① Linear Regression

$$L = \frac{1}{2n}(x_i - \hat{x}_i)$$

Cost fxn / or Loss fxn / or Error fxn

$$L = \frac{1}{2n}\sum_{i=1}^{n}(\text{actual} - \text{predicted})^2$$

predicted - actual

↳ $(mx+c)$

why this $\frac{1}{2n}$? extra $\frac{1}{2}$?

↳ During gradient calculation that is $\frac{\partial}{\partial} L$, square term reduced

$\Rightarrow 2$ so, $2 \times \frac{1}{2n} \Rightarrow \frac{1}{n}\sum(y-\hat{y}_i)^2$ ↳ MSE

Convergence Algo

↳ $w_{new} = w_{old} - \eta \frac{\partial L}{\partial w_{old}}$

Q Why optimizers are used in DL not in ML?

↳ Optimizer speed up the training & prevent models from getting stuck in local minima

↳ ML model are simple so, don't face local minima problem mostly.

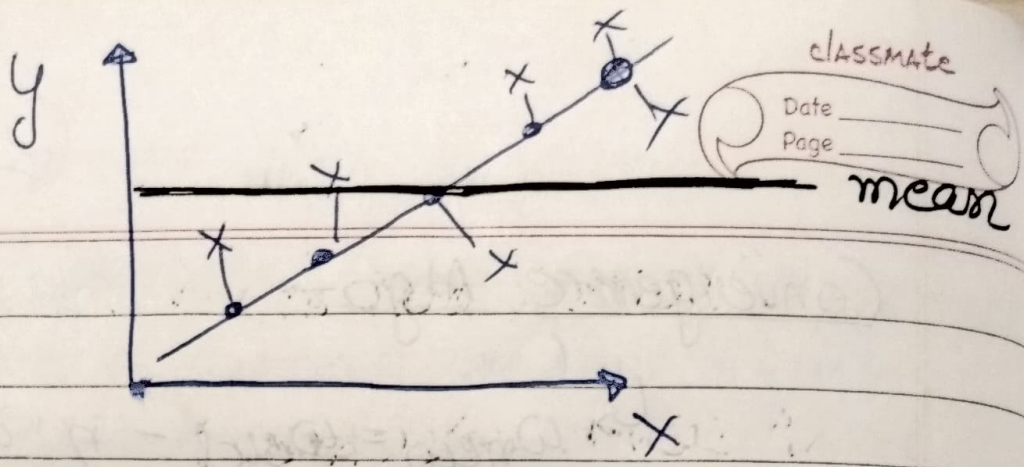
Performance Matrix for L.R

① $R^2 = 1 - \frac{\sum (actual - predicted)^2}{\sum (actual - mean)^2}$ → of y.

↳ Coefficient of Determination

$R^2 = 1$ => perfect fit => model explain all the variance in data

$R^2 = 0$ => Model explain no variance (as



bad as predicting mean)

$R^2 < 0$ => Model is worse than just using mean as a predicto

Q Why we use adjusted R^2 rather than R^2?

↳ Adding more features always ↑ R^2, even if the new variable are useless.

↳ Adjusted R^2 fixes the issue by penalizing model with unnecessary variables

$$\text{Adjusted } R^2 = 1 - \frac{((1-R^2)(n-1))}{n-f-1}$$

n = no. of datapoints
f = no. of feature

--> R^2 (with 2 feature) => 0.80
Adjusted R^2 => 0.80
--> R^2 (with 10 feature) => 0.90
--> Adjusted R^2 => 0.75

Overfitting

↳ (1) Lasso & Ridge Regularization
↳ Absolute + square value

★ Lasso also used for feature selection

Things to Do Before Applying Linear Regression

i) Check if the Data is Linearly separable
(i) Handle Missing value
(ii) Feature Selection => Check correlation
(iii) Check outlier => Boxplot.
(iv) Scale Feature => Standardization (zScore)
↳ mean = 0, var = 1
↳ Normalization
↳ scaling data b/w 0 & 1.

(v) Split data into Train & Test