



Lecture 2

Classification: Feature Spaces and Linear Discriminants

Gabriele Pergola and Amina Asif
Machine Learning Algorithms and Practice
CS429

October 2026

Outline

- Cats
- Feature Space and the K-Nearest Neighbors (k-NN)
- Classification with k-NN
- Limitations of k-NN
- Linear Classification
- RAO: A General Framework for Machine Learning

How many PCs to recognise a cat?

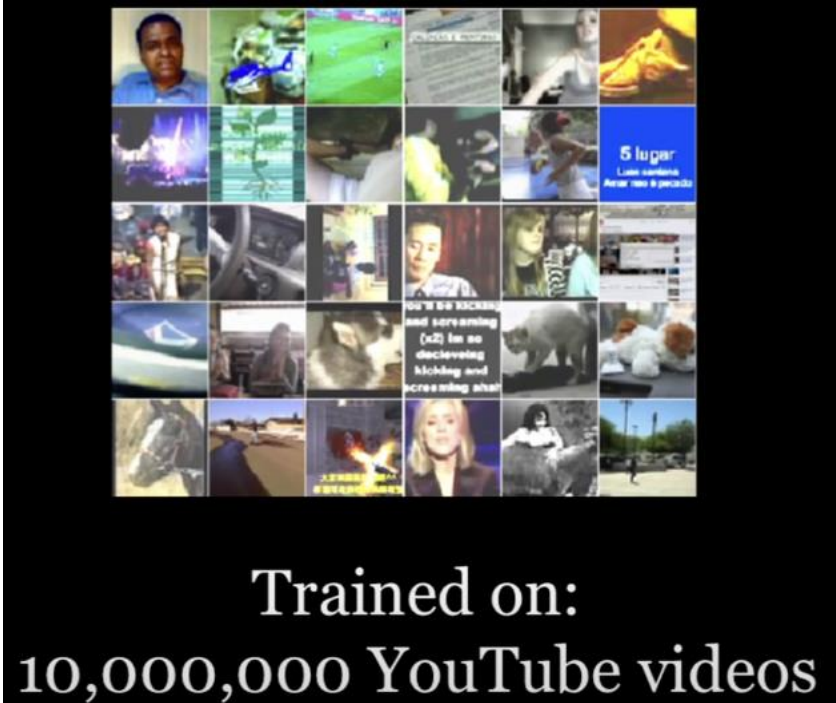
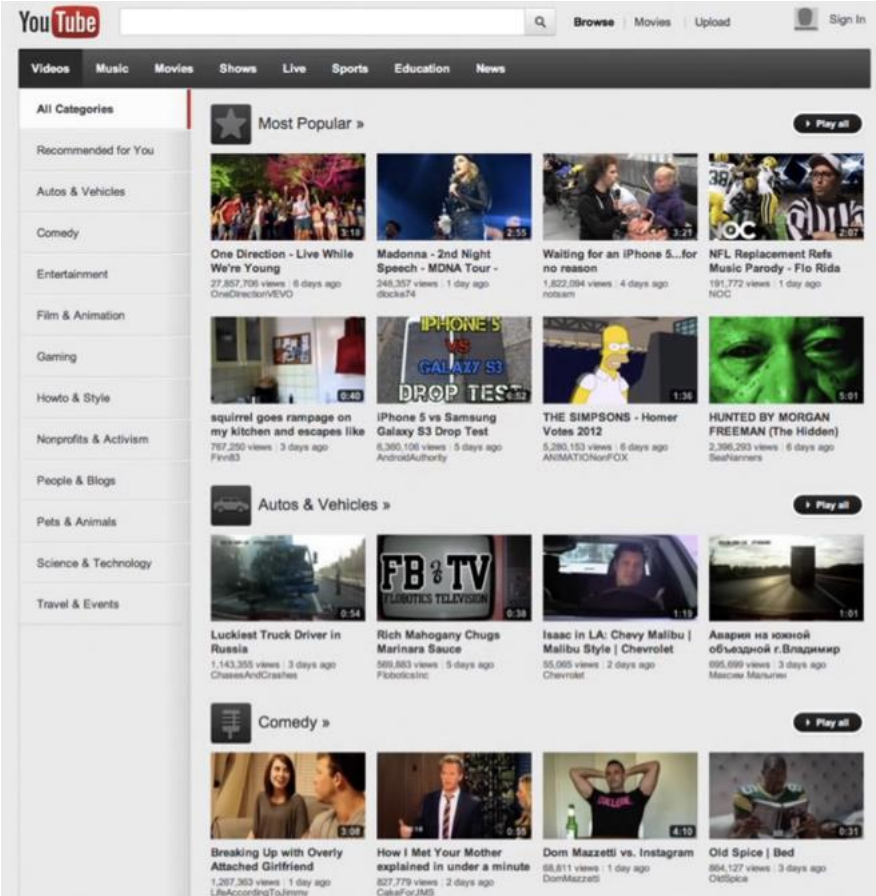
The New York Times

How Many Computers to Identify a Cat? 16,000

 Give this article  



An image of a cat that a neural network taught itself to recognize. Jim Wilson/The New York Times



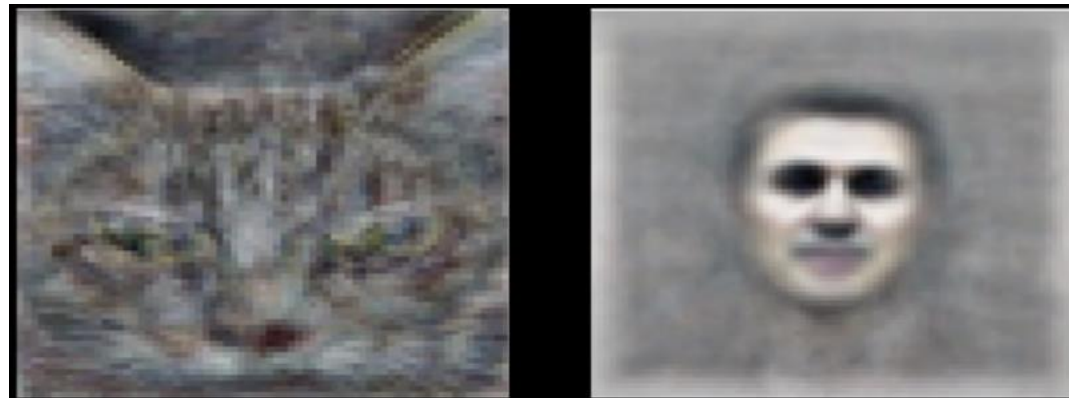
How many PCs to recognise a cat?

Top level features: images that most excite the nodes in the network



... first 'impressions' of a neural network!

