simple ↑

D_Tr → D_train → train

D_Tr → D_Test

every A { }

code ↑

Knn = KNN(n=3)

✓ knn.fit(train)

✓ Y_pred = knn.prediction(test)

✓ compare Y_pred == Ytest → Acc → 80%

Brute force

K=3 5 7 9 11 all

odd | even

K=3

K=5 → 84%

K=7 → 89%

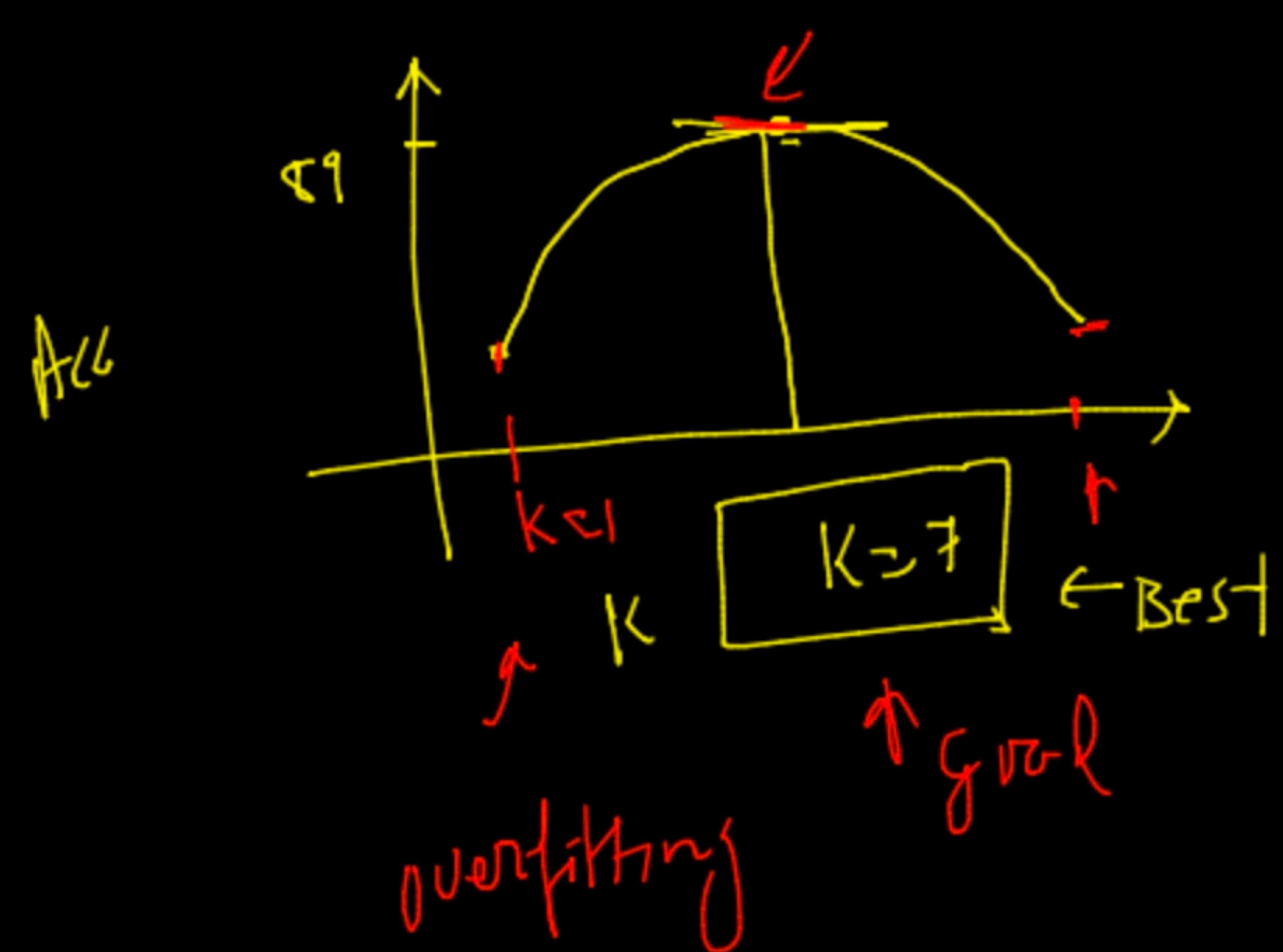
K=9 → 82%

K=11 → 78%

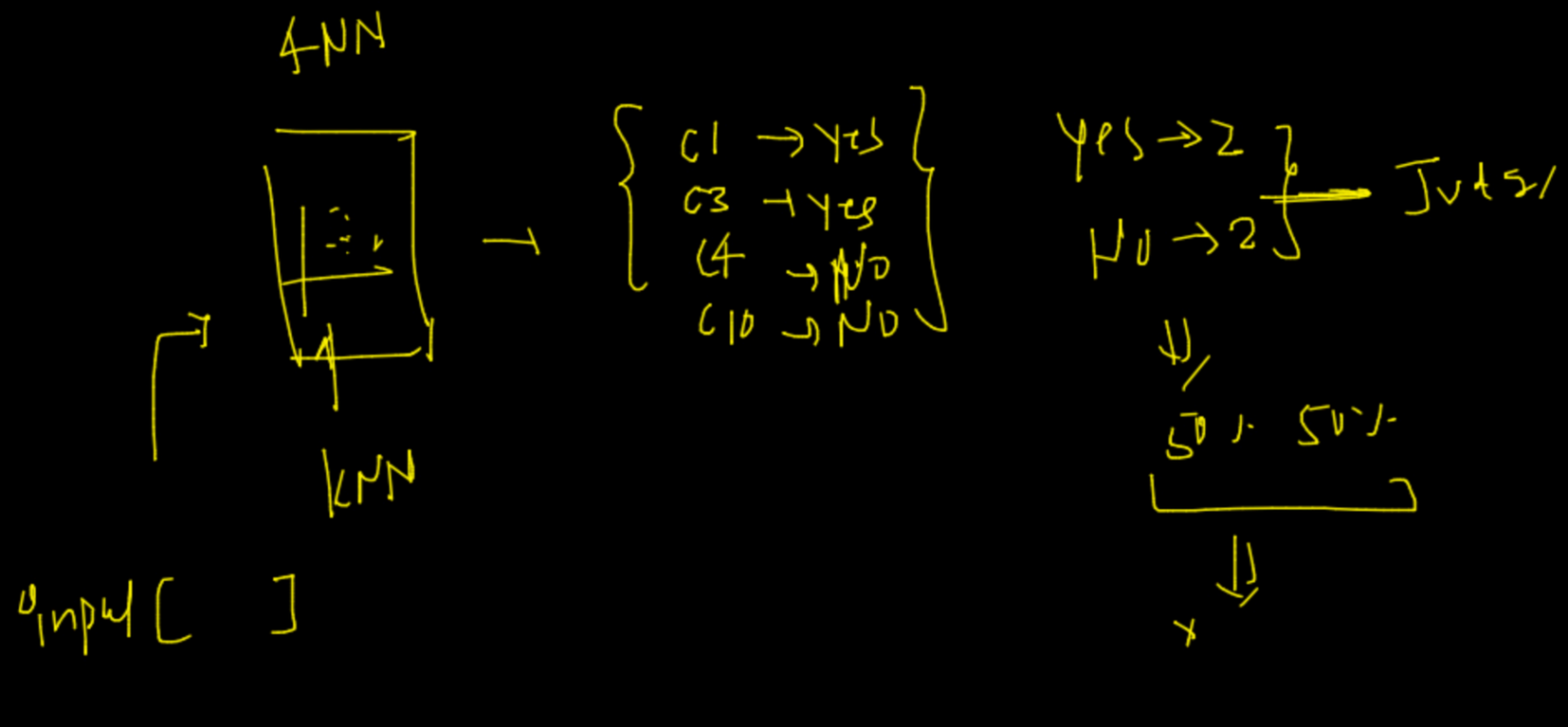
K find

Hyper-parameters

ML → Hyper-parameters tuning



1/4


4NN
1NN
input [ ]
C1 → Yes
C3 → Yes
C4 → NO
C10 → NO
Yes → 2
NO → 2
Just 2!
50 : 50 !
x

$D_x \to$ 1000pts

- $D_{Train} \to$ 800pts { ~~400 Yes~~ / ~~400 NO~~ } 500 Yes, 300 NO
- $D_{Test} \to$ 200pts { 150 Yes, 50 NO }

ML|DL

knn = knn(n=800)

knn.fit(Train)

= knn.predict(X-test)

