"SGD/GD"

Basics of optimization for ML/AI → for-loop
→ Num-methods

(i) ~15 code-walkthrough

(ii) pseudo-code + interactive → Lr. SVM

(iii) Logistic Reg + L_2 reg (SGD)
/Lr-Reg

Linear Regression:

$$\min_{w,b} \sum_{i=1}^{n} \left[y_i - (w^T x_i + b) \right]^2 + \lambda \sum_{j=1}^{d} w_j^2$$

for

$w \in \mathbb{R}^d$, $b \in \mathbb{R}$

squared-loss

L_2-reg

$\lambda w^T w$

GD/SGD

$D_{tr} = \{x_i, y_i\}_{i=1}^{n}$

$D_{te} = \{x_i, y_i\}_{i=1}^{m}$

$x_i \in \mathbb{R}^d$

GD

D_{tr}, w, b

$L(w, b)$

$\frac{\partial L}{\partial w}, \frac{\partial L}{\partial b}$

Derivative of $\mathcal{L}(w, b)$

param: w_i's & b

Diff. w.r.t w, b

$$\nabla_w L = \frac{\partial L}{\partial w} = \sum_{i=1}^{n} 2\left(y_i - (w^T x_i + b)\right)(-x_i) + \lambda(2w)$$

$n \rightarrow$ for

Vector

def dL/dw

$$\frac{\partial L}{\partial b} = \sum_{i=1}^{n} 2\left(y_i - (w^T x_i + b)\right)(-1)$$

$n \rightarrow$ for

Diff. w.r.t w_t, b_t

dL/db

Scalar

Update - eqns:

learning - rate

$$w_{t+1} = w_t - l \left.\frac{\partial L}{\partial w}\right|_{w_t}$$

$$b_{t+1} = b_t - l \left.\frac{\partial L}{\partial b}\right|_{b_t}$$

init : $t = 0 (w_0, b_0)$

$t = 1, 2, 3, \ldots$

w_1, b_1

Gradient - descent :

$\{(x_i, y_i)\}_{i=1}^{n} = D_{tr}$ (10 GB)

SGD

- prog-ability
- boundary
- internals

1. Pre-processing: Normalize data (D_{tr}) sklearn → Dim-red

2. init : hyper-params $\lambda = 1$ $\ell = 0.5$

✓ $n = \checkmark$
$d = \checkmark$

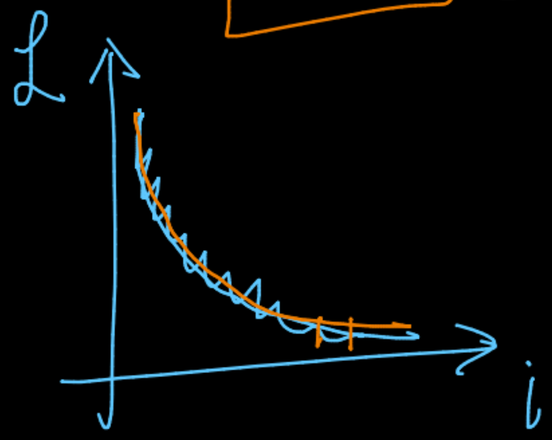
w_{old} = random-vec (d)

b_{old} = scalar - randomly

random.seed(10)

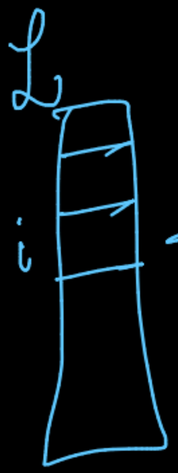
SGD

3. for i = 1 to max iter (100)



3a. $W_{new} = W_{old} - l * \frac{\partial L}{\partial W}(D_{tr}, W_{old}, b_{old})$

3b. $b_{new} = b_{old} - l * \frac{\partial L}{\partial b}(D_{tr}, W_{old}, b_{old})$



3c. $L_{new}(W_{new}, b_{new}, D_{tr})$

3d. - - - - -

$L_{old} = L(W_{old}, b_{old}, D_{tr})$

while true:

- W_{new}
- b_{new}

$L_{new} = L(W_{new}, b_{new}, D_{tr})$

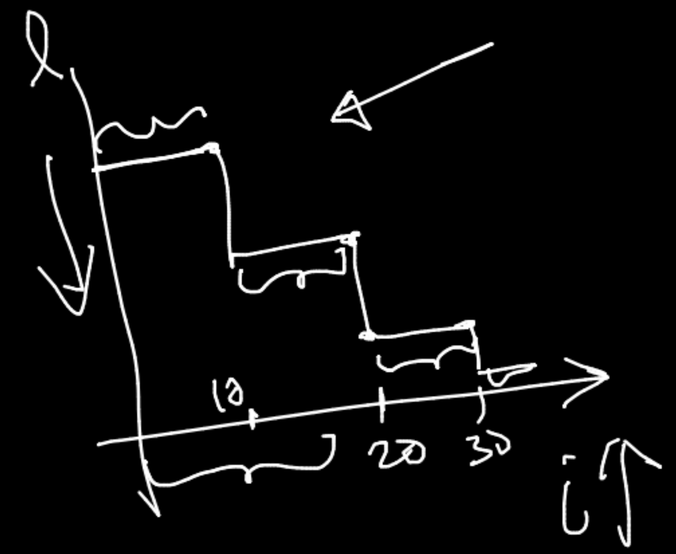
$L_{old} - L_{new} < 0.01\%$ of L_{old}

3e. $w_{old} = w_{new}$

$b_{old} = b_{new}$

list

3f. l = Updated (l, i)

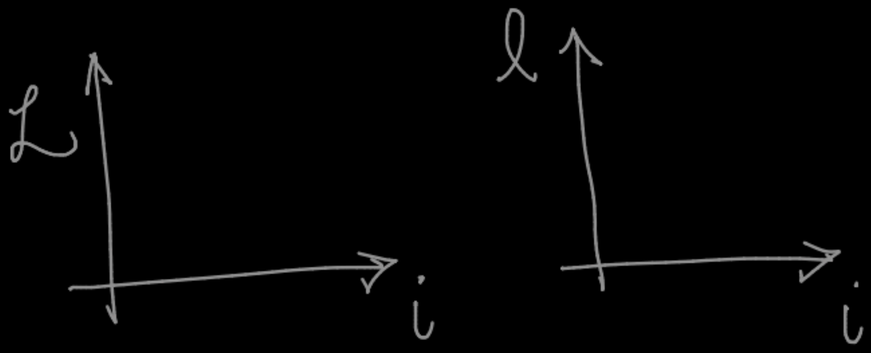


3g. end of loop

$l = 1/i$

w_{new}, b_{new}

4. plotting



5. return W_{new}, b_{new}

2 hrs

Eval Model ($W, b,$ D_{cv})

Sq. loss

SGD:

