Datascience → under
→ overfit } bias variance
~~why~~ saturday
→ hiccups
→ remedy → ~~catpsl~~
what?
why?

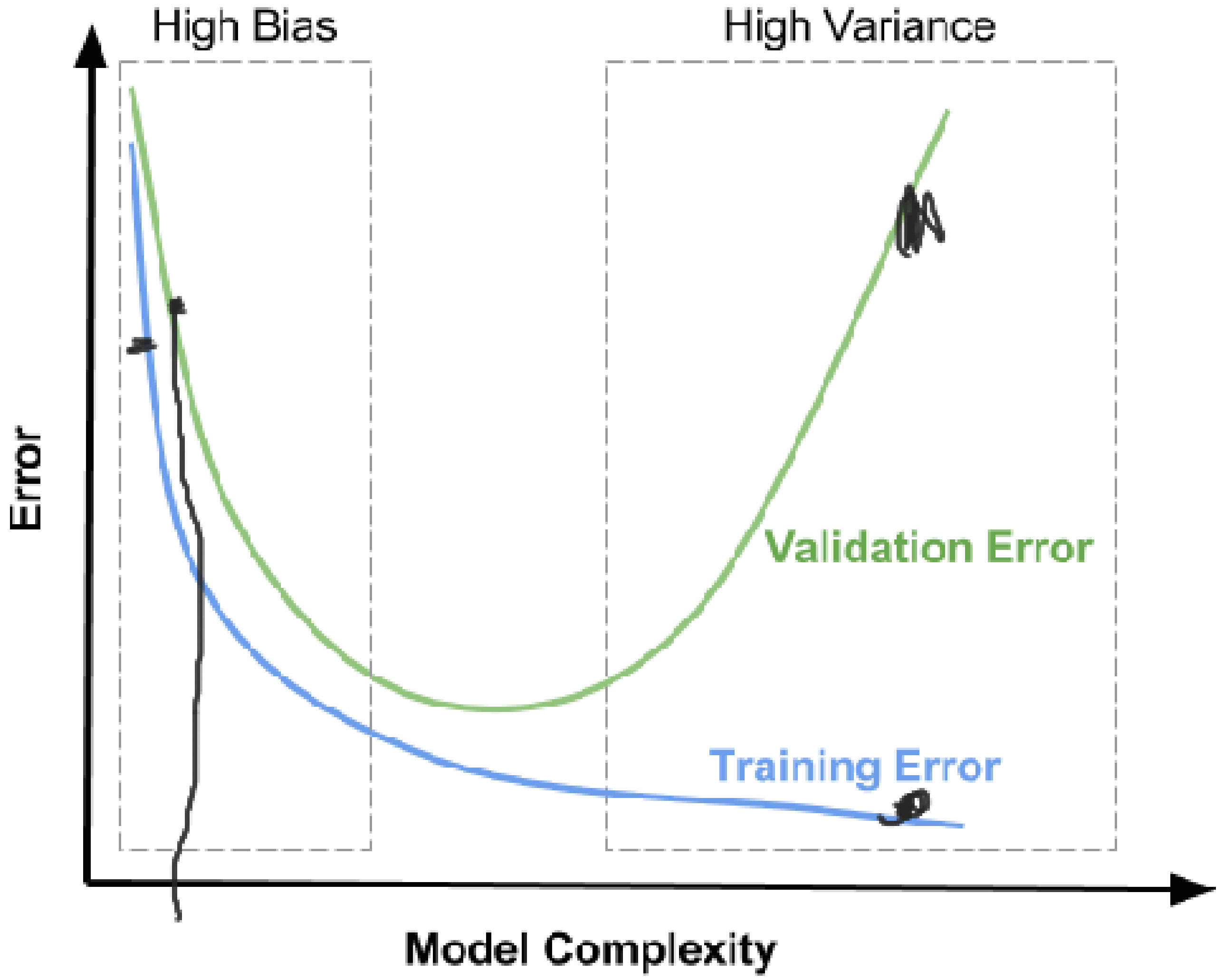
flow

---

complex ↑↑ linear reg → variables polynom

DS → underfit → variables polyfit ↑
→ overfit
penalize → something
TE → least

STOP!