↑ K = 800 ← 800

{ C4 … } → 500 Yes ✓, 300 NO

Input

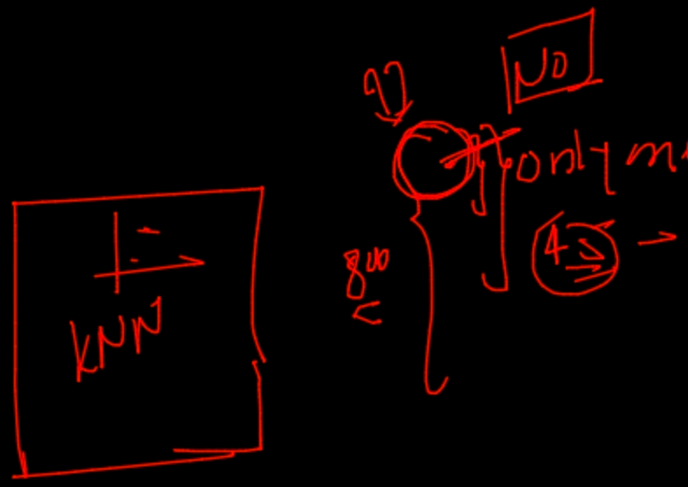
Query(Input) → Yes

3 / 3

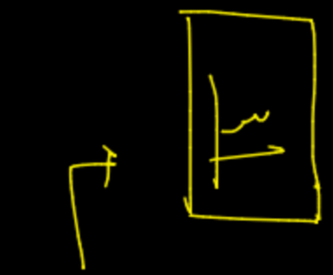
Overfitting and underfitting (interview)

K=1

K=800 →



[16, 78] → Yes/No → NO



Input → Yes

[6, 350] → Yes

[16, 780] → Yes

Why ?