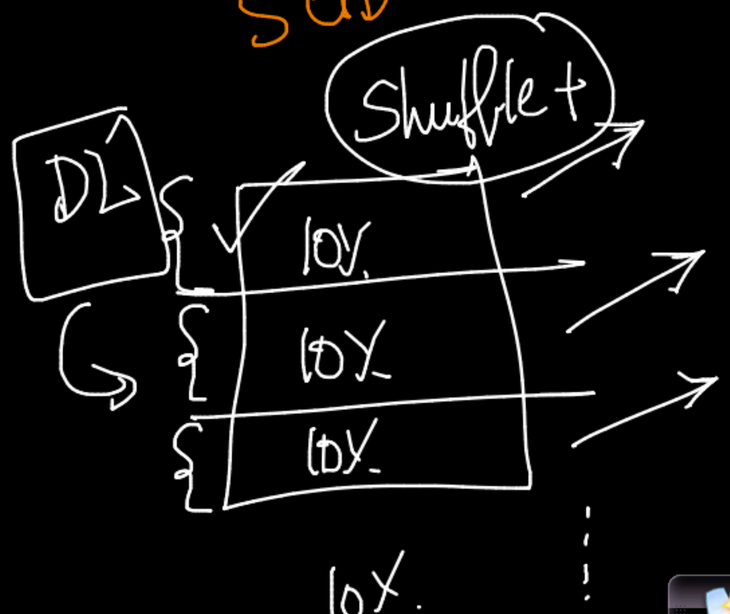
x_i, y_i

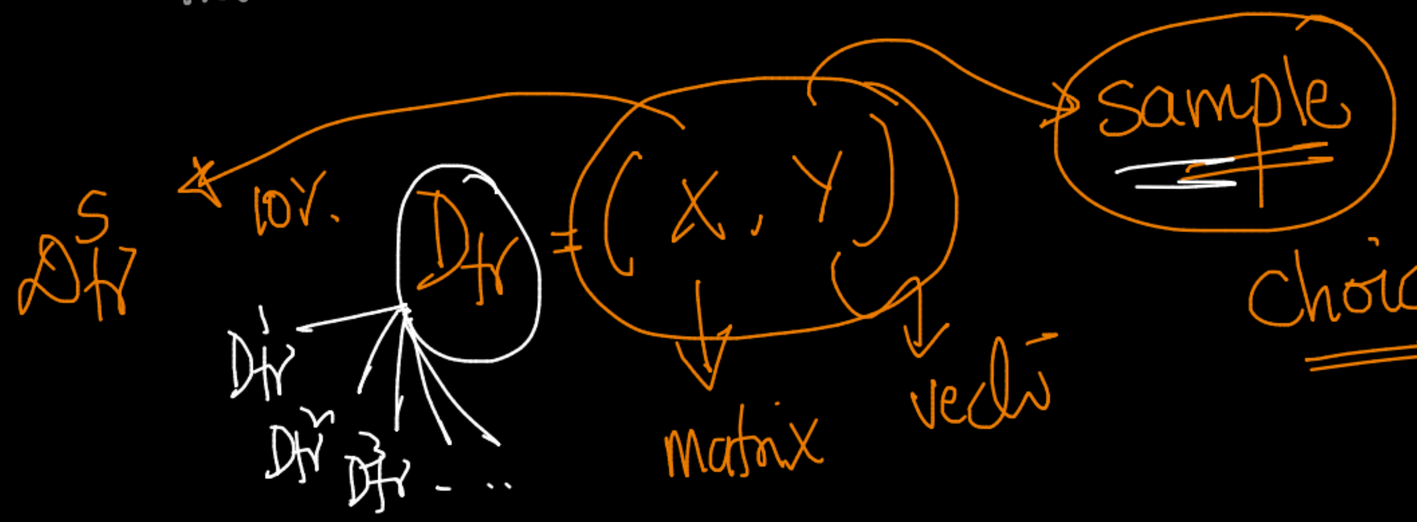
3a. $W_{new} = W_{old} - l * \frac{dL}{dW}(W_{old}, b_{old}, \hat{S}_{tr})$

