session 4:

Supervised learning

M. Kundegorski

3rd March 2020

Centre for Ecological Sciences
Indian Institute of Science
Bengaluru, India

Traditional supervised learning

Traditional supervised learning still proves itself useful

It is a framework

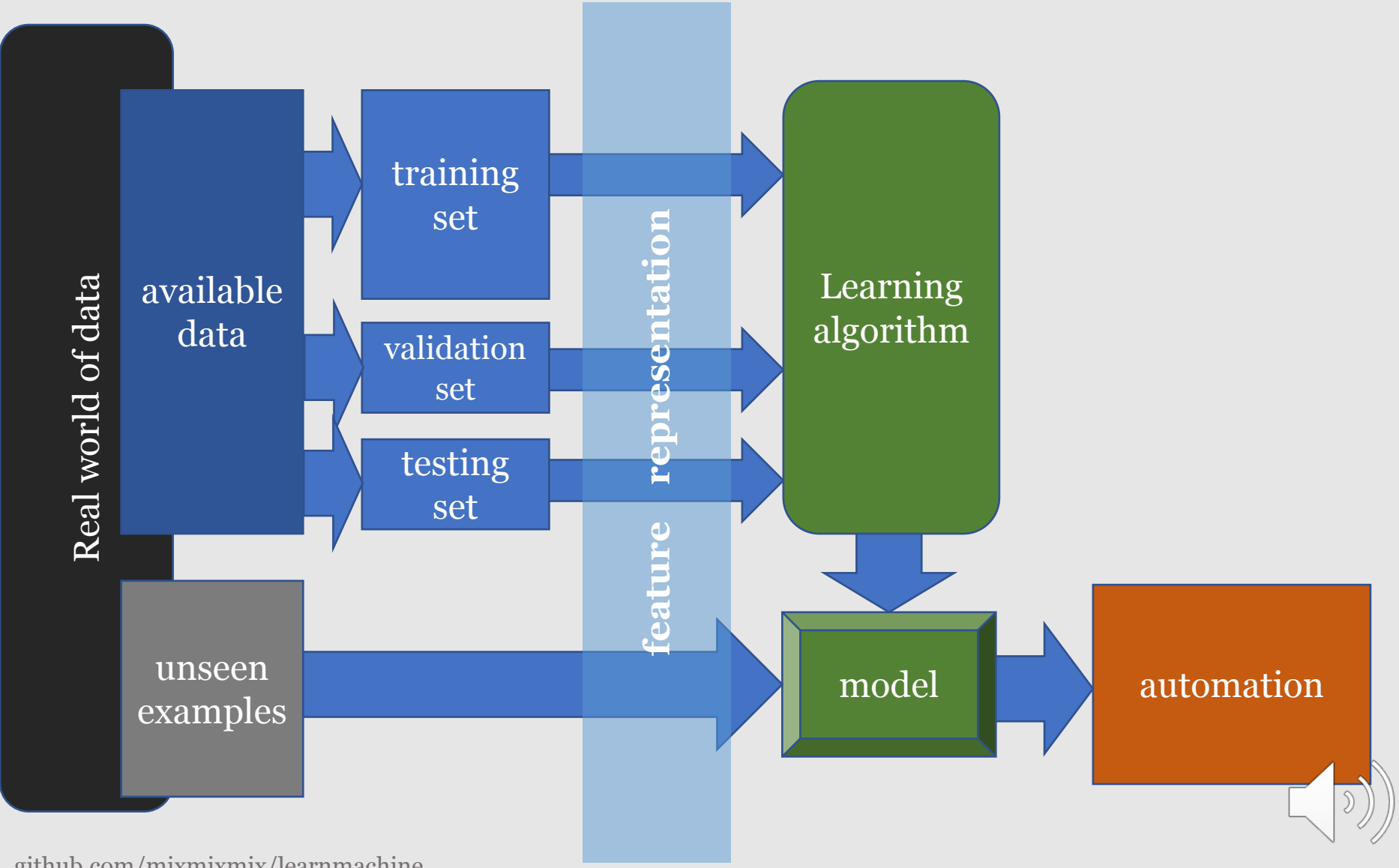
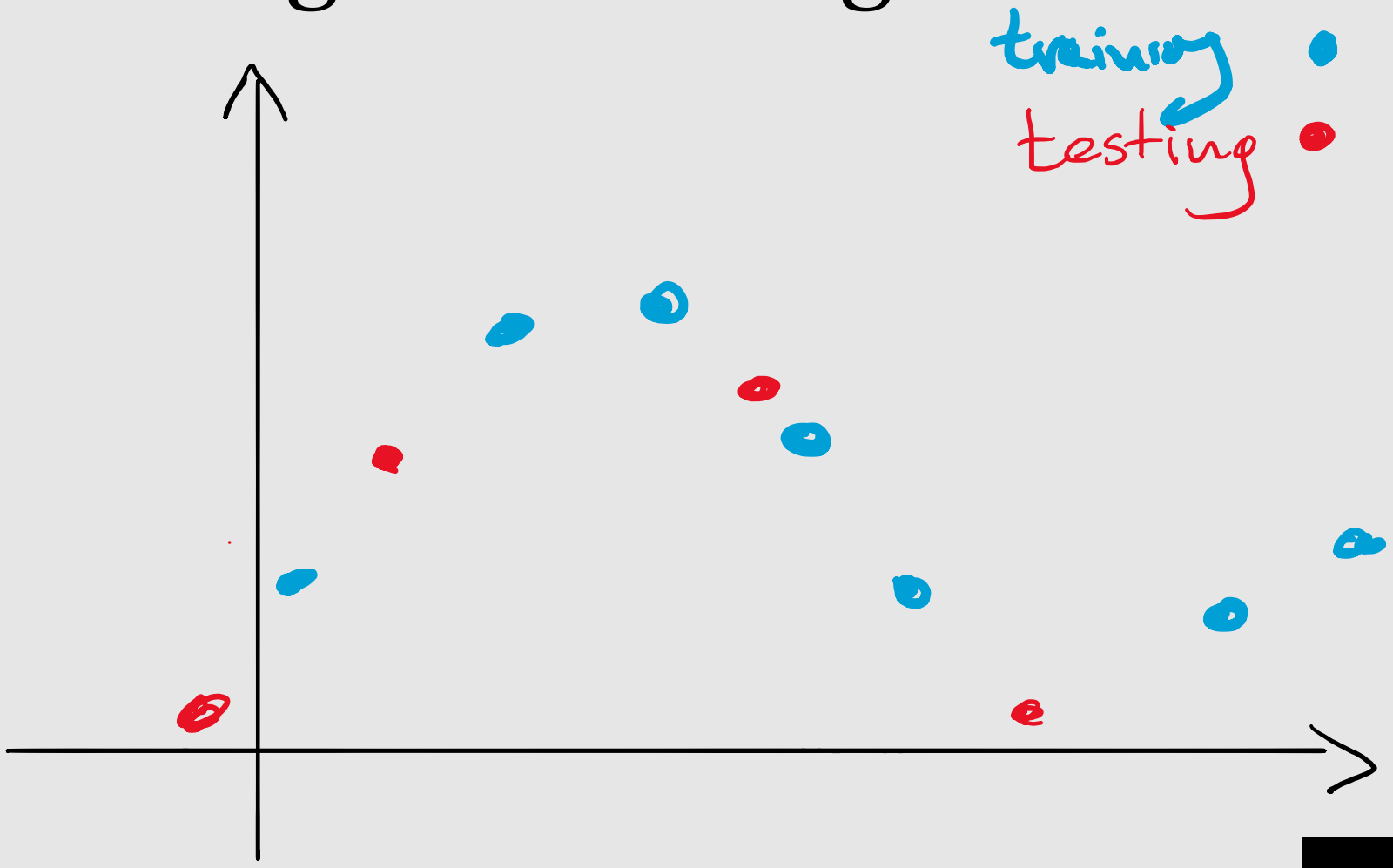


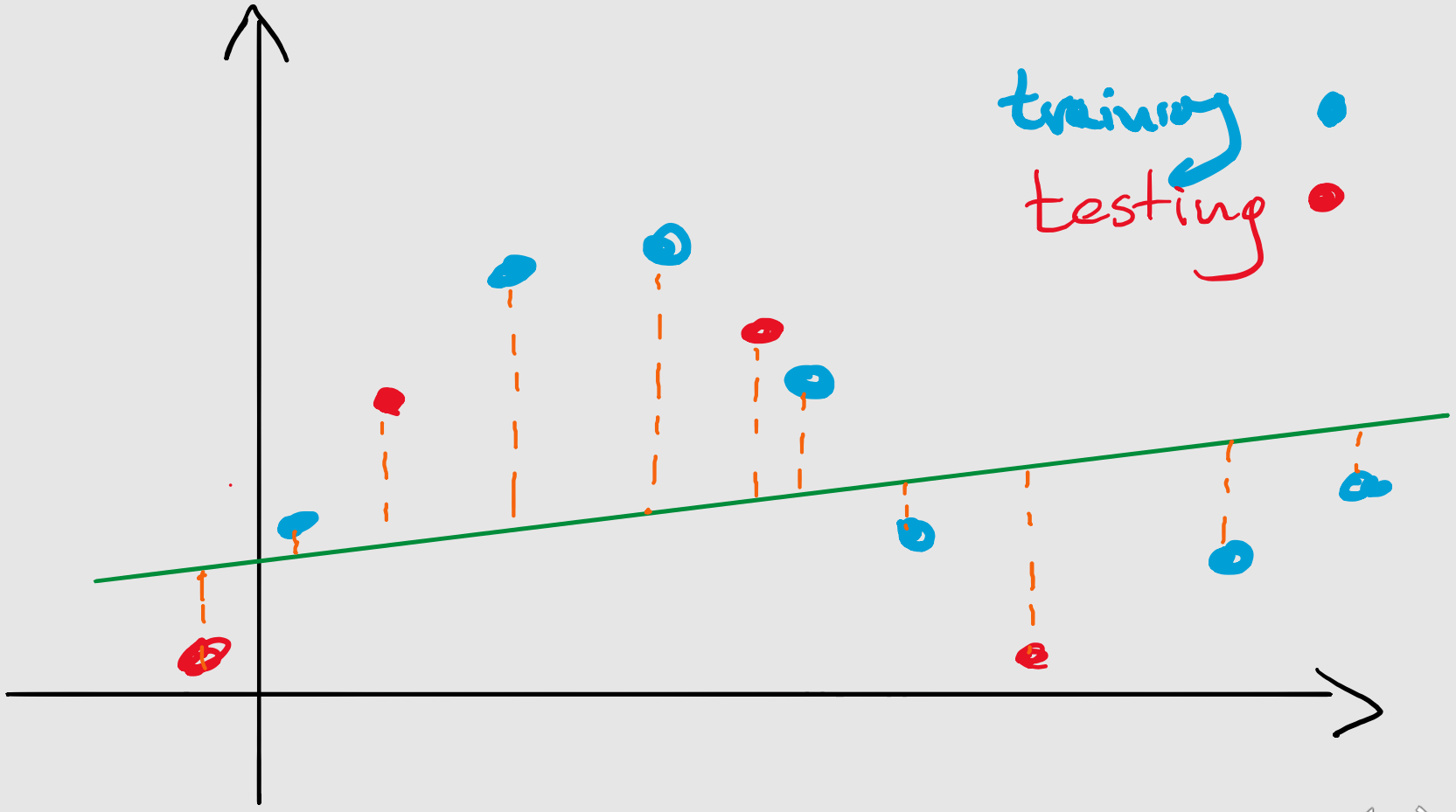
Image classification

- The most common and fundamental task
- two or more classes of objects
- object is already segmented and localised => presented as an image