equ^n

$$\text{Error} = \sum (y - \hat{y})^2$$

$$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2$$

$$\text{Error}_1 = \sum (y - \beta_0 - \beta_1 x_1)^2$$

$\Downarrow Er_2 < Er_1$

$$\Downarrow \text{Error}_2 = \sum (y - \beta_0 - \beta_1 x_1 - \beta_2 x_2)^2$$

↑ variable → Er_1 → x_1

eg

usecase

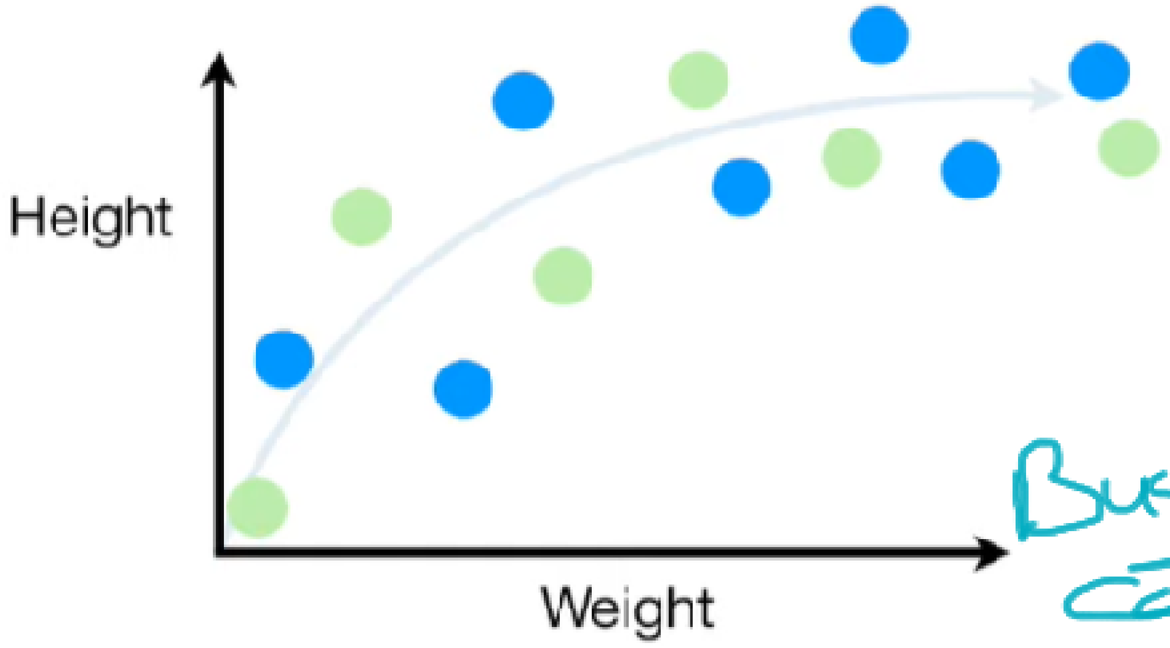
Er_2

x_1 x_2

underfit

$\beta_2 x_2$

overfit



$\beta_0 + \beta_1 x_1$

$\beta_0 + \beta_1 x_1 + \beta_2 x_2$

Business case { e_{tr} ↑, e_{te} ↑ }

$\boxed{\beta_2} x_2$

e_{tr} ↓

→ problem

root

e_{tes} ⇈

coeff DS

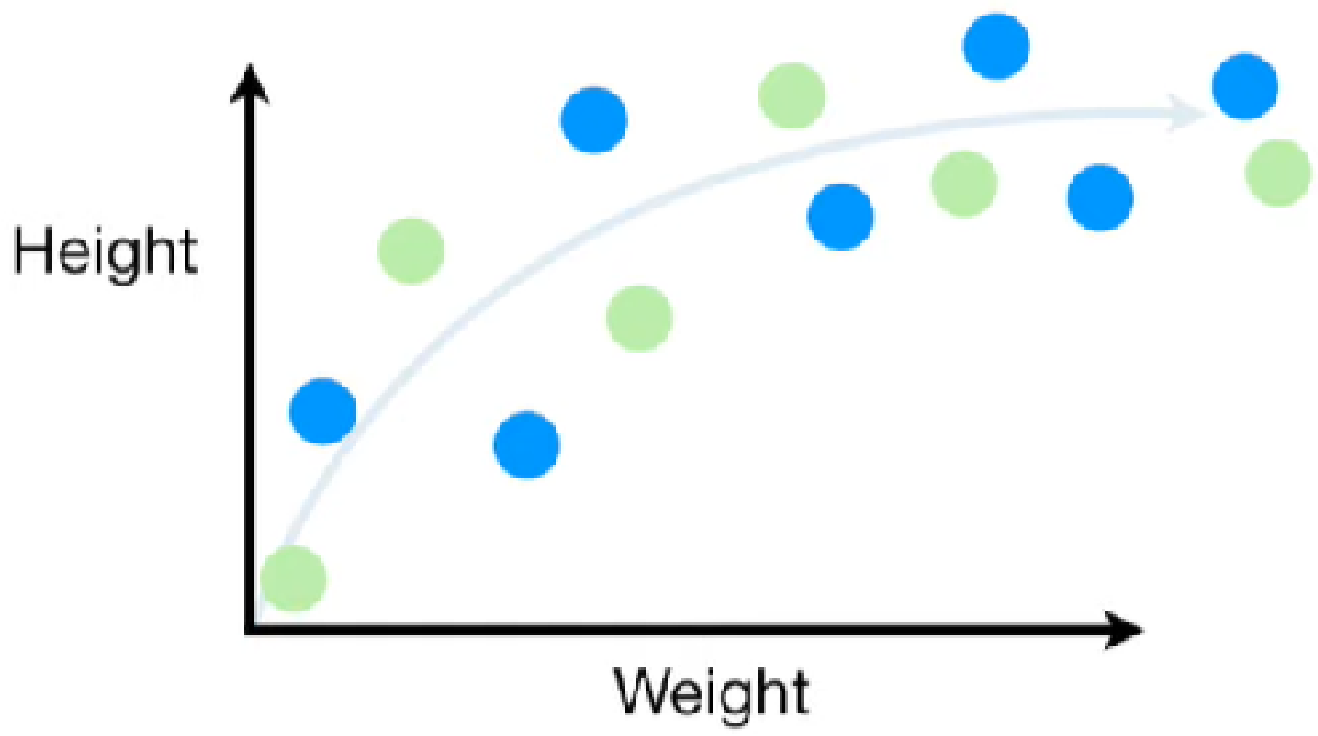
$$e_{tr} = \sum (y - \beta_0 - \beta_1 x_1)^2$$

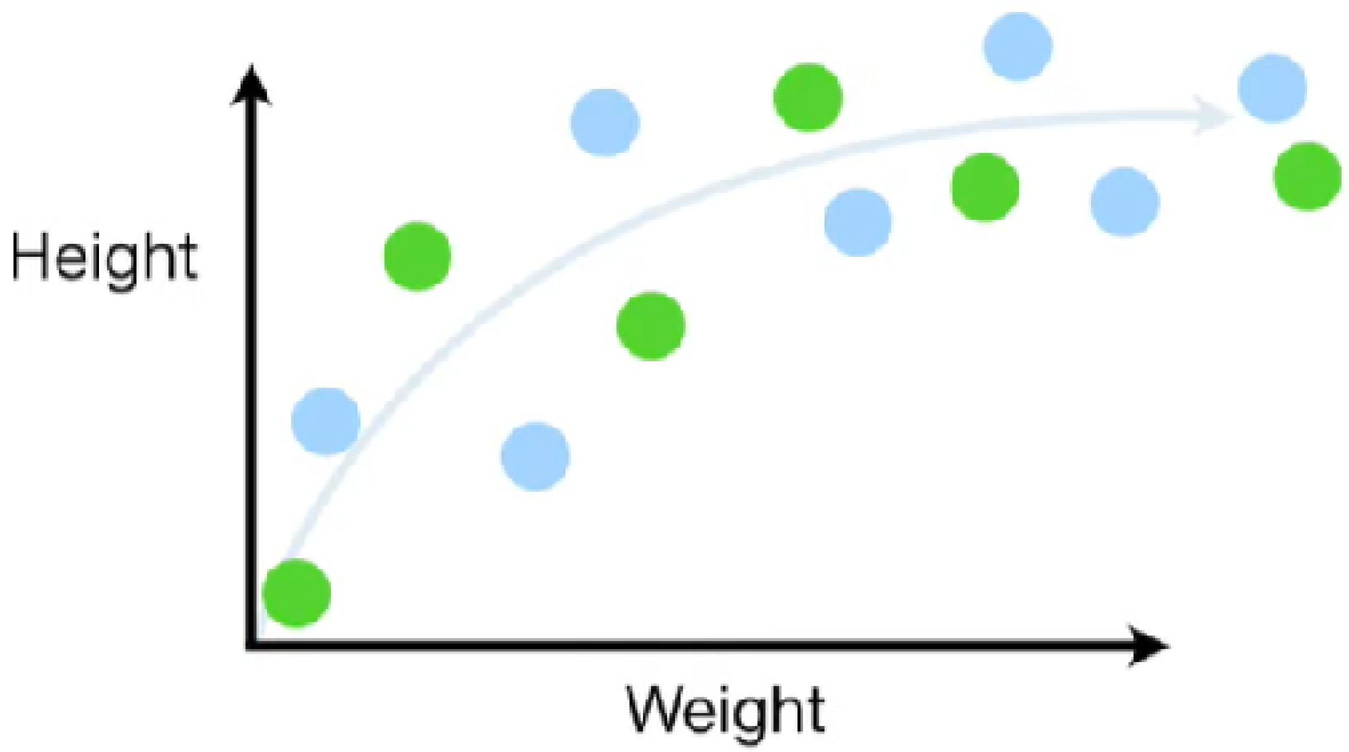
$$\downarrow e_{tr} \sum (y - \beta_0 - \beta_1 x_1 - \beta_2 x_2)^2$$