D_x → 80% D_{train} → 64% D_{train} → ☐ ← train my model

D_{train} → 16% D_{cv} → I will find Best value of k

D_x → 20% D_{Test}

k = 7

knn(7) → get y

KNN → Regression

↑

-

↓

eg

Day	Discount	Brand	Volume
1	-	-	340
2	-	-	386
3	-		
4	-	-	
100	-	-	

X → Y → Continuous / Real

3NN

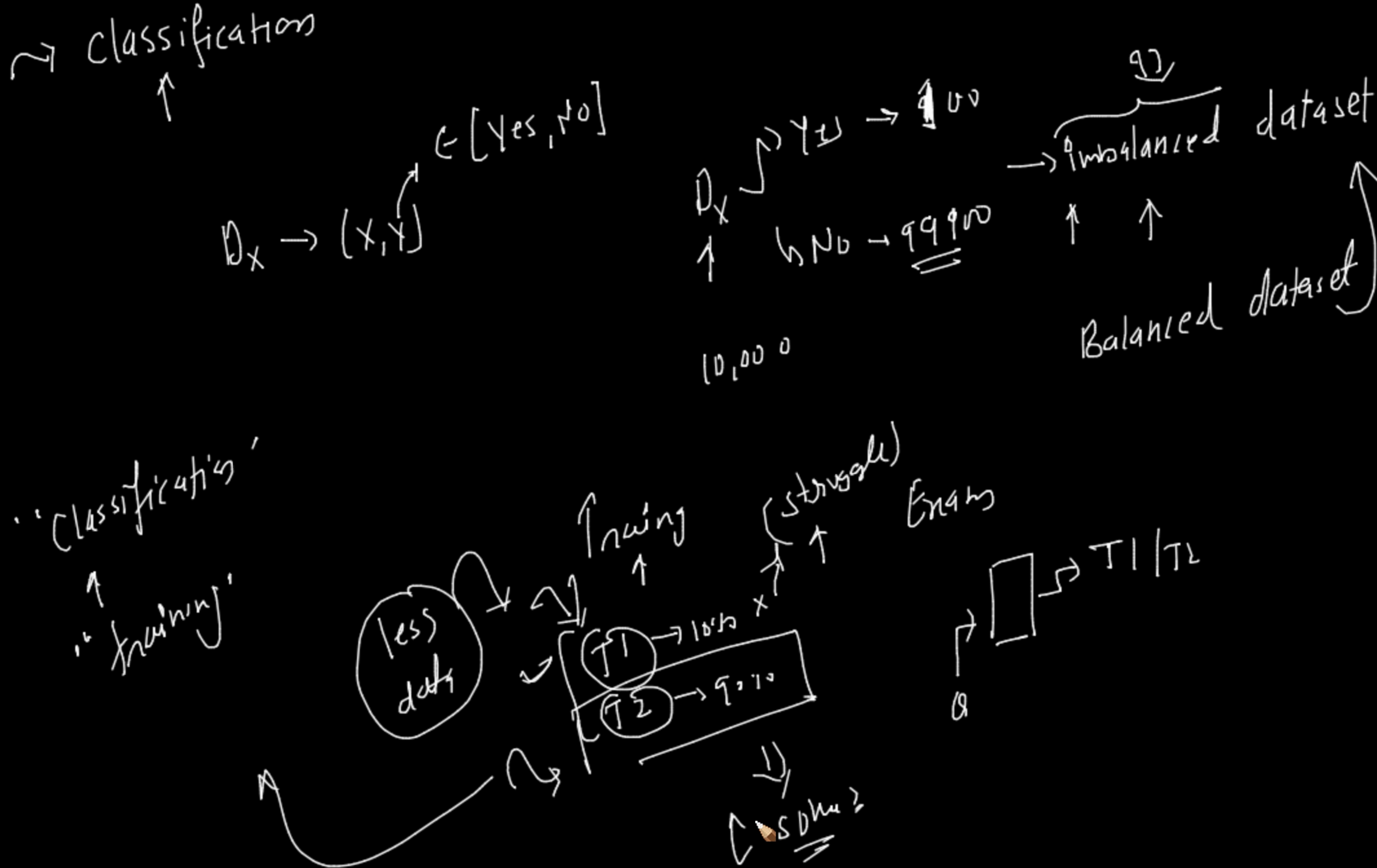
{ 2 → 380
4 → 400
100 → 280 }

Avg

Median

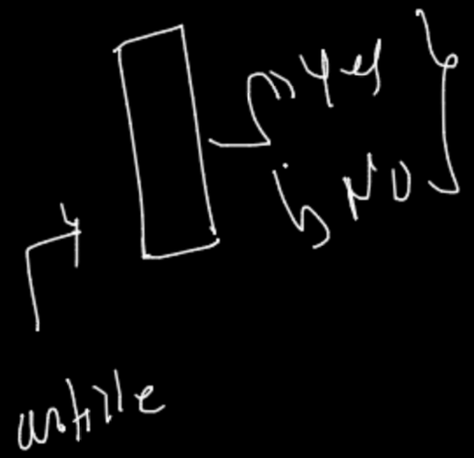
[10%, ?] → ?

Classification / Regression


Classification
Dx → (x, y)
∈ [Yes, No]
Yes → 100
No → 99900
imbalanced dataset
Balanced dataset
10,000
'Classification'
'training'
less data
Training
(struggle)
Exam
T1 → 100%
T2 → 90%
T1 / T2

Classification

1. i) Newspaper article → Real / False

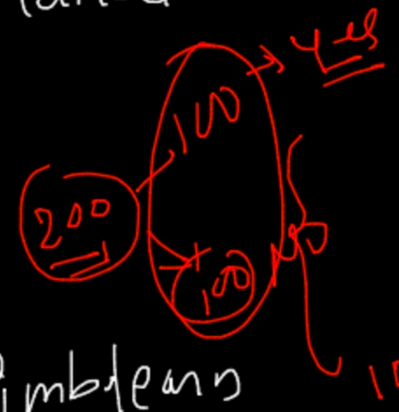


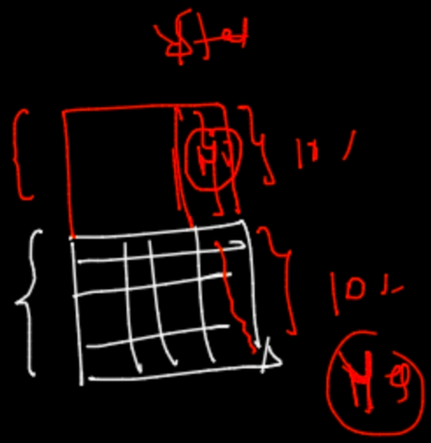
Imbalance dataset ⟶ Balance dataset

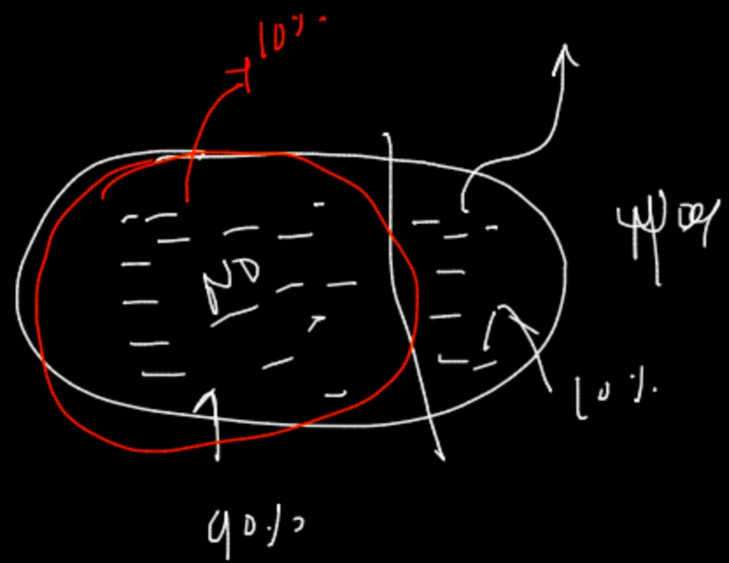
1) undersampling
2) Oversampling

→ imblearn

$D_x(1000)$ → 100 (Yes)
→ 900 (No)