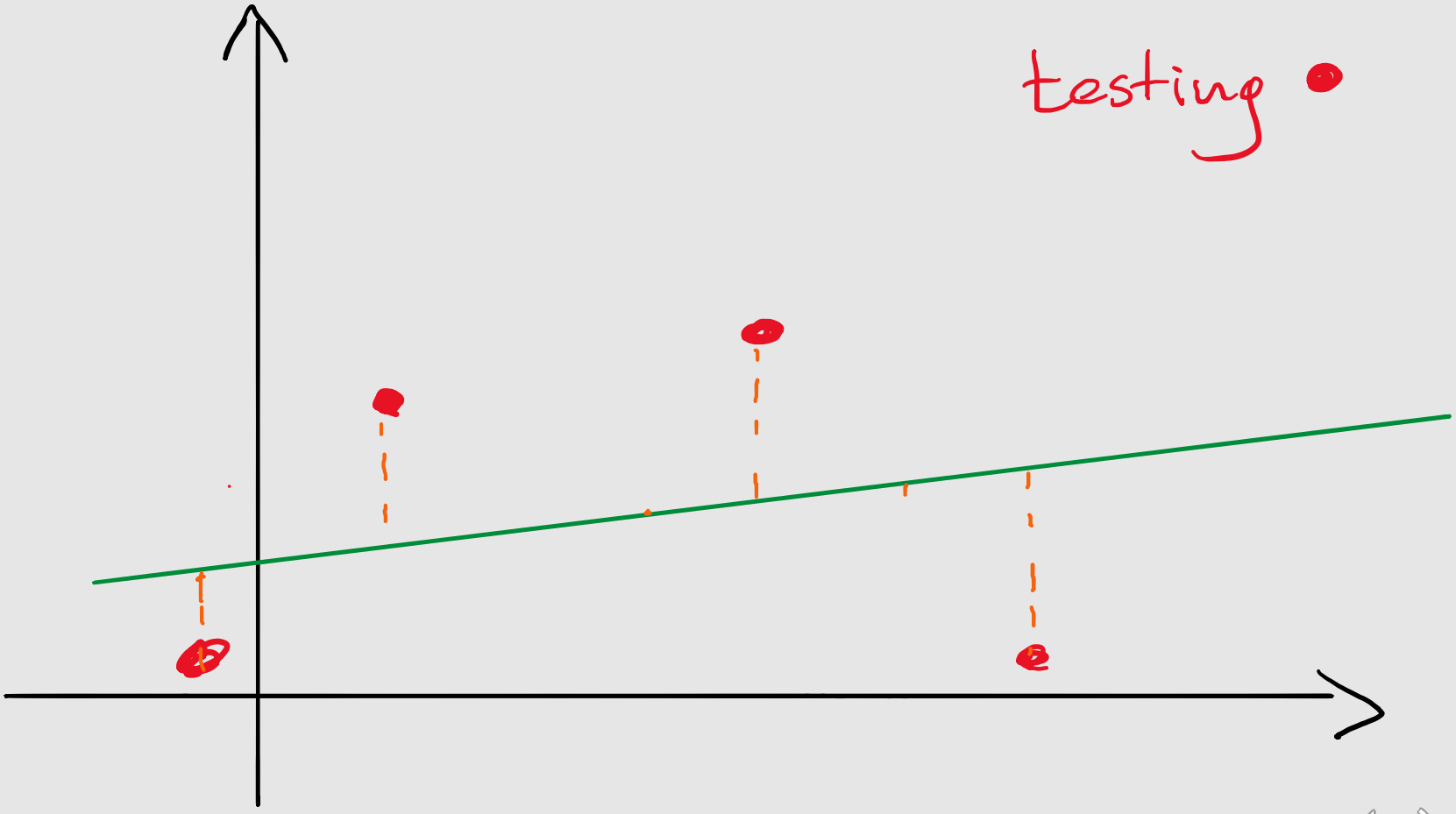
Training and Testing Errors



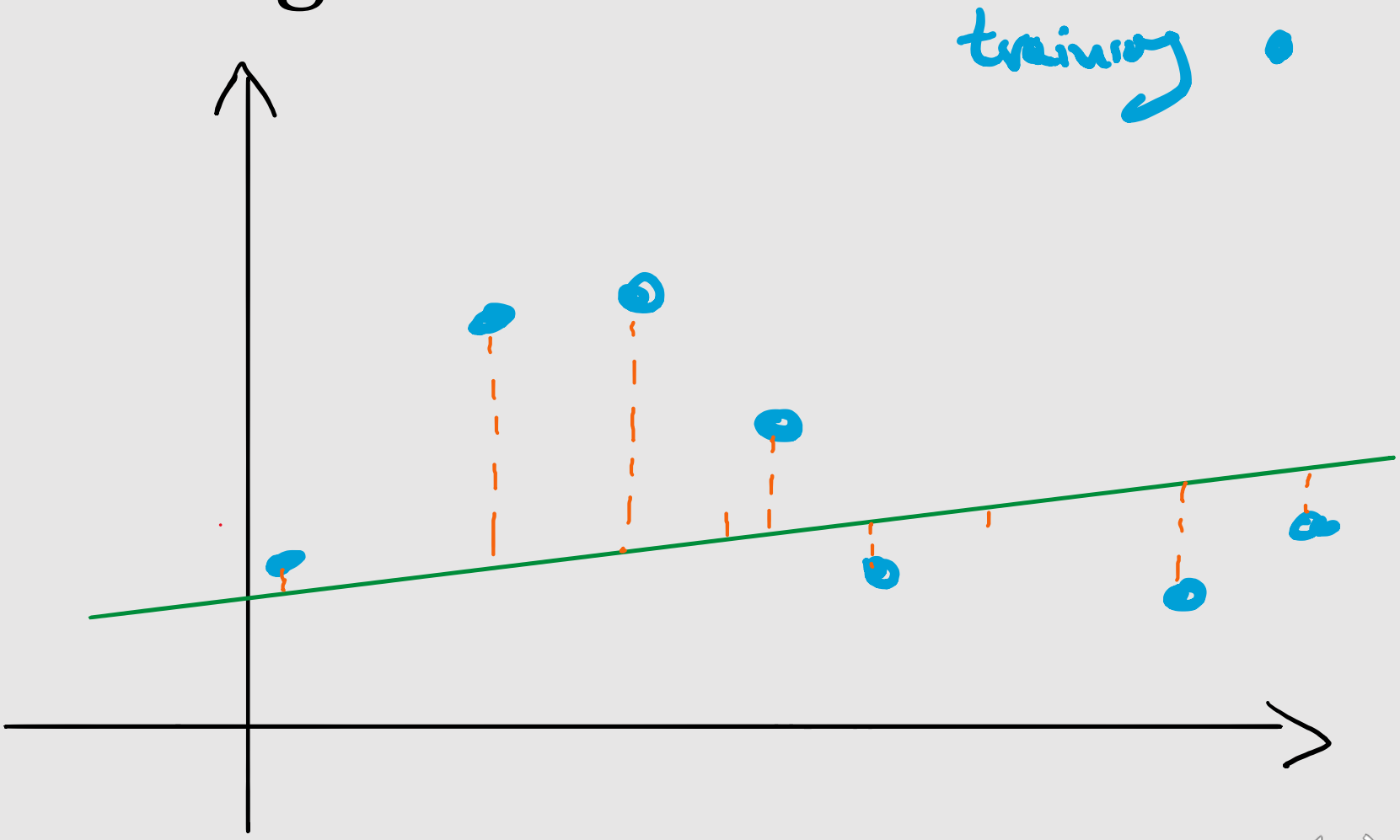
Training and Testing Errors



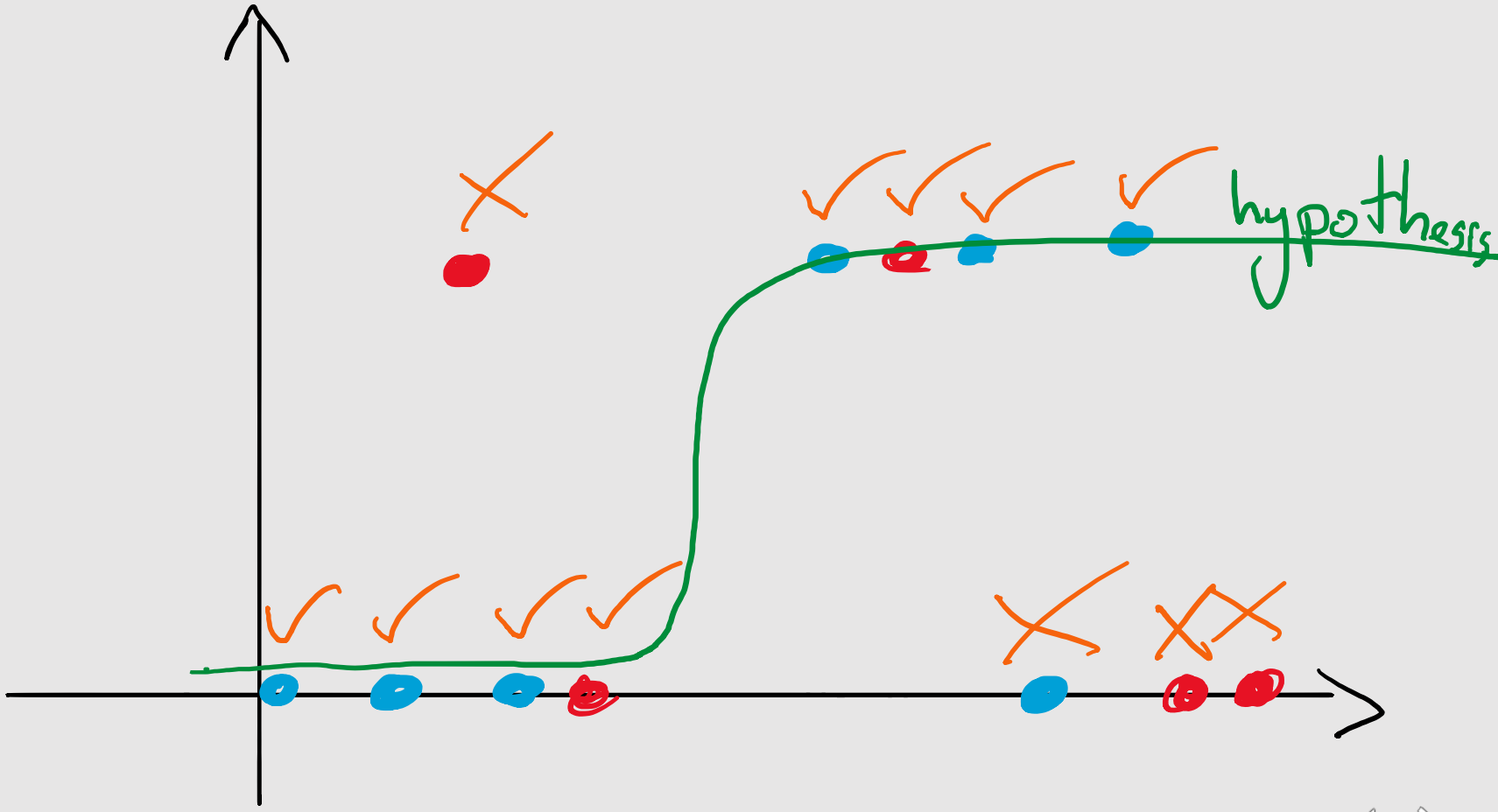
Testing Error



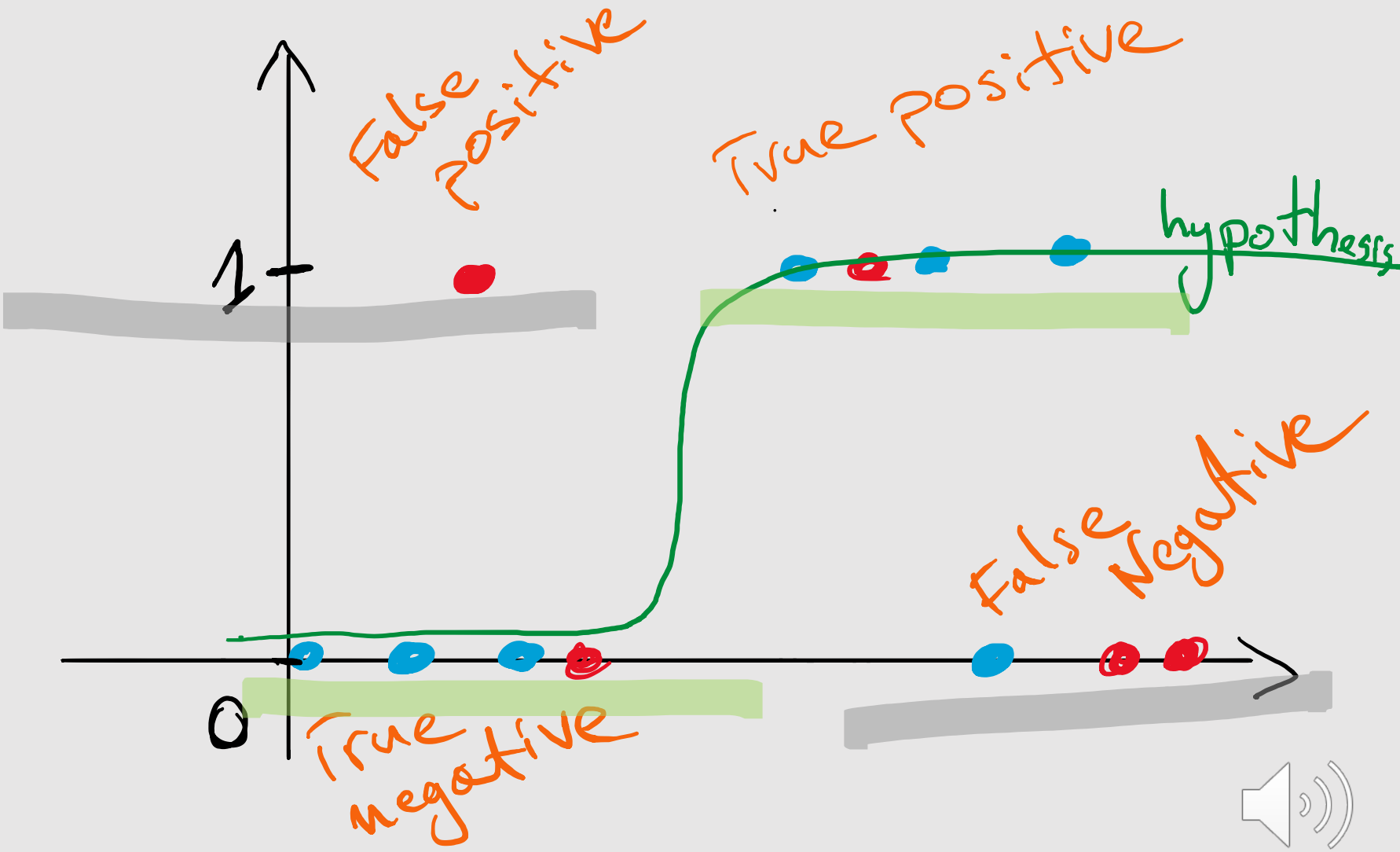
Training Error



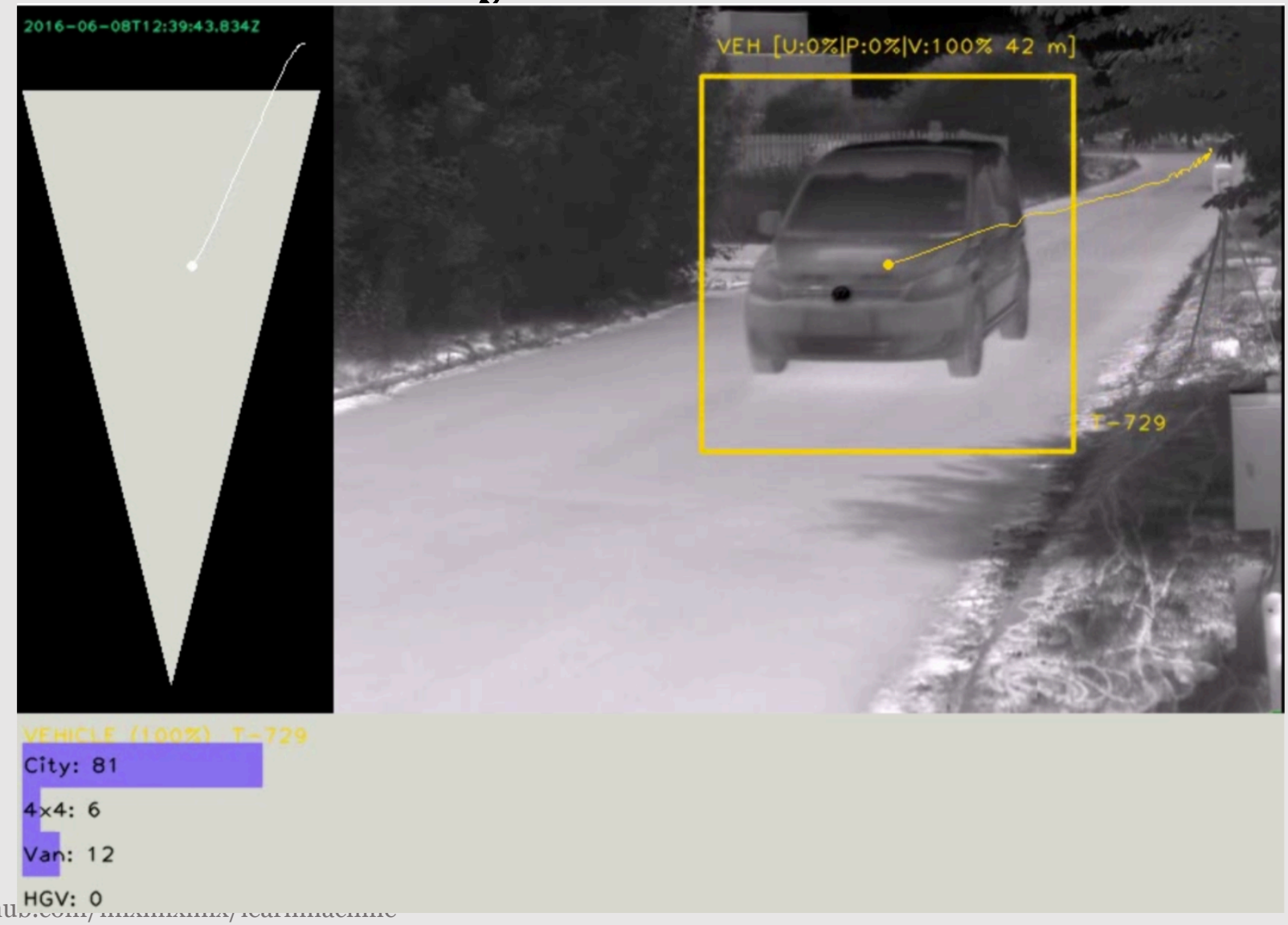
Training and Testing Errors

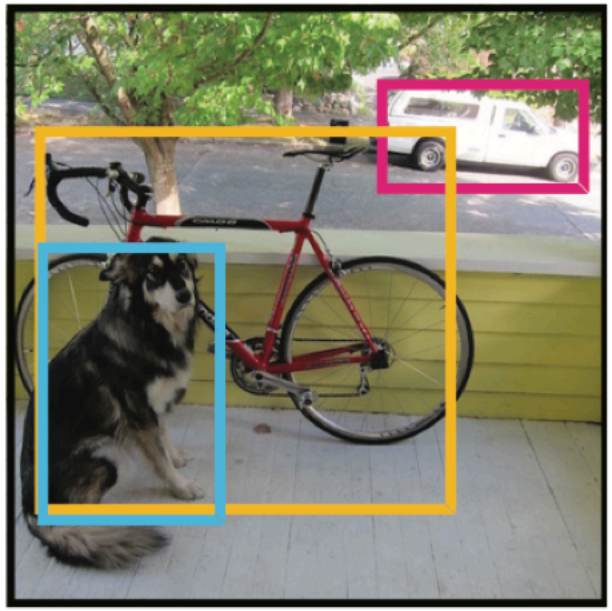


Training and Testing Errors



Probability of detection





Intersection of Union

$$IoU = \frac{A \cap B}{A \cup B}$$

https://www.pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/

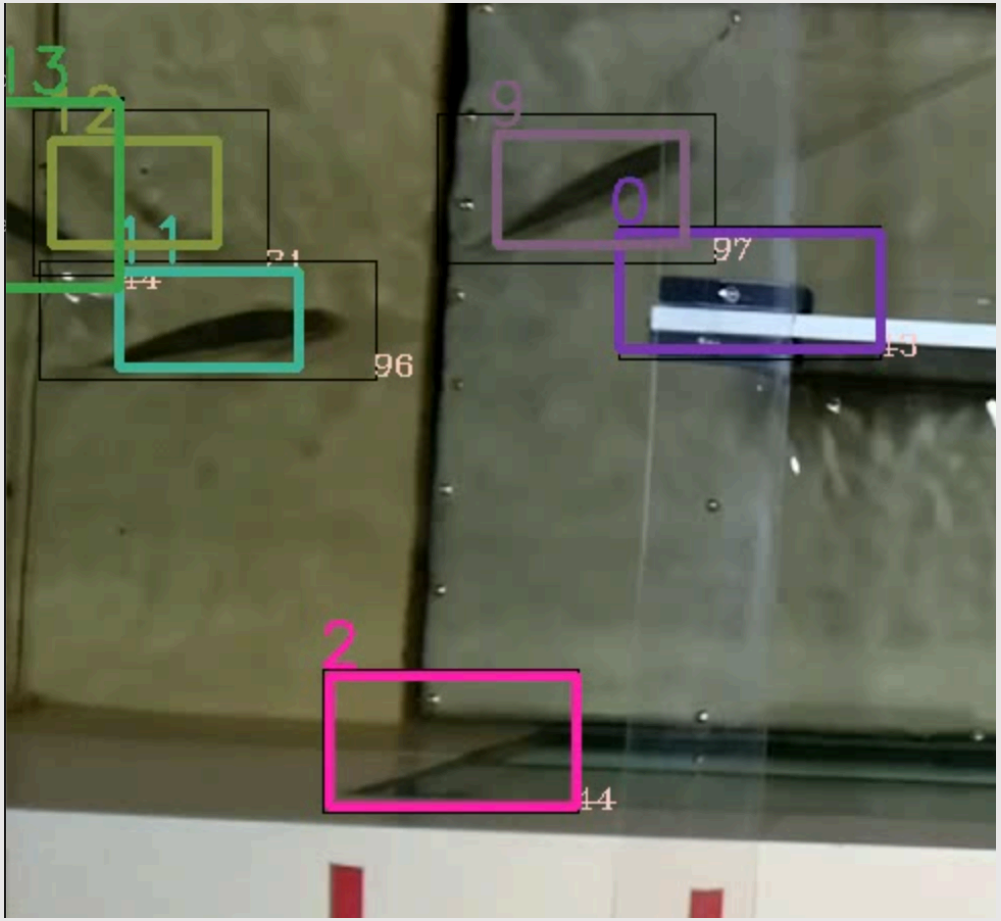
Thresholds

- prioritise:

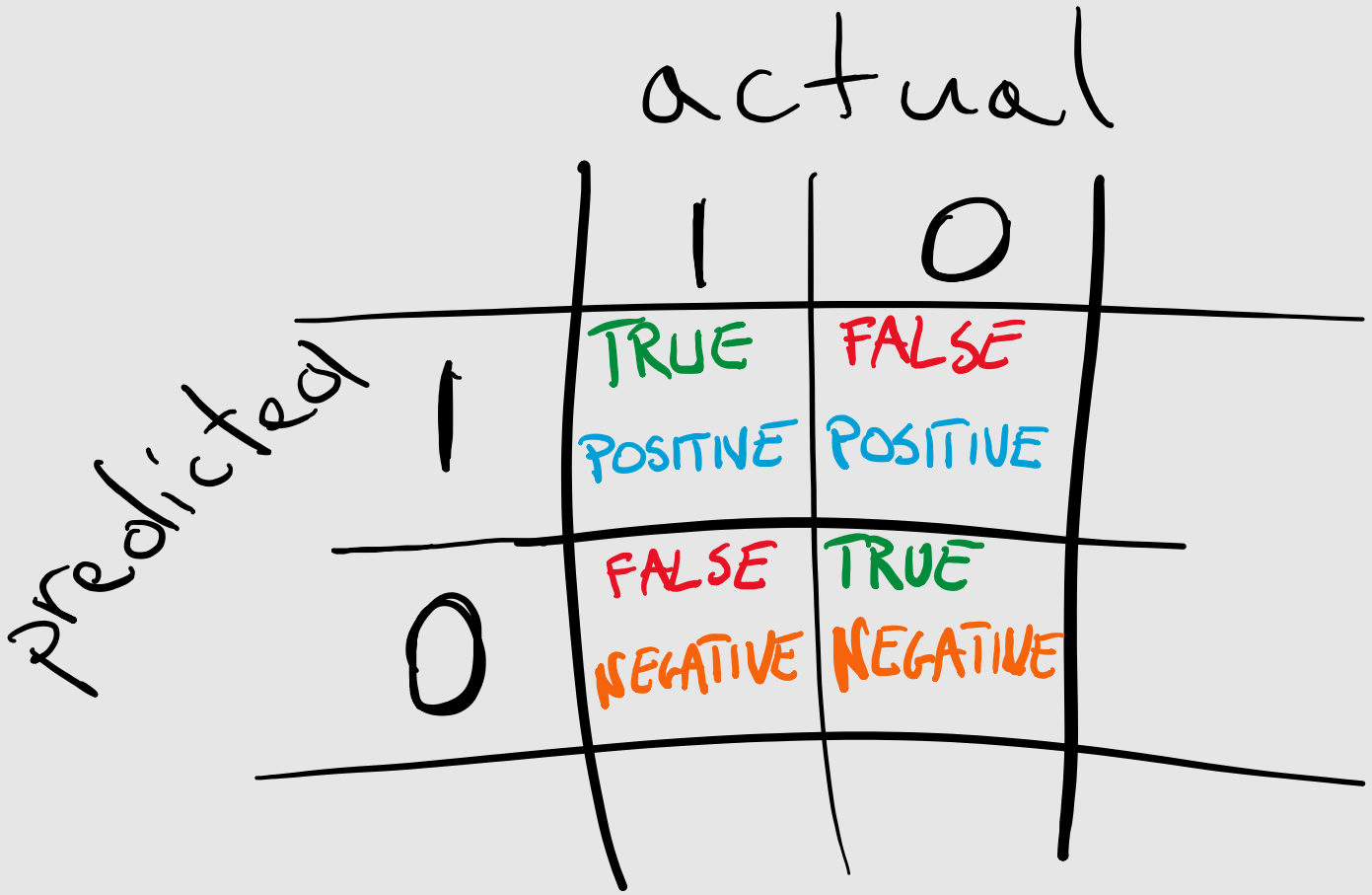
missed detections vs false detections