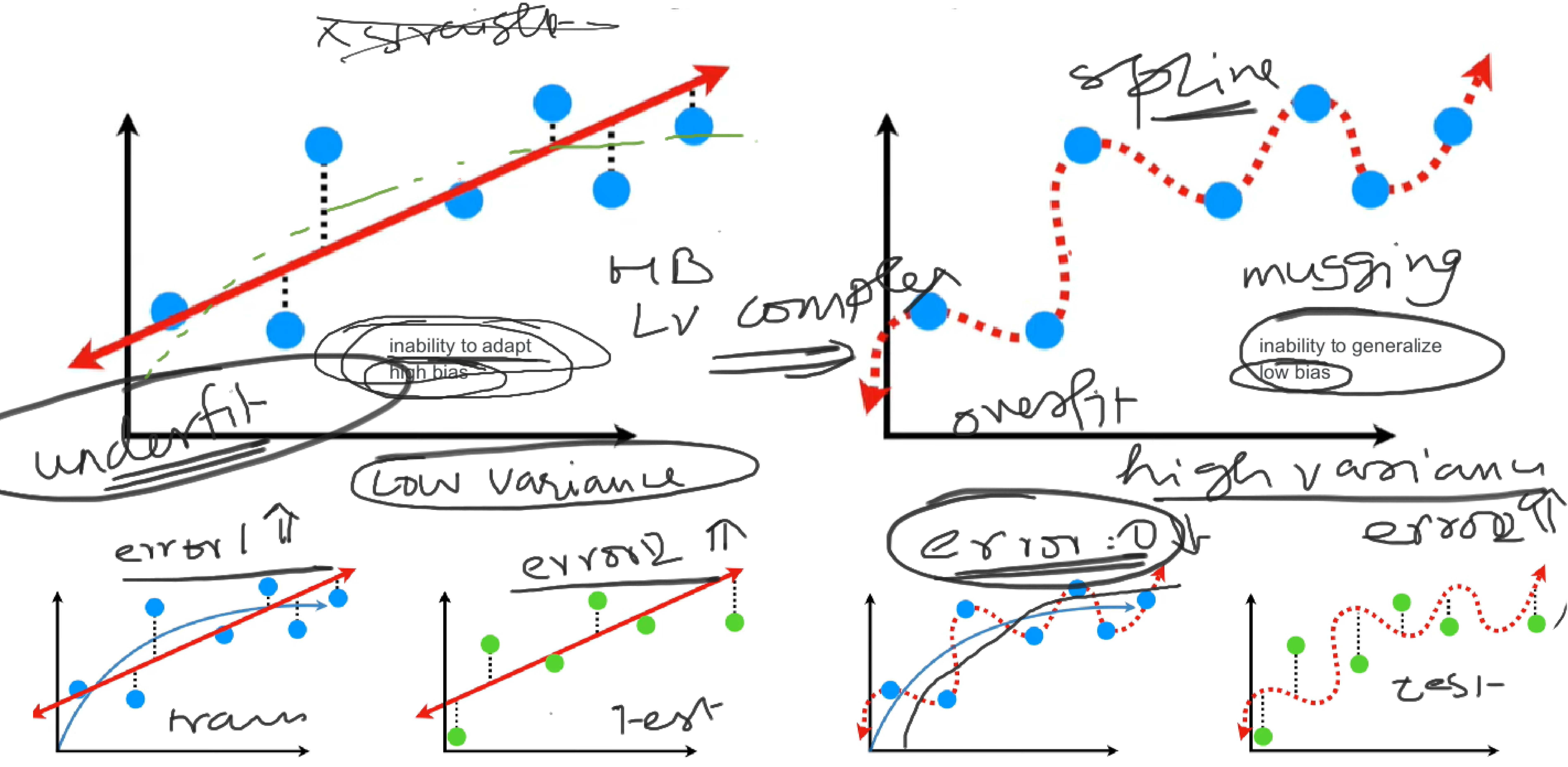
dataset $\beta = 5 x_2$ 5

hicum ← $0.5 x_2$ → 0.5

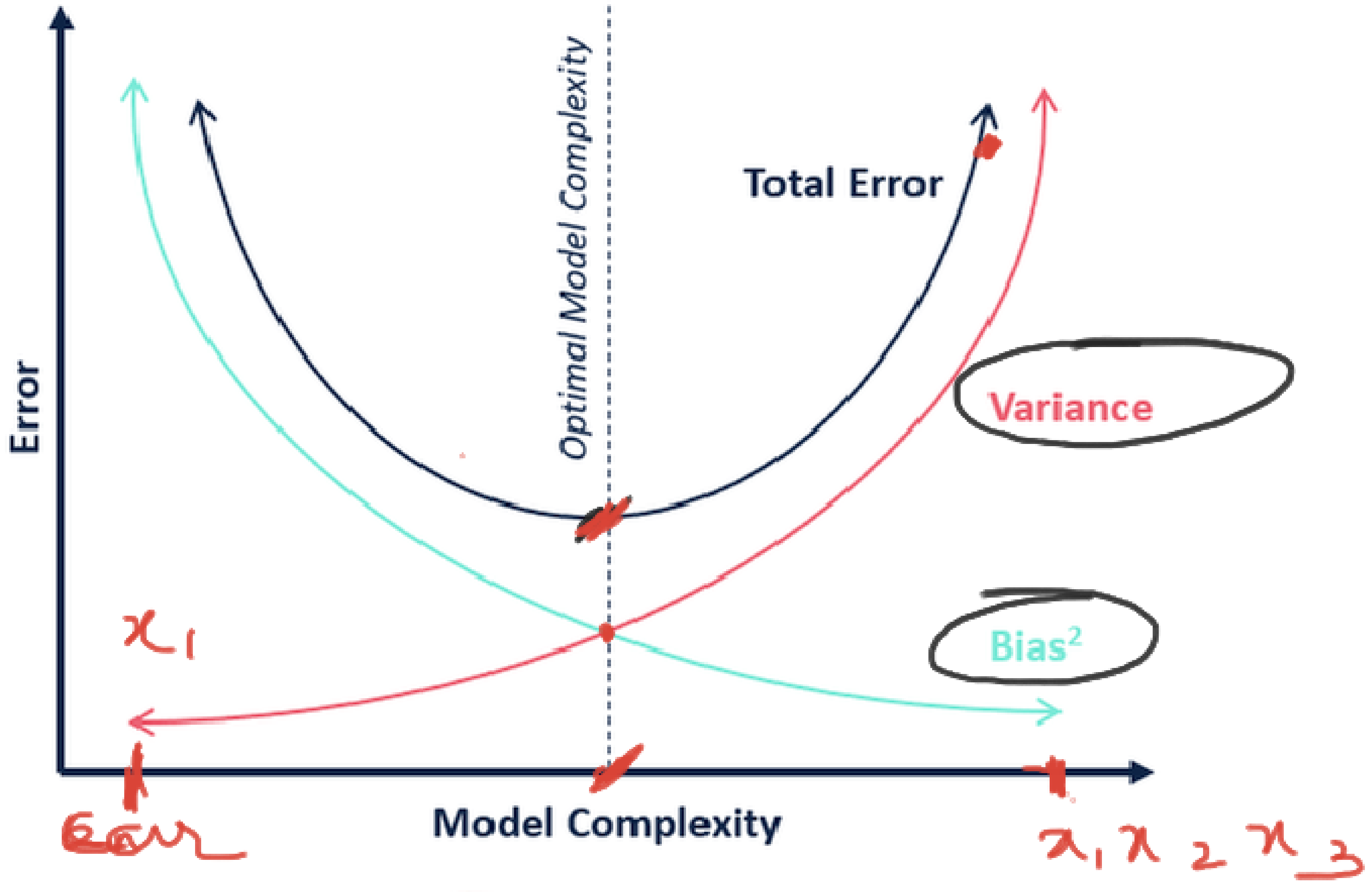
The **Blue Dots** are the **training set**… …and the **Green Dots** are the **testing set**.







spline
HB
LV
complex
inability to adapt
high bias
underfit
Low variance
overfit
inability to generalize
low bias
high variance
error1
error2
error : 0
error2
