ML M1

Famous
Page No.:
Date.:

Feature Engineering Model

* Simple linear Regression *

Supervised ML
- → Regression — o/p OR Dependent feature ⇒ Continuous
- → Classification — O/p OR Dependent feature ⇒ Categorical

Dataset

Independent feature	Dependant feature
Weight	hight
74	170
80	180
75	175.5
–	–
–	–

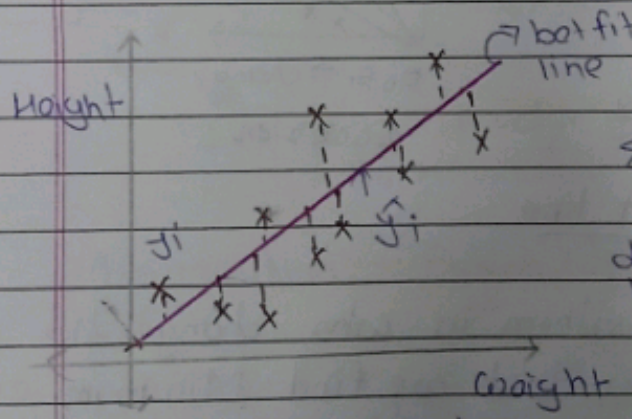
How Model work

Train Dataset

(Prediction)

Max weight → Model → height

Accuracy ↑↑



→ best fit line predictions

Error ↓↓ Minimal

distance between → best fit line and y_i = error

y_i = Actual Point

$\hat{y}_i$ = Predicted Point

Q What is the mean AIM of Simple Linear Regression?

↳ The mean AIM of simple linear Regression is to find the best fit line in such a way when we do the sumition of Actual Point and Predicted Point find the difference between tose must be minimal

Shot on Y12
Vivo AI camera

$\hat{y} = mx + c$ (Indicate best fit line)

$\hat{y} = \beta_0 + \beta_1 x_1$

$\boxed{h_\theta(x) = \theta_0 + \theta_1 x_1}$

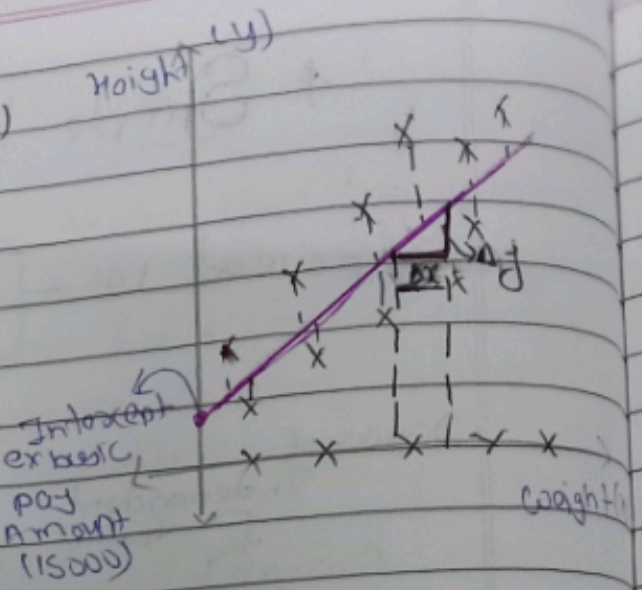
$\hat{y}_i = h_\theta(x)$ = predicted point

$\theta_0 \Rightarrow$ Intercept

$\theta_1 \Rightarrow$ Slope or coefficent

$x_1 \Rightarrow$ Datapoint

$\boxed{\text{Slope} = \frac{dy}{dx}}$



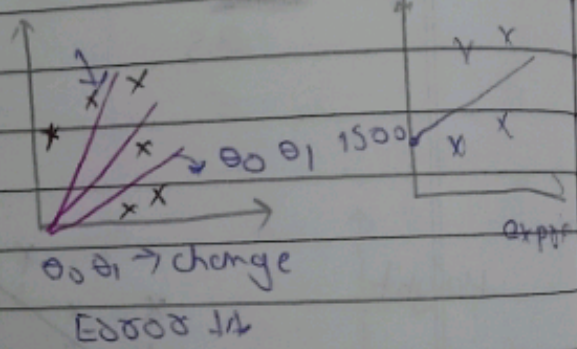
simple linear Regression

↓

Model

↓

Hypo Thesis Testing



* How to get best fit line

We create an algorithem in who change the values of θ_1 and θ_2 so that we find Minimum error और जब अपने कम error वाला find होगा तो हमे लास्ट best fit line बन जायेगा।

और stop हो जायेगा → Algorithem

Shot on Y12
Vivo AI camera

Date:

→ Why use to minimum error

- **Cost function** [Error]

$$\boxed{J(\theta_0, \theta_1) = \frac{1}{n} \sum_{i=1}^{n} (\underbrace{y_i}_{\text{Actual}} - \underbrace{h_\theta(x)_i}_{\text{predicted}})^2}$$