- Pedestrian detection? Better false detections, high recall

- Data presentation, best candidate selection? high precision

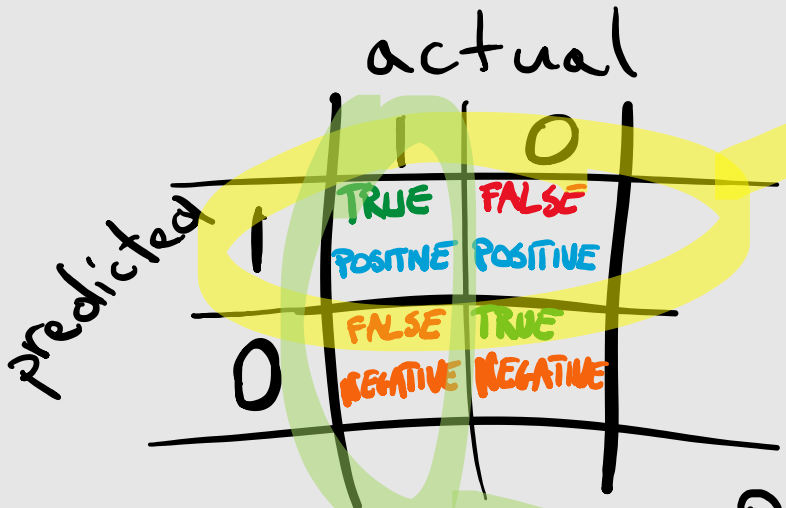


Performance measures



class ∈ {1, 0}

Performance measures



Precision

$$\frac{TP}{\text{Predicted Pos}} = \frac{TP}{TP + FP}$$

Recall

$$\frac{TP}{\text{Actual Pos}} = \frac{TP}{TP + FN}$$

class $\in \{1, 0\}$

- small precision: high false positives, false detections
- small recall: high false negatives, missed detections

F1-score

$\frac{P+R}{2}$ $2\frac{PR}{P+R}$

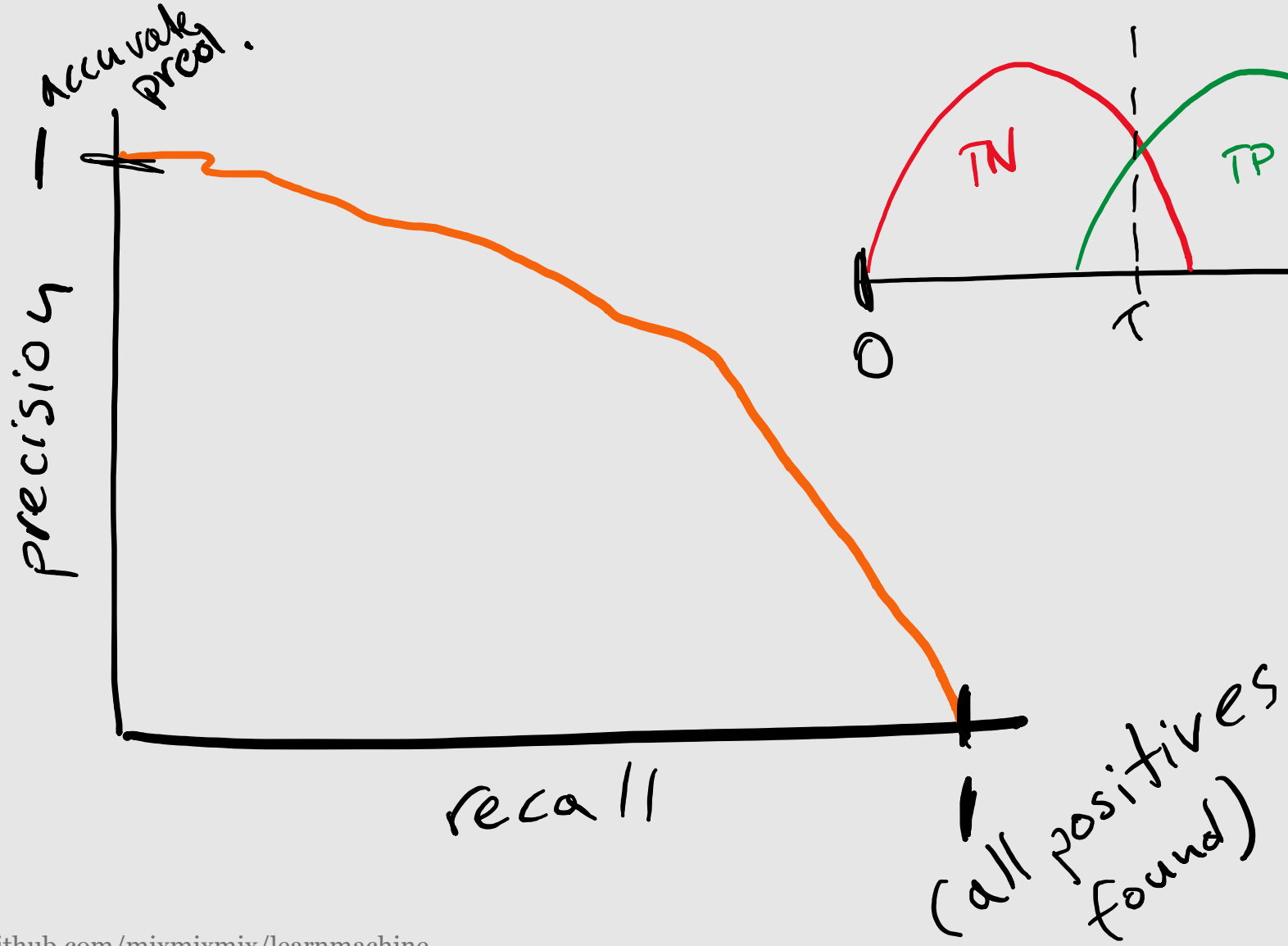
	Precision	Recall	Average	F1-score
Method A	0.5	0.4	0.45	0.44
Method B	0.7	0.1	0.4	0.18
Method C	0.02	1.0	0.51	0.04

