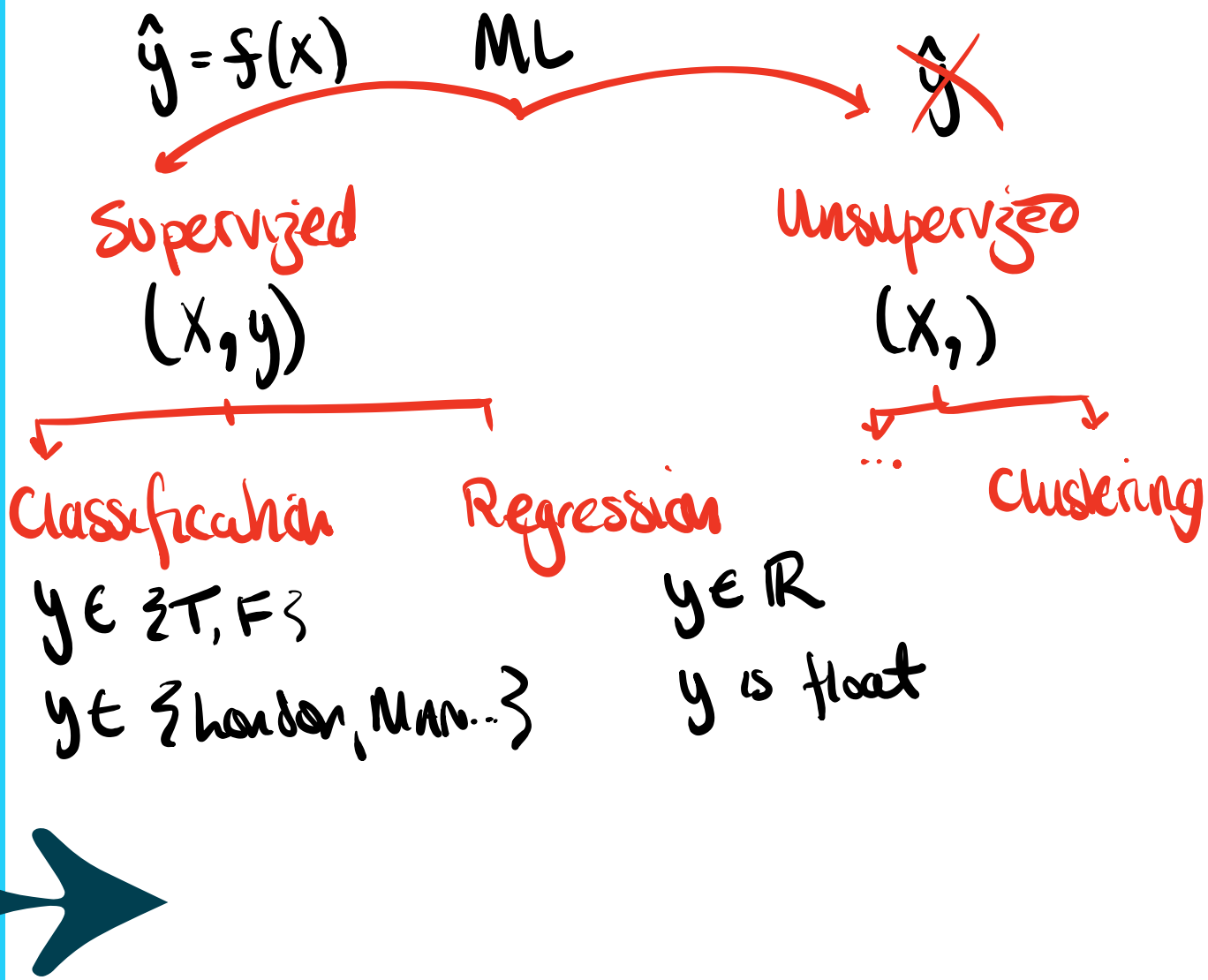
Intro to ML

Method & algorithms

Review


$\hat{y} = f(x)$
ML
$\hat{y}$
Supervised
(X, y)
Unsupervised
$(X,)$
...
Classification
Regression
Clustering
$y \in \{T, F\}$
$y \in \{London, Man...\}$
$y \in \mathbb{R}$
y is float

How does ML work?

Sup. Unsup.

Data (x, y) (x_i)

Alg

FOR el. in DATA —

$\hat{y} = \hat{f}(x^{NEW}; w, b)$ predict(x^{NEW}, w, b)

↑ template is pre-defined

Aside: Notation – Data

$$X : \{ (x_0, x_2)^0, (x_0, x_1)^1, \dots \}$$

Matrix

Row index

Row

Col 0, Col 1

$$\vec{y} : \{ y^0, y^1 \dots \}$$

Vector = "One Column"