[Mean Squared Error]

predicted ($\hat{y}_i$)

n = no. of Data points
y_i = Actual value
$h_\theta(x)$ = Predicted value

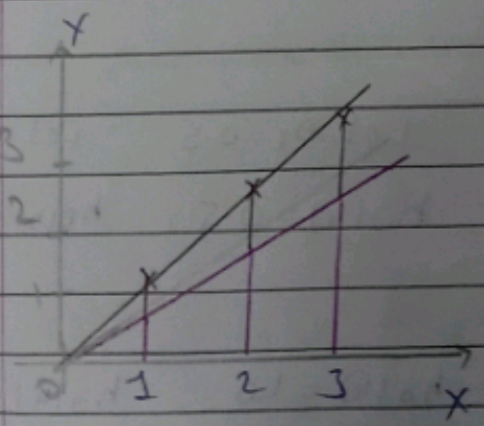
- **Final Aim** → [In order to get best fit line]

What to do:-

Minimize ⇒ $J(\theta_0, \theta_1) = \frac{1}{n} \sum_{i=1}^{n} (y_i - h_\theta(x)_i)^2$ ↓↓↓

by changing the value of θ_0 θ_1

Minimization - Error ↓↓↓



$h_\theta(x) = \theta_0 + \theta_1 x_1$

Let consider $\theta_0 = 0$

$h_\theta(x) = \theta_1 x_1$

Actual Dataset

x	y	$h_\theta(x)$
1	1	
2	2	
3	3	

Let $\theta_1 = 1$

Predicted
$x = 1$, $h_\theta(x_1) = 1(1) = 1$
$x = 2$, $h_\theta(x_2) = 1(2) = 2$
x_3, $h_\theta(x_3) = 1(3) = 3$

Shot on Y12
Vivo AI camera

Page No.:
Date.:

$$J(\theta_0, \theta_1) = \frac{1}{n} \sum_{i=1}^{n} (\text{Actual} - (\text{Pricted})^2)$$

* Cost function for this dataset

For $\theta_1 = 1$

$$J(\theta_1) = \frac{1}{n} \sum_{i=1}^{n} (y_i - h_\theta(x_i))^2$$

$$= \frac{1}{3} \left[(1-1)^2 + (2-2)^2 + (3-3)^2 \right]$$

$J(1) = 0$

cost fun^n $\theta_1 = 0.5$

$$J(\theta_1) = \frac{1}{3} \left[(1-0.5)^2 + (2-1)^2 + (3-1.5)^2 \right]$$

$J(\theta_1) = 1.16$

cost fun^n $\theta_1 = 0$

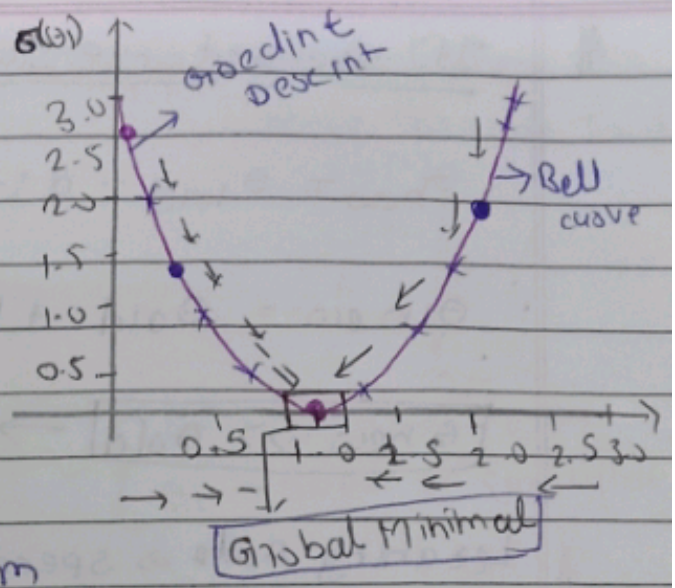
$$J(\theta_1) = \frac{1}{3} \left[(1-0)^2 + (2-0)^2 + (3-0)^2 \right]$$

$J(\theta_1) = 4.66$

	Let $\theta_1 = 1$	Let $\theta_1 = 0.5$	Let $\theta_1 = 0$
$x = 1$	$h_\theta(x) = 1(1) = 1$	$h_\theta(x) = 0.5$	$h_\theta(x) = 0$
$x = 2$	$h_\theta(x) = 1(2) = 2$	$h_\theta(x) = 1.0$	$h_\theta(x) = 0$
$x = 3$	$h_\theta(x) = 1(3) = 3$	$h_\theta(x) = 1.5$	$h_\theta(x) = 0$

predicted

Shot on Y12
Vivo AI camera



Convergence Algorithem

{optimize the change of θ_0, θ_1 to Global Minima

↓

Repeat until convergence

$$\theta_j := \theta_j - \alpha \boxed{\frac{\partial}{\partial \theta_j} J(\theta_j)}$$

→ Derivative slope of a point

$\boxed{\alpha = 0.01}$ Learning Rate → Speed of convergence

↓

* value data convergence की speed देता है।

If α value ↑↑ → en of day point out (Not Reach at Global Minimal

If α value → Minimum (Best) → Process to take time

* If tesion st Right side → facing down → -ve slope

* If tesion left side → facing upper → +ve slope.