- Images that induces the highest values on the neural units

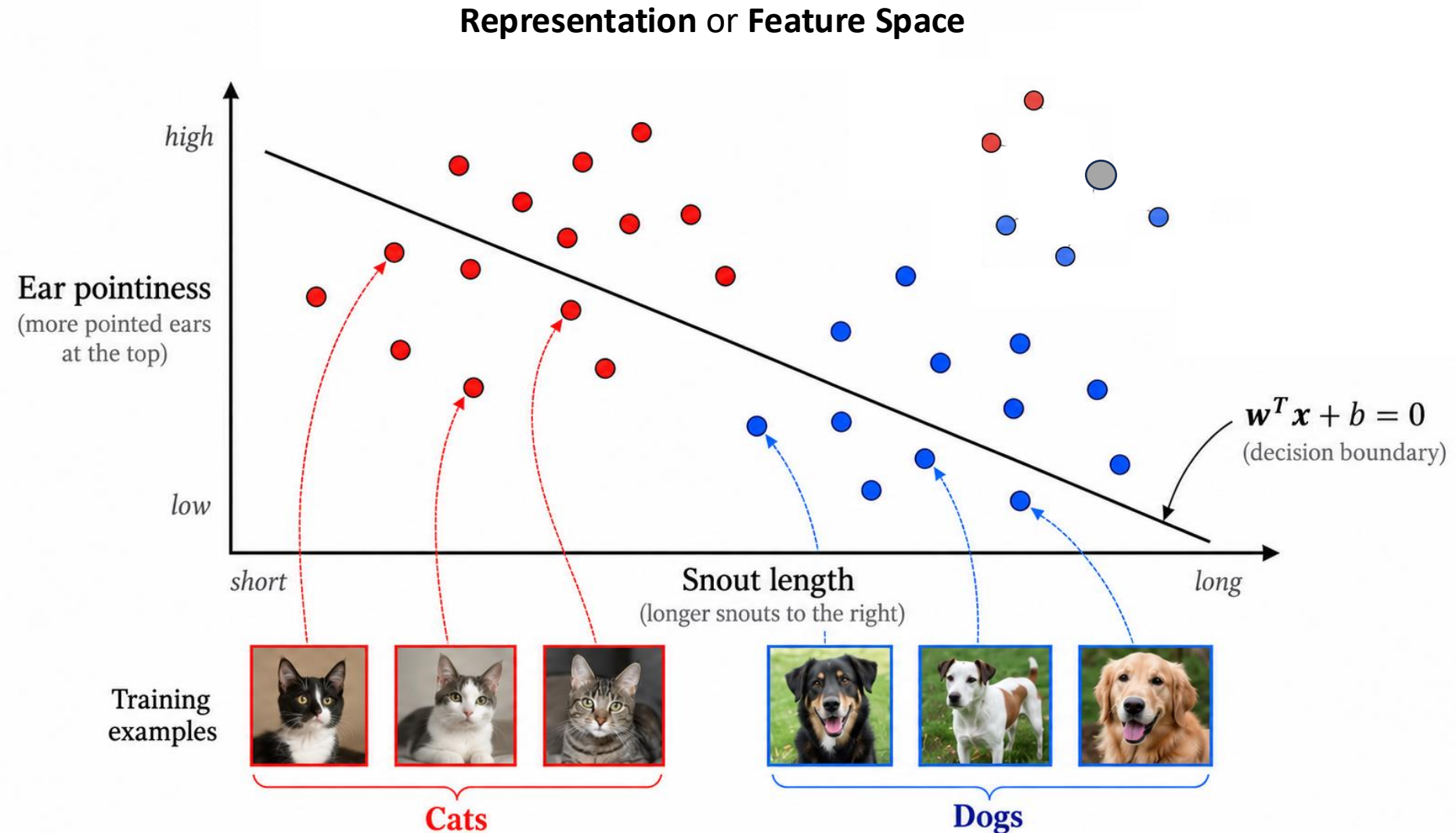


Machine Learning: Classifying Cats and Dogs

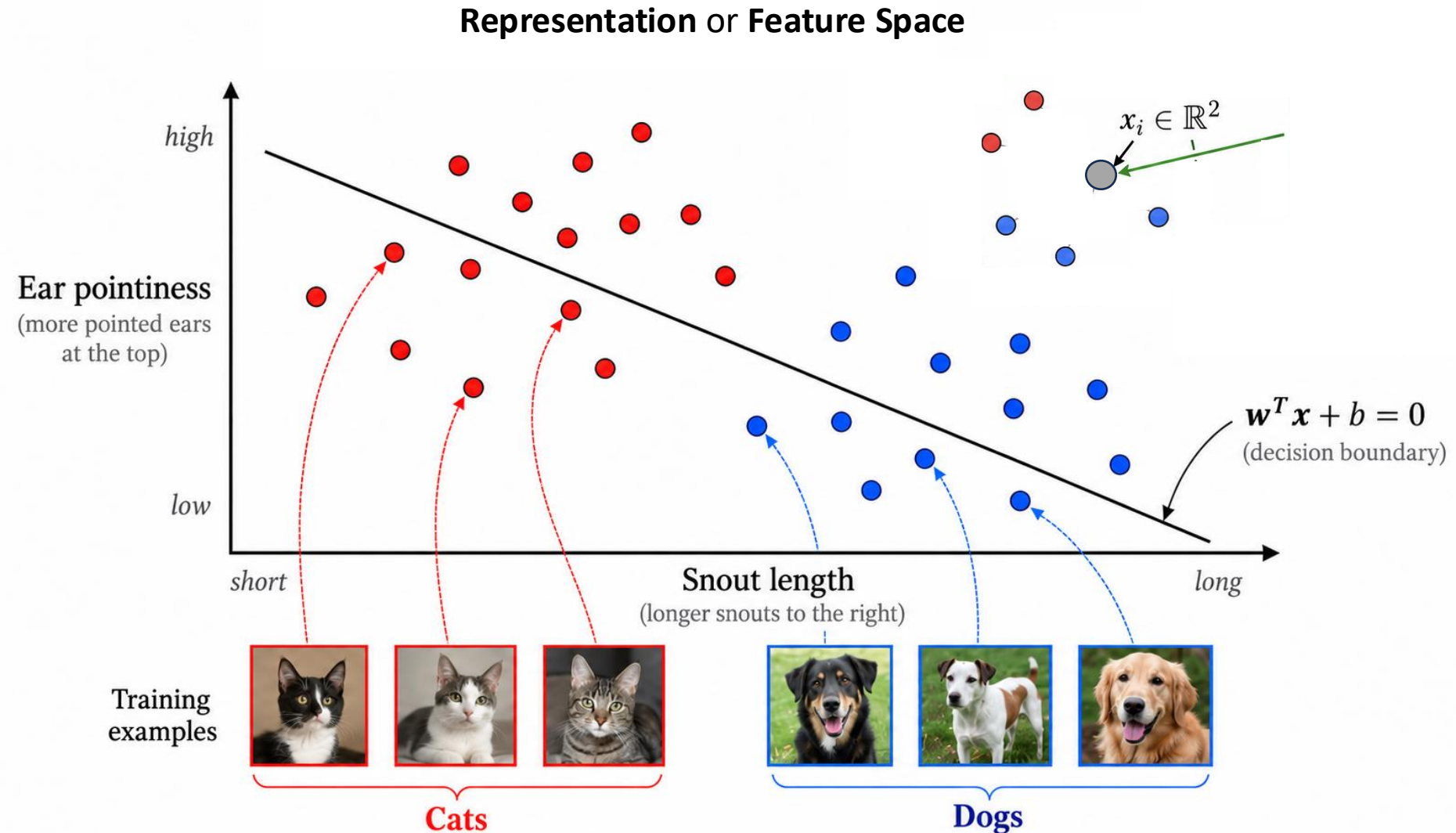
Basic idea:

If it walks like a duck, quacks like a duck, then
it's probably a duck.

