Ridge Regularization

→ Known as L2 Regularization

Overfitting:

→ ML model Training data par exceptionally well perform kar raha hai

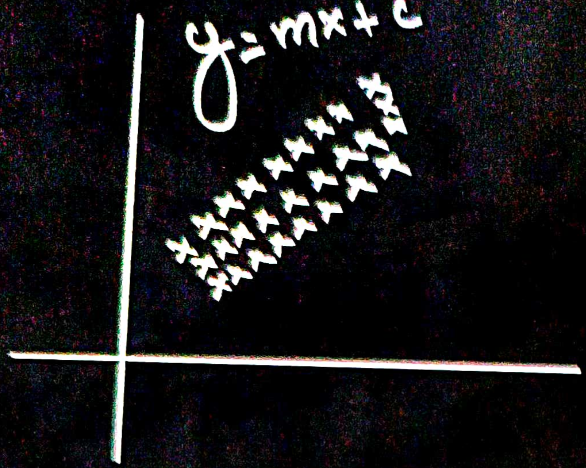
→ magar test data par it is unable to perform well

→ Implies aapke model mey variance bahut jyada hai i.e; its performance on one data set and its performance on another dataset is giving completely different results

⇒ Now coming to linear regression:

→ Idea is to find the best fit line

→ Idea is to find the best fit line

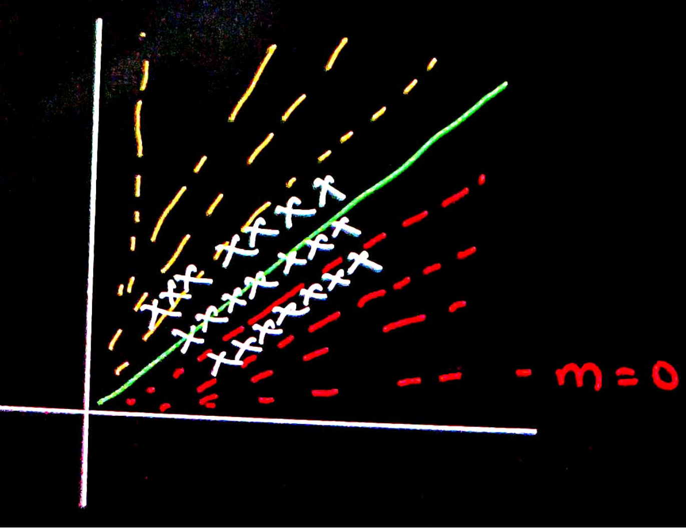


→ Ab agar check karna ho ki model overfitted hai ya underfitted
→ To check the value of m

$y = mx + c$ mey m bahut important hai kyuki use se pta chalega ki y nikaalne mey x ka kitna weightage hai.