Shot on Y12
Vivo AI camera

Page No.:
Date.:

* If we get -ve slope

$\theta_{new} = \theta_{old} - \alpha(-ve)$

$\theta_{1new} = \theta_{old} + (value)$

$\boxed{\theta_{new} >> \theta_{old}}$ → इस case में θ की value Decrease की जायेगी Toward to get global minima.

Learning Rate ⇒ speed of convergence

If we got +ve slope

$\theta_{1new} = \theta_{old} - \alpha(+ve)$

$\theta_{1new} = \theta_{old} - (value)$

$\boxed{\theta_{1new} << \theta_{old}}$ इस case θ की value Decrease की जायेगी Towards to get global minima.

Conclusion:-

Repeat until convergence

$$\left\{ \theta_J = \theta_J - \alpha \frac{\partial}{\partial \theta_J} J(\theta_0, \theta_1) \right\}$$

$$\boxed{J(\theta_0, \theta_1) = \frac{1}{n} \sum_{i=1}^{n} (y_i - h_\theta(x_i))^2}$$

↓ Mean squared error

Shot on Y12
Vivo AI camera

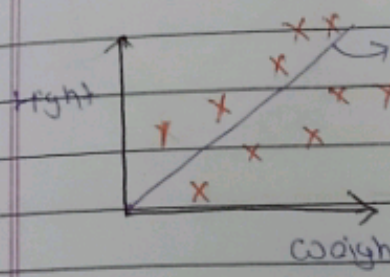
Page No.:
Date.:

* Multiple linear Regression * [where are more the one Feature (I/P)]

Dat aset

I/P → weight

O/P → hight



$h_\theta(x) = \theta_0 + \theta_1 x_1$

$\theta_0 \rightarrow$ Intercept

$\theta_1 \rightarrow$ Slope or coefficient

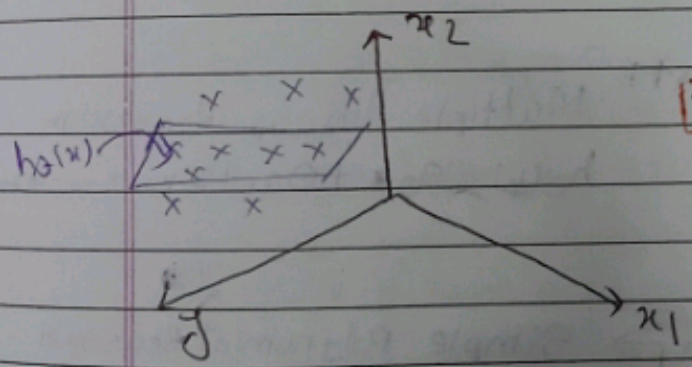
Ex:- House pricing Dataset

Independent: x_1, x_2 — Dependent feature: y

x_1: No. of Room

x_2: Size of Room

y: Price



$h_\theta(x) = \theta_0 x_1 + \theta_0$

$$\boxed{h_\theta(x) = \theta_0 + \theta_0 x_1 + \theta_2 x_2}$$

$\theta_1, \theta_2 \Rightarrow$ Slope or cofficent

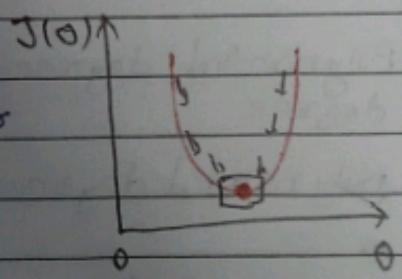
$\theta_0 \rightarrow$ Intercept

* Generic equation of Multiple linear Regression

$$\boxed{h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \cdots \theta_n x_n}$$

