Coefficient of Determination (R^2)

↳ goodness of fit. $\in [0, 1]$

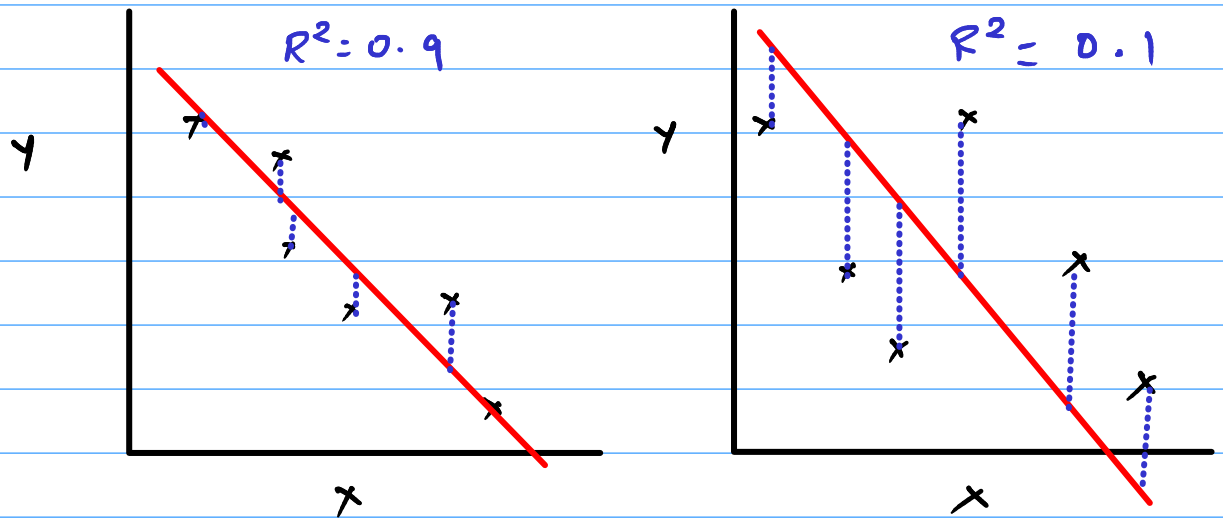
- 0: fails to model the data.
- 1: perfect fit

$R^2 = 0.2 \Rightarrow$ 20% of the dependent variable is predicted by the independent variable.

exam's score(s) $\rightarrow y$
time spent in studying $\rightarrow x$

→ $R^2 = 0$: Can't predict
→ $R^2 = 1$: perfect prediction
→ $R^2 \in (0,1)$: partial prediction.

R^2 is a goodness of fit.



NOTE: R^2 tells you how X & y are correlated with each other, but it shouldn't be taken as an evidence

for causality.

$$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n + \varepsilon$$

$$\varepsilon_i = \hat{y} - y$$

Supervised ML

- Regression
  - → Simple LR
  - → Multiple LR
- Classification
  - → Logistic Regression
  - → KNN
  - → SVM

Logistic Regression

Y (dependent variable) → categorical (Discrete)

Logistic Regression Model →

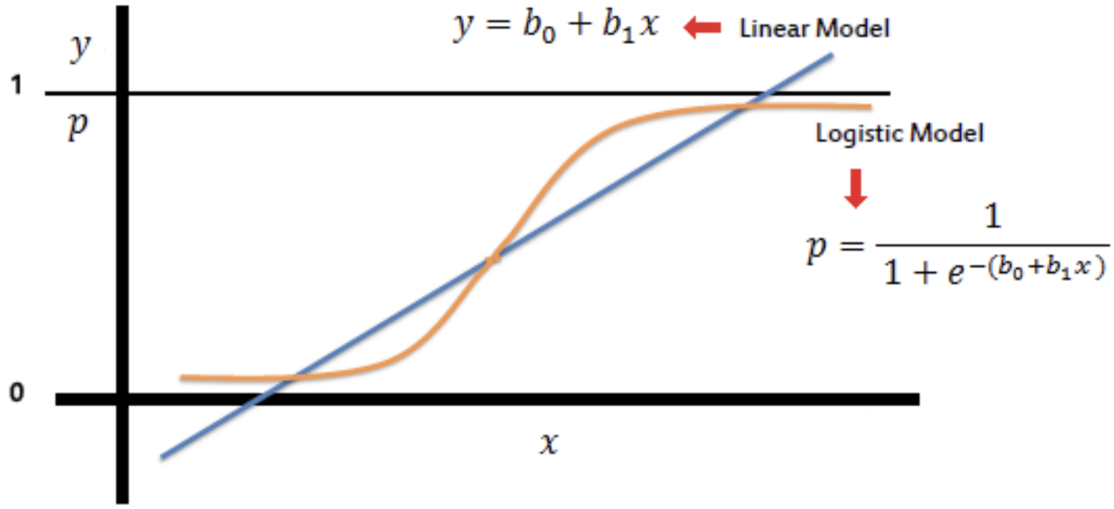
$$f(x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \cdots + \beta_n x_n)}}$$