Row	x_{Age}	$x_{Location}$	y_{Like}
1	18	300	✓
	x^1_0	x^1_1	y^2
2	21	400	X

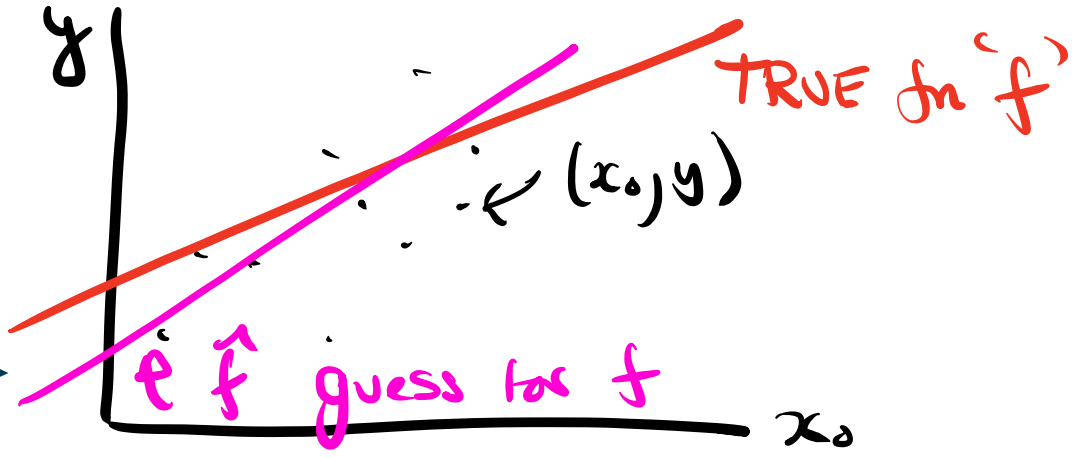
Notation: functions & predictions

$(X, y) \rightarrow Alg \rightarrow \hat{f}(X; w, b)$

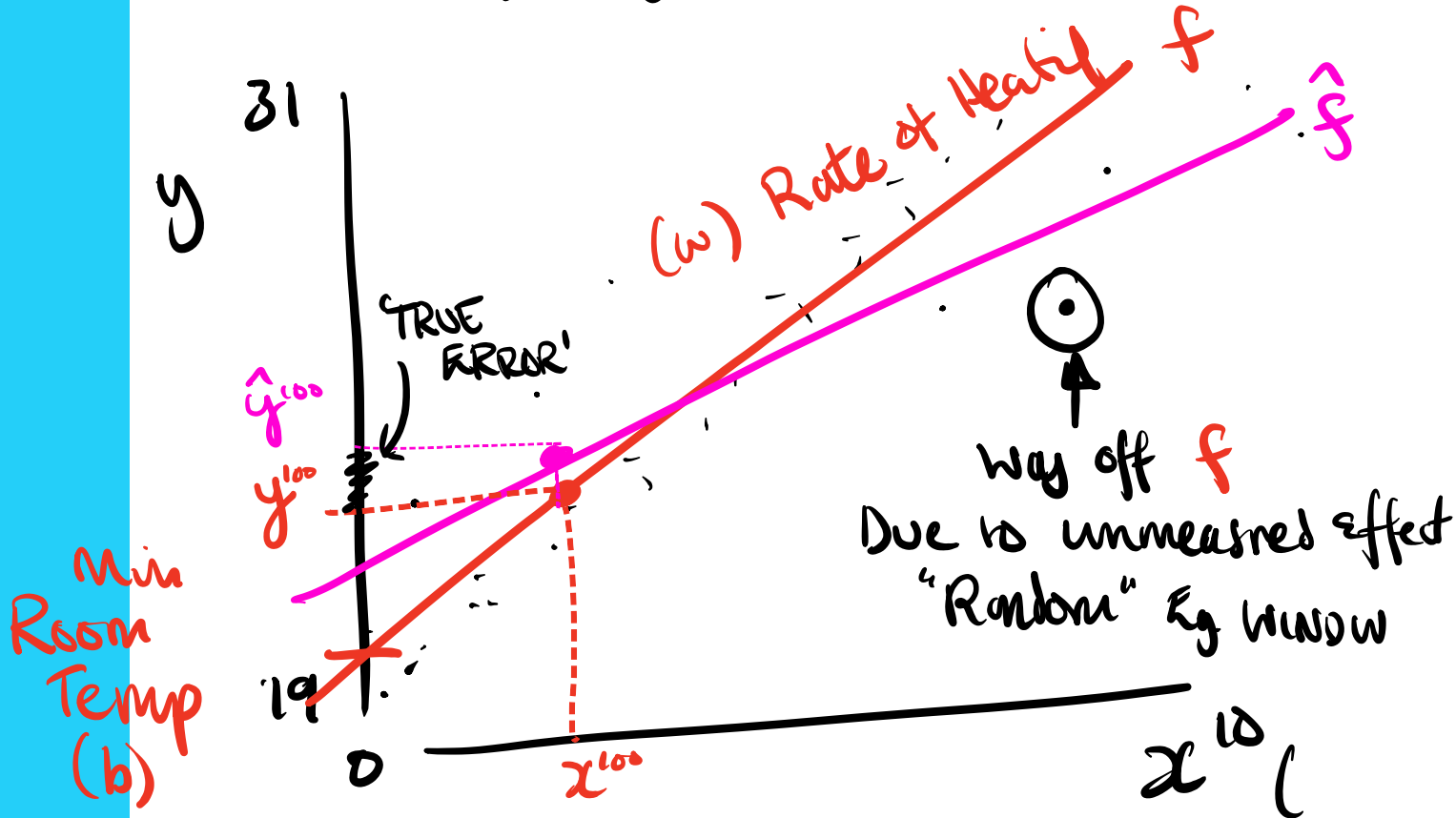
$$\hat{y} = \hat{f}(X \ldots)$$

$\hat{y}$: a guess for y (unknown)

$\hat{f}$: a guess for f ← guess



Eg. temp (y) dial (x)



IN ML. f is mostly unknown.

Cont..