Precision == Positive Pred. Power

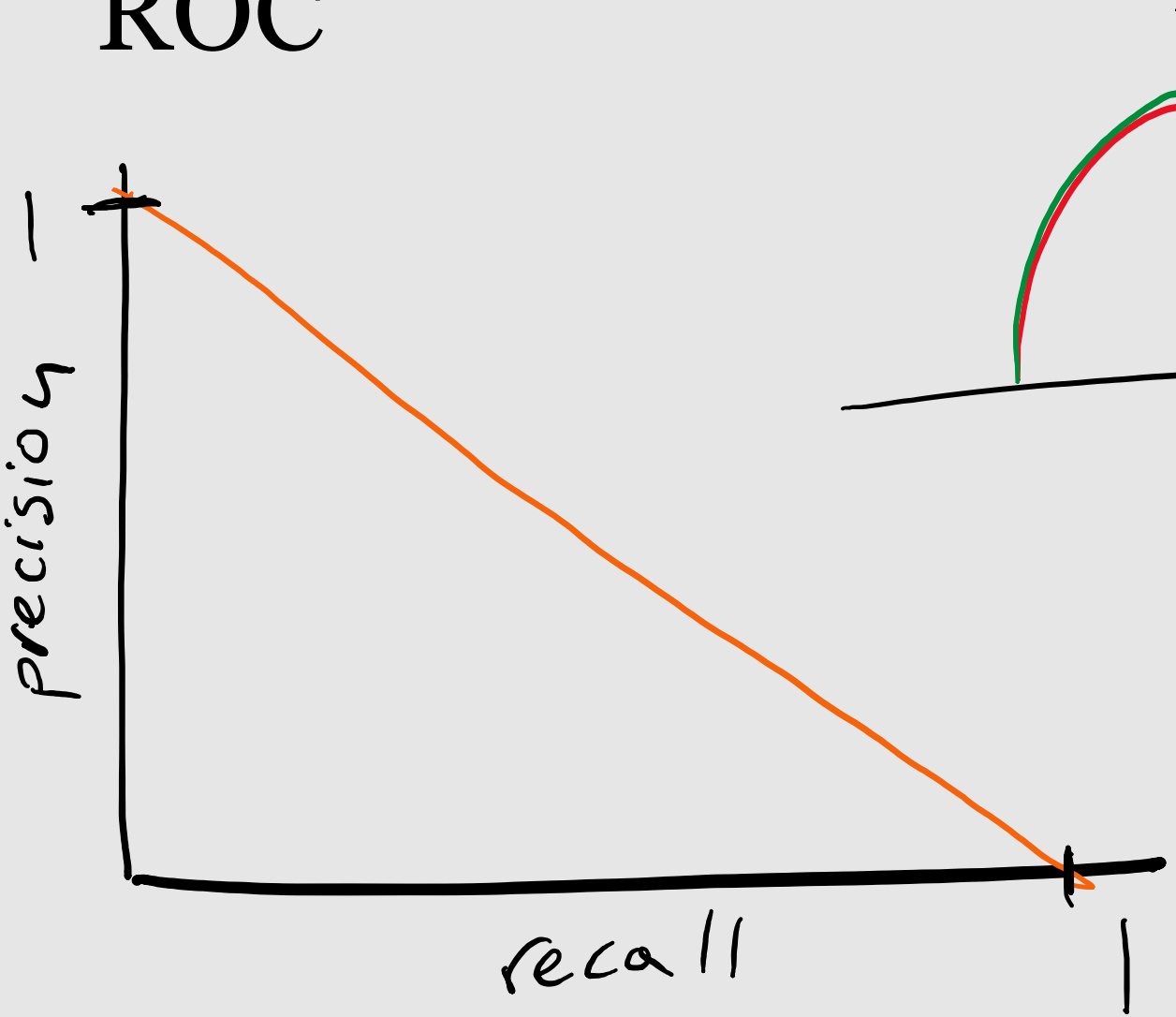
↑ harmonic mean

Recall == Sensitivity

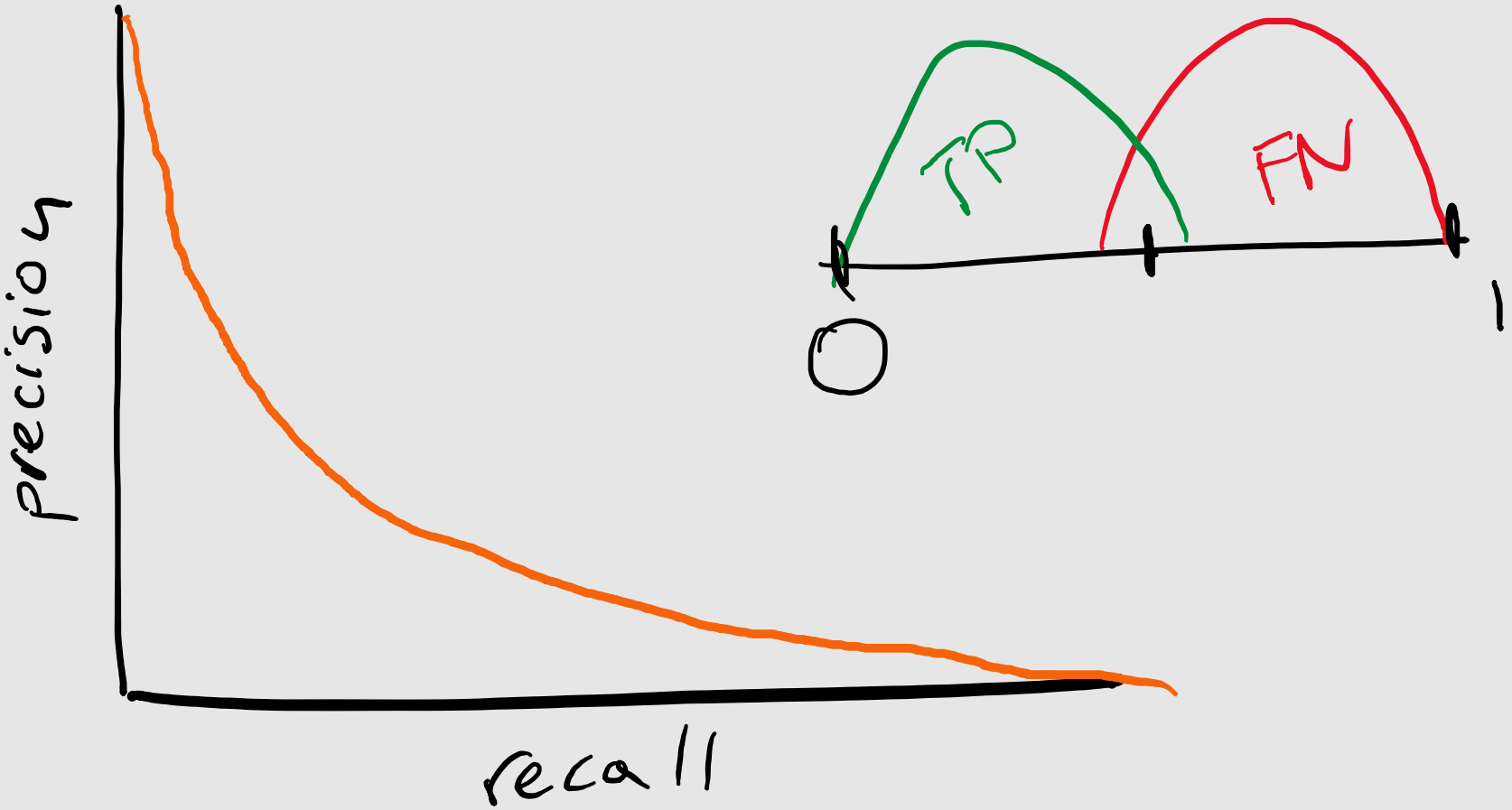
Recall-Precision Curve, Receiver Operating Characteristic (ROC)