3b. $b_{new} = b_{old} - l * \frac{dL}{db}(W_{old}, b_{old}, \hat{S}_{tr})$

GD

SGD





cache elem → $\frac{\partial L}{\partial w}$ w_t

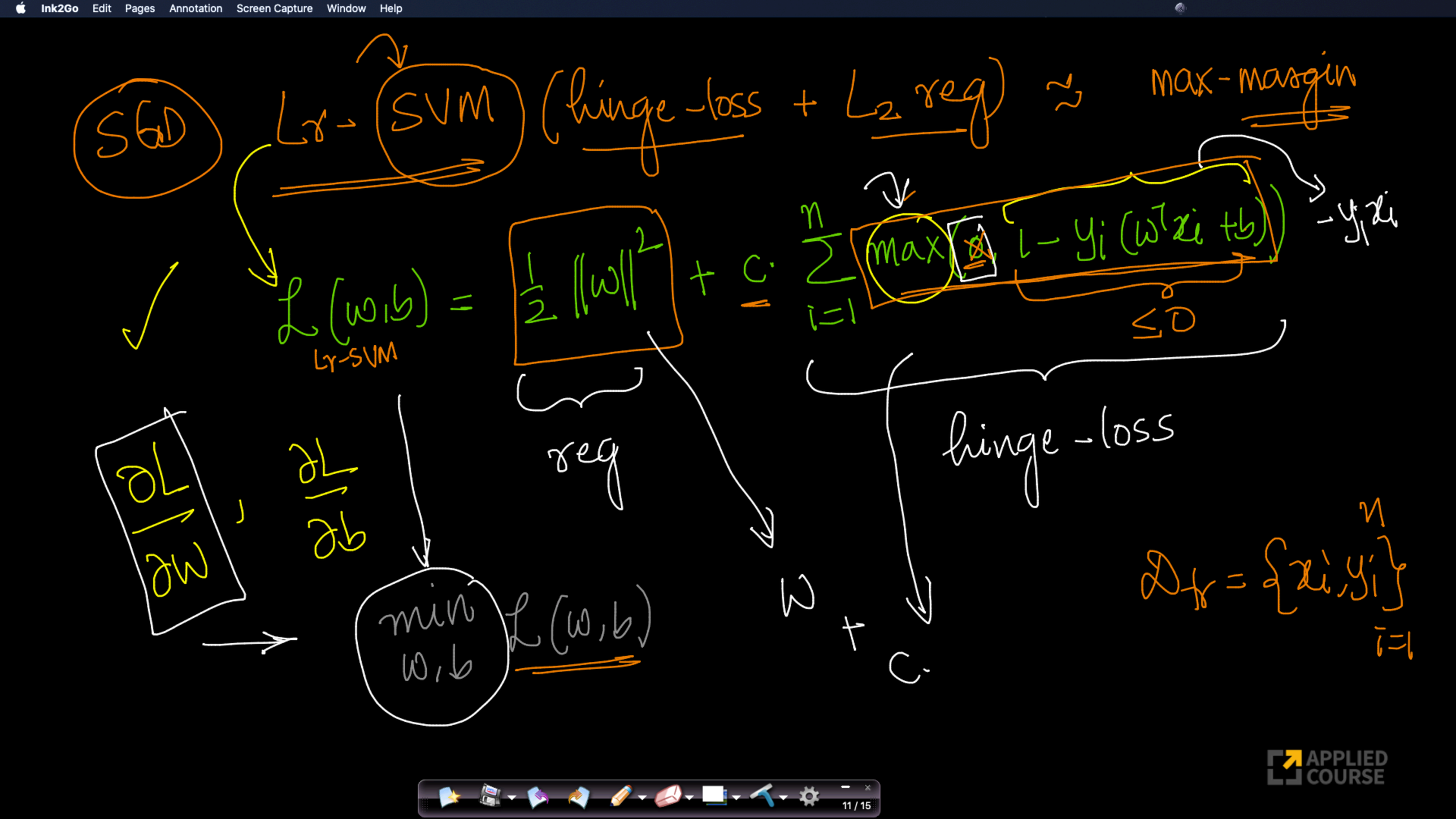
faster

more - elems

GD

slower / Memory issues

