Encoding the Categorical features

Categorical features Example: 1) Marital Status : [unmarried, married, divorced, widowed]

2) Colour of a car : [Red, Blue, sky blue, Grey, Black etc.]

3) Smoking habit : [smoker, non-smoker]

for binary features : 0, 1

0 → non-smoker ; 0 → female

1 → smoker ; 1 → male

for feature having multiple possible values / categories :

Label Encoding :

unmarried → 0

married → 1

divorced → 2

widowed → 3

One-Hot Encoding :-

marital Status (MS) }
[U, M, D, W]

Observation	MS
1	U
2	D
3	M
4	W

Dummy Variables.

MS_U , MS_M , MS_D , MS_W

MS_U	MS_M	MS_D	MS_W
1	0	0	0
0	0	1	0
0	1	0	0
0	0	0	1

Redundant

U = [1, 0, 0]

D = [0, 0, 1]

M = [0, 1, 0]

W = [0, 0, 0]

n - features

(n-1) : non-redundant

Binary number Encoding:

$(4)_{10} = (1\,0\,0)_2$; $(12)_{10} = (1100)_2$

$2^4 \quad 2^3 \quad 2^2 \quad 2^1 \quad 2^0$

$(1 \quad 0 \quad 0 \quad 1 \quad 1)_2 \rightarrow 1\times2^4 + 0\times2^3 + 0\times2^2 + 1\times2^1 + 1\times2^0$

$4 \quad 3 \quad 2 \quad 1 \quad 0$

$= 16 + 0 + 0 + 2 + 1 = (19)_{10}$

37 → $32 \rightarrow 2^5$

5 → $4 \rightarrow 2^2$

1 → $1 \rightarrow 2^0$

$(37)_{10} = (1\,0\,0\,1\,0\,1)_2$

6 bits

4 bits

1 1 1 1

$2^3 + 2^2 + 2^1 + 2^0$

$= 2^4 - 1$

l - bits

max value: $(2^l - 1)$

n - different categories

Label Enc. { 1, 2, 3, ⋮, n }

$\lceil \log_2(n) \rceil$ → bits are required to represent those categories.

Marital Status

Values	Label Enc	binary Enc		Encoding	E_2	E_1
U	0	0	0		0	0
M	1	0	1		0	1
W	2	1	0		1	1
D	3	1	1		1	1

Multi-class Classification (C-class) [C ≥ 3]

One-vs-Rest (OVR) / One-vs-All (OVA)

train C- no. of binary classifiers. Each of them will be trained for one class.

Logistic Regression (binary / Dichotomous)

X (features)

x_1 x_2 x_3 .... x_n

Y (target)

y

Logit

z

$$z = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$$

Predicted

$\hat{y}_p = \sigma(z)$

$\text{logit}(x) = \ln\left(\frac{x}{1-x}\right)$ inverse of sigmoid

Multinomial Regression / Softmax Regression / Polytomous Regression / Maximum Entropy classifier :

features

x_1	x_2	x_3	...	x_n
$x_1^{(1)}$	$x_2^{(1)}$	$x_3^{(1)}$		$x_n^{(1)}$
$x_1^{(i)}$	$x_2^{(i)}$	$x_3^{(i)}$		$x_n^{(i)}$

One-hot Encoded target

y_1	y_2	....	y_C
0	1	0	0 0
0	0	0	1 0

Logit

$\ln\left(\frac{x}{1-x}\right)$

$\text{sigmoid}\left(\ln\left(\frac{x}{1-x}\right)\right) = x$

features: $x_1 \quad x_2 \quad x_3 \quad \cdots \quad x_n$

targets: $y_1 \quad y_2 \cdots y_c$

Logits: $z_1 \quad z_2 \cdots z_c$

$$z_k = \theta_{k,0} + \theta_{k,1} x_1 + \theta_{k,2} x_2 + \cdots + \theta_{k,n} x_n$$

$$z_k = \vec{\theta}_k^{\,T} \vec{x} + \theta_{k,0} \qquad \left[\vec{\theta}_k = [\theta_{k,1}, \theta_{k,2}, \cdots, \theta_{k,n}]^T \right]$$