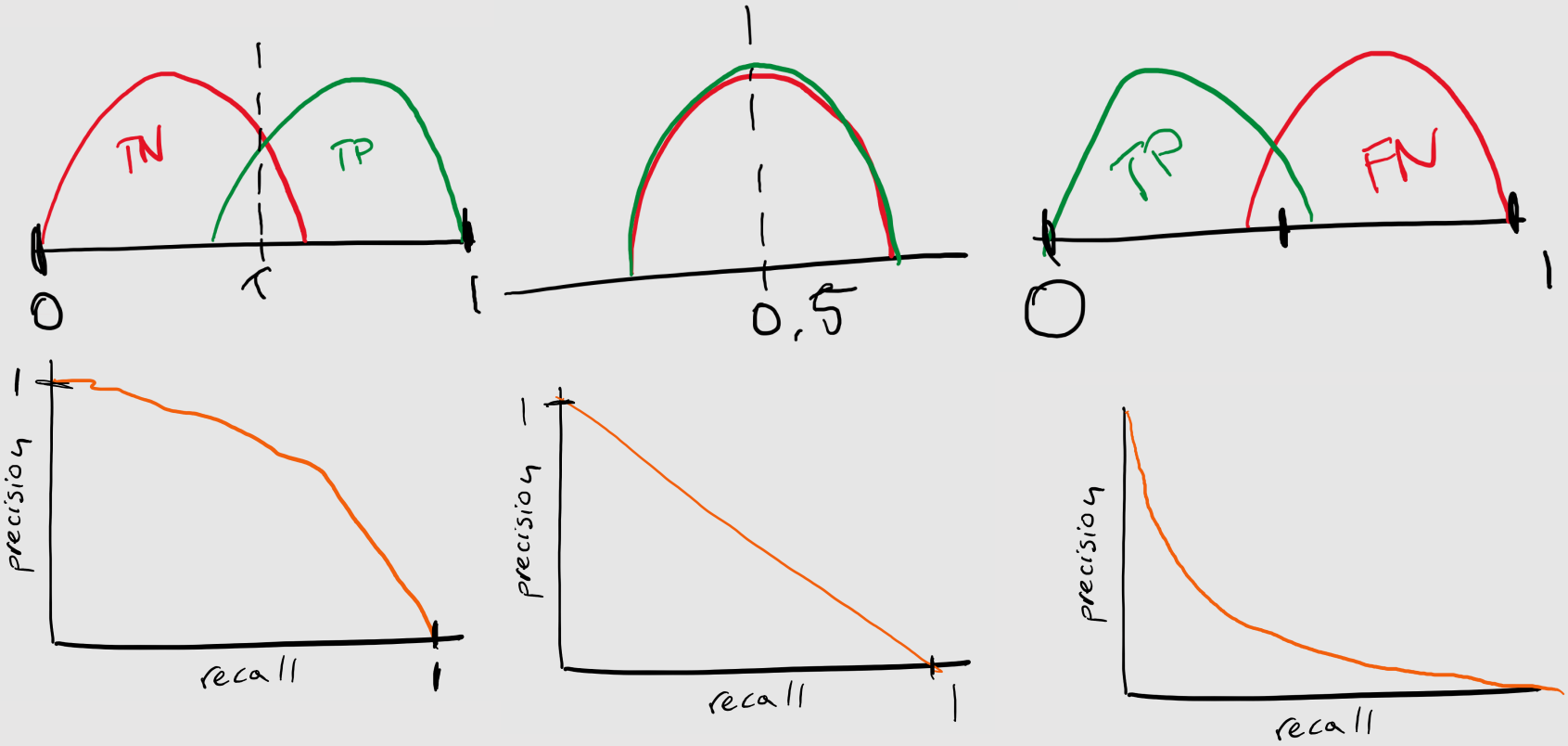


ROC



ROC



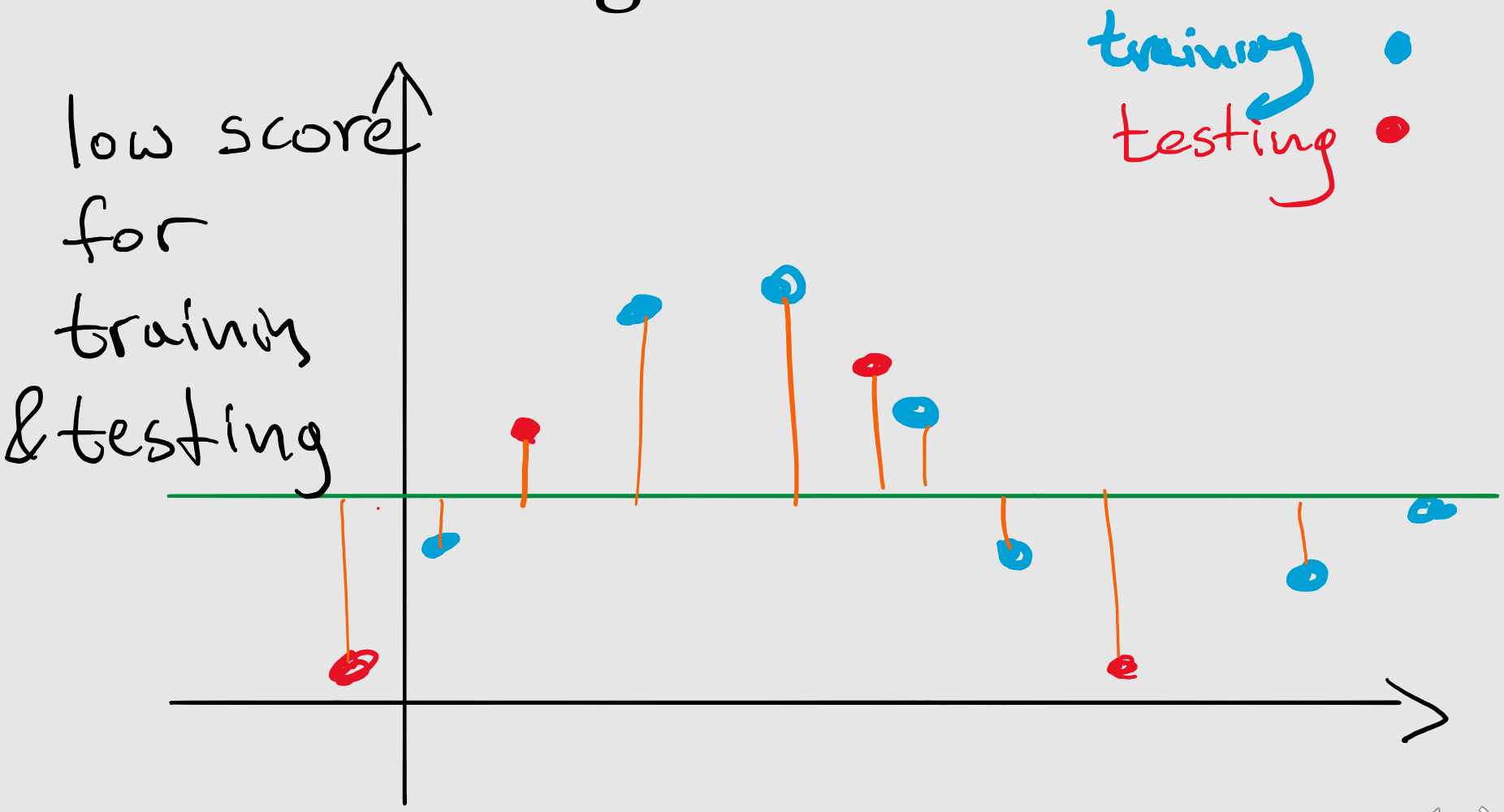

TN
TP
0
τ
1
0.5
TP
FN
0
1
precision
recall
1
precision
recall
1
precision
recall

mean Average Precision

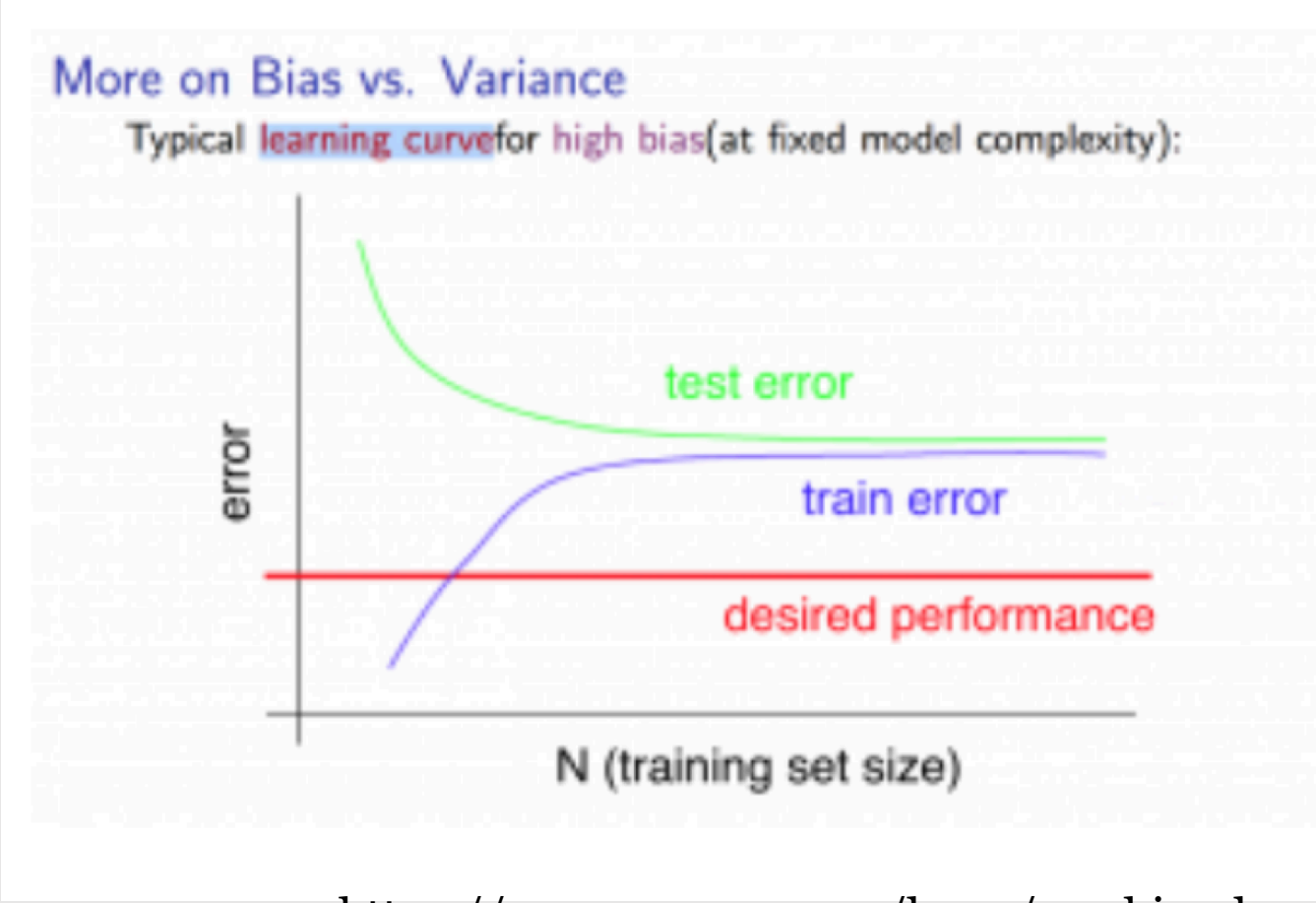
Fitting model

- overfitting
- underfitting
- perfect fit?