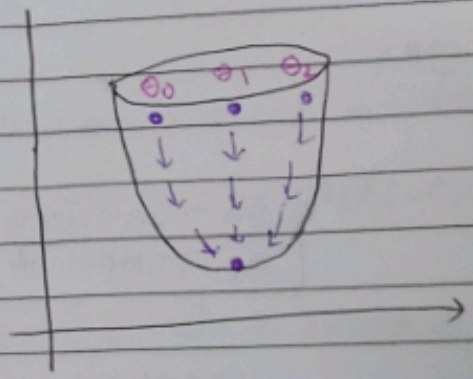
Simple linear Regression

$h_\theta(x) = \theta_0 + \theta_1 x_1 \Rightarrow$ for

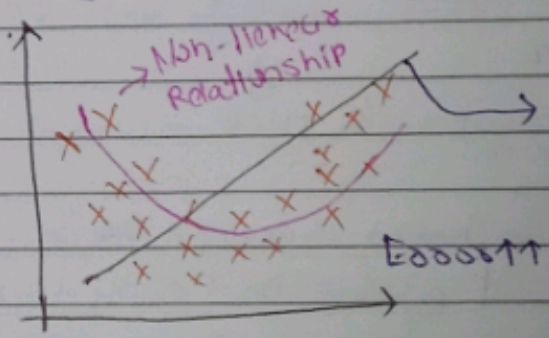


Shot on Y12
Vivo AI camera

For Multiple linear Regression



* Polynomial Regression *



Simple Linear Regression

$$\boxed{h_\theta(x) = \theta_0 + \theta_1 x_1}$$

Multiple linear Regression

$$h_\theta(x) = \theta_0 x + \theta x_1 + \theta_2 x_2 + \cdots \theta_n x_n$$

Interview

Polynomial Regression → Simple Polynomial Regression
→ Multiple polynomial Regression

⇓

Polynomial Degre

* 1) Simple Polynomial Regression ⇒ $h_\theta(x) = \theta_0 + \theta x_1$

Polynomial degree=0 degree $h_\theta(x) = \theta_0 x_1 = \theta_0 x_1^0$ ⇒ constant

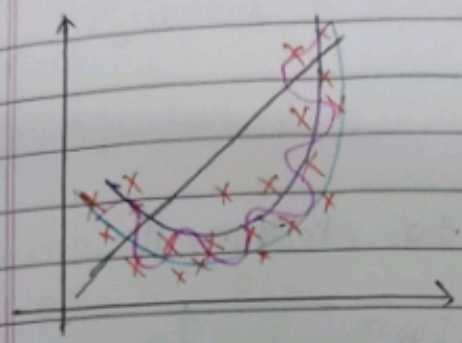
polynomial degree=1 $h_\theta(x) = \theta_0 x_1^0 + \theta_1 x_1^1$ ⇒ simple linear Regression

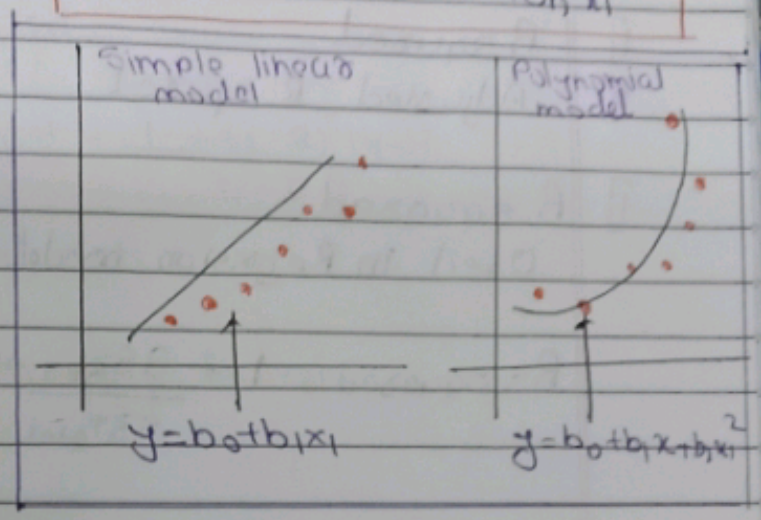
Polynomial degree=2 $h_\theta(x) = \theta_0 x_1^0 + \theta_1 x_1^1 + \theta_2 x_1^2$

$h_\theta(x) = \theta_0 x_1^0 + \theta_1 x_1^1 + \theta_2 x_1^2 + \theta_3 x_3^3$

Shot on Y12
Vivo AI camera

polynomial degree = n $\boxed{h_\theta(x) = \theta_0 x_1^0 + \theta_1 x_1^1 + \theta_2 x_1^2 + \dots + \theta_n x_1^n}$





2] Mutiple Polynomial Regression

$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_2$ [Multiple Linear Regression]

Polynomial Degree = 2

$$\boxed{h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \theta_4 x_1^2 + \theta_5 x_2^2 + \theta_6 x_3^2}$$