80% Data

Model

'200'

Yes + 100

No + 100

Balance

undersampling

'1000'

Yes

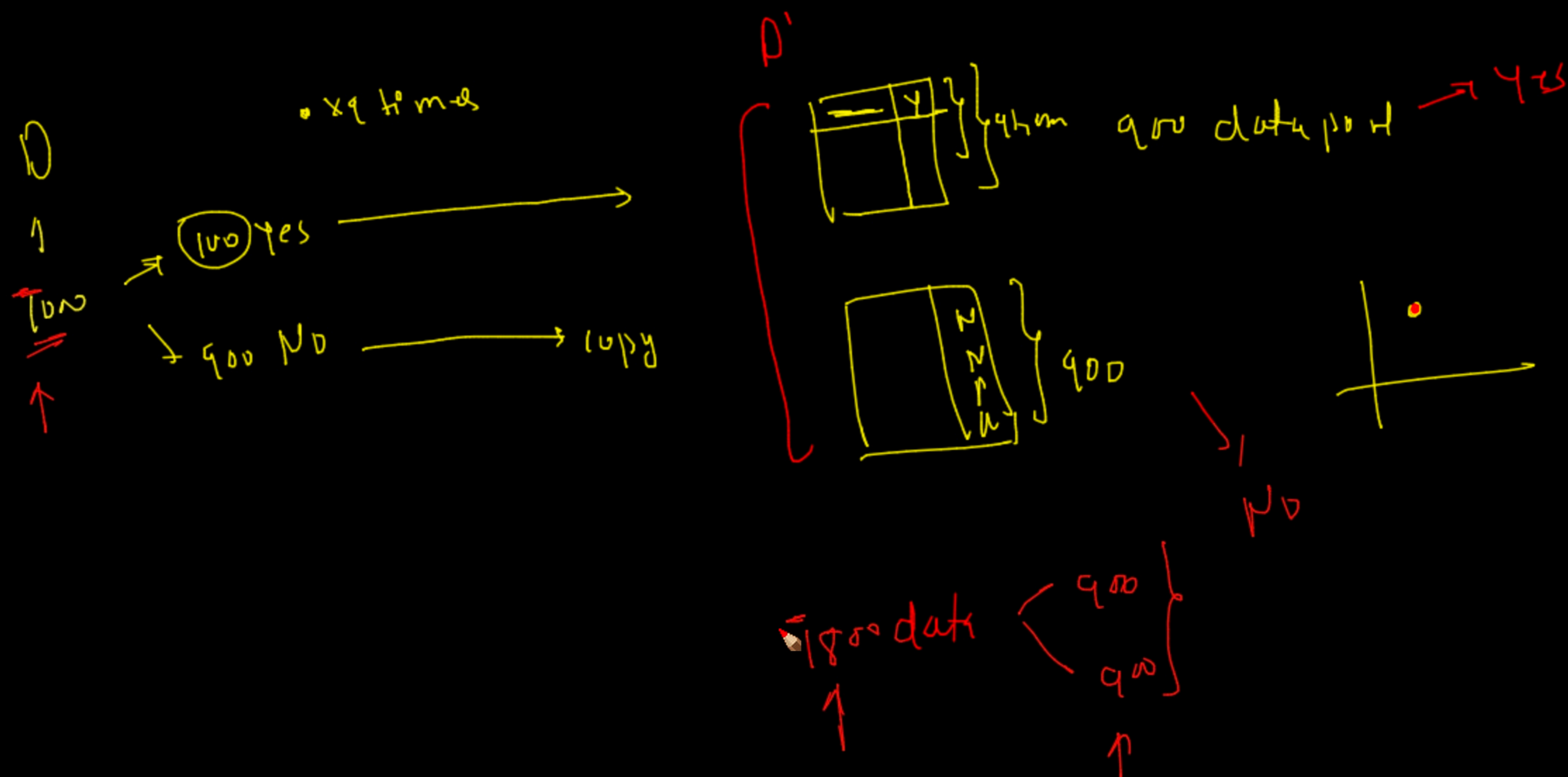
No

Random 100 data

100

100

200



D
1000
100 yes
900 No
copy
x9 times
D'
900 data point
yes
900
No
1800 data
900
900