Underfitting



Underfitting – high bias

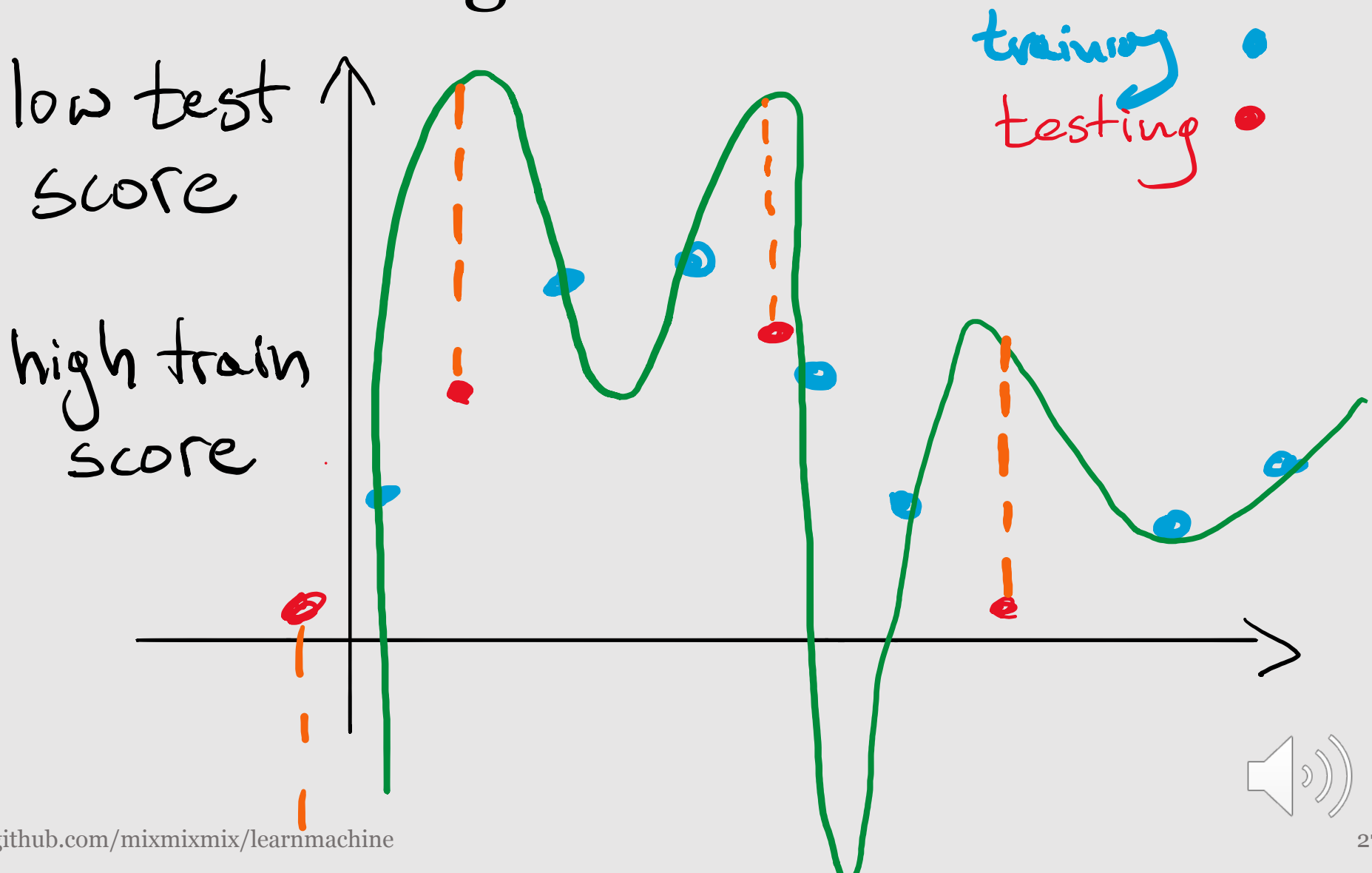


source: https://www.coursera.org/learn/machine-learning

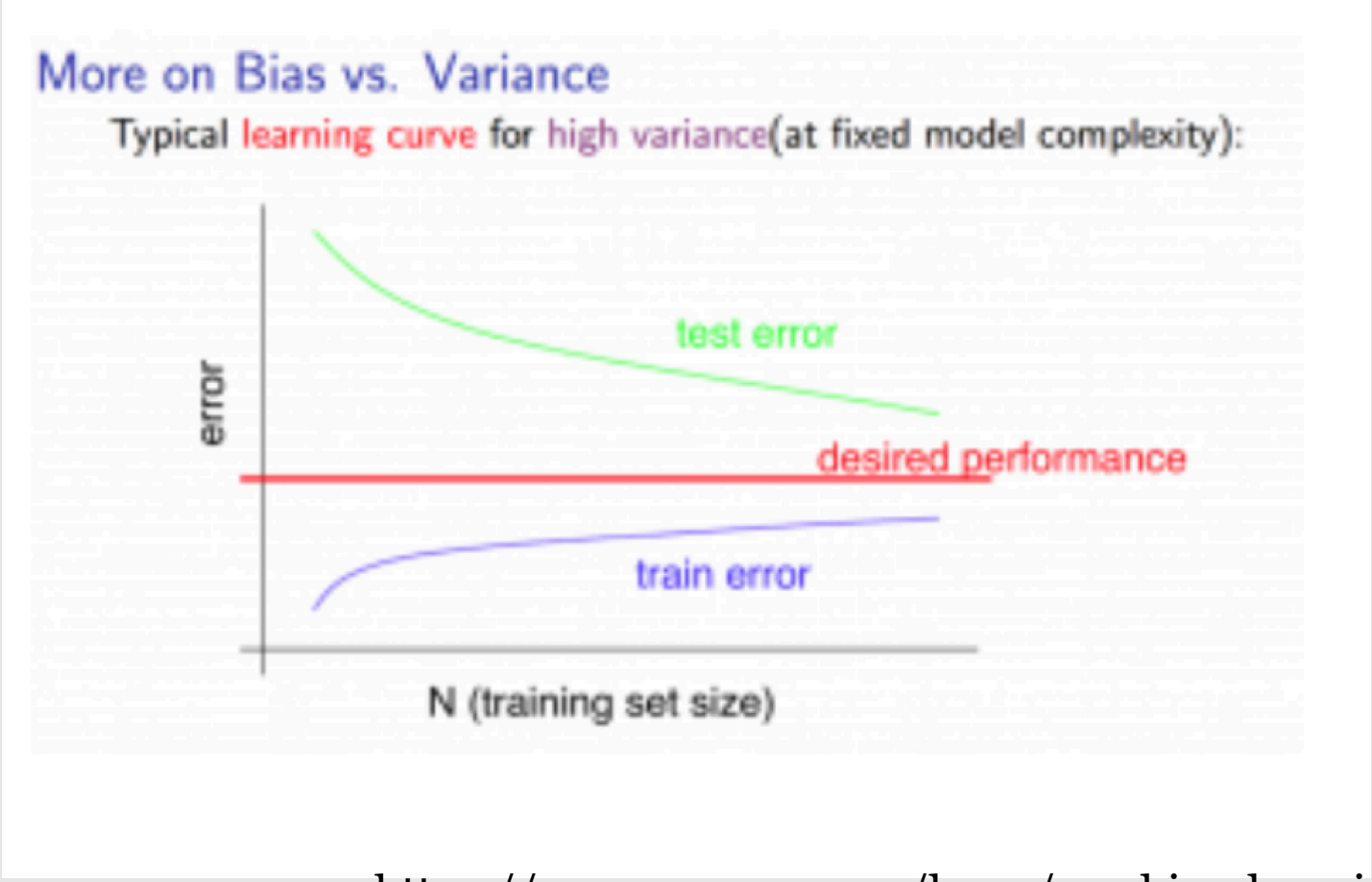
Underfitting

- more complex model
- more complex features
- more data

Overfitting

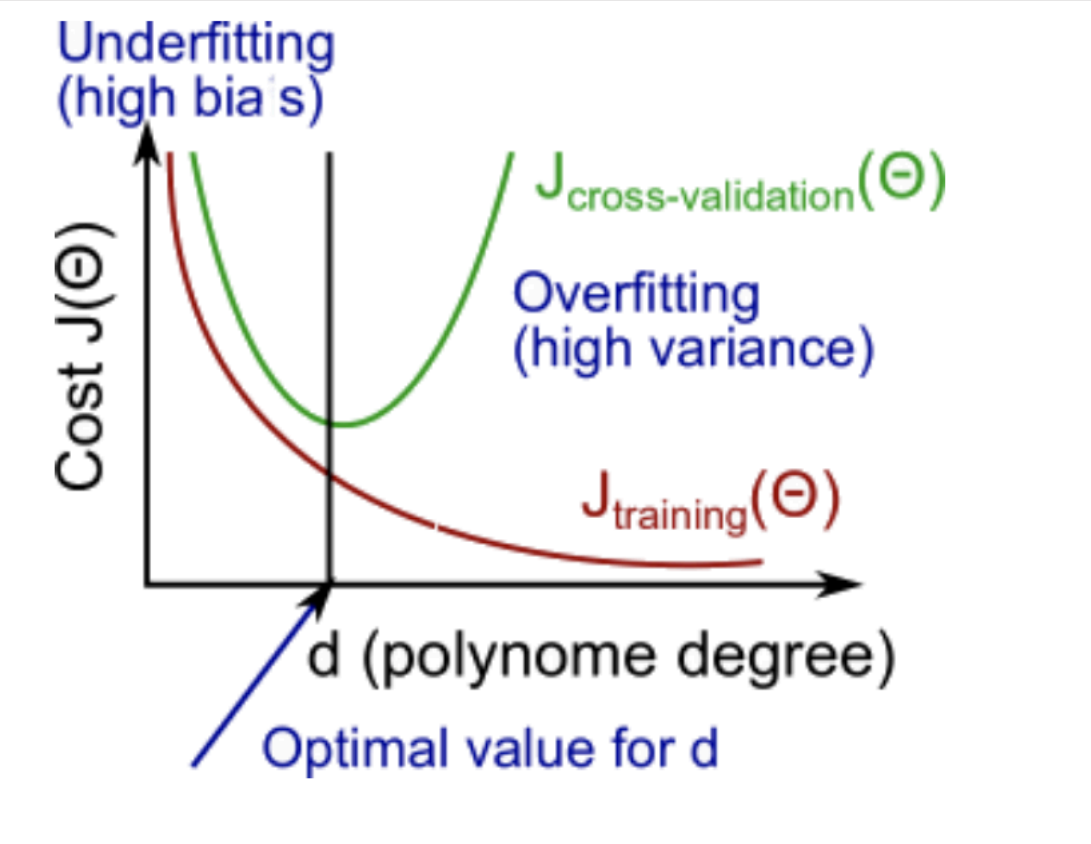


Overfitting – high variance



source: https://www.coursera.org/learn/machine-learning

Overfitting – high variance



source: https://www.coursera.org/learn/machine-learning

Overfitting

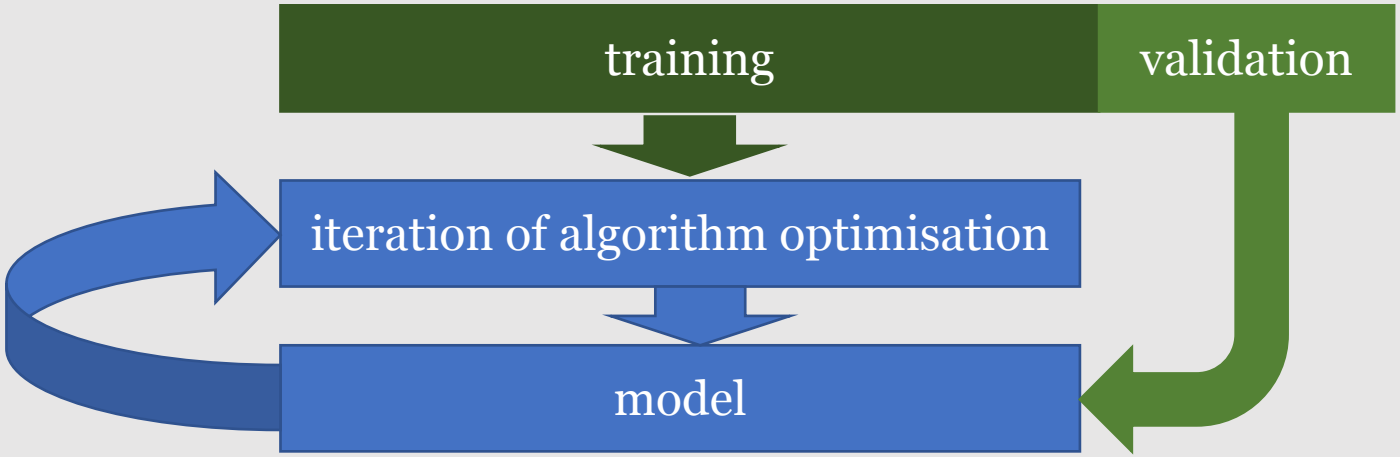
- simpler model
- regularisation
- cross-validation
- data augmentation

- Overfitting -> Underfitting? More data!

Regularisation

$$J(\theta) = cost(h_\theta(x), y) + \lambda \sum_j \theta_j^2$$

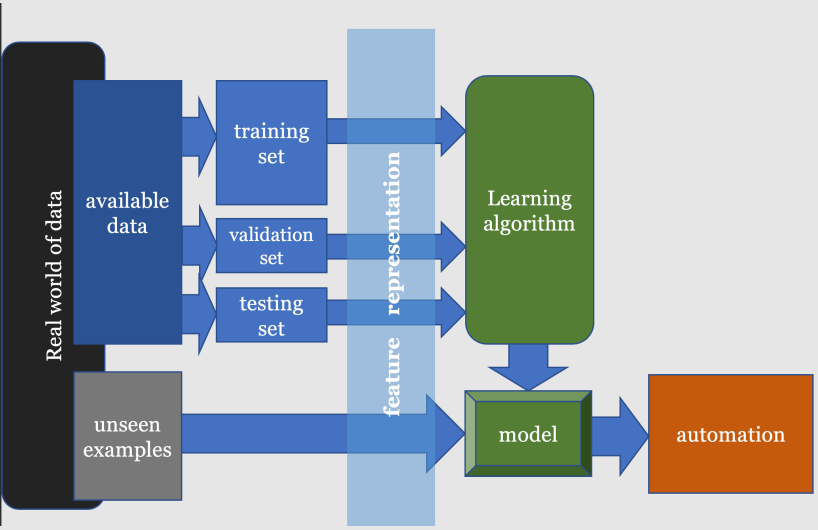
Cross-validation

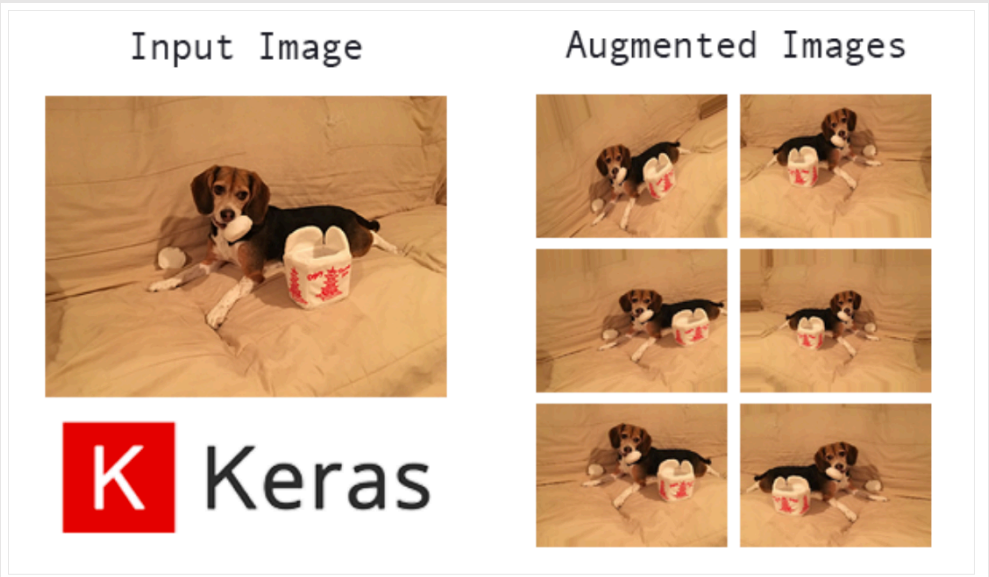


k-fold CV: every iteration take different (k^{th}) part of the dataset

Data augmentation

- extending training set by artificially simulating more robust data





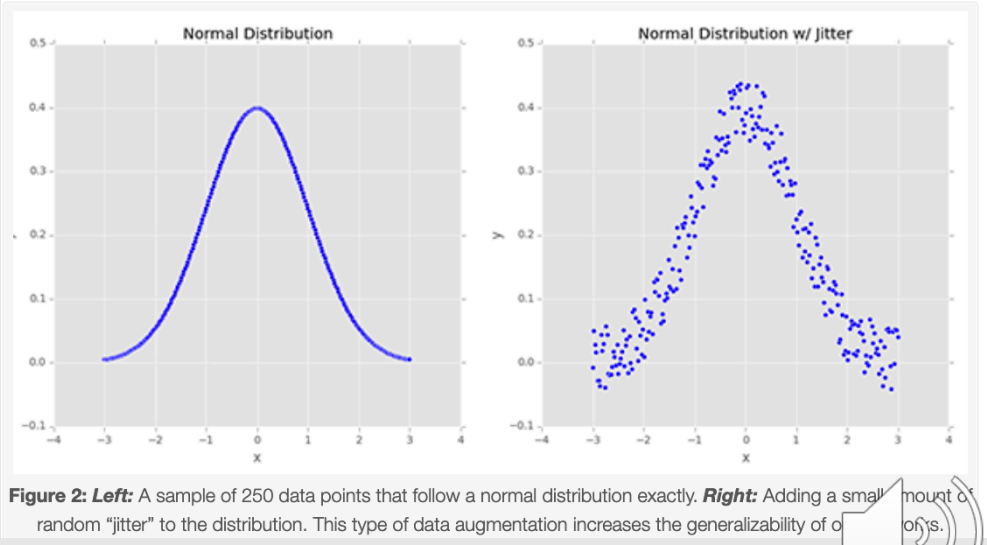
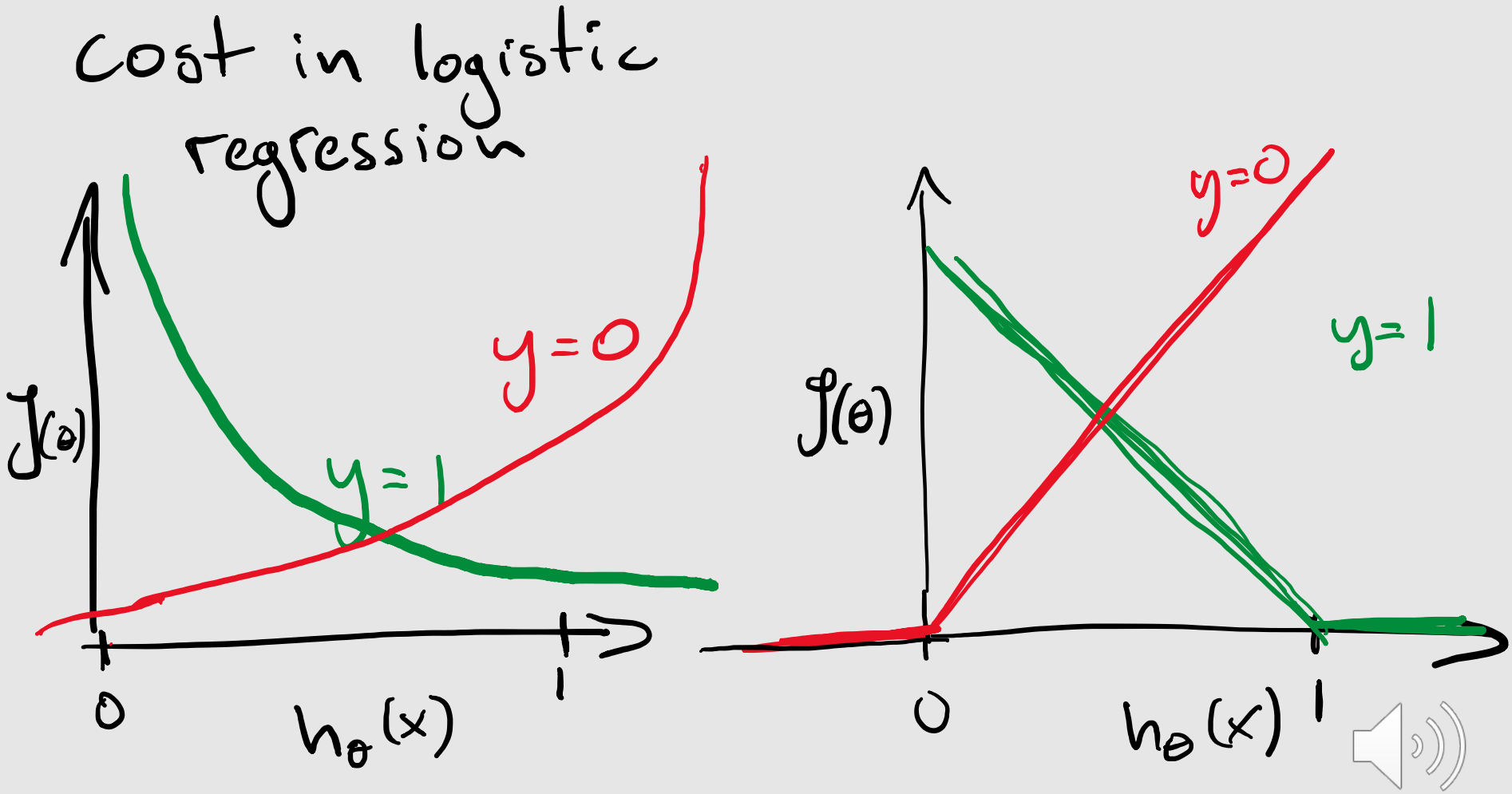


Figure 2: ***Left:*** A sample of 250 data points that follow a normal distribution exactly. ***Right:*** Adding a small amount of random "jitter" to the distribution. This type of data augmentation increases the generalizability of our models.

