apsara

Date: __.__.__

* Linear Regression

Linear Regression → method to find straight-line relationship b/w input (X) & output (Y).
↳ If X changes, how does Y changes

* It is used only when
  ↳ output is continous value
  ↳ Dataset must be linear in nature
  i.e. Relationship b/w X & Y is straight line.

$y = mx + c$ → eq^n of straight line

in ML,

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

- x → input feature
- $\hat{y}$ → y - predicted → output
- β_0 → intercept → value of y at $x = 0$.
- β_1 → slope / weight.

* How it works

Tries many possible lines and measure error and continue doing till error converges and picks the line with minimum error.

* Loss function / Cost function → To measure total error, here we use MSE (Mean squared error).

$$\left\{ MSE = \frac{1}{n} \sum (\underset{\text{Actual}}{y} - \underset{\text{Predicted}}{\hat{y}})^2 \right\}$$

* Two types
  ↳ simple LR → One input feature $[\hat{y} = \beta_0 + \beta_1 x]$
  ↳ Multiple LR → More than one feature $[\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots]$

Page No. ____

apsara

Date:___.___.___

* Evaluation metric → MSE → mean squared error
  R^2 score → goodness of fit (0 to 1)
  0 → model explain nothing
  1 → perfect fit.

* When to use
  - Relationship is linear (can be increasing / decreasing) can't be curved / random
  - Output is continuos

* Avoid when
  - Data is highly non-linear
  - Many outliers
  - Classification problem.

$$\hat{y} = \underbrace{\beta_0 + \beta_1}_{\text{trainable}} x_1$$

Prediction ($\hat{y}$), Bias (β_0), Weight (β_1), input feature (x_1)

* Mostly training in LR follow Gradient Descent method.

idea
↳ Start with random value of β
↳ Move step by step toward min error
↳ Stop when improvement is very small.

$$\boxed{\text{Update formula} \rightarrow \theta = \theta - \alpha \frac{\partial J}{\partial \theta}}$$

[α → learning rate
  J → Cost function]

α or η both are learning rate symbol

apsara

Date: __.__.__

example → Properly random simple example just to understand functioning.

x	y
1	5
2	7
3	9
4	11

$\hat{y} = mx + c$

taking $m = 0$, $c = 0$ - (trainable).

calculate, $\hat{y}$ with x as 1, 2, 3, 4

$\hat{y} = 0$

Calculate error,

$$MSE = \frac{1}{4}\left[(5-0)^2 + (7-0)^2 + (9-0)^2 + (11-0)^2\right]$$

$$= 69 \rightarrow \text{loss}.$$

now, we will update both m & c (trainable parameter) by gradient descent.

$$\underset{m_{new}}{\underset{\downarrow}{m_n}} = \underset{m_{old}}{\underset{\downarrow}{m_o}} - \eta \frac{\partial MSE}{\partial m} \quad - (1)$$

{ Formula $\theta = \theta - \eta \frac{\partial \text{ cost Func}}{\partial \theta}$ }

$$C_n = C_0 - \eta \frac{\partial MSE}{\partial c} \quad - (2)$$

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

$$= \frac{1}{n}\sum\left(y - (mx_i + c)\right)^2 \quad (\hat{y} = mx + c)$$

Put $y - (mx_i + c) = E$

$$= \frac{1}{n}\sum (E)^2$$

now Partial derivative of MSE with m.

$$\frac{\partial MSE}{\partial m} = \frac{1}{n}\sum 2E\left(\frac{\partial E}{\partial m}\right)$$

$$= \frac{1}{n}\sum 2E(-x)$$

$$= \frac{1}{n}\sum 2(y - (mx + c))(-x)$$

$$\left\{ \begin{aligned} \frac{\partial E}{\partial m} &= \frac{\partial}{\partial m}(y - (mx + c)) \\ &= \frac{\partial}{\partial m}(y - mx - c) \\ &= -x \end{aligned} \right.$$

$= \frac{1}{n} \sum (2y - 2mx - 2c)(-x)$

$= \frac{1}{n} \sum (-2yx + 2mx^2 + 2xc)$

$= \frac{-2x}{n} \sum (y - mx - c)$

$= \frac{-2x}{n} \sum (y - (mx + c))$

$$\frac{\partial mse}{\partial m} = -\frac{2}{n} \sum x (y - \hat{y})$$

Put in ①

$$m_n = m_0 - \eta \frac{\partial mse}{\partial m}$$

$$\left\{ m_n = m_0 - \eta \left(\frac{-2}{n} \sum x (y - \hat{y}) \right) \right\}$$

calculate new m

$$m_n = 0 - 0.1 \left(\frac{-2}{4} \left[1(5-0) + 2(7-0) + 3(9-0) + 4(11-0) \right] \right)$$

$= 4.5$

- Like that we will find $C_n = C_0 - \eta \frac{\partial mse}{\partial c}$

Similarly, C_n will come out like solve as we solve for m_n.

$$\frac{\partial mse}{\partial c} = \frac{-2}{n} \sum (y - \hat{y})$$

$$C_n = C_0 - \frac{\partial mse}{\partial c}$$

$$C_n = C_0 - \eta \left(\frac{-2}{n} \sum y - \hat{y} \right)$$

- $\left\{ C_n = C_0 + \eta \left(\frac{2}{n} \sum (y - \hat{y}) \right) \right\}$

apsara

Date: __.__.__

$$c_n = 0 + 0.1\left(\frac{2}{4}(5-0 + 7-0 + 9-0 + 11-0)\right)$$

$$c_n = 1.6$$

We got $m_n = 4.5$, $c_n = 1.6$

$$eq^n : \hat{y} = 4.5x + 1.6$$

x	y	$\hat{y}$
1	5	6.1
2	7	10.6
3	9	15.1
4	11	19.6

By using new eq^n

Again calculate Loss

$$MSE = \frac{1}{n}\sum (y - \hat{y}_i)^2$$

$$= \frac{1}{4}\left((5-6.1)^2 + (7-10.6)^2 + (9-15.1)^2 + (11-19.6)^2\right)$$

$$= 31.33$$

In one iteration we decrease loss from 69 to 31.33

Like this ~~we~~ this algorithm works & keep on doing till Loss get minimal & converge properly