Parameters / weights

$$\hat{y} = \hat{f}(\vec{x}^{NEW}; \vec{w}, b)$$

Variable input

Constant input

Note

$$\vec{w} = [w_0, w_1, \dots]$$

$$\vec{x} = [x_0, x_1, \dots]$$

NB. $\vec{c}$ means 'a col of c' (vector)

Review

$$(x, y) \rightarrow \text{Alg} \rightarrow \hat{y} = \hat{f}(x^{NEW}; w, b)$$

Q&A where does 'f' come from?

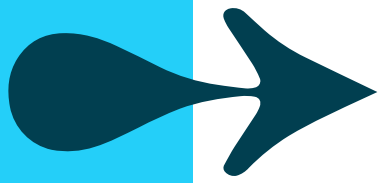
Eg. heating flat

$$\text{temp} = f(\text{dial}) = \underbrace{3}_{\text{Effect of Heater.}} * \text{dial} + \underbrace{19}_{\text{Min temp.}}$$

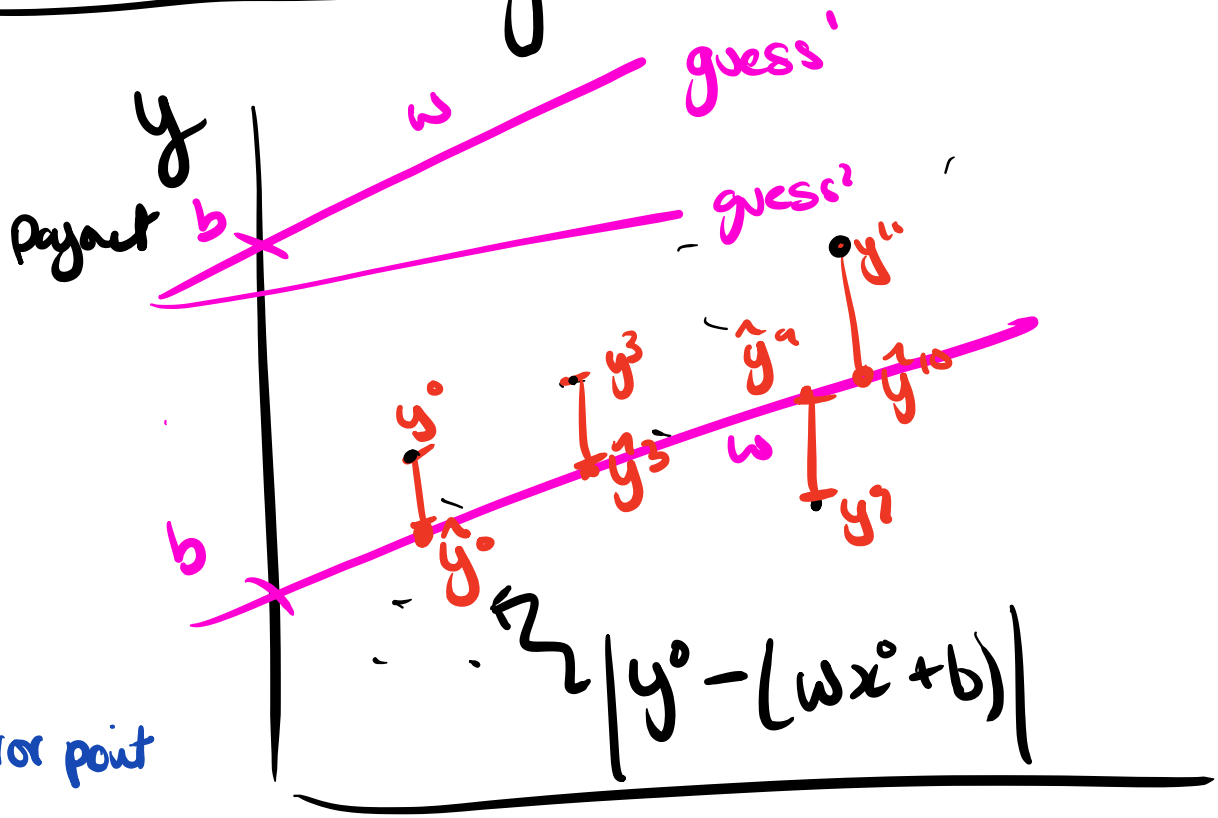
$$\hat{f}(x, 2.9, 18) = 2.9x + 18$$

Algorithms & Models

- Linear Regression
- k-Nearest Neighbors



Linear Regression



loss = error point

h - total loss

$$h = \sum_{r}^{\text{All Rows}} |y^r - wx^r + b|$$

Corresponding Python to Mathematics

$$h = \sum_{r}^{\text{All Rows}} |y^r - wx^r + b|$$

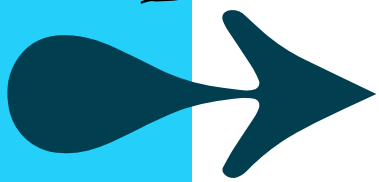
Where is x, y? MISSING

$\forall x, y$

```
Sum( [abs(yr - (w * xr + b))
      for xr, yr in zip(X,y)] )
```

Programming MORE Explicit

$\forall \ldots \quad x, y$



Alg

fixed data

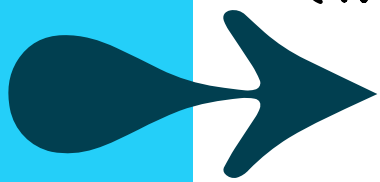
$$L(w,b) = \sum_{r}^{\text{All Rows}} |y^r - wx^r + b|$$