Shot on Y12
Vivo AI camera

* Performance Metrix Used in Regression:-

1] R squared
2] Adjusted R squared

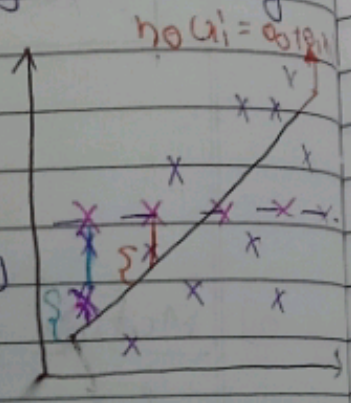
1] R squared

Used in Regression model to checking its accuracy

$$R\text{-squared} = 1 - \frac{SS_{Res}}{SS_{Total}}$$

$SS_{Res} \rightarrow$ Error

$SS_{Total} \rightarrow$ Average of y_i line $\rightarrow$ find error



$SS_{Res} \rightarrow$ Sum of square Residual (Error)

$SS_{Total} \rightarrow$ Sum of square Total

$$R_{squared} = 1 - \frac{\overbrace{\sum_{i=1}^{n} (y_i - h_\theta(x))^2}^{SS_{Res}}}{\sum_{i=1}^{n} (y_i - \bar{y}_i)^2}$$

$\rightarrow$ small

$\rightarrow$ Big

$$\boxed{R\text{ squared} = 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2}}$$

R squared range between 0 to 1

Highly correlated feature

(Low correlation)

No of Room | Size of Room | Location | state | Gender | Price

f_1 | f_2 | f_3 | f_4 | f_5 | O/P

$f_3 = 0.85$ $f_4 = 90\%$ $f_5 = 0.2$

$f_1 = R$ squared $= 75\% = 0.75$

$f_2 = $ squared $= 80 = 0.80$

Shot on Y12
Vivo AI camera

Good R - squared = 85% = 0.85
R - squared = 90% = 0.90
R - squared = 95% = 0.95

$\boxed{\text{correlation} \uparrow\uparrow = \text{R. squared} \uparrow\uparrow}$

2] Adjusted R squared

$$\text{Ajusted R. squared} = 1 - \frac{(1-R)^2 (N-1)}{N-P-1}$$

N = no. of data points
R^2 = R squared
P = No. of Independent feature.

Ex:-

$R^2 = 0.8$, $N = 11$, $P = 2$

$$\text{Adjusted R squared} = 1 - \frac{[(0.2)(10)]}{11-2-1} = 0.75$$

$R^2 >> \text{Ajusted } R^2$

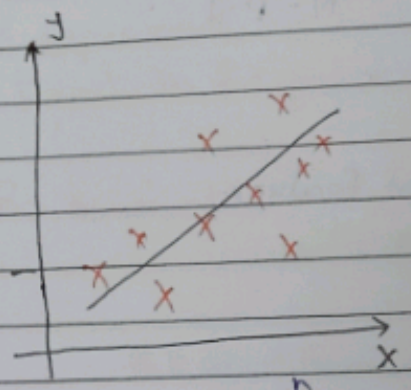
Adding P_3 (IMP) $R^2 = 80\%$ Adjusted $R^2 = 75\%$
$R^2 = 85\%$ Adjusted $R^2 = 78\%$

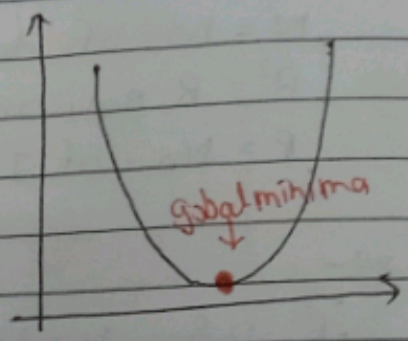
Adding P_4 $\uparrow\uparrow\uparrow$ $R^2 = 87\%$ Adjusted $R^2 = 76\%$ $\downarrow\downarrow\downarrow$
Independent feature is not that important

Shot on Y12
Vivo AI camera

* MSE, MAE, RMSE [cost function]

1] Mean Squared Error [MSE]
2] Mean Absolute Error [MAE]
3] Root Mean Squared [RMSE]





$$J(\theta_0, \theta_1) = \frac{1}{n} \sum_{i=1}^{n} (y_i - h_\theta(x_i))^2$$

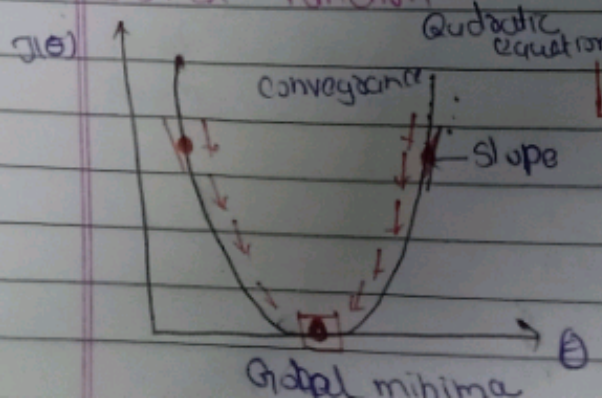
⇓
Mean squared Error

1] Mean squared Error (MSE)