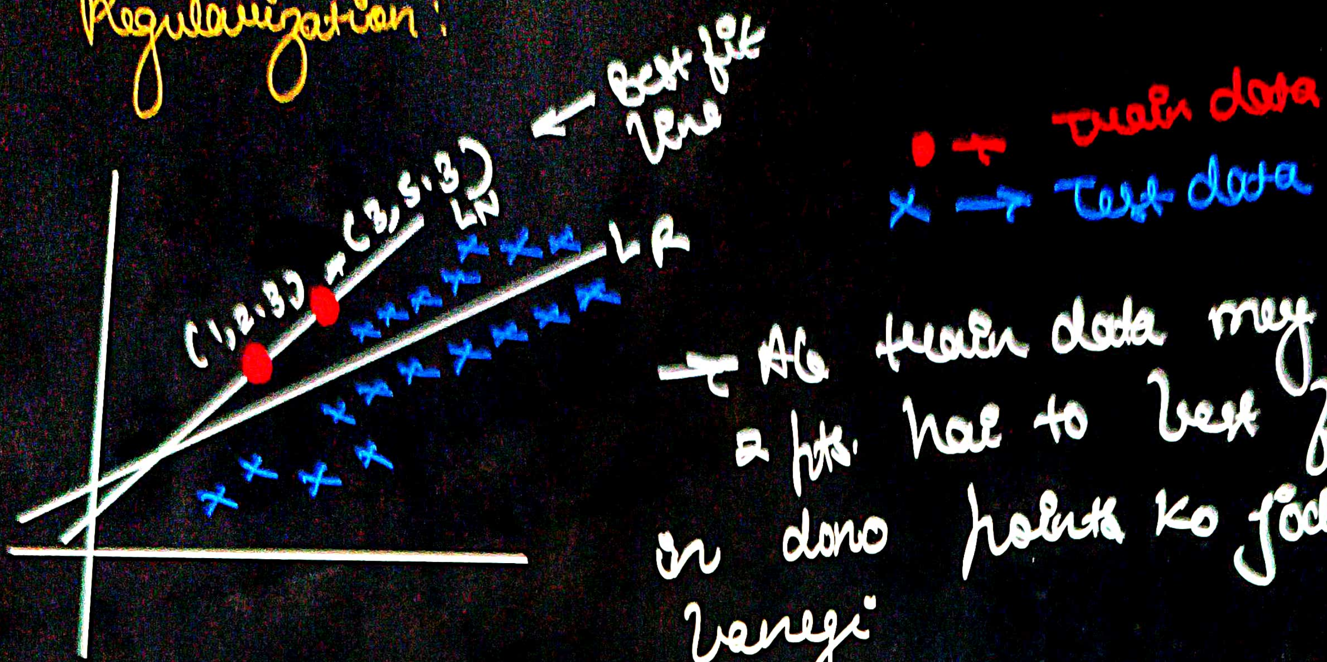
→ generally linear regression mey overfitted models mey m ka value bahut high rehta hai.

→ aur agar m ka value bahut kam hai to vo underfitted hai



— best fit line
— slope kam $= 0$ = underfit
— slope jyada (close to y) = overfit

Regularization :



→ Ab train data mey sirf 2 pts. hai to best fit line in dono points ko jod kar banegi

→ Ab whale hi line LN training data pare acha perform karega but on testing data it will perform poorly.

→ Whereas LR whale hi training data par bahut poor perform karega but testing data par result achcha dega.

→ Now what I have to do is by any means hume humare ML model ko samjhana hai ki dont choose line LN but choose line

LR :

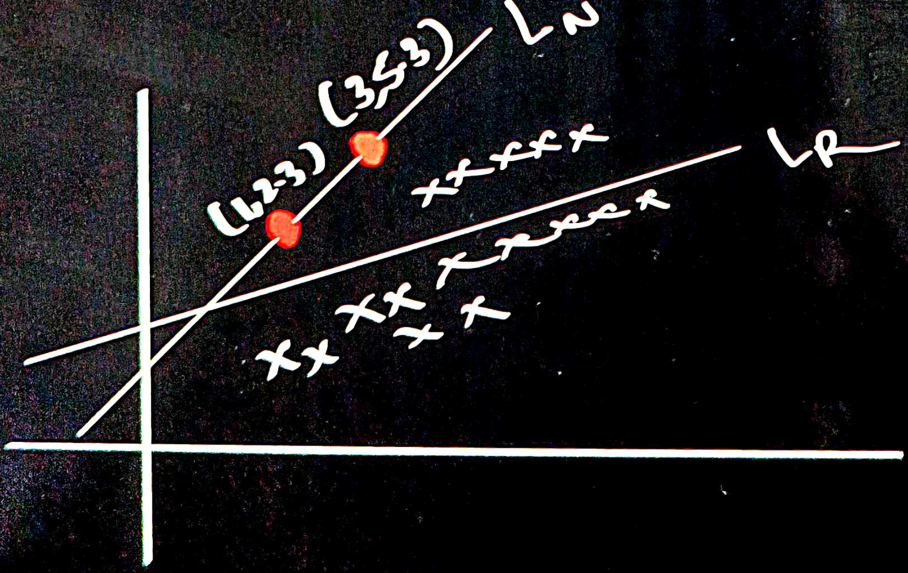
Loss funcⁿ: $$\boxed{L = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 + \lambda (m^2)}$$