i^{th} observation

$x_1 \quad x_2 \quad x_3 \cdots x_n$

$x_1^{(i)} \quad x_2^{(i)} \quad x_3^{(i)} \cdots x_n^{(i)}$

$z_1 \quad z_2 \cdots z_k \cdots z_c$

2	-0.5	1	0.7

Logits

$z_1 \quad z_2 \cdots z_k \cdots z_c$

$(\cdot) \quad (\cdot) \cdots \vec{\theta}_k^{\,T} \vec{x} + \theta_{k,0} \cdots (\cdot)$

z_1	z_2	z_3	..	z_c
0	2	-1		
0.5	1.2	0.1		
:	:	:		

$\sigma([z_1, z_2, z_3]) = (\hat{y}_1, \hat{y}_2, \hat{y}_3)$

s.t. $\hat{y}_1 + \hat{y}_2 + \hat{y}_3 = 1$

and $0 \leq \hat{y}_i \leq 1$

$\sigma(z) = \hat{y}$ s.t. $\sum_i \hat{y}_i = 1$ & $0 \leq \hat{y}_i \leq 1$

$[0, 2, -1] \xrightarrow{e^x} [e^0, e^2, e^{-1}] \xrightarrow{\text{normalize}} \left[\frac{e^0}{e^0+e^2+e^{-1}}, \frac{e^2}{e^0+e^2+e^{-1}}, \frac{e^{-1}}{e^0+e^2+e^{-1}}\right]$

$\hat{y}_1 \quad \hat{y}_2 \quad \hat{y}_3$

$z_1, z_2, \ldots, z_c$

$\hat{y}_k = \sigma(z_k) = \frac{e^{z_k}}{\sum_{j=1}^{c} e^{z_j}}$ } Softmax function.

$$\sigma(z_k) = \frac{e^{z_k}}{\sum_{j=1}^{C} e^{z_j}}$$

if $C = 2$ } binary classification

$$\sigma(z_1) = \frac{e^{z_1}}{e^{z_1} + e^{z_2}} = \frac{1}{1 + e^{-(z_1 - z_2)}}$$

$$\sigma(z_2) = \frac{e^{z_2}}{e^{z_1} + e^{z_2}} = \frac{1}{1 + e^{(z_1 - z_2)}}$$