Machine Learning: Classifying Cats and Dogs

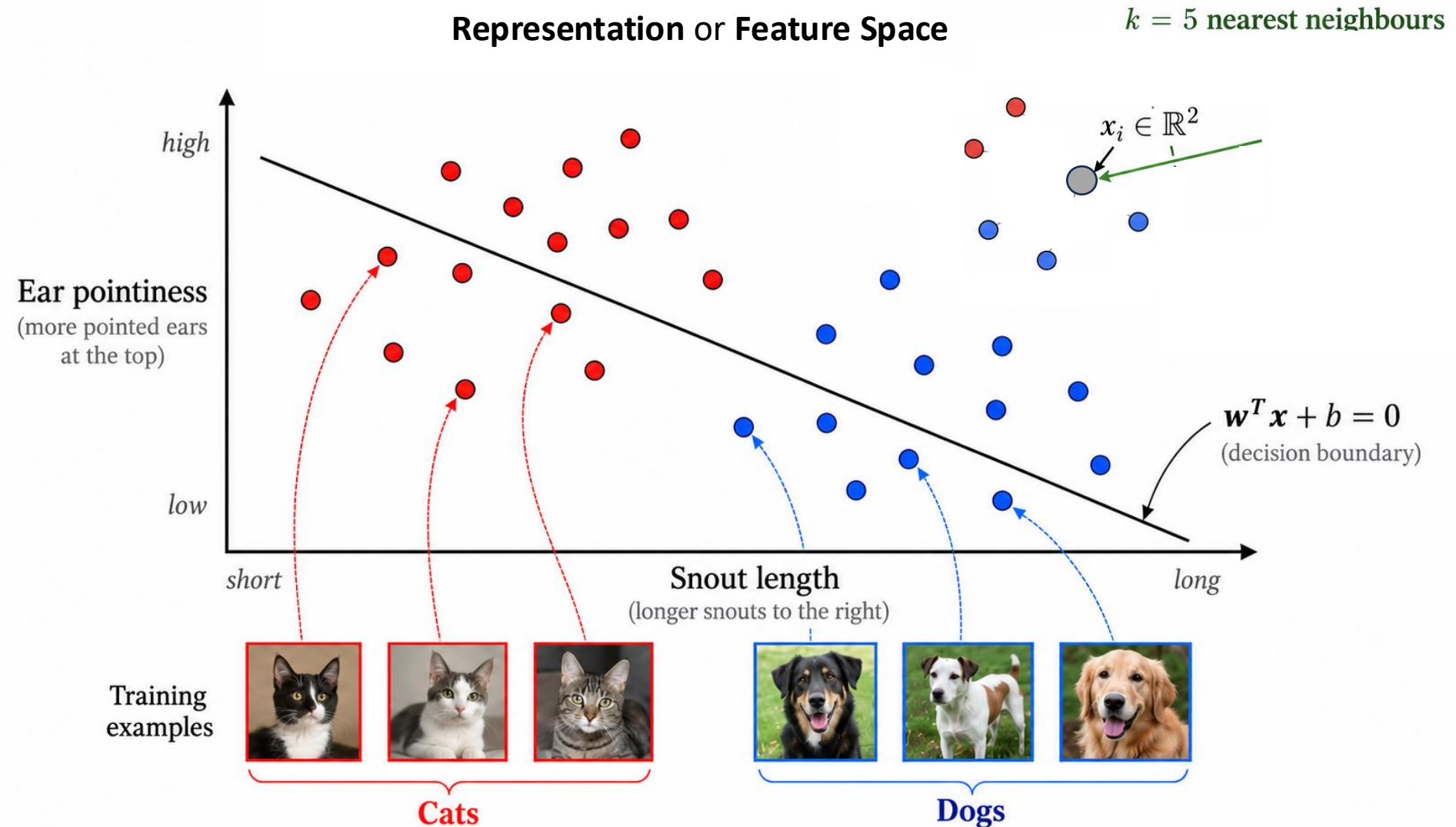


Machine Learning: Classifying Cats and Dogs



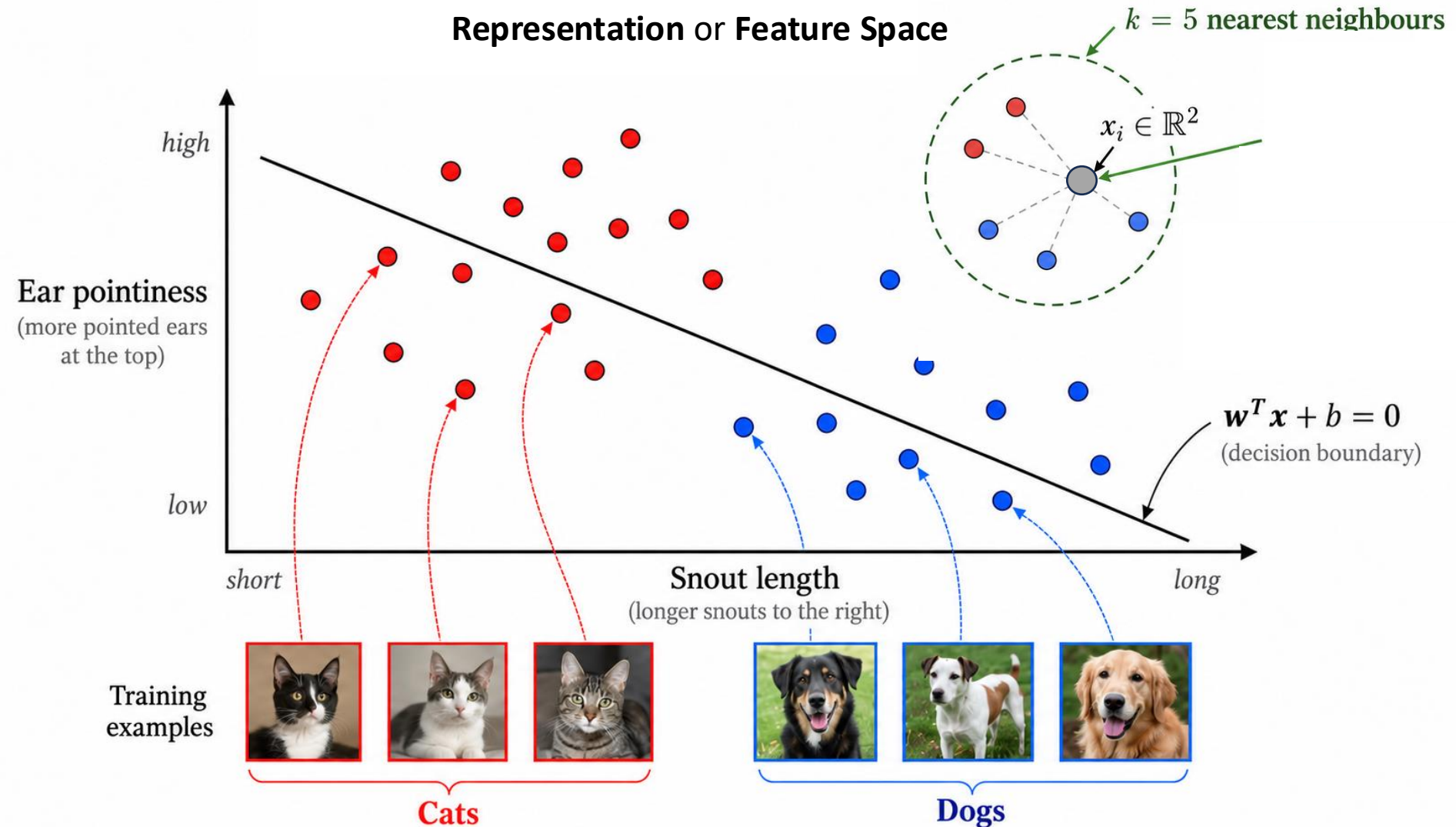
$$D(x_a, x_b) = \sqrt{\left(x_a^{(1)} - x_b^{(1)}\right)^2 + \left(x_a^{(2)} - x_b^{(2)}\right)^2}$$

Machine Learning: Classifying Cats and Dogs



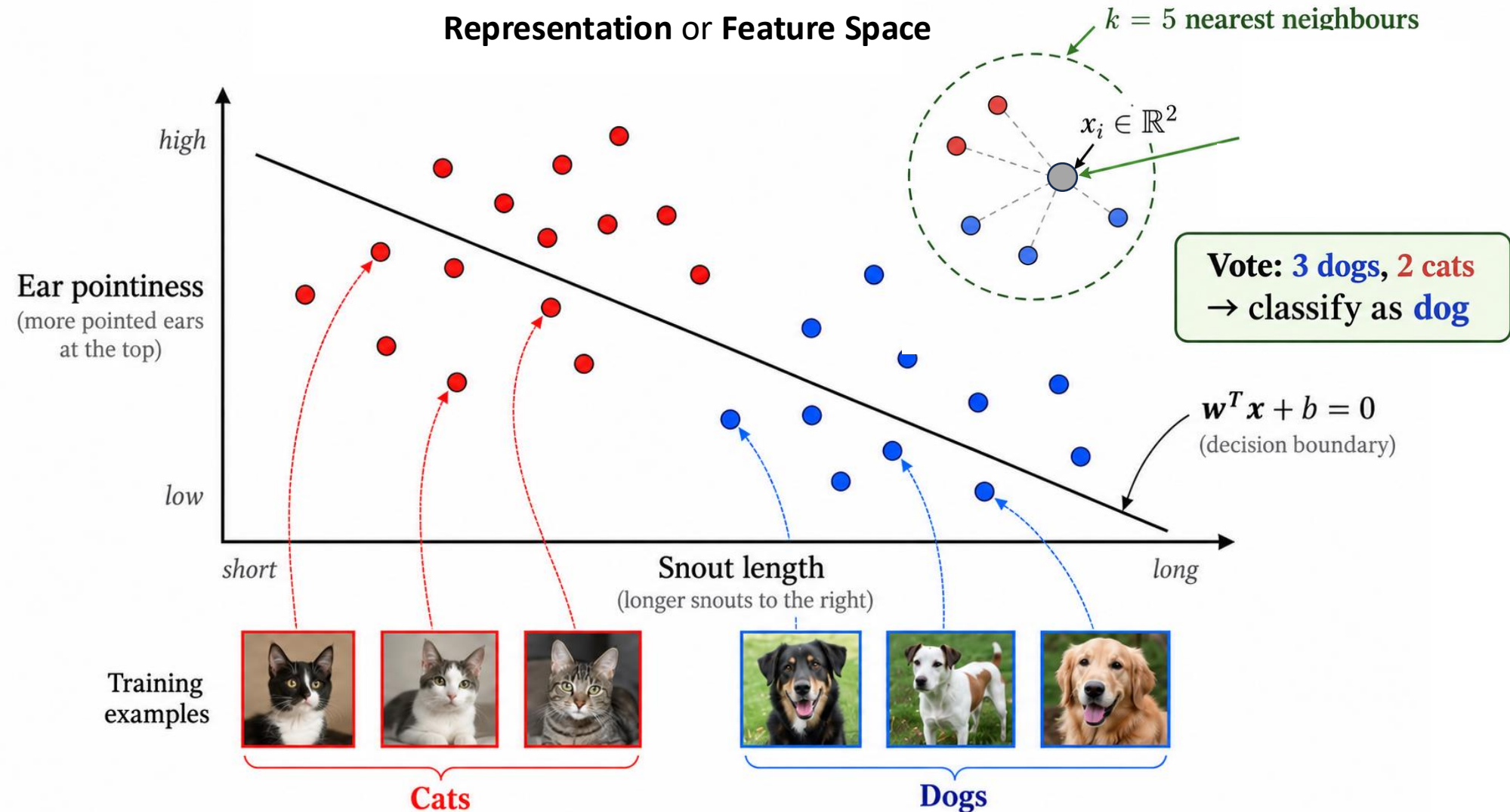
$$D(x_a, x_b) = \sqrt{\left(x_a^{(1)} - x_b^{(1)}\right)^2 + \left(x_a^{(2)} - x_b^{(2)}\right)^2}$$