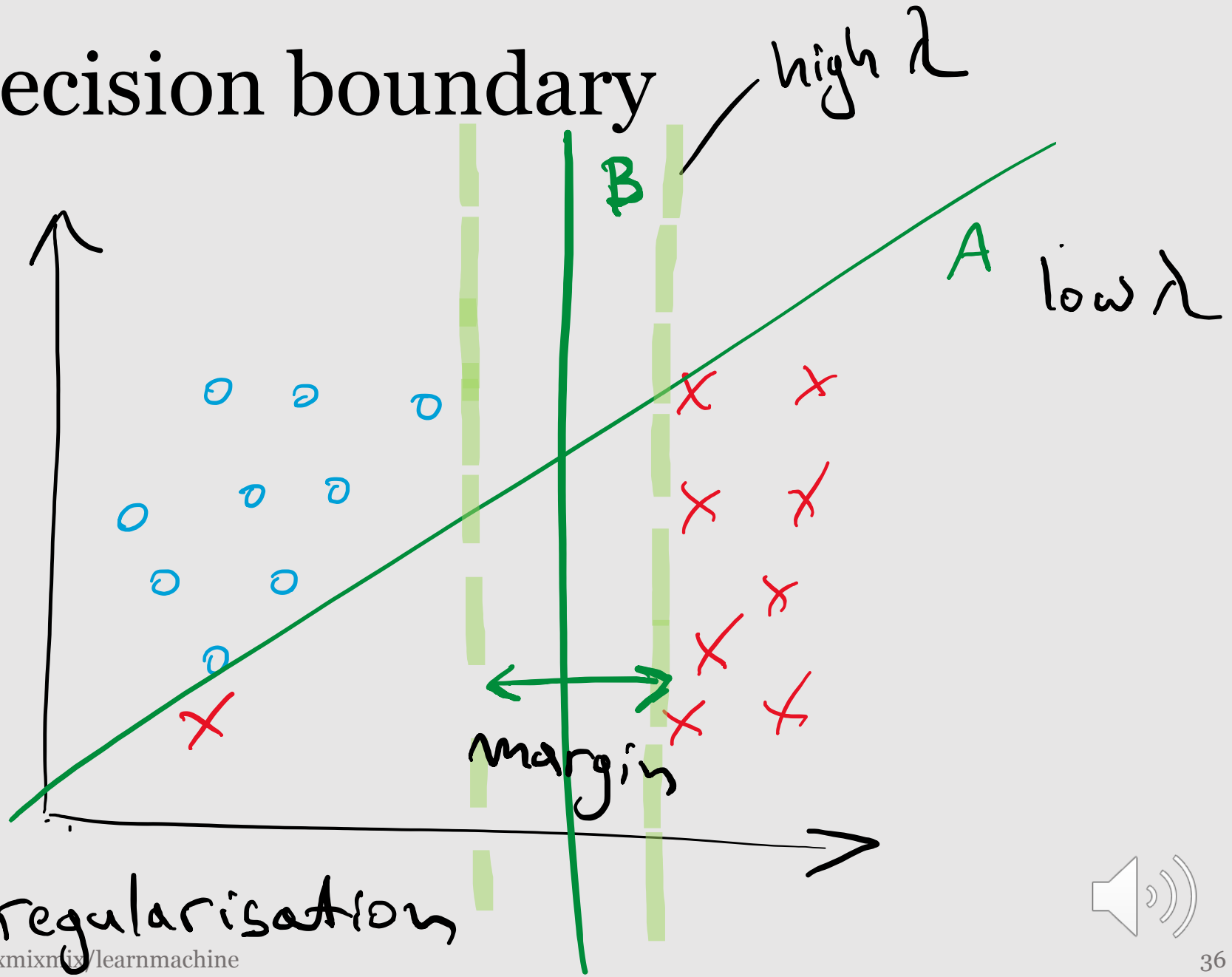
https://www.pyimagesearch.com/2019/07/08/keras-imagedatagenerator-and-data-augmentation/

Support Vector Machine or Large Margin Classifier

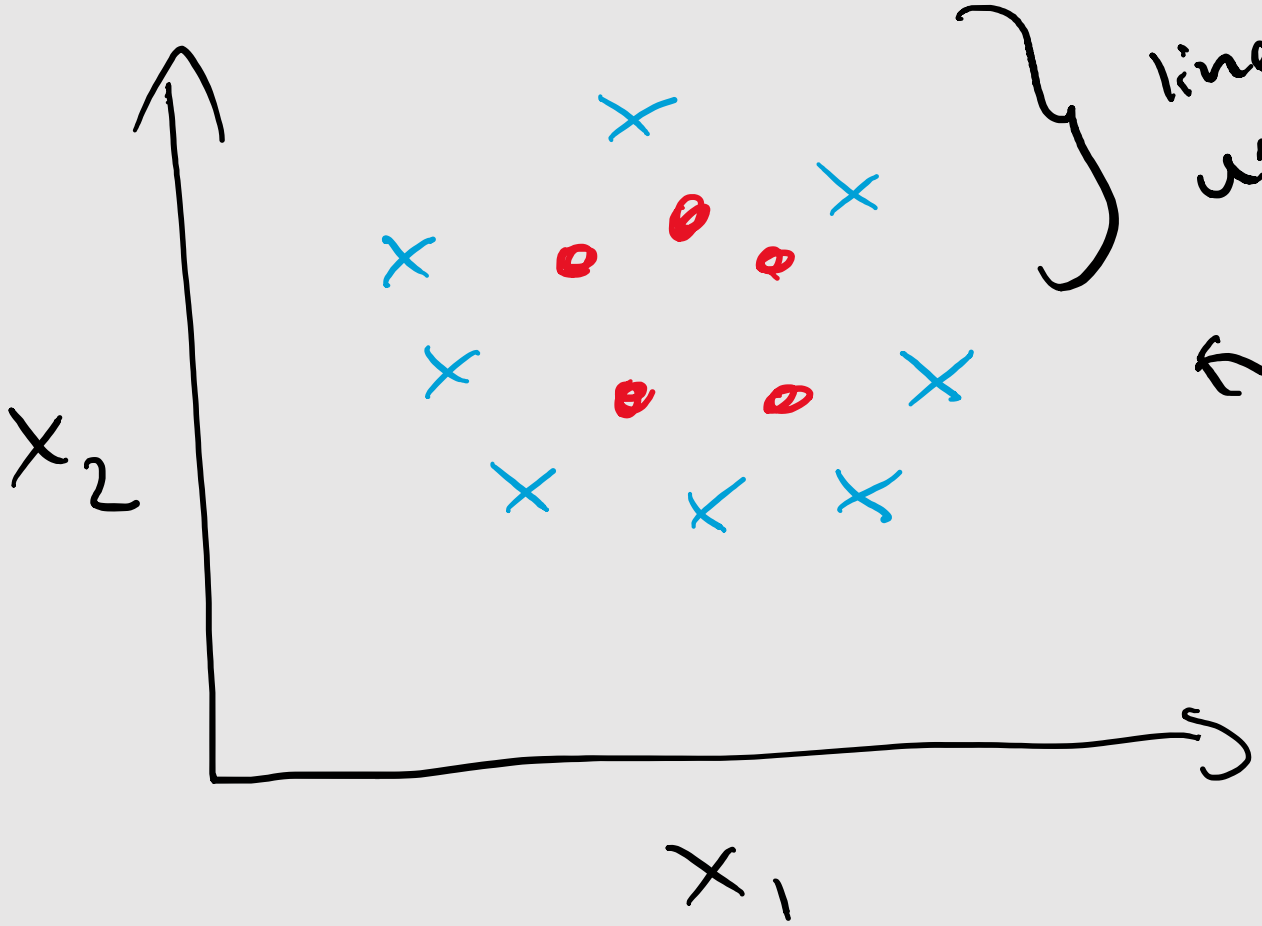
From logistic regression to kernel methods



Decision boundary

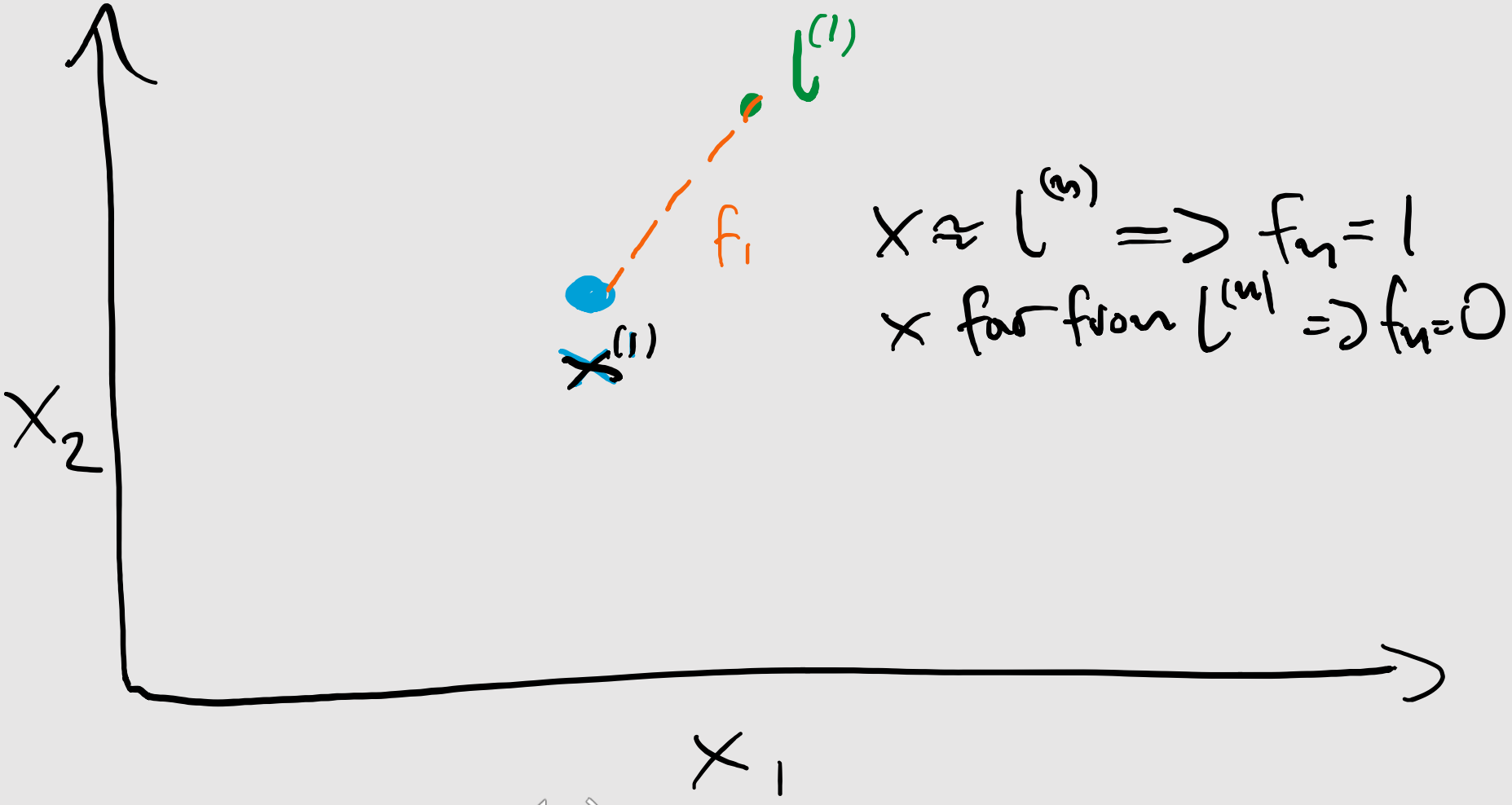


Kernel "trick"



Kernel "trick"

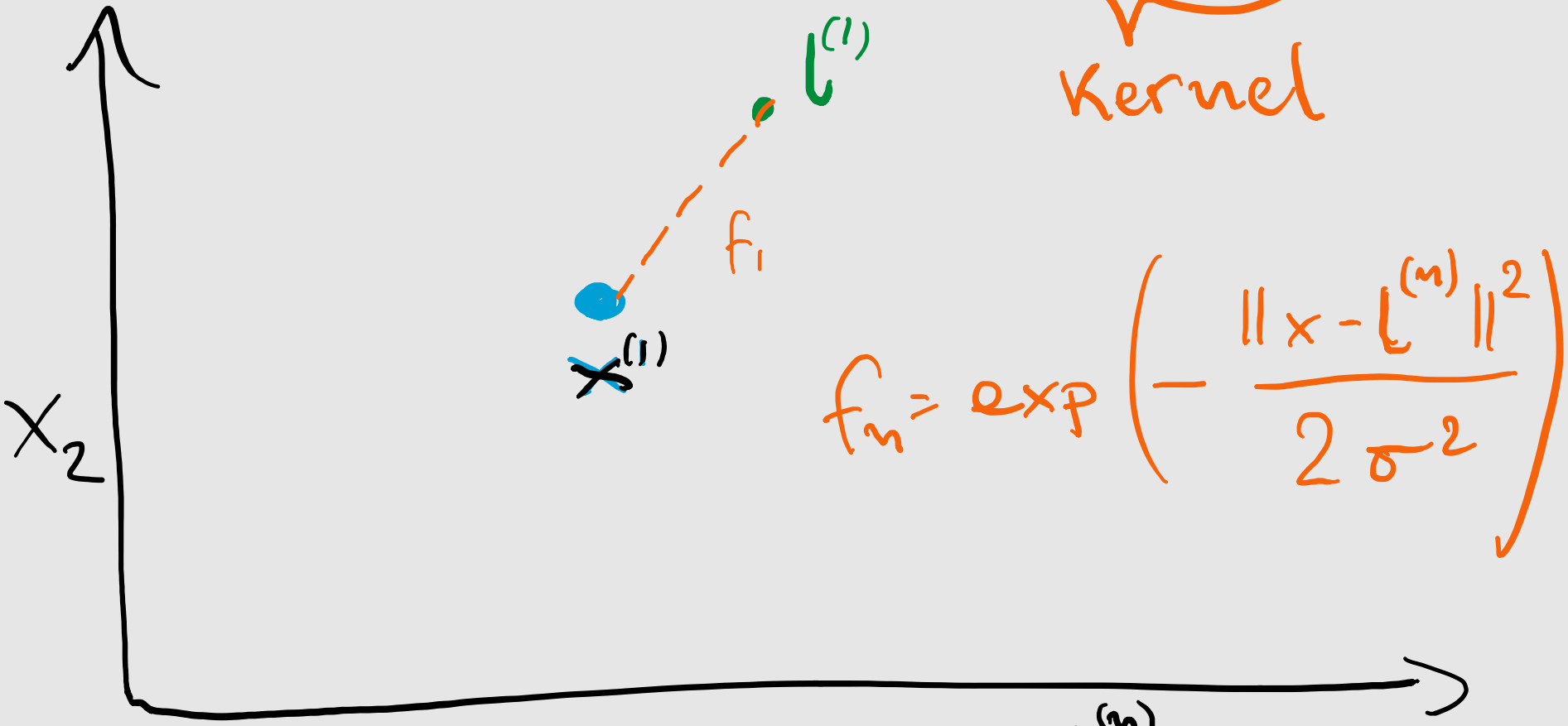
$f_n = \text{similarity}(x, l^{(n)})$



Kernel "trick"