λ → hyperparameter (λ)

m^2 → Slope (m)

eg: $L_N \Rightarrow y = 1.5x + 0.8$

$L_R \Rightarrow y = 0.9x + 1.5$



Calculate Loss:

(dono mey λ same)

Loss LN	Loss LR

Loss LN

$\lambda = 1$

$$L = \sum_{i=1}^{n} \cancel{(y_i - \hat{y}_i)} + \lambda(m^2)$$ → 0

$y_i = y_{predicted}$

$\hat{y}_i = y_{actual}$

Since LN dono points se pass kar rahi hai to y_i aur $\hat{y}_i$ mey koi difference hi nahi hoga

$\Rightarrow y = 1.5x + 0.8$

$y = mx + c$

$\Rightarrow m = 1.5$

$\Rightarrow 0 + \lambda(m^2)$

$0 + 1.(1.5)^2$

$\Rightarrow 0 +$

Loss LR

$\lambda = 1$

$y = 0.9x + 1.5$

$P_1 = (1, 2.3)$

$P_2 = (3, 5.3)$

$y_i = 2.3$ — (i) case

$y_i = 5.3$ — (ii) case

$\hat{y}_i$ ((i) case) $\Rightarrow$ use P_1

$\hat{y}_i = 0.9x + 1.5$

$\hat{y}_i = 0.9 \times 1 + 1.5$

$\hat{y}_i \Rightarrow 0.9 + 1.5 \Rightarrow$ case (i)

$\hat{y}_i$ ((ii) case) $\Rightarrow$ use P_2

$\hat{y}_i = 0.9x + 1.5$

$\hat{y}_i = 0.9 \times 3 + 1.5$

$\hat{y}_i = \ldots + 1.5 \Rightarrow$ case (ii)

$\hat{y} = m x + ...$
$\Rightarrow m = 1.5$

$\Rightarrow 0 + \lambda(m^2)$
$\Rightarrow 0 + 1(1.5)^2$
$= \boxed{2.25 = \text{Loss}}$

$\hat{y}_i$ (ii case) $\Rightarrow$ for Y_2

$\hat{y}_i = 0.9x + 1.5$

$\hat{y}_i = 0.9 \times 3 + 1.5$

$= 2.7 + 1.5 \rightarrow$ case (ii)

now, Loss =

$$L = \sum_{i=1}^{2} (y_i - \hat{y}_i)^2 + \lambda(m^2)$$

$\Rightarrow (2.3 - 0.9 - 1.5)^2 +$
$(5.3 - 2.7 - 1.5)^2 +$
$1 \cdot (0.9)^2$

$\Rightarrow \boxed{L = 2.03}$

$\Rightarrow$ Ab hoga kya ki aapka ML model LR ko choose karega LN ko nahi kyuki loss ki value kam aa rahi hai

⇒ Ab hoga kya ki aapka ML model LR ko choose karega LN ko nahi kyuki loss ki value kam aa rahi hai

⇒ Loss ki value kyu kam aa rahi hai kyuki aapne regularization ($\lambda(m^2)$) ki value add ki hai

⇒ Bias - Variance - Tradeoff

⇒ Ab isey L_2 kyu bolte hai?

means agar multiple axis rahi to what we will do is

single axis ⇒ $\lambda(m^2)$

double axis ⇒ $\lambda(m_1^2 + m_2^2)$

multiple axis ⇒ $\lambda(m_1^2 + m_2^2 + m_3^2 + \cdots)$

what we will do is

single axis $\Rightarrow$ $\lambda(m^2)$

double axis $\Rightarrow$ $\lambda(m_1^2 + m_2^2)$

multiple axis $\Rightarrow$ $\lambda(m_1^2 + m_2^2 + m_3^2 + \ldots)$

$\Rightarrow$ implies har slope ke term ko square (L^2) kar rahe hai isiliye isey L_2 norm se bolaya jaata hai

$\rightarrow$ Lasso ko L_1 se jaana jaata hai kyuki waha slope ka square nahi balki absolute value use ki jaati hai