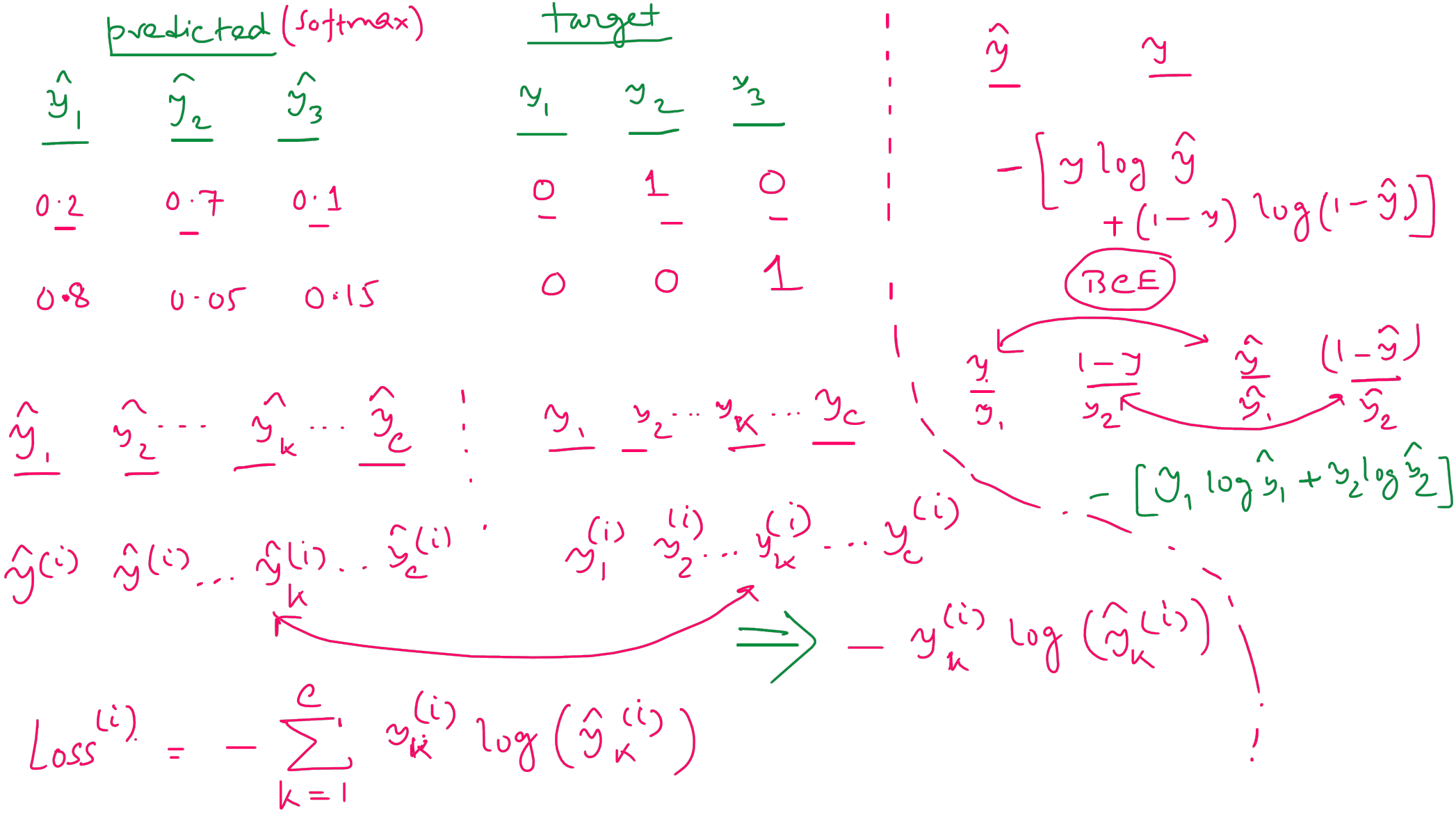
$z = z_1 - z_2$

z_1	z_2	$z_1 - z_2$
1	0	1
0	1	(-1)

$$= \frac{1}{1 + e^{-z}}$$

predicted

$\hat{y}_1$ $\hat{y}_2$ ... $\hat{y}_k$ ... $\hat{y}_C$: y_1 y_2 ... y_k ... y_C } one-hot enc. target


predicted (softmax)
target
$\hat{y}_1$ $\hat{y}_2$ $\hat{y}_3$
y_1 y_2 y_3
0.2 0.7 0.1
0 1 0
0.8 0.05 0.15
0 0 1
$\hat{y}_1$ $\hat{y}_2 \cdots \hat{y}_k \cdots \hat{y}_c$: y_1 $y_2 \cdots y_k \cdots y_c$
$\hat{y}^{(i)}$ $\hat{y}^{(i)} \cdots \hat{y}_k^{(i)} \cdots \hat{y}_c^{(i)}$: $y_1^{(i)}$ $y_2^{(i)} \cdots y_k^{(i)} \cdots y_c^{(i)}$
$\Rightarrow -y_k^{(i)} \log(\hat{y}_k^{(i)})$
$\text{Loss}^{(i)} = -\sum_{k=1}^{c} y_k^{(i)} \log(\hat{y}_k^{(i)})$
$\hat{y}$ y
$-[y \log \hat{y} + (1-y)\log(1-\hat{y})]$
BCE
y $1-y$ $\hat{y}$ $(1-\hat{y})$
y_1 y_2 $\hat{y}_1$ $\hat{y}_2$
$-[y_1 \log \hat{y}_1 + y_2 \log \hat{y}_2]$

$$\text{Loss}^{(i)} = -\sum_{k=1}^{c} y_k^{(i)} \log\left(\hat{y}_k^{(i)}\right) \longrightarrow i^{th} \text{ instance.}$$

total loss = average of all the losses

$$J = -\frac{1}{m} \sum_{i=1}^{m} \sum_{k=1}^{c} y_k^{(i)} \log\left(\hat{y}_k^{(i)}\right)$$

Loss function for multi-class classification.