fewer elem



Derivative of hinge-loss:

$\partial L / \partial w$ ✓

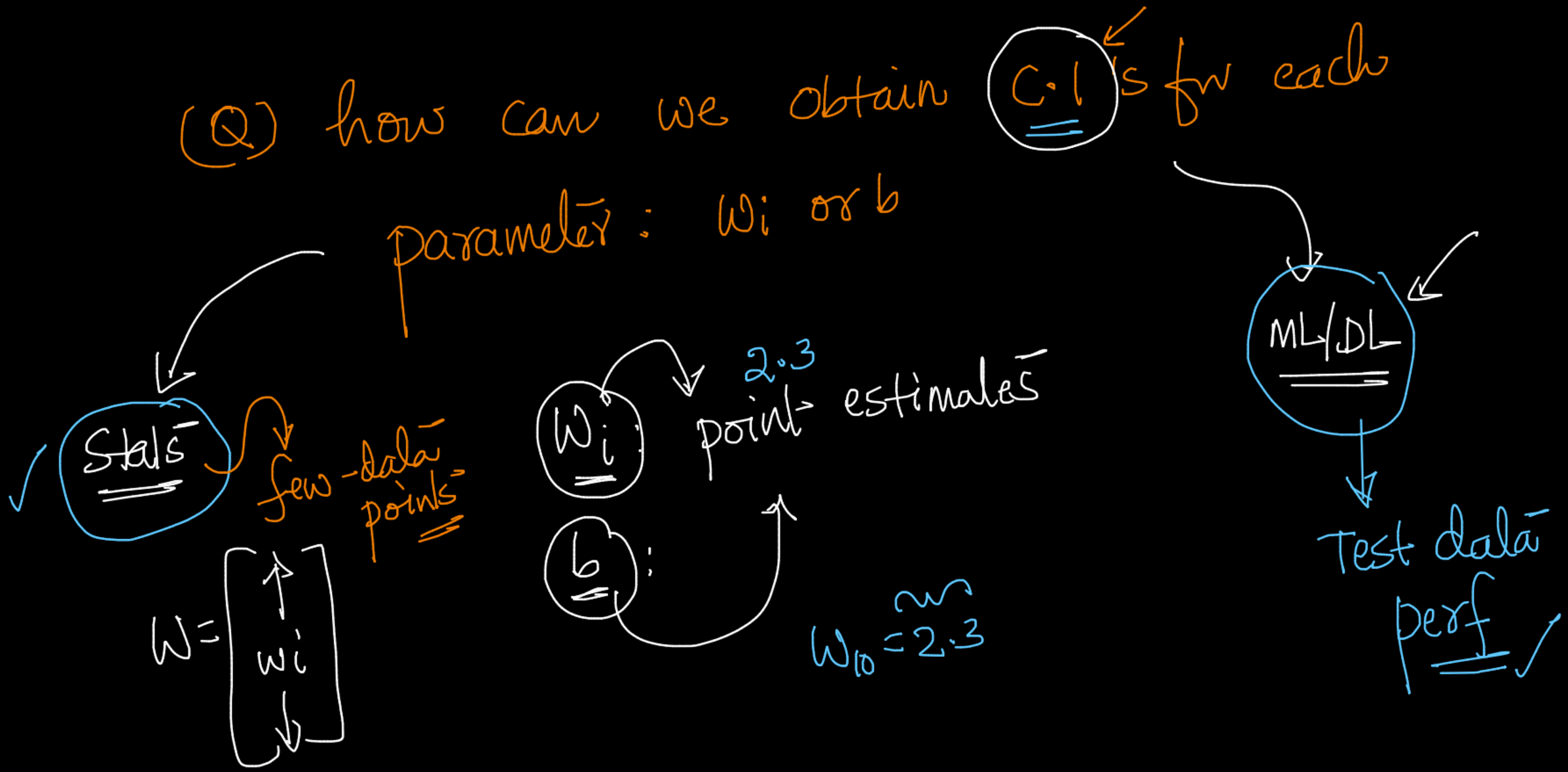
Vector → $\frac{\partial L}{\partial w} = w + C \sum_{i=1}^{n}$ { 0 if $1 - y_i (w^T x_i + b) \leq 0$; $-y_i x_i$ otherwise } for