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (y_i - h_\theta(x)_i)^2$$

$$\boxed{MSE = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2}$$ → Quadratic Equation / convex function

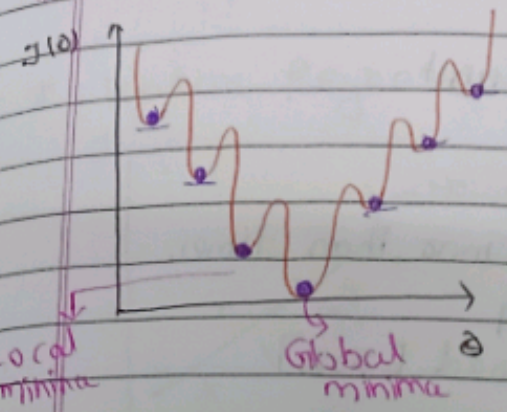
* Convex function



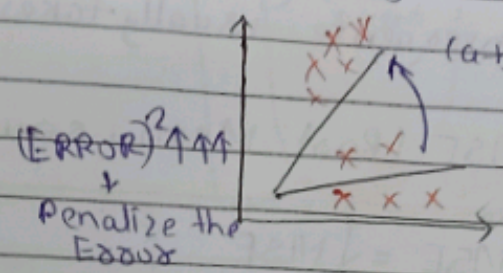
1] to make convergence
2] to calculating slope

Shot on Y12
Vivo AI camera

* No Convex function :-



In local minima's case मे मेरी रुक जाता है, मुझ से weight change (w) कर आगे लिए बहुत मुश्किल हो जाता है।



* MSE

* Advantage

1] Equation is differentiable so that it helps in finding the slope and then in calculating the convergence.

2] It has only one Local or Global minima.

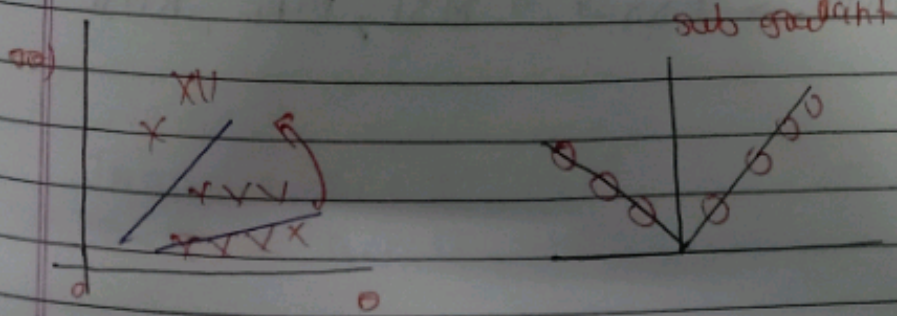
* Disadvantage

1] Not Robust to outliers

2] It is not in the same unit.

2] Mean absolute Error [MAE]

$$\boxed{MAE = \frac{1}{n}\sum_{i=1}^{n}|y - \hat{y}|}$$



Date:

* Advantage
1] Robust to outliers
2] It will be in the same unit

* Disadvantage
1] convergence usually takes more than times.

3] RMSE (Root Mean, squared Error)

$$\boxed{RMSE = \sqrt{MSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - h_\theta(x)_i)^2}}$$

* Advantage
1] same unit
2] Diffentiable

* Disadvantage
1] Not Robust to outliers

Note: Linear Regression

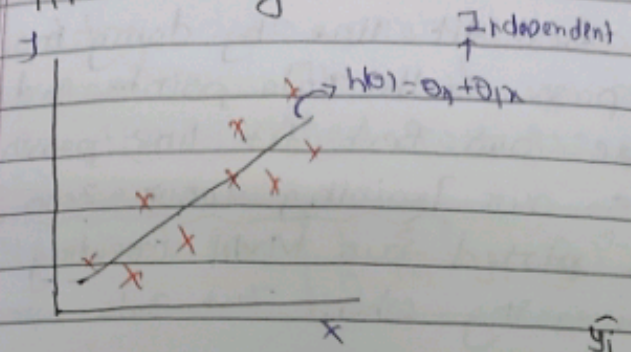
Performace metrices = R^2 and Adjusted R^2 → Acc of model

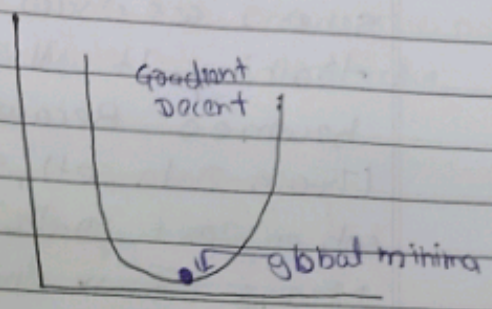
cost function → Error → MSE, MAE, RMSE

Shot on Y12
Vivo AI camera

* Ridge / Lasso And Elasticnet Regression } Sets of Regression

* linear Regression





$$\boxed{\text{Cost function} = \frac{1}{n}\sum_{i=1}^{n}(y_i - h_\theta(x)_i)^2}$$ [Mean Squared Error]