Machine Learning: Classifying Cats and Dogs



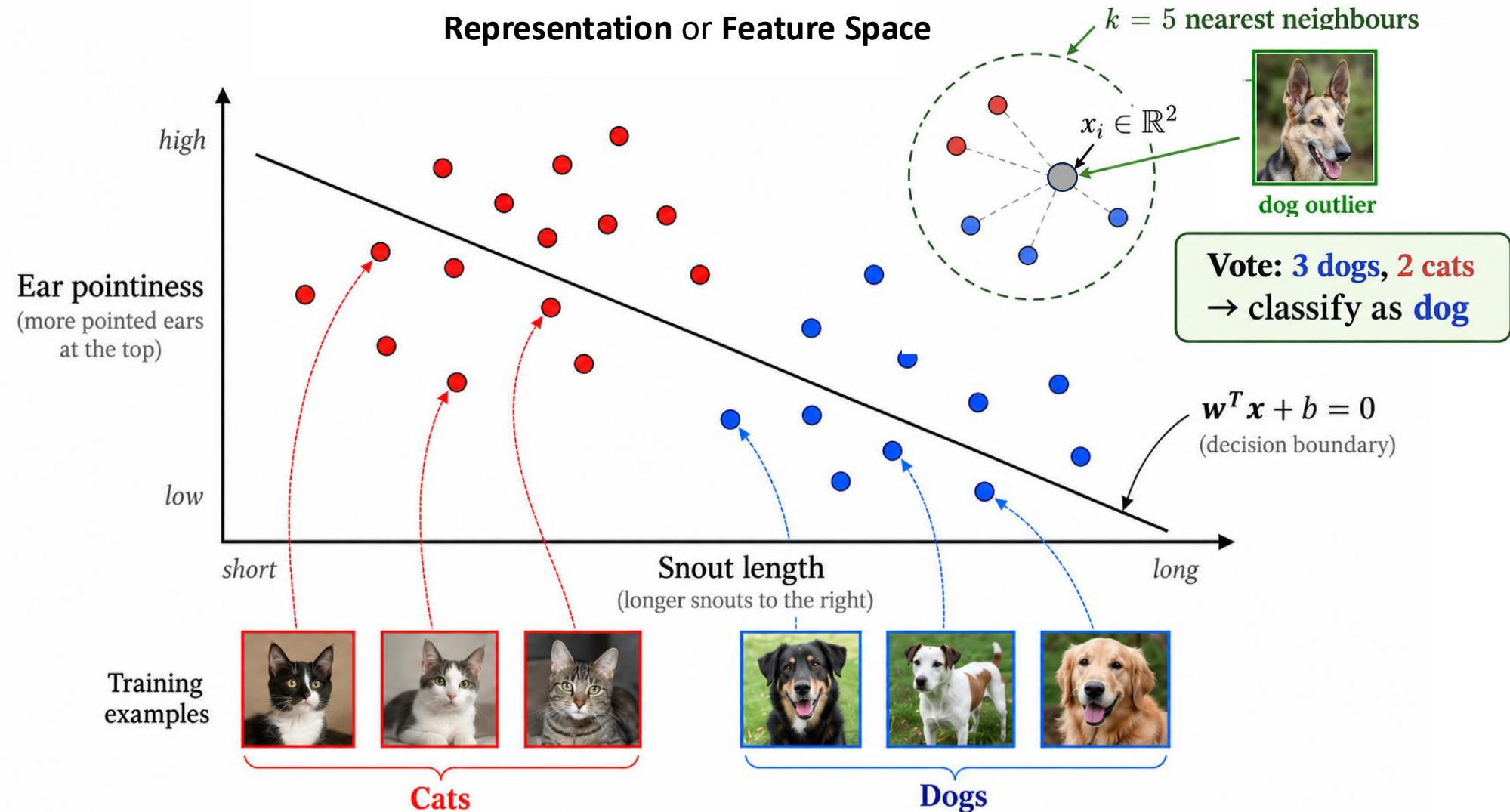
$$D(x_a, x_b) = \sqrt{\left(x_a^{(1)} - x_b^{(1)}\right)^2 + \left(x_a^{(2)} - x_b^{(2)}\right)^2}$$

Machine Learning: Classifying Cats and Dogs



$$D(x_a, x_b) = \sqrt{\left(x_a^{(1)} - x_b^{(1)}\right)^2 + \left(x_a^{(2)} - x_b^{(2)}\right)^2}$$

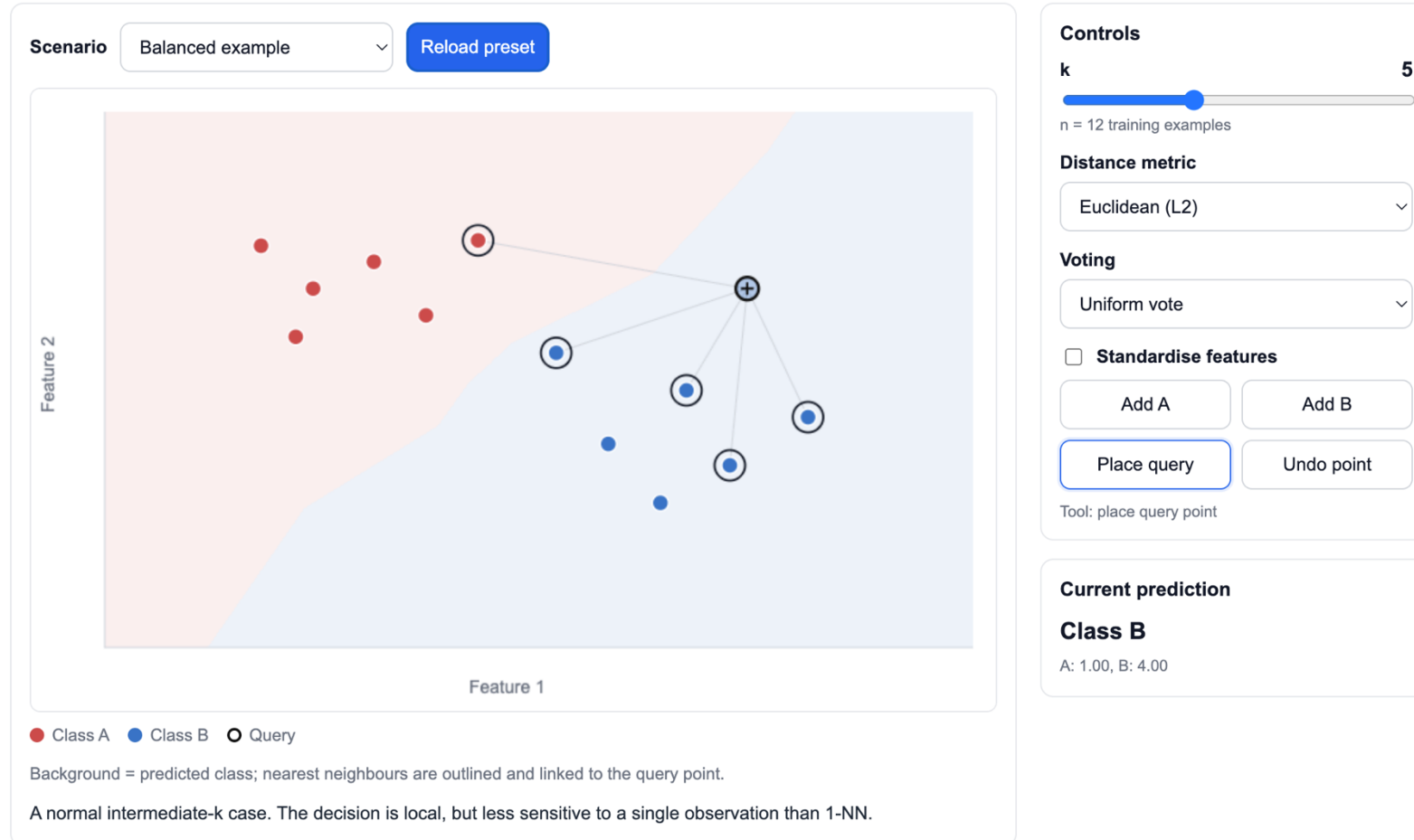
Machine Learning: Classifying Cats and Dogs



What happens if? - kNN Limitations

Interactive k-NN Edge Cases

Explore 1-NN, $k = n$, ties, duplicates, imbalance, feature scaling, and the curse of dimensionality.



[DEMO](#)

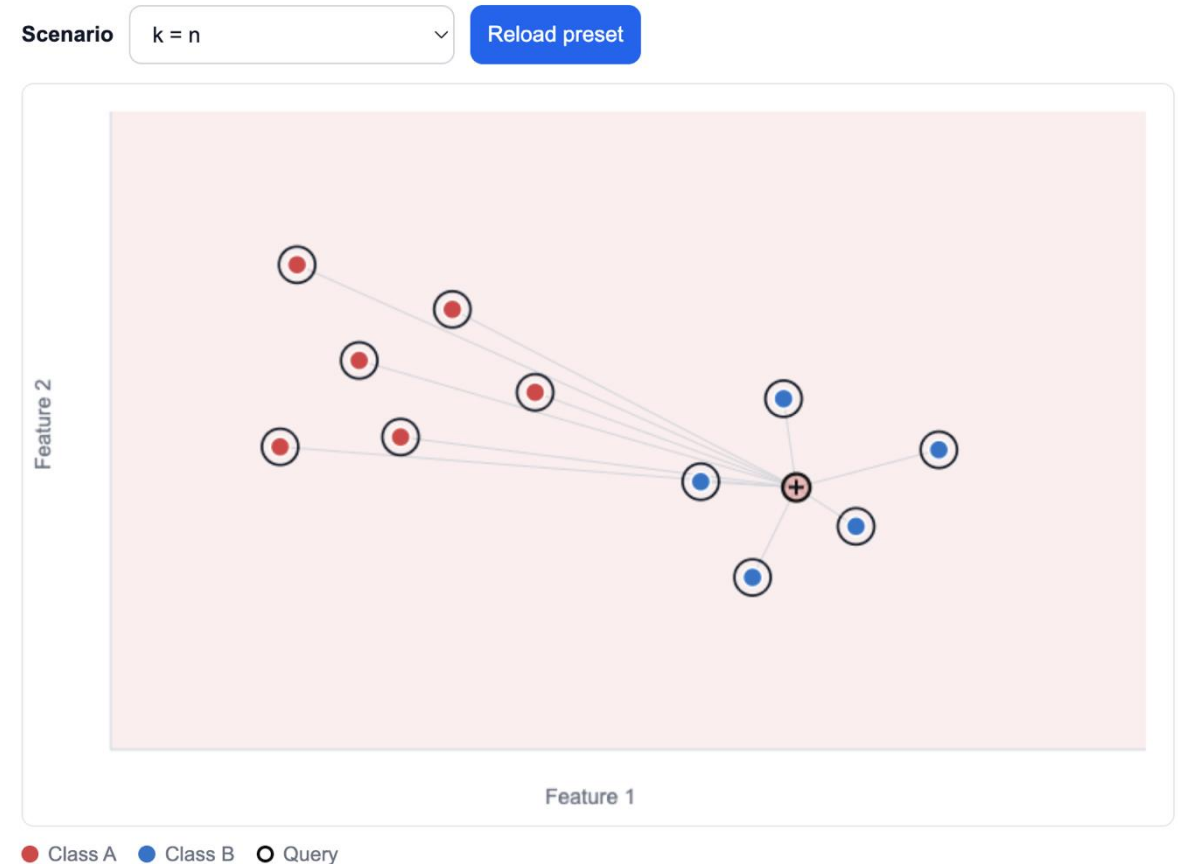
What happens if? - kNN Limitations

- If $k = 1$, each point gets the label of its single nearest training example
- Maximally **local**, low bias / high variance
- However, very susceptible to outliers and noise