Goal of most ML,

$$\arg\min_{w,b} L(w,b)$$

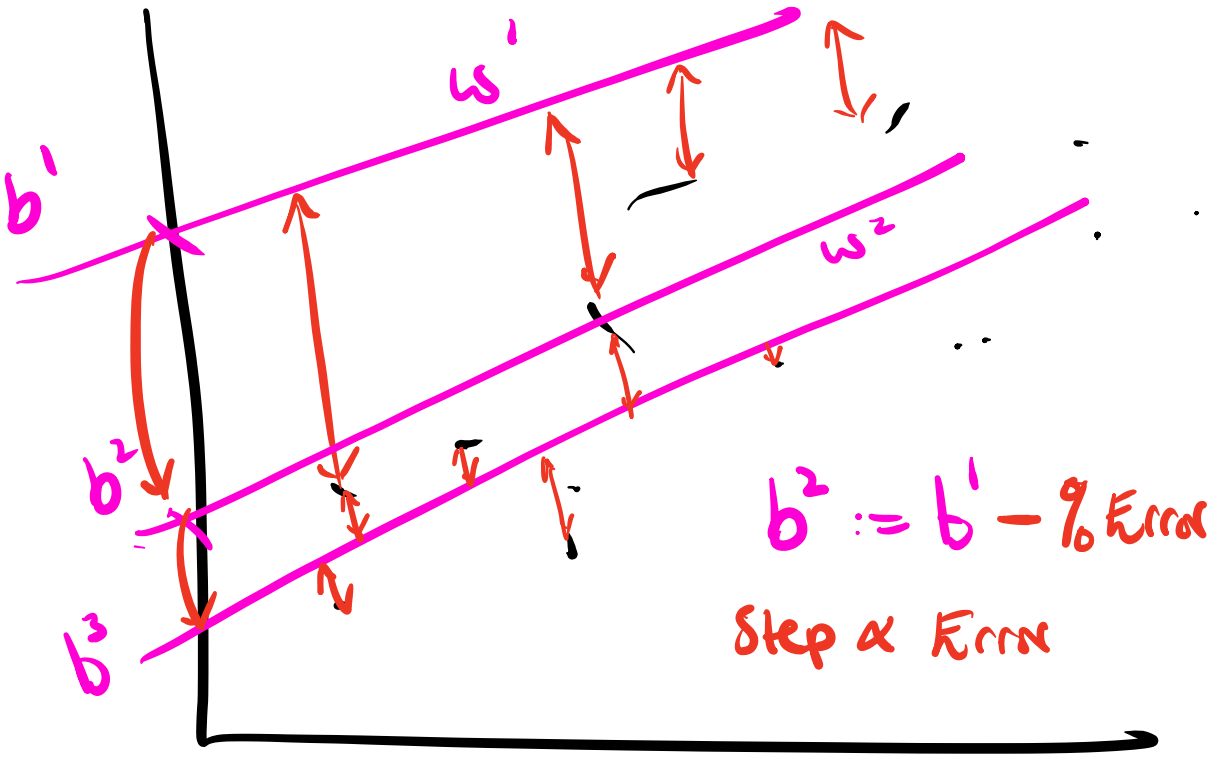
arguments which minimize

(INPUT) (OUTPUT)