where $e = 2.7$

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \text{predictor variables}$$

β_0 = Bias

$\beta_1, \cdots, \beta_n$ = Weights



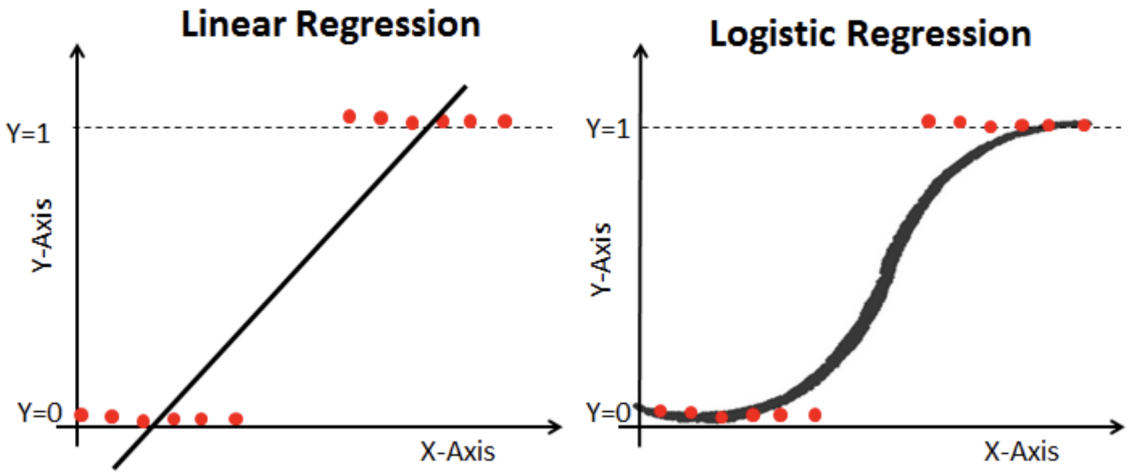


Image Source www.datacamp.com

$$p = \frac{1}{1 + e^{-(B_0 + B_1 x)}}$$

where x : Age

if $p \geq 0.5$ => The person has CHD or 1

else $p < 0.5$ => The person don't have CHD or 0.

x_1 y_1 70: 30

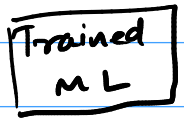
x_2 y_2

x_3 y_3 Training $(x_1, x_2, \dots x_{70})$

x_{98} y_{98} $(y_1, y_2, \dots y_{70})$

x_{99} y_{99}

x_{100} y_{100}



Testing (30%) $(x_{71}, x_{72} \dots, x_{100})$

predicted Values: $(\hat{y}_{71}, \hat{y}_{72}, \dots, \hat{y}_{100})$

Actual Values: $(y_{71}, y_{72}, \dots, y_{100})$

70 : 30

80 : 20

Training : Testing should be depend on the size of the dataset.

Dataset : Training : Testing.

70: 30 ✗ 10,000,000 ✓ 7,000,000 3,000,000

9,990,000 (99.9%) 10,000 (0.1%)

Training set ✓ Validation set Test Set ✓

500 5

if he fails ⇒ he failed to generalise

(Overfitting)

500 [5-20] ≈ 60% 5 ≈ 85%

Train the model → Evaluation within the env., with the available dataset → Rolling out to the world & getting the feedback.

Random State = k

dataset = [a, b, c, d, e]

Splitting: 8:2 ⇒ test-size = 0.2

I: Random state = 0
training = [a, b, c, d]
Test = [e]

II Random state = k
training = [a, b, d, e]
test = [c]