What happens if? - kNN Limitations

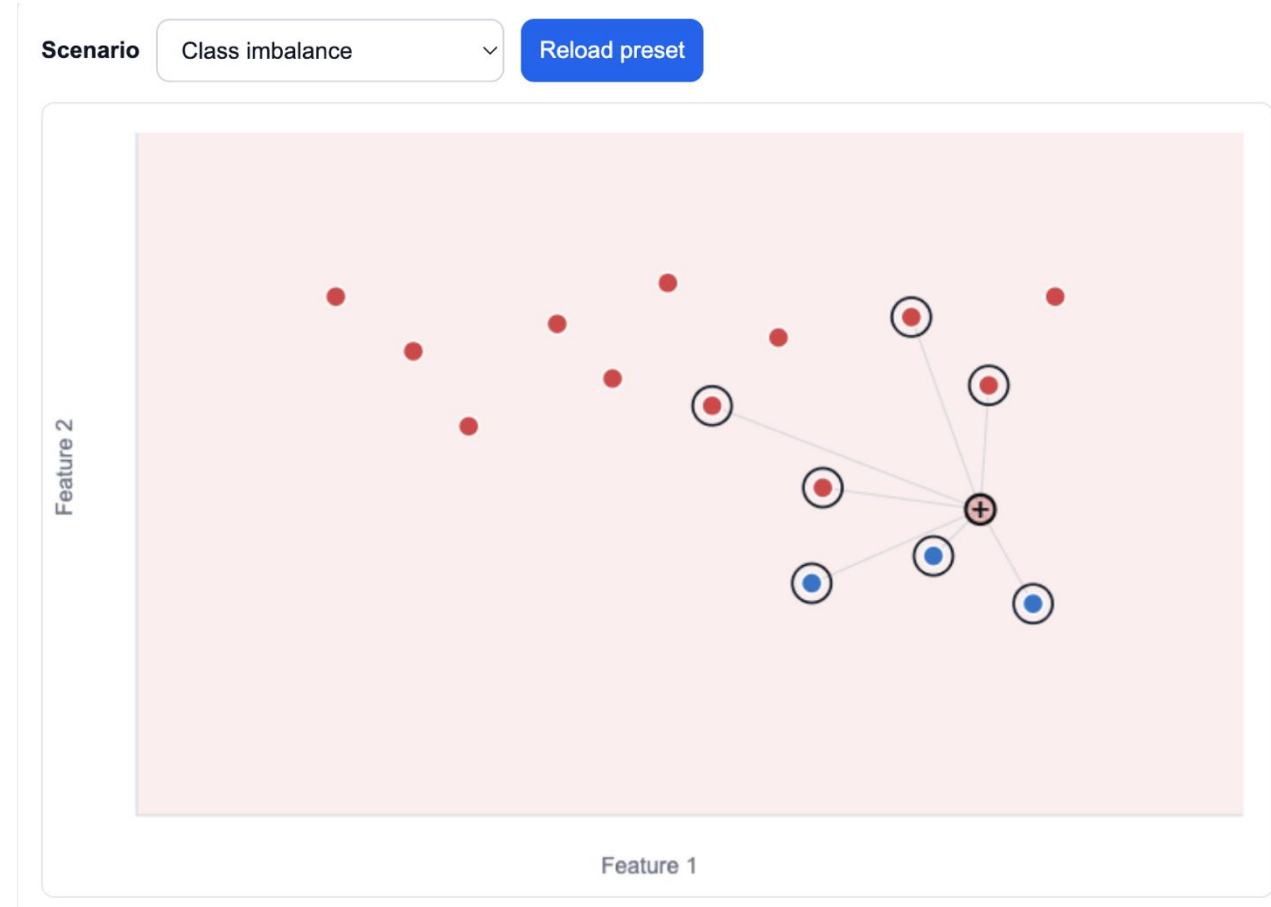
- If $k = n$: Every prediction uses the entire training set.
- **Always** predicts the global majority class.
- Maximally global: high bias / low variance



kNN Limitations

Class Imbalance:

A locally meaningful minority class can be overwhelmed as k becomes larger.



kNN Limitations

Scaling issues

- Attributes may have to be scaled to prevent distance measures from being dominated by one of the attributes

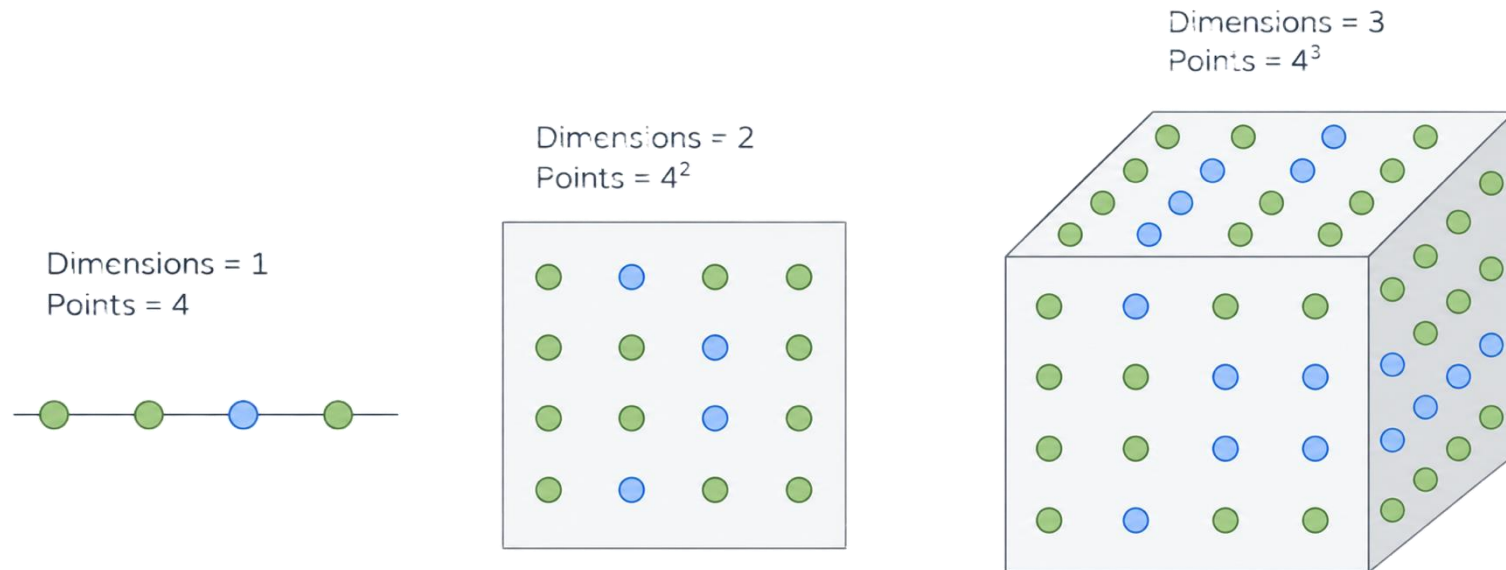
Example:

- height of a person may vary from 1.5m to 1.8m
- weight of a person may vary from 90lb to 300lb
- income of a person may vary from £15k to £200k

kNN Limitations – High Dimensionality

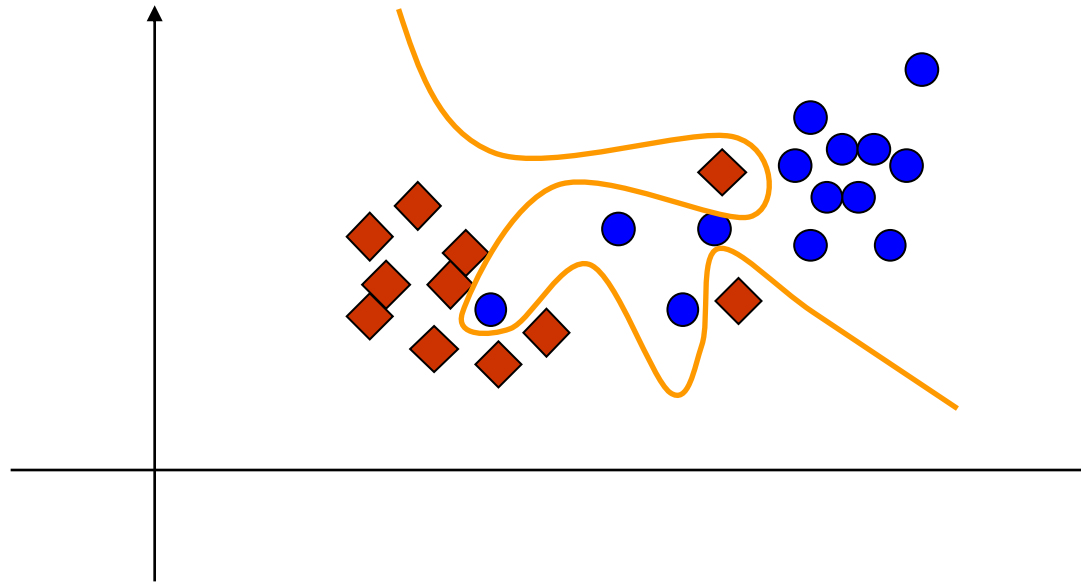
Curse of Dimensionality

- kNN quickly starts losing accuracy as the num. of dimensions goes up.
- If there are many irrelevant attributes, the accidental similarity “shadow” meaningful similarity and k-NN is no better than random guessing.