train
Test
test



$TE = Bias^2 + var + \text{Irreducible}$

error ↓

$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2$

Bias Variance

Remedy → Penalize

β_i → hiccup

OLS → β_i

GD → β_i

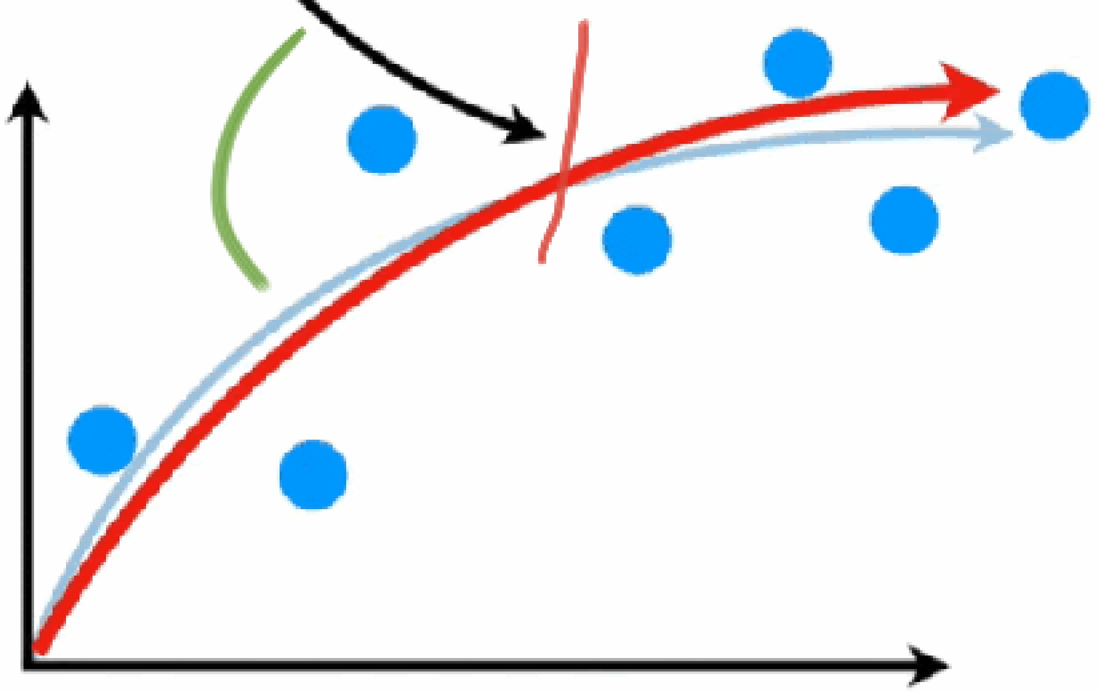
β_i

$5\, x_2$

$0.5\, x_2$ → overfit

Penalize

In machine learning, the ideal algorithm has **low bias** and can accurately model the true relationship…