(y_i)

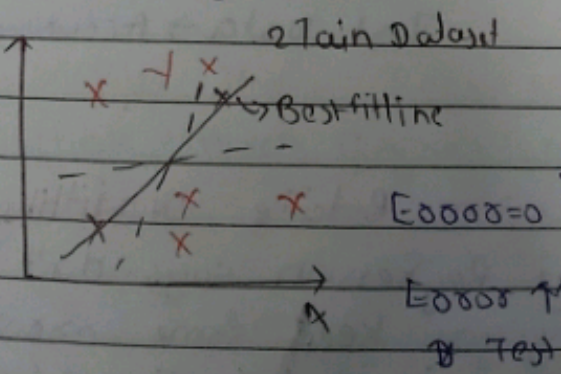
1) Ridge Regression (L_2 Regularization) :-

→ [Reducing overfitting]

* Why Using Ridge Regression

It is used for Reducing overfitting

* overfitting *



Dataset → Train, Test

overfitting

TRAIN → ACC ↑↑ → R^2

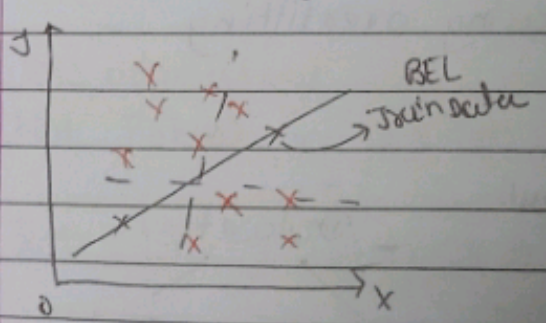
TEST → ACC ↓↓ → R^2

In this case:-

There are two train Data set ploted. In this senerio we draw Best fit line by doing model training. It will pass both Data points. And error becomes 0. Because our Best fit line passing points (Train Data set), So our Training error = zero, But when Test Data ploted our Model creating Error Means Error increasing when Test Data comes.

In short:-

In the training Data our model performe very well (error = 0), But In Test Data our model performence Decreasing (Error = ↑↑↑)
This senerio called ⇒ overfitting, when In New data come this model don't perform well because of overfitting.



overfitting

Train Data → Accuracy ↑↑
Test Data → Accuracy ↓↓

Accuracy Check by R^2 OR Adjusted R^2

* How Ridge Regression Reduce overfitting *

In overfitting Ridge Regression says. don't make error by doing something you keep some error and find Best fit line in dotted form By this our model perform well some error and Predict train Data easily

Famous
Page No.:
Date.:

* What to do Ridge model *

We modify our cost fuction. In modifacation we adding two permeter so that our error does not come zero. There always avaible some cost value, so the Modified cost is:

New cost fun $\boxed{\frac{1}{n}\sum_{i=1}^{n}(y_i-\hat{y}_i)^2+\lambda\sum_{i=1}^{n}(\text{slope})^2}$

λ → slopes parameters

Let In

In simple linear Regression (cost function)

(∵ simple linear Regression = slope = θ_1)

↳ $h_\theta(x) = \theta_0 + \theta_1 x$ ↳ slope

In Intialse $\lambda=1$, slope θ_1, Best fit line error $=0$

So, Now

Cost function $= 0 + 1[\theta_1]^2$ by this we penalize the cost function so that error $\neq 0$

* In Multiple linear Regression * (Cost Function)

Multiple linear Regression equation $= h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3$

So

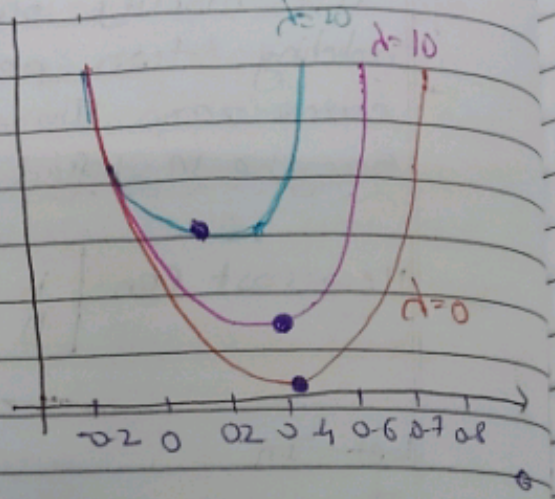
$$\boxed{\text{Cost function} = \frac{1}{n}\sum_{i=1}^{n}(y_i-\hat{y}_i)^2+\lambda\left((\theta_1)^2+(\theta_2)^2+(\theta_3)^2\right)}$$