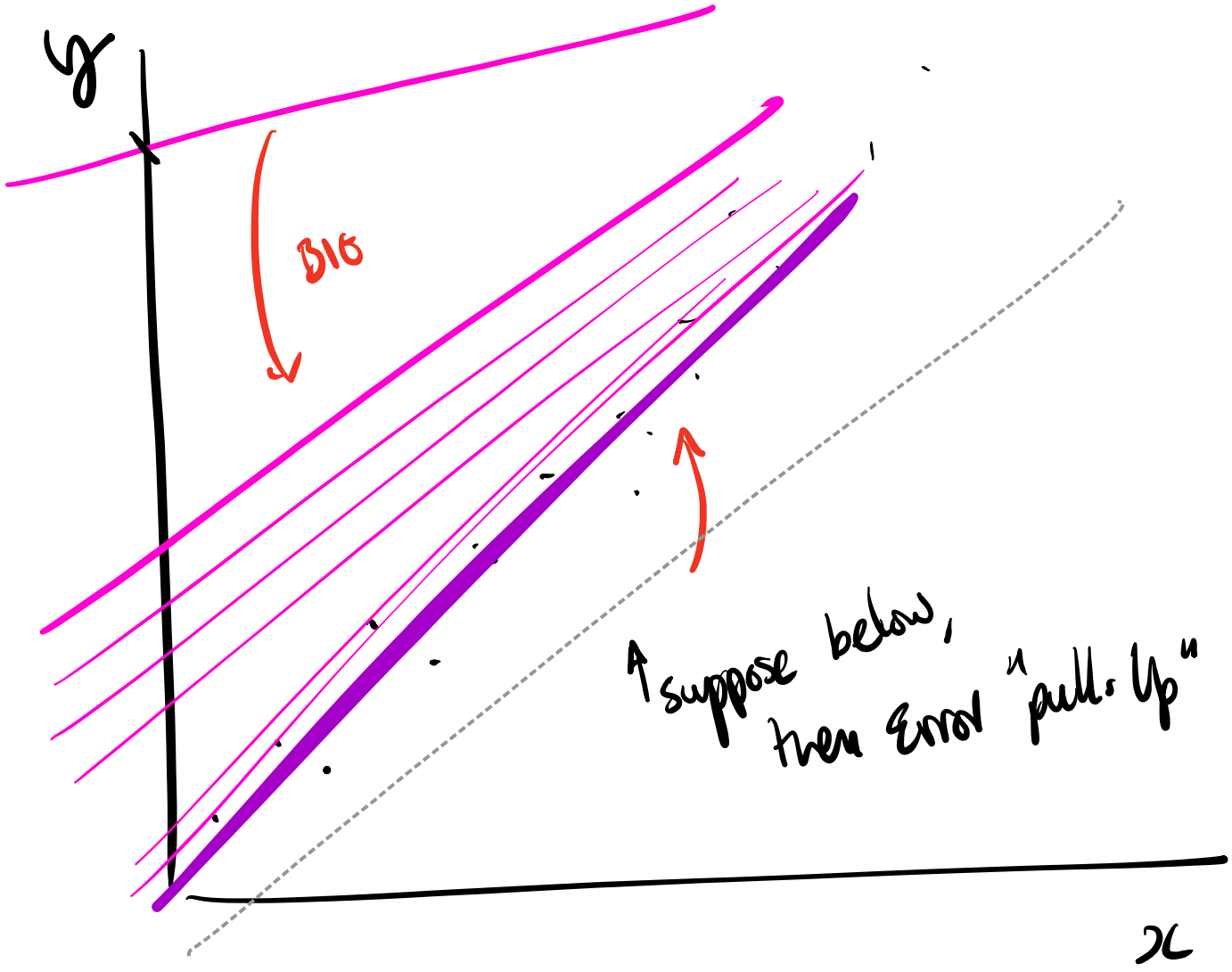


Linear Regression

$w^1, b^1 := \text{Random}()$



"Distance from $\hat{f}$ to Data is L (i.e. total error)"


y
BIG
suppose below,
then Error "pull. Up"
x

K-Nearest Neighbors

Aside

Alg