Three commonly used methods for finding the sweet spot between simple and complicated models are: **regularization**, **boosting** and **bagging**.

class Decision

$\beta_1 \quad \beta_2$

$$RSS = \sum (y - \hat{y})^2 + \boxed{\lambda f(\beta_i)}$$

$\beta_i \Uparrow$

$$\Downarrow err = \sum (y - \beta_0 - \beta_1 x_1 - \beta_2 x_2)^2 + \lambda(\Downarrow\beta_1^2 + \Downarrow\beta_2^2)$$

$$\boxed{\beta_i = \beta_i + \alpha \frac{\partial e}{\partial \beta_i}}$$

$$\beta_i^2 \qquad \frac{\partial e}{\partial \beta_i} = \_ + \_$$

$$2\lambda\beta_i$$

quick

$\Uparrow$

train

$$err \Downarrow = \sum (y - \beta_0 - \beta_1 x_1 - \beta_2 x_2)^2 + \lambda(\beta_1^2 + \beta_2^2)$$

$5x_2$

$0.5\, x_2$

$\Downarrow \beta_2 \rightarrow$ underfit $\rightarrow$ overfit

overfit

x

β_0

$f(\beta_i)$

error $\Sigma(y - \beta_0 - \beta_1 x_1 - \beta_2 x_2)^2 + \lambda(\beta_1^2 + \beta_2^2)$

term

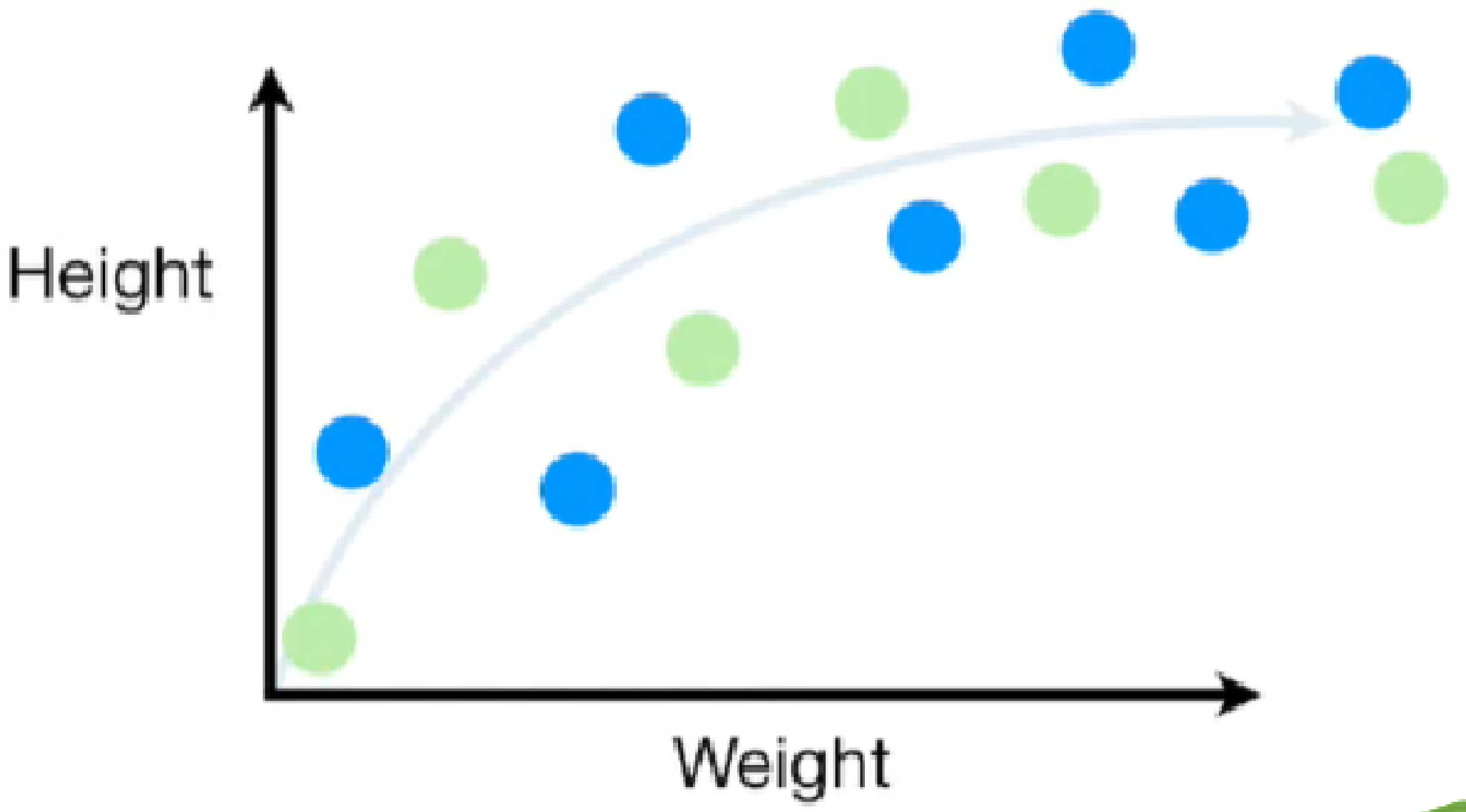
underfit β_2 overfit β_2 varian

"λ" $\Rightarrow$ confidence error intro

minize

1

0.001



$\lambda \uparrow$ $\quad f(\beta_i) \uparrow$

$\lambda \downarrow$ $\quad f(\beta_i) \downarrow$

How? → gridsearch

λ

model → 0.01, 0.1, 1, 10, 100

lowest — train test

e_{tr}
e_{test}

λ

$f(\beta_i) = |\beta_i|$ / β_i^2

1 2

mod

L 1

$|\beta_i|$

Lasso

Ridge Lasso

Bitwise diff

f'

