m+1) \times n)$

$(m+1 \times m+1)\ (m+1 \times n)\ (n \times 1)$

equal

$(m+1 \times n)\ (n \times 1)$

$((m+1) \times 1)$

Why Gradient Desient

SKlearn

- OLS
- Gradient Desent

(-1) inv ki

T. Complexity

$O(n^3)$

problematic using Direct formula

(✓)

Dheere Sahi value pe jaati hai

OLS

↳ ~~Static~~ linear Regression

SGDRegr

↳ Gradient Desent

```
class MyLR:
    def __init__(self):
        self.coeff_ = None
        self.intercept_ = None

    def fit(self, X_train, y_train):
        X_train = np.insert(X_train, 0, 1, axis=1)

        # Calculate coeff
        betas = np.linalg.inv(np.dot(X_train.T, X_train)).dot(X_train.T)
                                                .dot(y_train)
        self.intercept_ = betas[0]
        self.coeff_ = betas[1:]

    def predict(self, X_test):
        y_pred = np.dot(X_test, self.coef_) + self.intercept_
        return y_pred
```