Shot on Y12
Vivo AI camera

Page No.:
Date.:

* Relation between λ and slope:

λ=0, (Then it is MSE) J(θ)
(Simple linear Regression)



But,
when we increasing λ (Landa's value, we observed that In Gradient Desent our line shift upward, and θ value Decressing but never get 0.
So we can say that:
When

$$\boxed{\lambda \uparrow\uparrow = \text{Slope} \downarrow\downarrow}$$

By λ and slope we Reducing overfitting of the mold

B] Lasso Regression (L1 Regularization): {feature selection}

Lasso regression also called L1 Regularization. It it used for feature selection process. The feature who does not too much correlated with output feature. It coefficient is small

In the conversion algorithm as we increase the value of λ Lamda(λ) θ value became decrease at one point of time the feature who does not much correlated that's θ value becomes zero. By this we directly Remove that feature and use remaining feature for the model.

Shot on Y12
Vivo AI camera

In this way Lasso regression works

cost fn for Lasso Regression $\boxed{\frac{1}{n}\sum_{i=1}^{n}(y_i-\hat{y}_i)^2+\lambda\sum_{i=1}^{n}|\text{slope}|}$

* In Multiple linear Regression *

$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3$

Lets,

$\theta_0 = 0.5$, $\theta_2 = 1.5$, $\theta_3 = 0.2$

$= 0.5 + 1.5x_2 + \boxed{0.2}\, x_3$ → Feature remove → Feature selection

Lets, In

x_1 → unit moment

y → 1.5 x_2 move

→ 1.5 time moves x

x_3 → unit move

y_3 → 0.2 x_3

$0.5 + 1.5x_2 + 0.2\, x_3$ x_3 → 1 unit move

y_3 → 0.2 x_3

0.2 and y_3 not too much correlated so Remove x_3 feature This process called "feature selection".

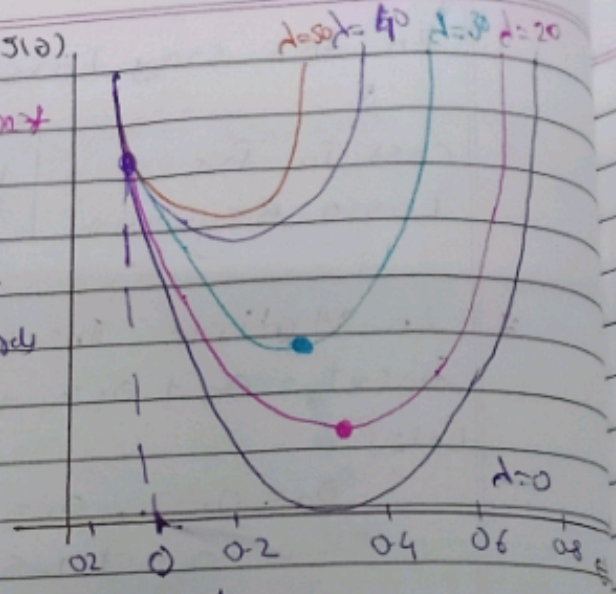
* How to Remove Feature :-

Cost fn → $\boxed{\frac{1}{n}\sum_{i=1}^{n}(y_i-\hat{y}_i)^2+\lambda\sum_{i=1}^{n}|\text{slope}|}$ Hyperparameter

Lets $\lambda = 1$

cost fn = $[\text{Error} + 1[|\theta_1| + |\theta_2| + |\theta_3|]]$

Date:



* Relation between λ and slope *

When we increase the value of λ our coefficient value (θ) Decreasing towards zero

The feature who is not too much correlated By cost-faction we saw when be optimized and merging then - we Lasso regression it will Reduce the 0.2 (smaller) θ's value exactly, to the zero and smaller θ's detected by Lasso regression

So we can say that

When

$$\boxed{\lambda \uparrow\uparrow \quad \theta \downarrow\downarrow \rightarrow \text{It will become } 0}$$

3] Elastic Net Regression:-

In this Regression we combine both Ridge and Lasso Regression

Elastic Net Regression → Ridge → Reducing overfitting
→ Lasso → Feature selection

$$\boxed{\text{Cost fn} = \underbrace{\frac{1}{n}\sum_{i=1}^{n}(y_i-\hat{y}_i)^2}_{\text{MSE}} + \underbrace{\lambda_1\sum_{i=1}^{n}(\text{slope})^2}_{\text{Reducing overfitting}} + \underbrace{\lambda_2\sum_{i=1}^{n}|\text{slope}|}_{\text{Feature selection}}}$$

Shot on Y12
Vivo AI camera