L2

Square

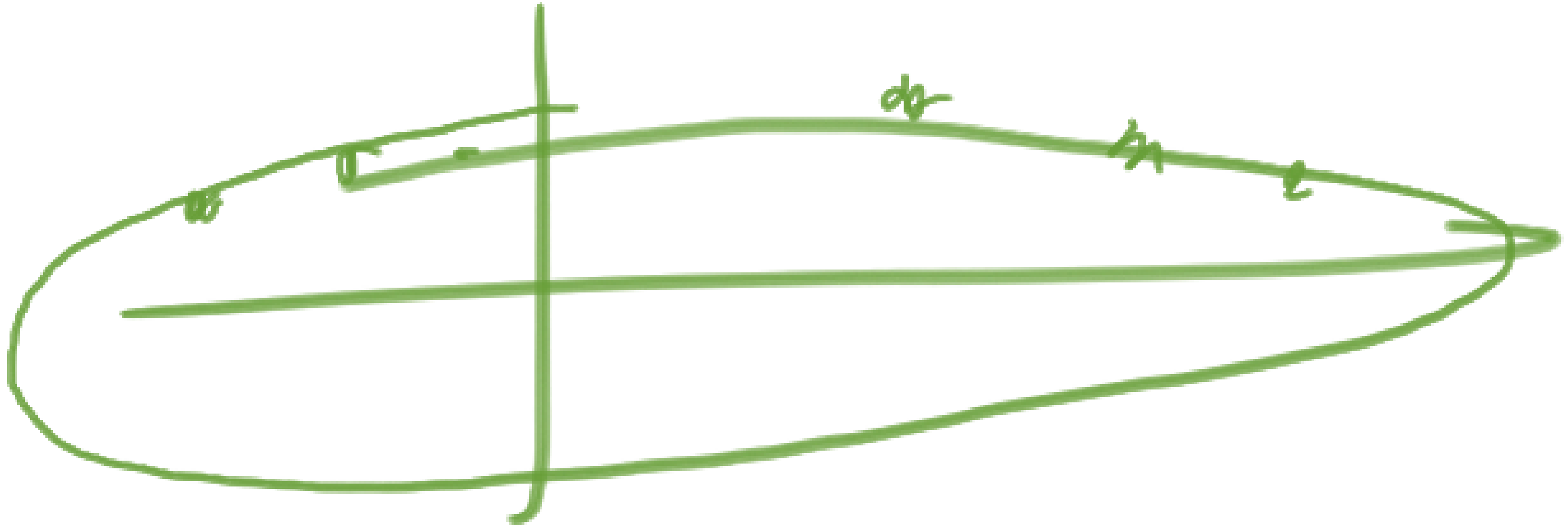
β_i^2

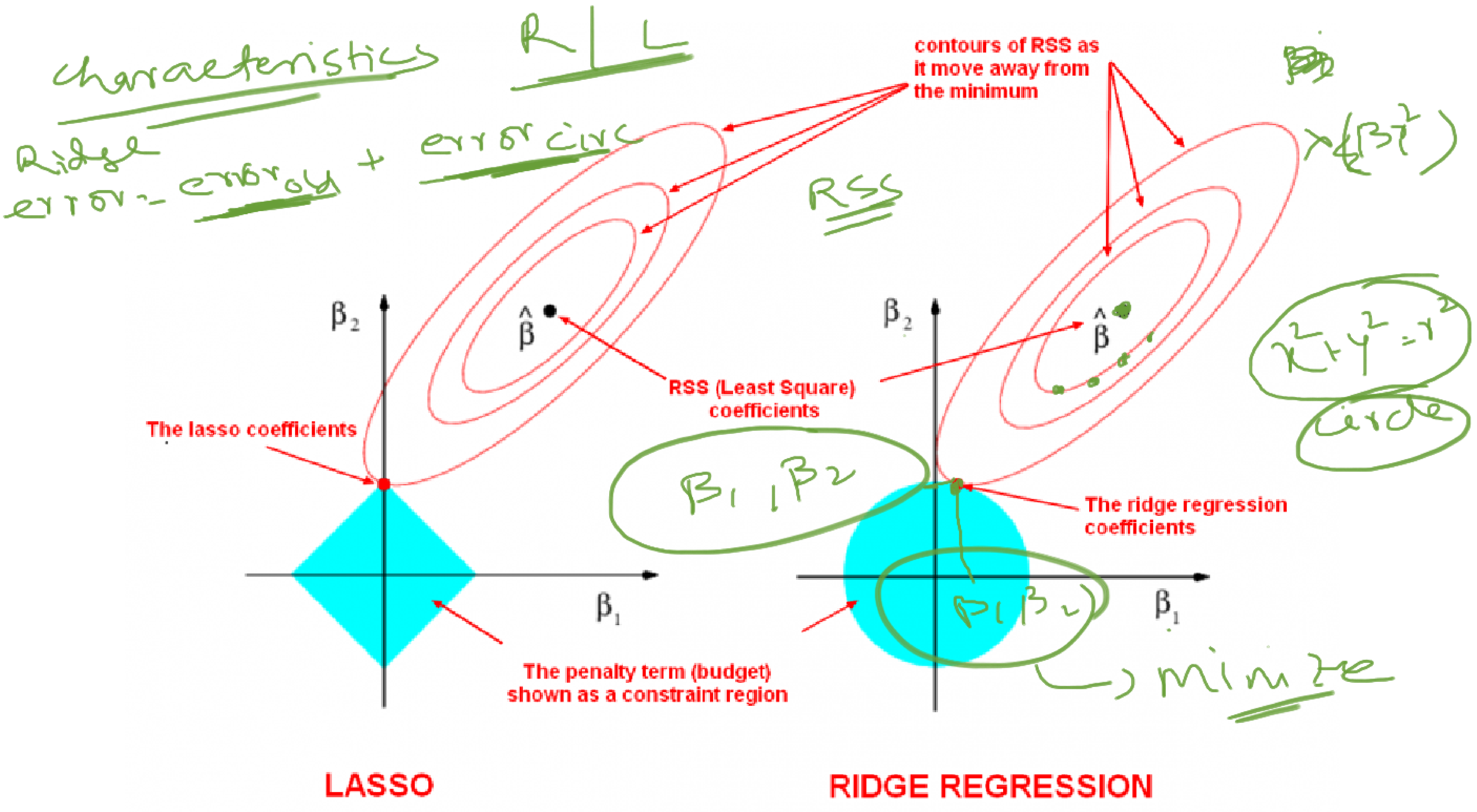
Ridge

$err_{old} = (y - \beta_0 - \beta_1 x - \beta_2 x_2)^2$

$(a+b+c)^2$

elliptical → $a^2 + b^2 + c^2 + 2ab + 2bc + 2ca$

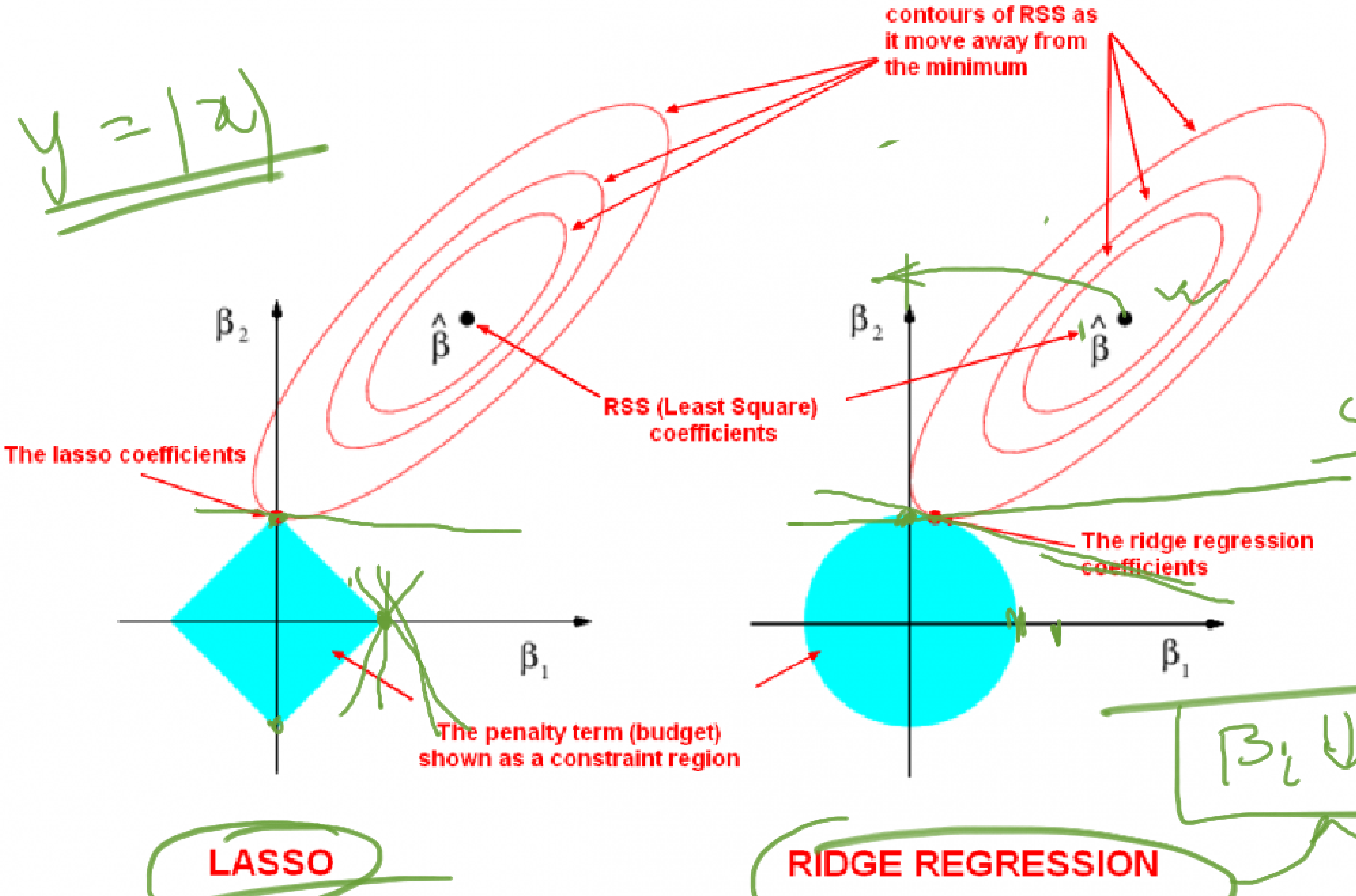



characteristics
R | L
Ridge
error = error_old + error circ
RSS
contours of RSS as it move away from the minimum
λ(β²)
β₂
β̂
RSS (Least Square) coefficients
The lasso coefficients
β₁, β₂
x² + y² = r²
circle
The ridge regression coefficients
β₁
β₁ β₂
minimize
The penalty term (budget) shown as a constraint region
LASSO
RIDGE REGRESSION