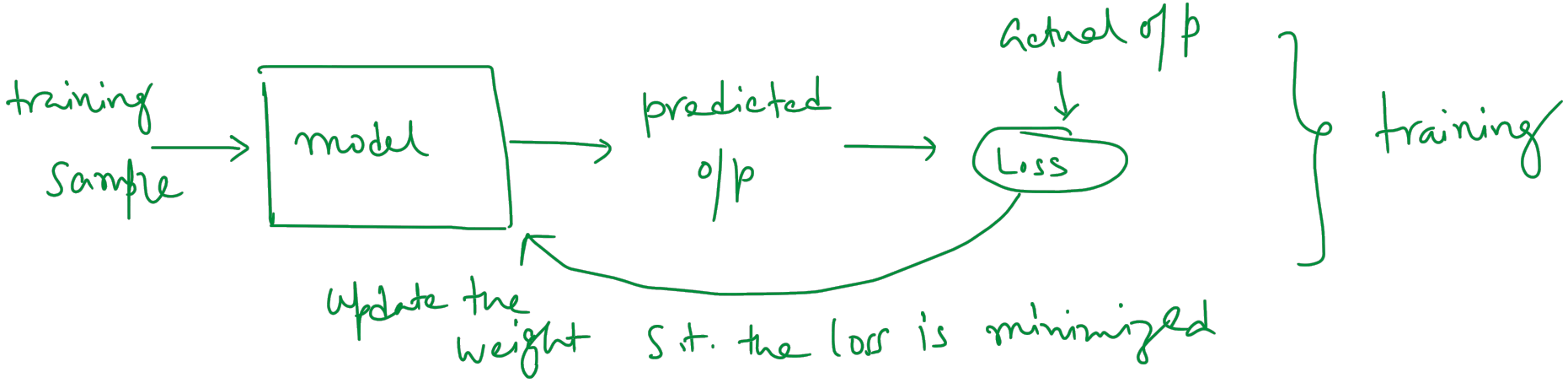
Categorical Cross Entropy Loss (CCE)

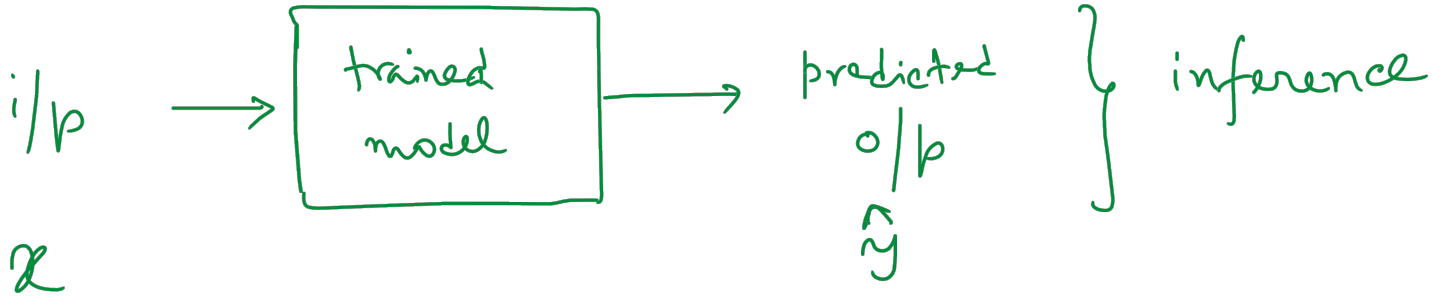
$\sum_{k=1}^{c}$ } all the class

$\sum_{i=1}^{m}$ } all the observations.