Classification Approaches: A Linear Classifier?

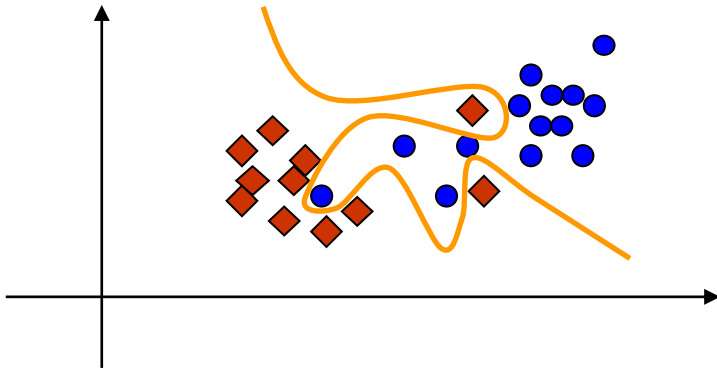
- Does k-NN classify with a straight line? (is it a linear classifier?)
- Nonlinear Classification boundary



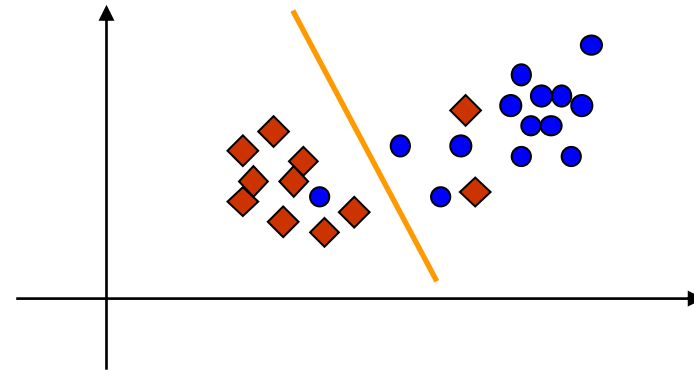
Classification Approaches: Linear Classifier

- **Generalization and Memorization**

- Remembering everything is not learning
- The true test of learning is handling similar but unseen cases, i.e., Generalization



Has great memorization but may generalize poorly (under certain assumptions)



Has lesser memorization but may generalize better (under certain assumptions)

Discriminant based Classification

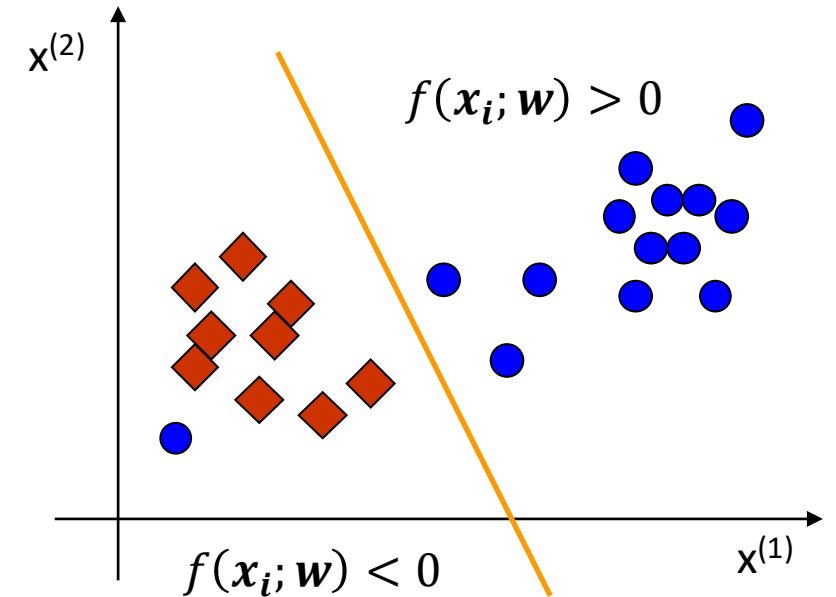
- In this type of classification, the objective is to **learn a function f** or, in the case of more than 2 classes, a set of functions from training data which can generate decisions for test data such that the classes in the data can be separated.

- Class label assignment: $c(x) = \operatorname{argmax}_{k=1,\dots,M} f_k(x)$
 - $f_k(x)$ tells you the ' k -classiness' of an example x
- If the number of classes $M = 2$:
 - Choose class-1 if $f_1(x) \geq f_2(x)$, i.e., $f_1(x) - f_2(x) \geq 0$
 - Otherwise assign it to class-2, i.e., $f_1(x) - f_2(x) < 0$
 - We can thus replace the two functions with a single function
$$f(x) = f_1(x) - f_2(x)$$

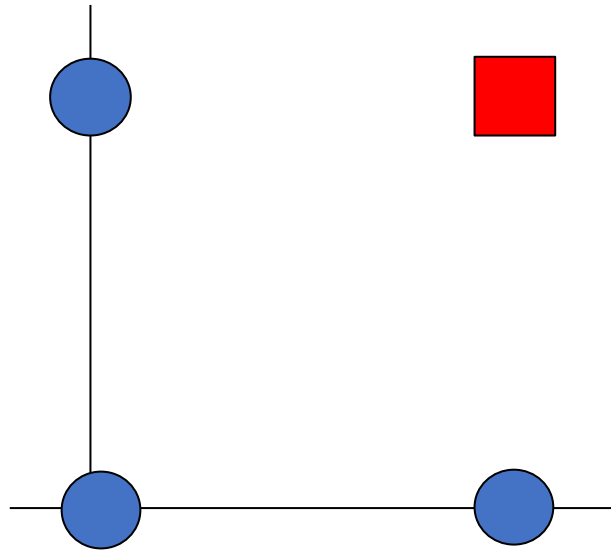
Assign to positive class if $f(x) \geq 0$, otherwise negative

$f(x) = 0$ separates the two classes and is called the discriminant

If the function(s) are linear, the classifier is called a **linear discriminant**



Exercise



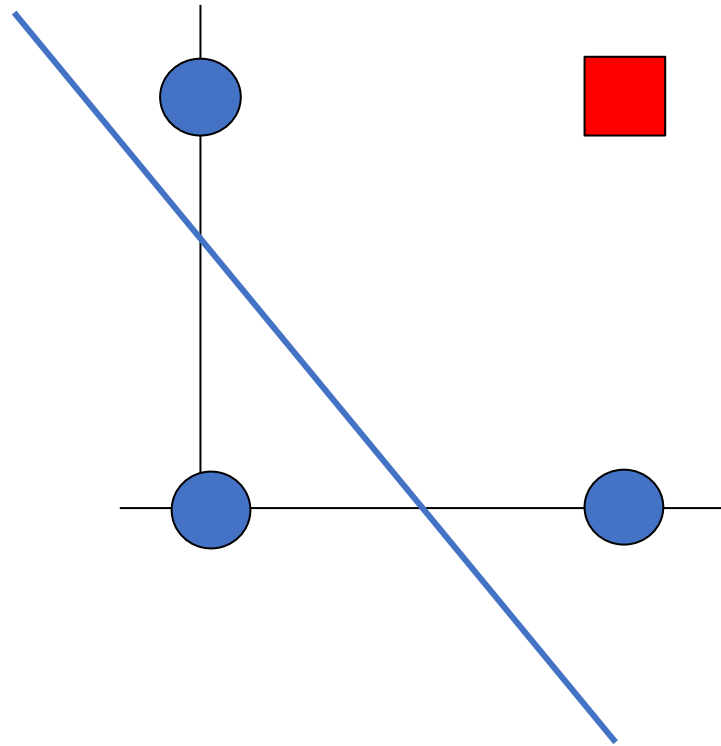
$$f(\mathbf{x}; \mathbf{w}) = w_1 x^{(1)} + w_2 x^{(2)} + \dots + w_d x^{(d)} + b = 0$$

such that

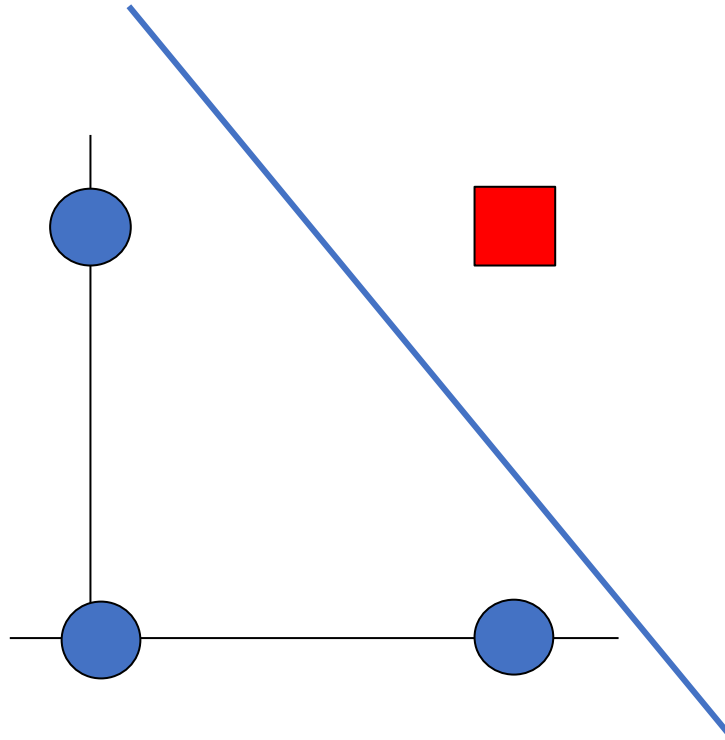
If $y_i = +1$, then $f(\mathbf{x}_i; \mathbf{w}) > 0$