$f_n = \text{similarity}(x, l^{(n)})$

Kernel



$$f_n = \exp\left(-\frac{\|x - l^{(n)}\|^2}{2\sigma^2}\right)$$

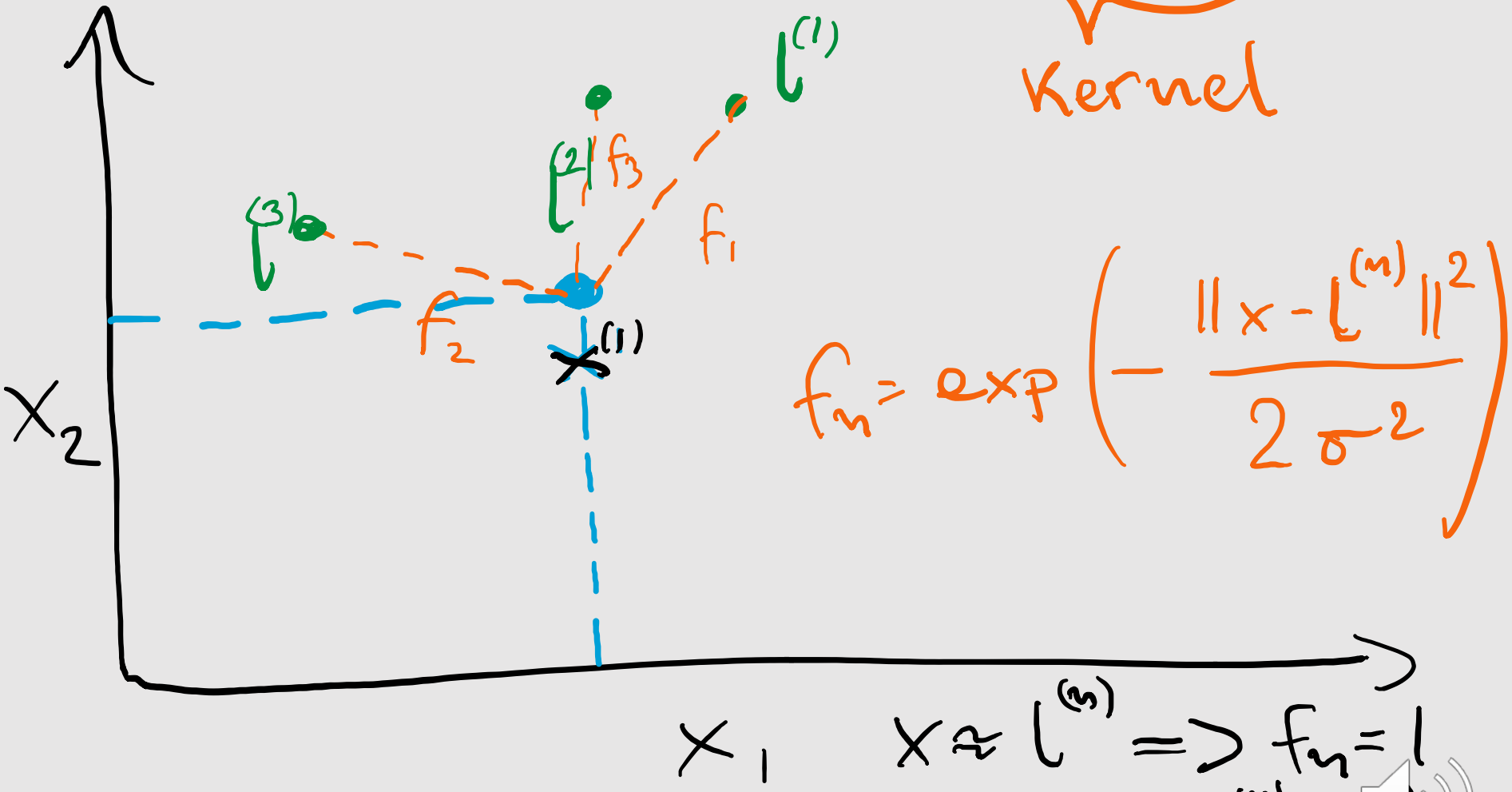
$x \approx l^{(n)} \Rightarrow f_n = 1$

x far from $l^{(n)} \Rightarrow f_n = 0$

Kernel "trick"

$f_n = \text{similarity}(x, l^{(n)})$

Kernel



$$f_n = \exp\left(-\frac{\|x - l^{(n)}\|^2}{2\sigma^2}\right)$$

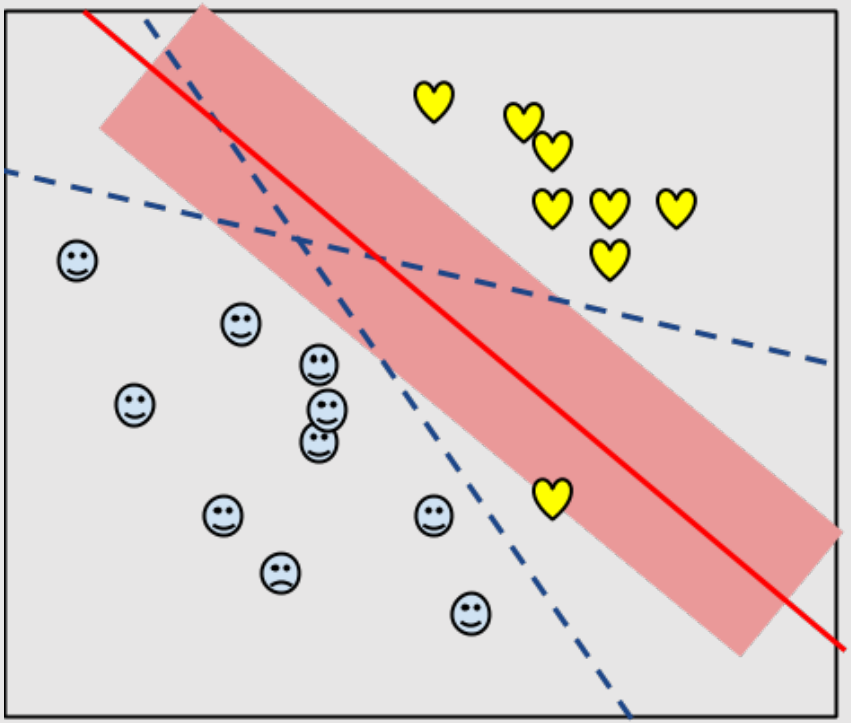
$x \approx l^{(n)} \Rightarrow f_n = 1$

x far from $l^{(n)} \Rightarrow f_n = 0$

Similarity

- If $x \approx l^{(n)} \Rightarrow f_n = 1$, x far from $l^{(n)} \Rightarrow f_n \approx 0$
- What if we have a lot of landmarks?
- What if all training examples are landmarks?
- Different kernels (Gaussian, Linear, Polynomial) depending on relationship in data

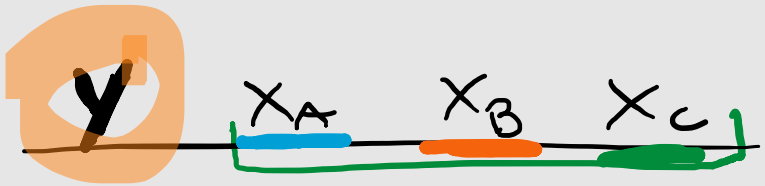
Support Vector (...machine) == Large Margin Classifier

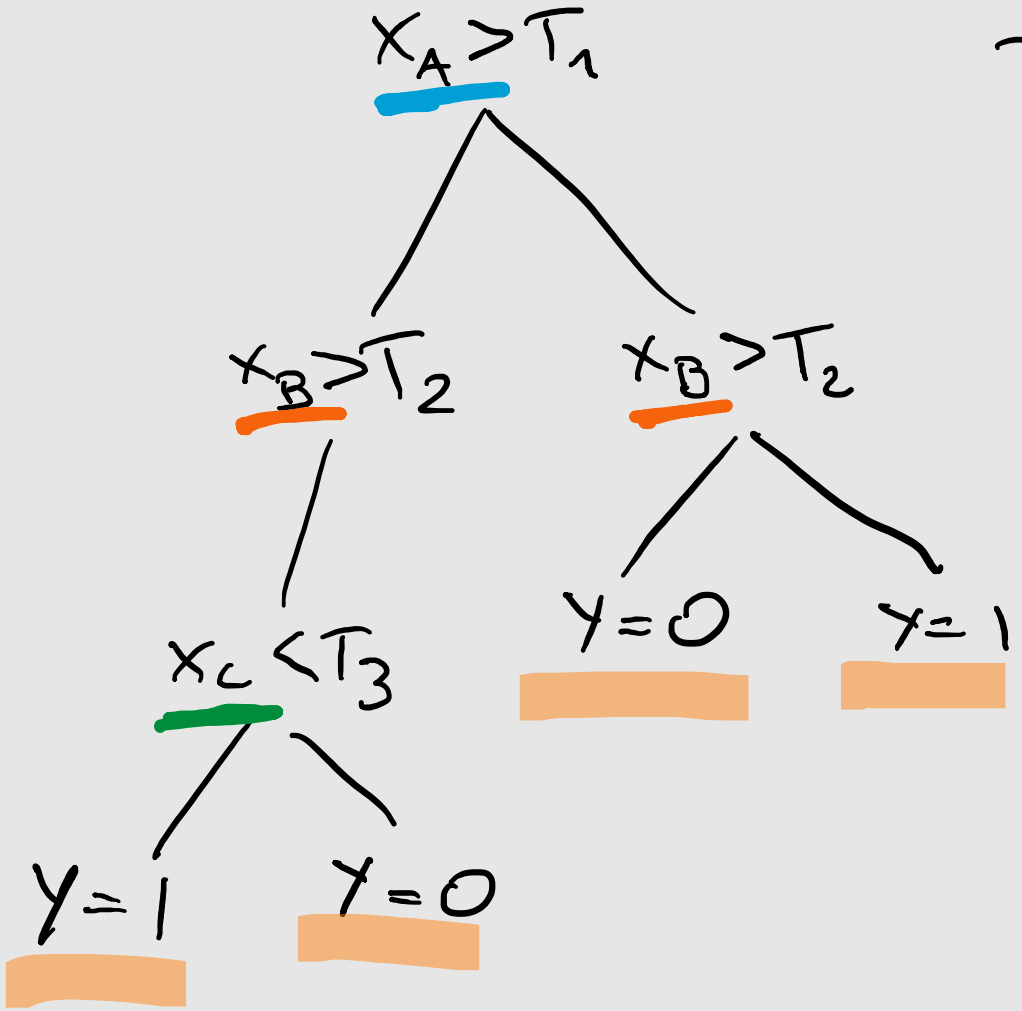


Basically, linear regression with slightly modified cost function and transformed features...

Let's make features out of data!

Decision tree

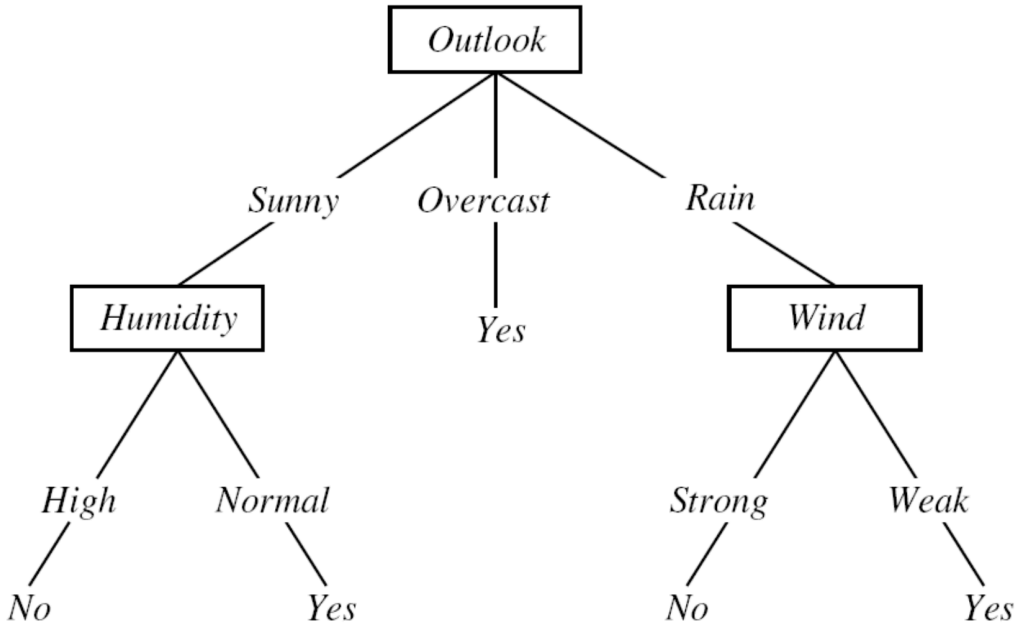




- optimise to minimize entropy with every division

Decision tree

Safe conditions to fly ?



Attributes **Classification**

Outlook	Temperature	Humidity	Windy	Fly
Sunny	85	85	False	No
Sunny	80	90	True	No
Overcast	83	86	False	Yes
Rainy	75	80	False	Yes
…	…	…	…	…

source: http://breckon.eu/toby/teaching/mltutorial/

Random Forest

- One decision tree is a basic classifier
- Train multiple small trees (each on subset of features)
- Let them vote on the final outcome
- Principles of “many wrong”
- Small trees == regularisation, no overfitting
- Routinely used to classify vast datasets from particle physics and astronomy