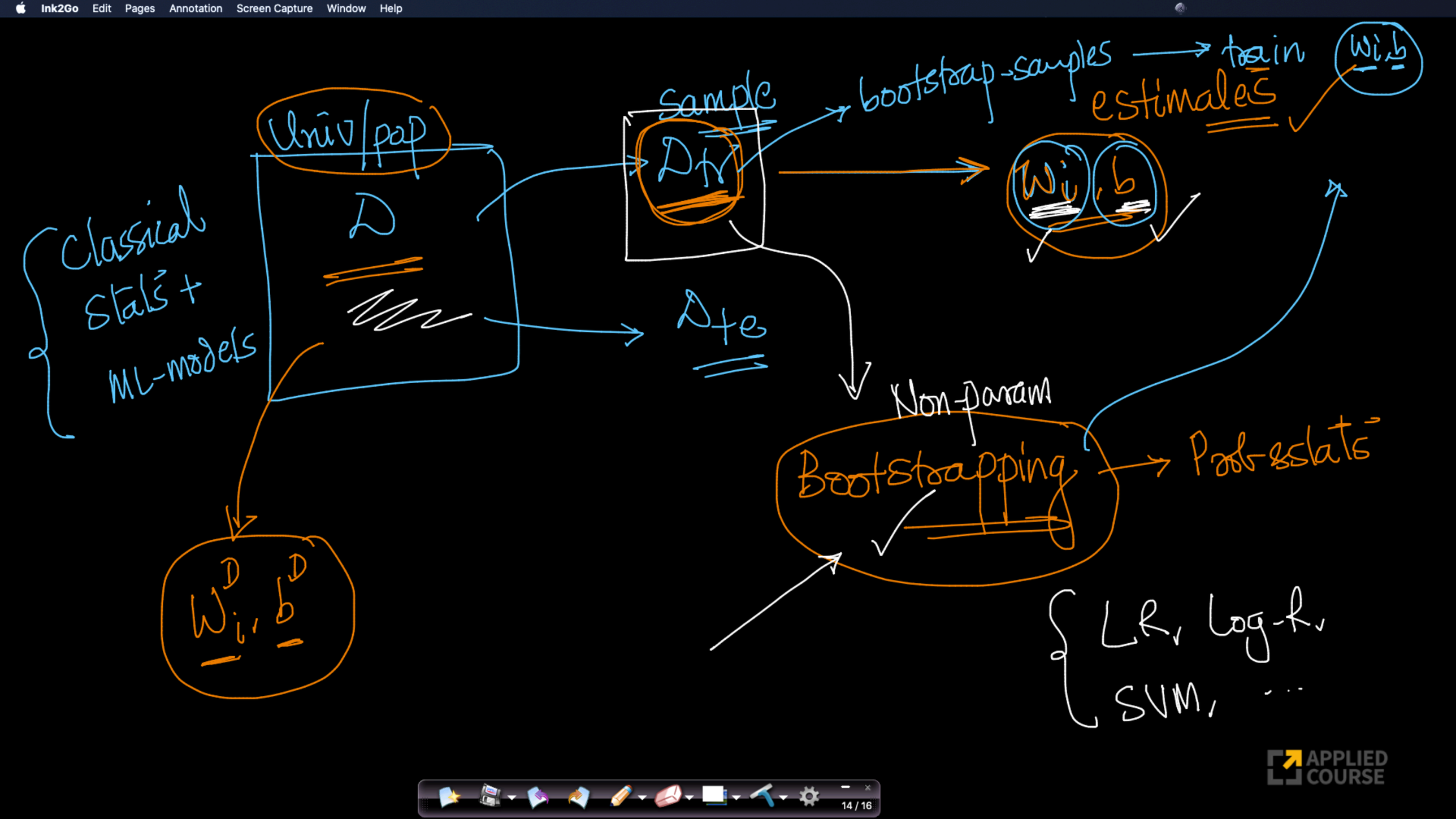
word

$\partial L / \partial b$ ✓

$\frac{\partial L}{\partial b} = 0 + C \sum_{i=1}^{n}$ { 0 if $1 - y_i (w^T x_i + b) \leq 0$; $-y_i$ otherwise }

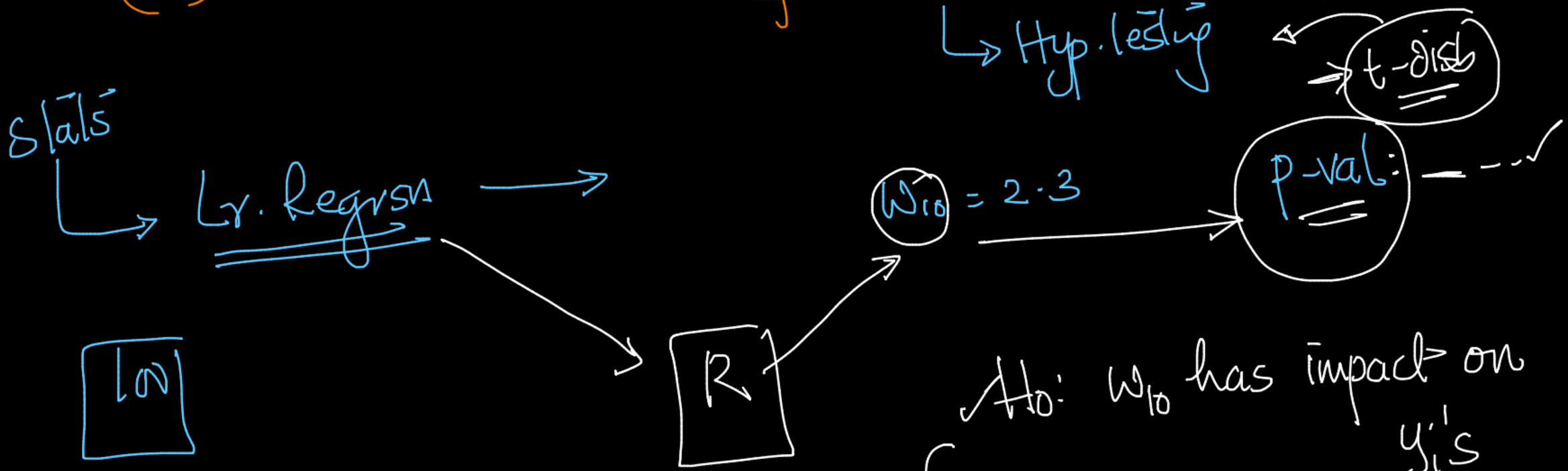
(Q) how can we obtain C.I's for each parameter: w_i or b




Univ/pop
D
sample
D_{tr}
bootstrap-samples
train
estimates
w, b
D_{te}
Classical stats + ML-models
Non-param
Bootstrapping
Prob-stats
w^D_i, b^D
LR, Log-R, SVM, ...

(Q) Can we obtain p-values for each param ?

↳ Hyp. testing



H_0: w_{10} has impact on y_i's

✓ { Permutation test + Resamp