And If $y_i = -1$, then $f(\mathbf{x}_i; \mathbf{w}) < 0$

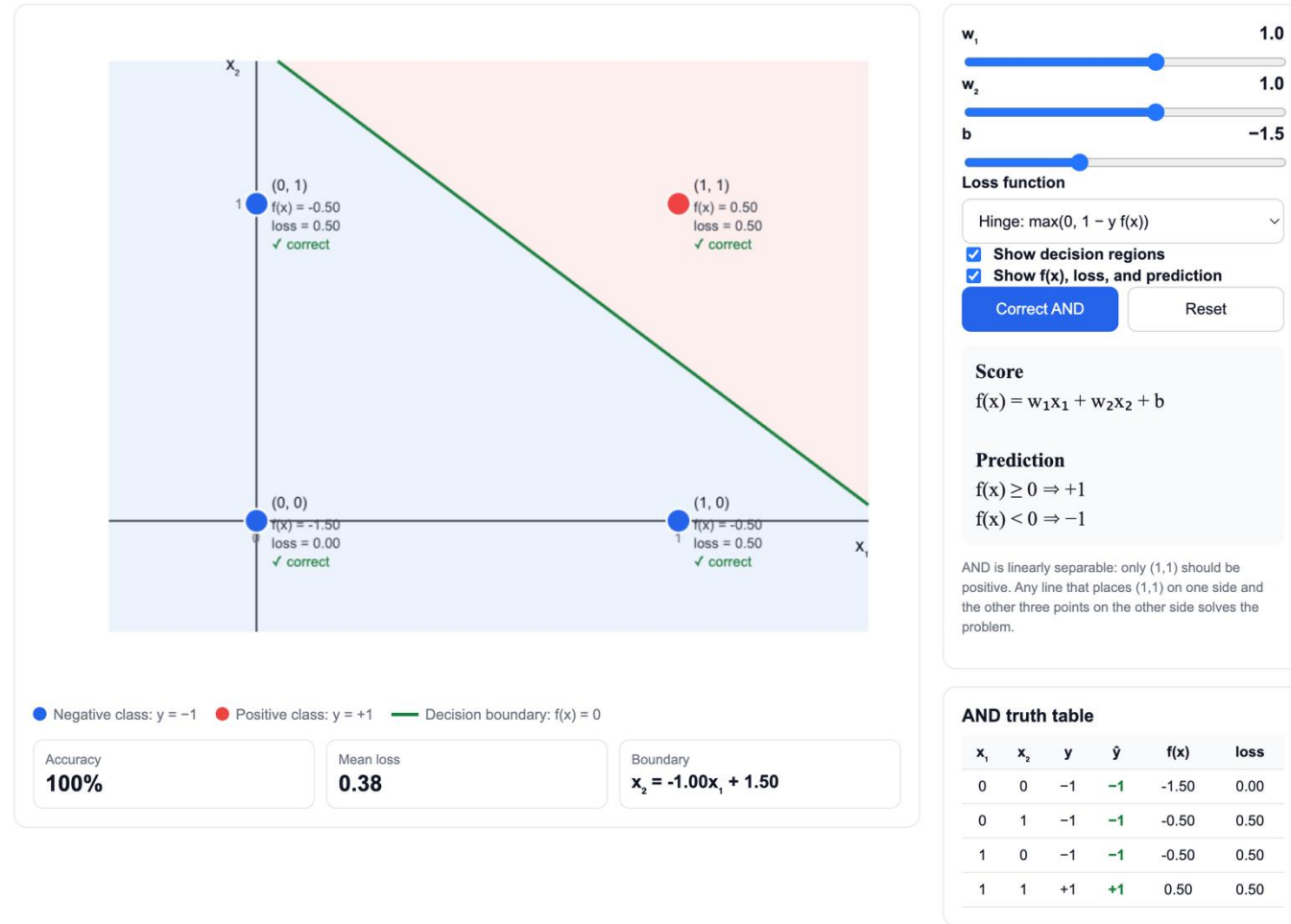
Exercise



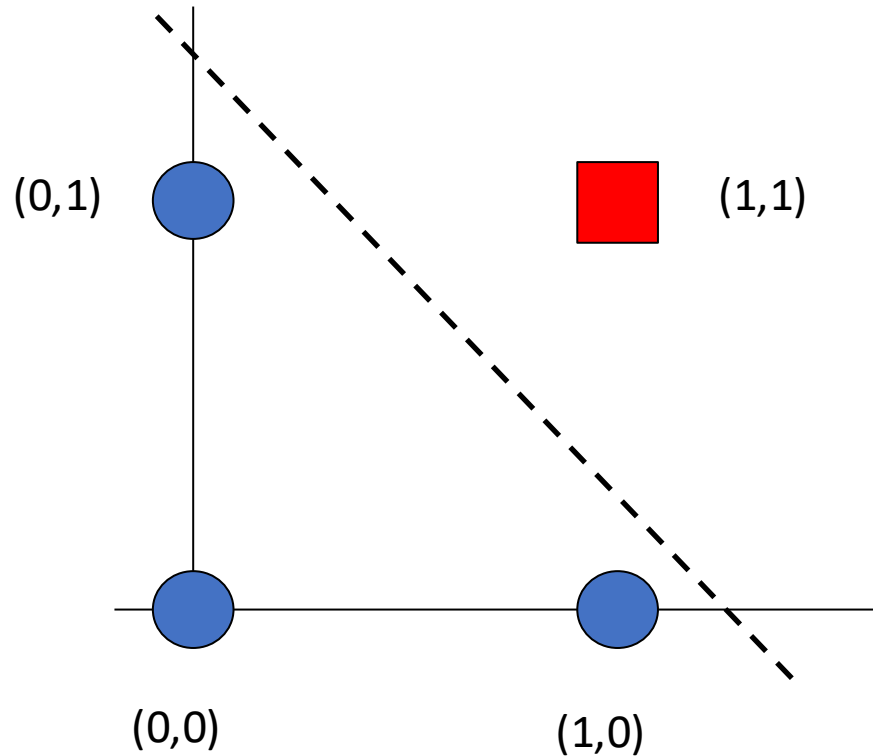
Exercise



Doing it interactively



Example: Another Way (Algebraic Constraint Satisfaction to find equation)



$$f(\mathbf{x}; \mathbf{w}) = w_1 x^{(1)} + w_2 x^{(2)} + b = 0$$



$$\begin{aligned} (1,1): & w_1(1.0) + w_2(1.0) + b > 0 \\ (1,0): & w_1(1.0) + w_2(0.0) + b < 0 \\ (0,1): & w_1(0.0) + w_2(1.0) + b < 0 \\ (0,0): & w_1(0.0) + w_2(0.0) + b < 0 \end{aligned}$$



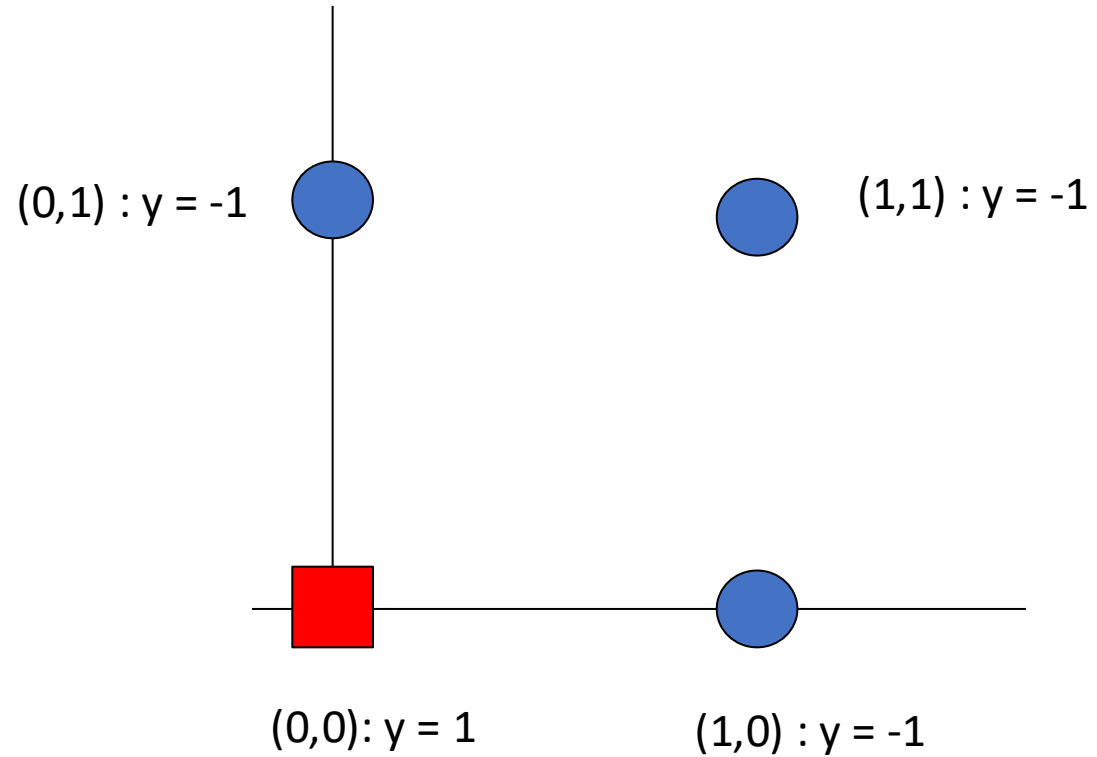
$$\begin{aligned} w_1 + w_2 + b &> 0 \\ w_1 + b &< 0 \\ w_2 + b &< 0 \\ b &< 0 \end{aligned}$$