Bias & Variance

x → input $\hat{y}$ → predicted o/p y → actual output

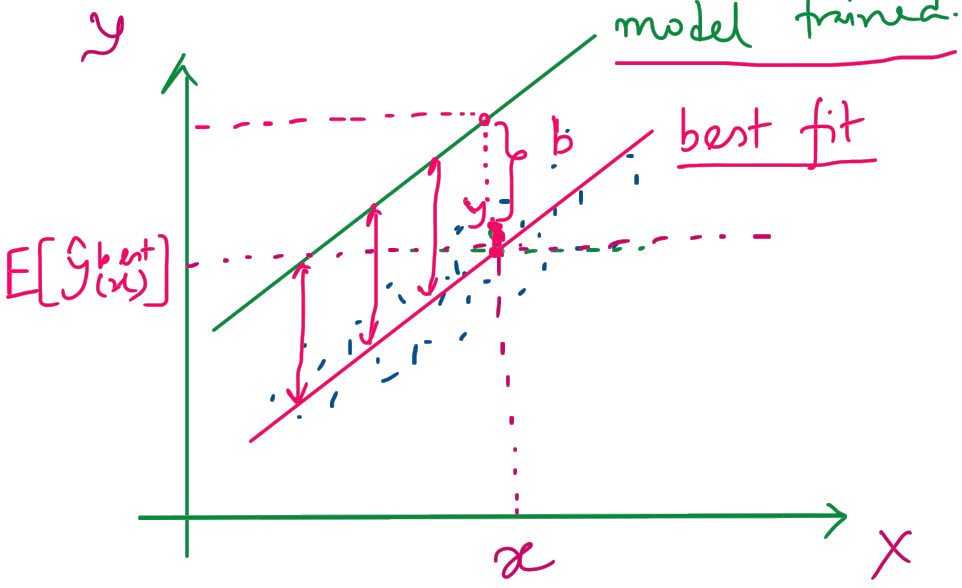




Bias: $\text{Bias}(\text{model}) = \mathbb{E}[\hat{y}] - y$

$\mathbb{E}[\hat{y}]$ → the expected (average) prediction of the model for input x over different training dataset.

y → the true value for input x.

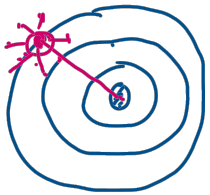


High bias: Our predicted o/p will be widely different from actual o/p.

Low bias: our predicted o/p is close to the actual o/p.

Variance : $\mathbb{E}\left[\left(\hat{y} - \mathbb{E}(\hat{y})\right)^2\right]$

low bias
high variance



high bias
low variance

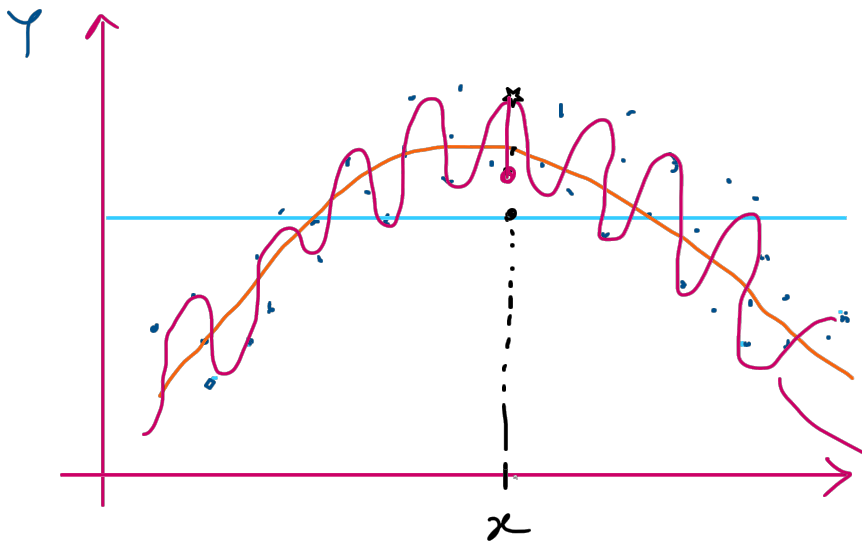
Bias

- Error between avg. model prediction & ground truth (actual value)
- $\mathbb{E}[\hat{y}] - y$
- We want a low bias because more bias means more deviant the model is from actual o/p

Variance

- Average variability of the model prediction
- $\mathbb{E}\left[\left(\hat{y} - \mathbb{E}[\hat{y}]\right)^2\right]$
- we want to reduce variance because we want our model to be robust, i.e. not affected by slight variation in i/p data.