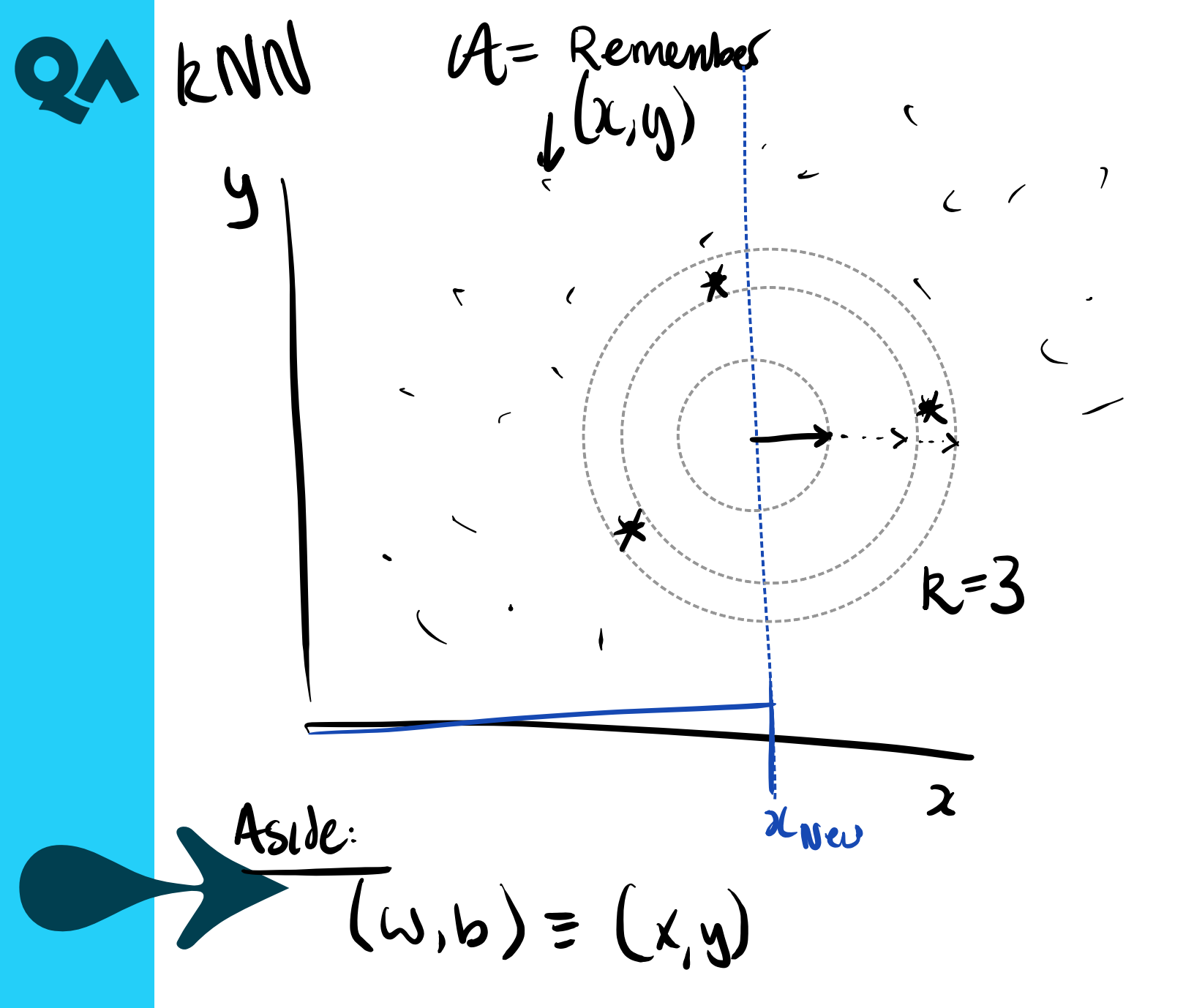
Parametric

assume shape

Non-Parametric

assumes no shape

1. Data
2. Alg = "Remember Data"
3. $\hat{f}$ = find similar

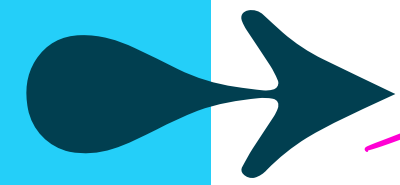

kNN
A= Remember
(x,y)
y
x
x_{New}
R=3
Aside:
(w,b) ≡ (x,y)