$$\begin{aligned} b &= -1.5 \\ w_1 &= 1.0 \\ w_2 &= 1.0 \end{aligned}$$

Exercise

- Is this problem linearly separable?

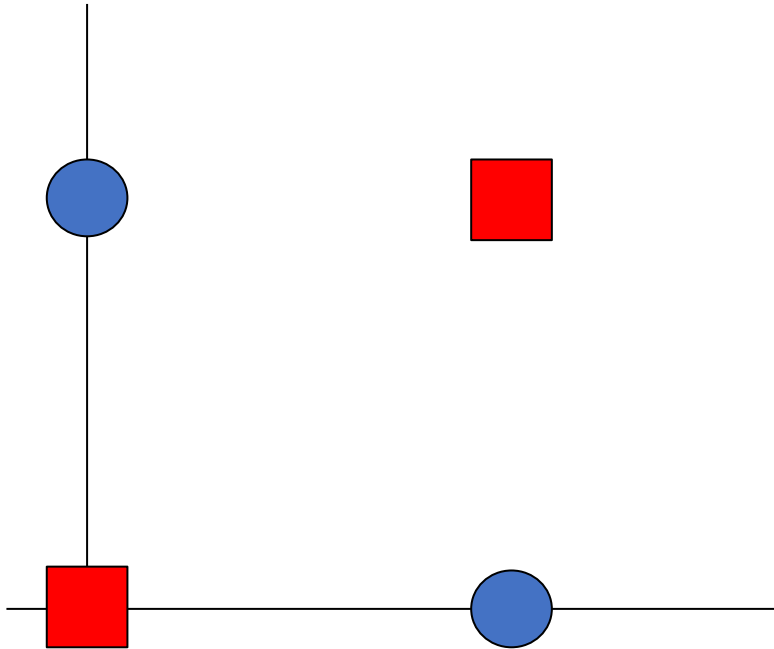


What about this one?

- (0,0,0): -1
- (1,0,0): +1
- (0,1,0): -1
- (0,0,1): +1
- (1,0,1): +1
- (1,1,0): +1
- (0,1,1): +1
- (1,1,1): +1

$$f(\mathbf{x};\mathbf{w}) = w_1x^{(1)}+w_2x^{(2)}+w_3x^{(3)}+b = 0$$

- What about this one?



$$\begin{aligned}(1,1): & w_1(1.0)+w_2(1.0)+b > 0 \\(1,0): & w_1(1.0)+w_2(0.0)+b < 0 \\(0,1): & w_1(0.0)+w_2(1.0)+b < 0 \\(0,0): & w_1(0.0)+w_2(0.0)+b > 0\end{aligned}$$

REO: A general framework to look at Linear Classification

- We would like to minimize the number of errors a discriminant function $f(x)$ makes

- **Representation:**

Assume we look at only linear functions

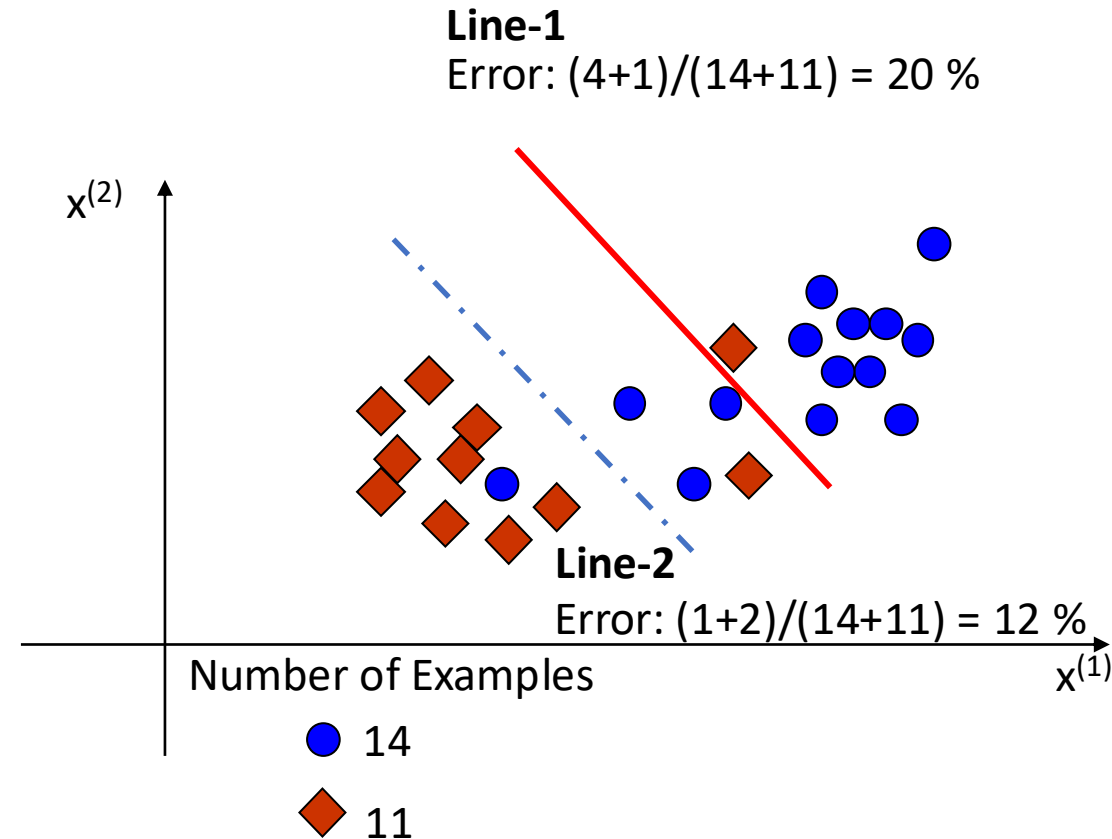
$$f(x; w) = w_0 + w_1 x^{(1)} + w_2 x^{(2)} + \dots + w_d x^{(d)} = 0$$

- **Evaluation:**

We need to define error that a particular $f(x; w)$ makes

- **Optimization:**

We need to minimize the error by tuning w



Building Linear Discriminants

- Representation

- Features
- Linear Function

$$f(\mathbf{x}; \mathbf{w}) = w_1x^{(1)} + w_2x^{(2)} + \dots + w_dx^{(d)} + b = 0$$

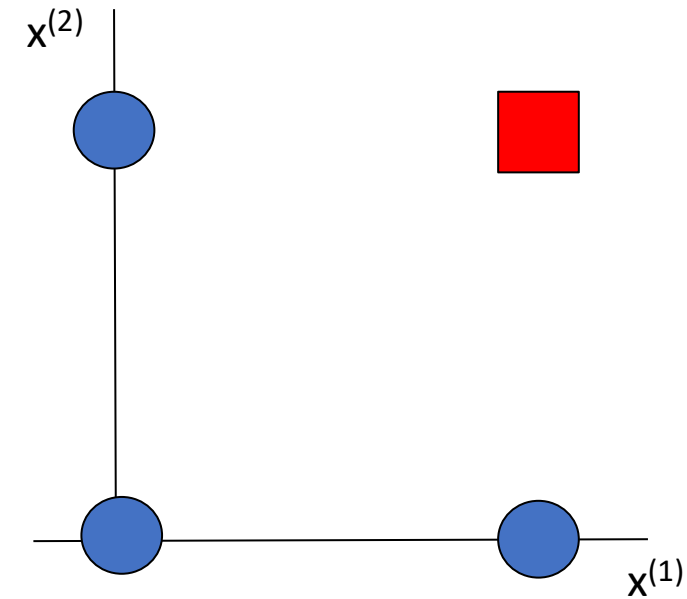
- Evaluation

- Misclassification Error



- Optimization

- Find a line that minimizes misclassifications
- How done: Visual reckoning / Constraint Satisfaction



(1,1):	$w_1(1.0) + w_2(1.0) + b > 0$
(1,0):	$w_1(1.0) + w_2(0.0) + b < 0$
(0,1):	$w_1(0.0) + w_2(1.0) + b < 0$
(0,0):	$w_1(0.0) + w_2(0.0) + b < 0$