Problems with high bias & high variance



high bias / low variance

OLS fit → underfits the data

polynomial of degree = 2 (low bias)
→ perfectly fit.

polynomial of higher degrees.
(high variance / low bias)
↓
overfits the data.

High Bias

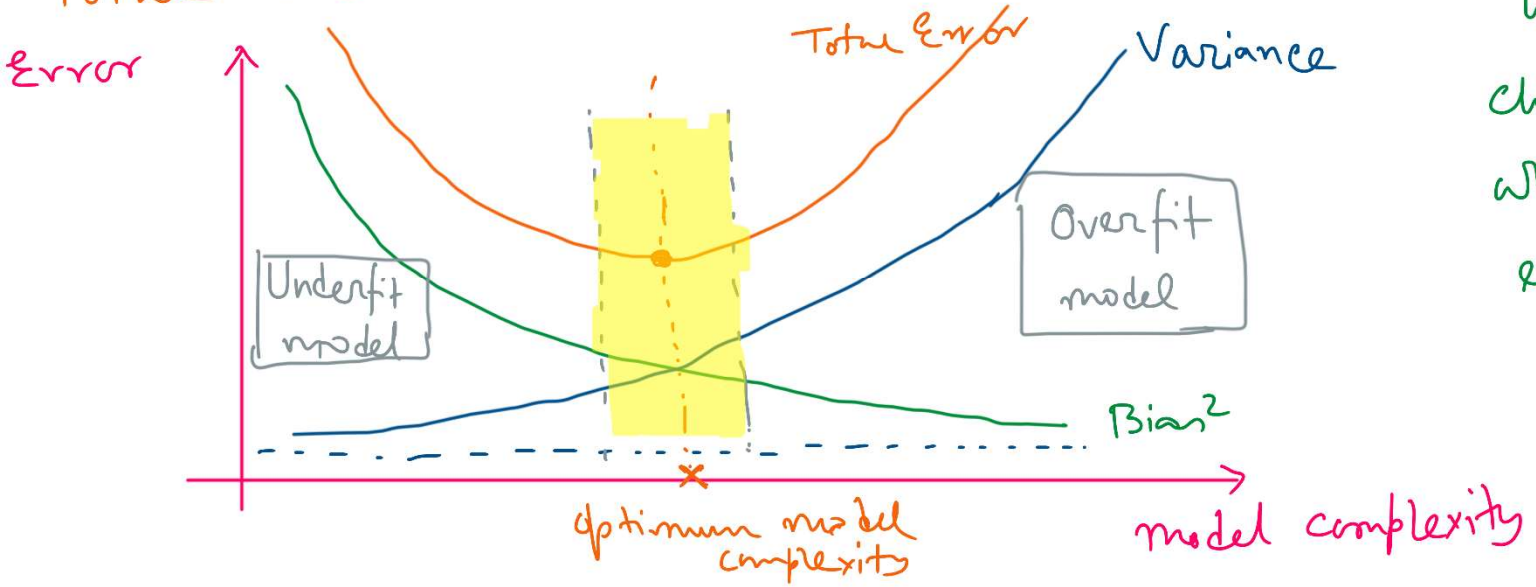
1) Simpler models
2) Underfits the data
3) High error in both training & test dataset

High Variance

1) Complex model.
2) Overfits the data
3) Low error in training / High error in testing dataset

(MSE)

Total Error = $Bias^2$ + Variance + Irreducible Error



We should always choose the model which has lowest error in test dataset