$\hat{f}$:

```
SELECT MEAN (y)
FROM (x, y)
ORDER BY abs(x - x_NEW)
LIMIT R
```

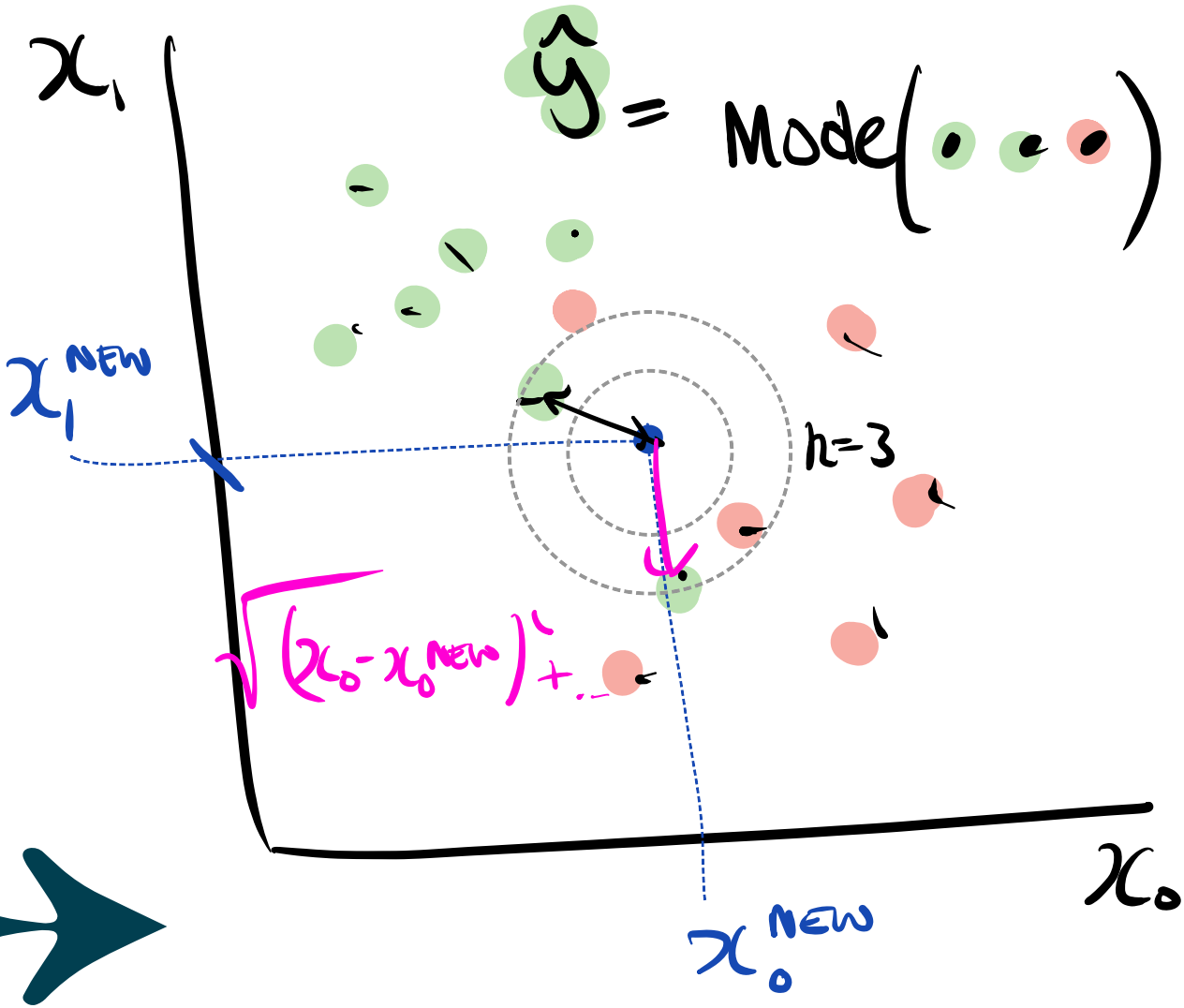
ORDER BY

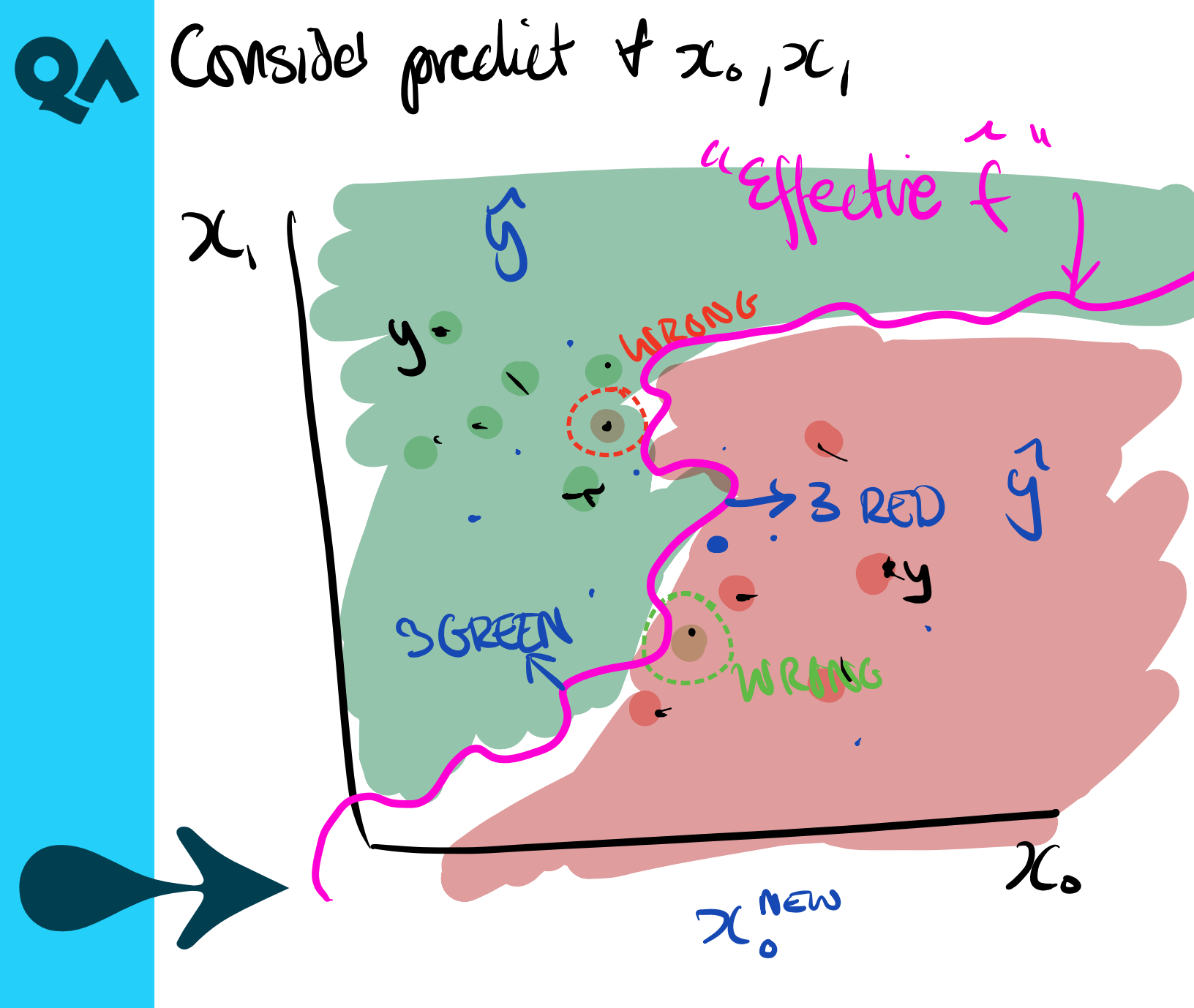
$$\sqrt{\left(x_0 - x_0^{NEW}\right)^2 + \left(x_1 - x_1^{NEW}\right)^2}$$