$y = |x|$

Lasso → Diminishing

β_i



curves

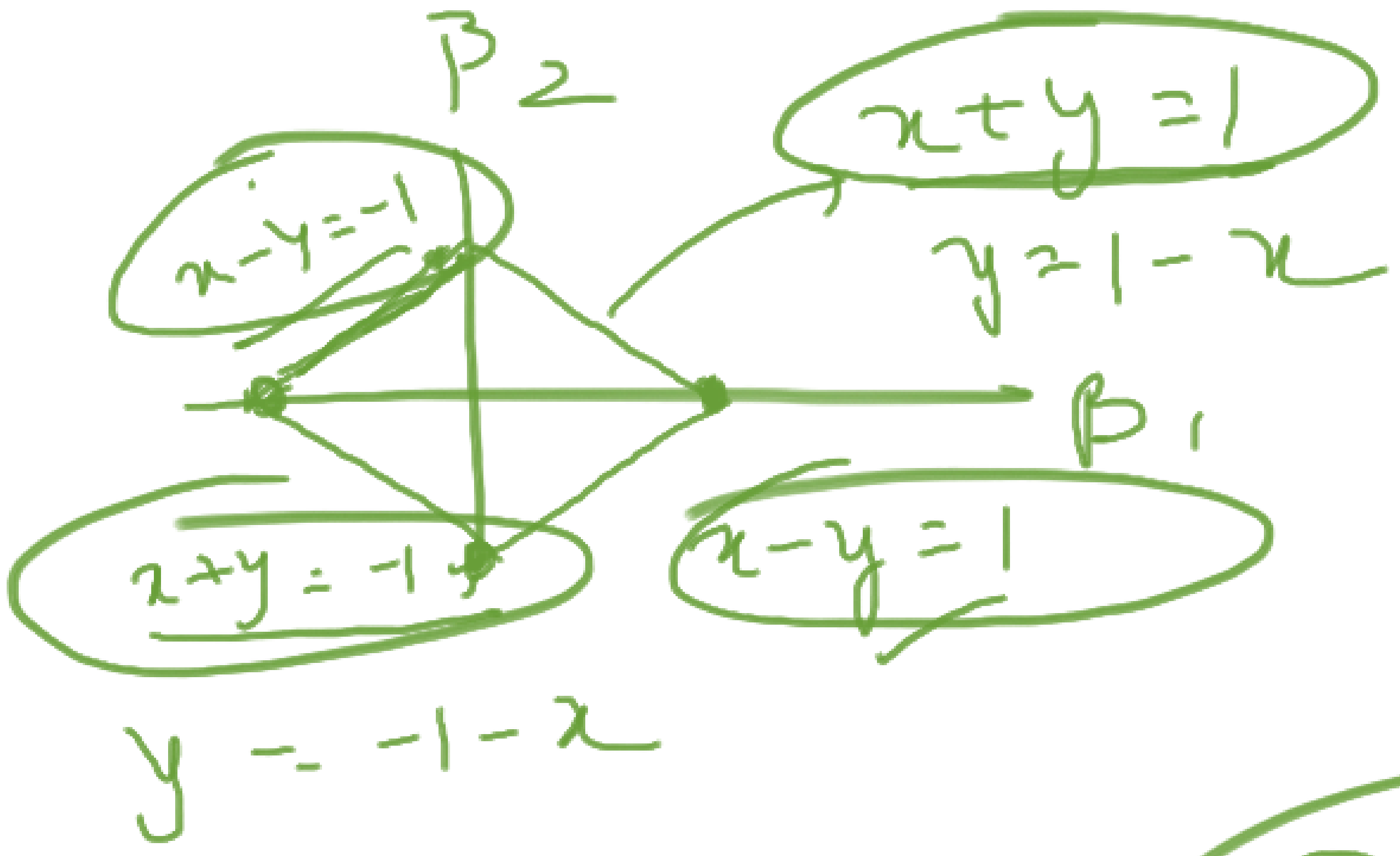
ellipse
circle

$\beta_i \downarrow = 0$

$\beta_i = 0$

$y = |x|$

$$\boxed{\begin{aligned} x = 4 &\Rightarrow y = 4 \\ x = -4 &\Rightarrow y = 4 \end{aligned}}$$



$$y = \begin{cases} 1x & x > 0 \\ -1x & x < 0 \end{cases}$$

"x" $\longrightarrow$ IV

Bitwise shift

$\lambda(\beta_1^2 + \beta_2^2)$ ⇓ x_1 x_2 ⇓

indep → 0 → 65

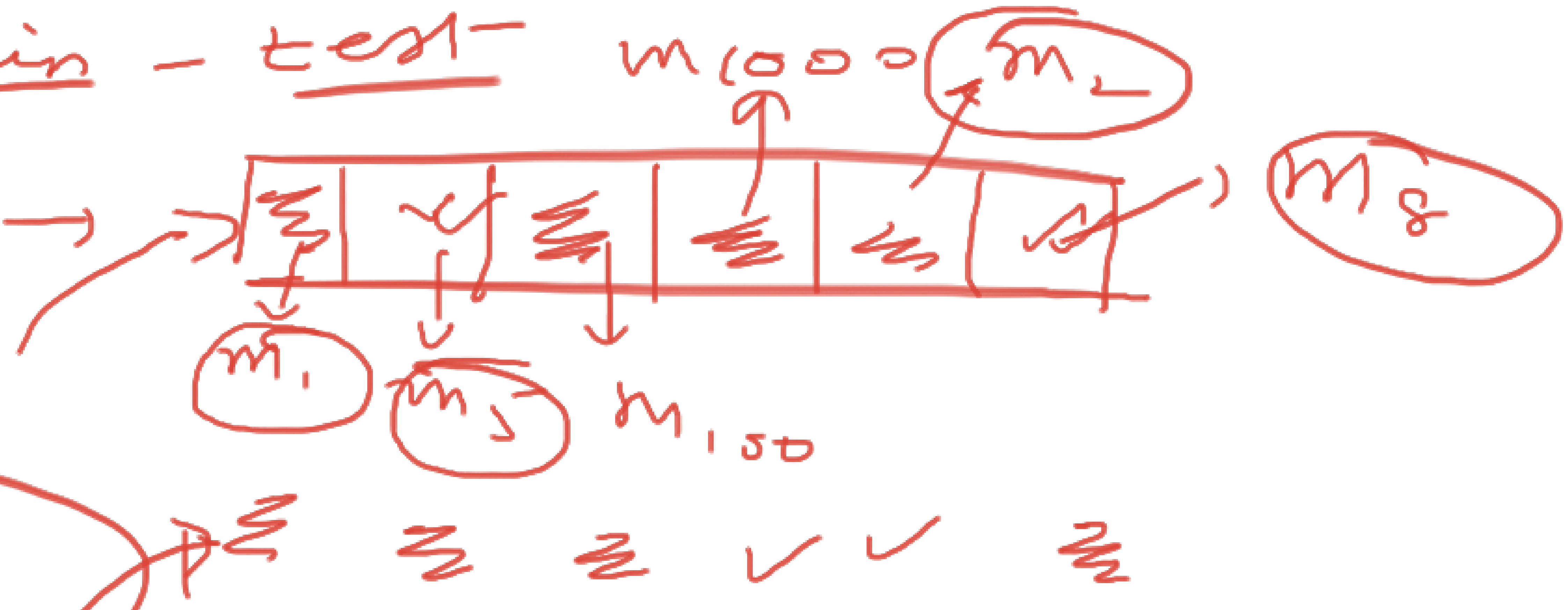
β_i ⇓ $\beta_i \downarrow$ x_i → Bell ← correct

std scaler

```
[array([-1.49246448,  0.37088936, -0.70836731,  1.08568161, -0.8097063
10      4.4075122 , -0.80450999]),
 array([-1.30486468,  0.5170331 , -0.85951603,  0.96594376, -0.8100684
100     3.54696735, -0.74796938]),
 array([-0.6486645 ,  0.4993468 , -0.67576213,  0.39800779, -0.61503278,
1000    1.33852138, -0.56022566]),
 array([-0.11842591,  0.11262874, -0.1443168 ,  0.05431575, -0.1355300
10000   0.20563081, -0.12574102])]
```

train - test

random seed →



random seed = 43