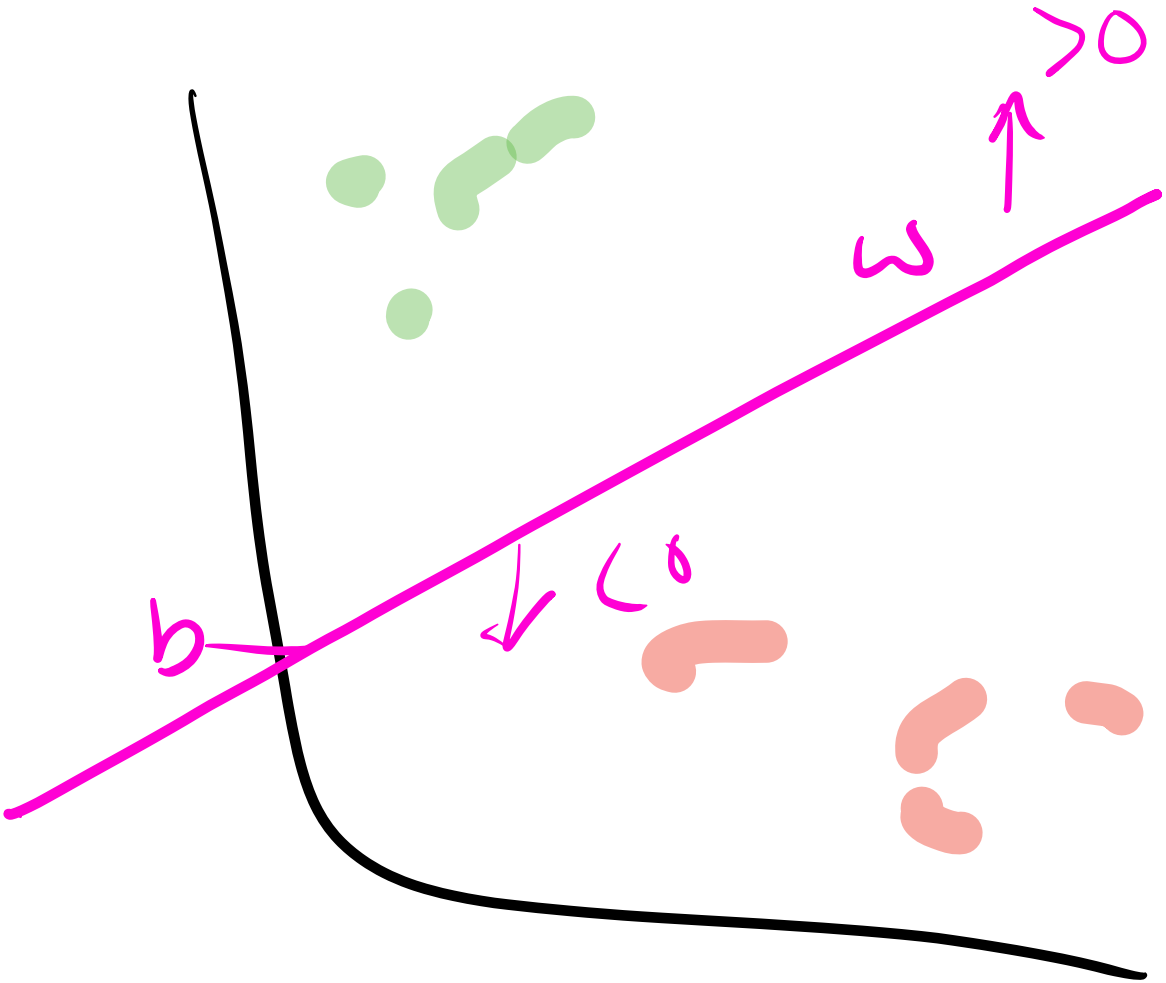


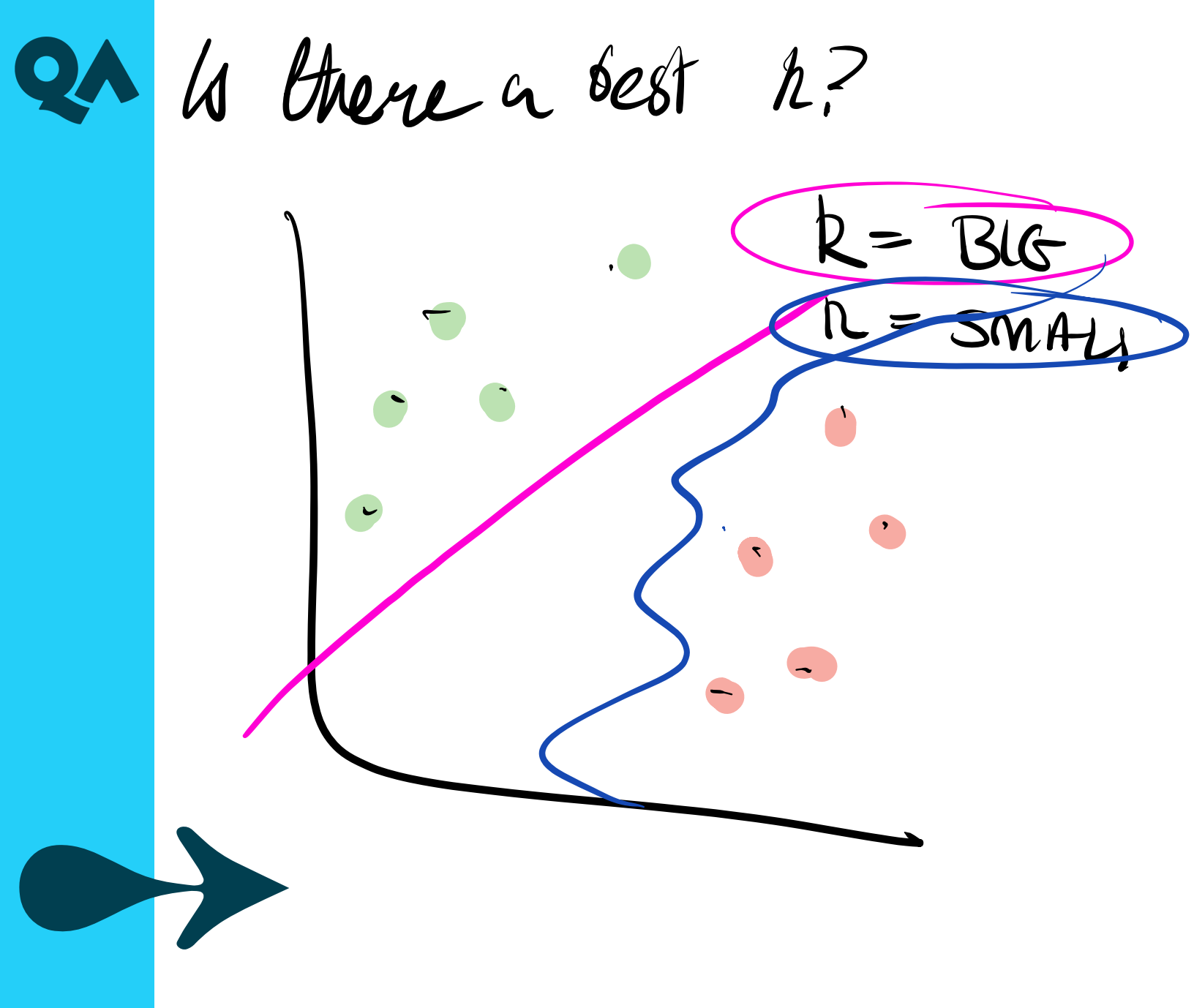
"Radical (ie. Cruelw) distance"

KNN for Classification




Consider predict $\forall x_0, x_1$
"Effective $\hat{f}$"
x_1
$\hat{y}$
y
WRONG
3 RED
$\hat{y}$
y
3 GREEN
WRONG
x_0
x_0^{NEW}


w
>0
b
<0


Is there a best k?
k = BIG
k = SMALL