COS 126

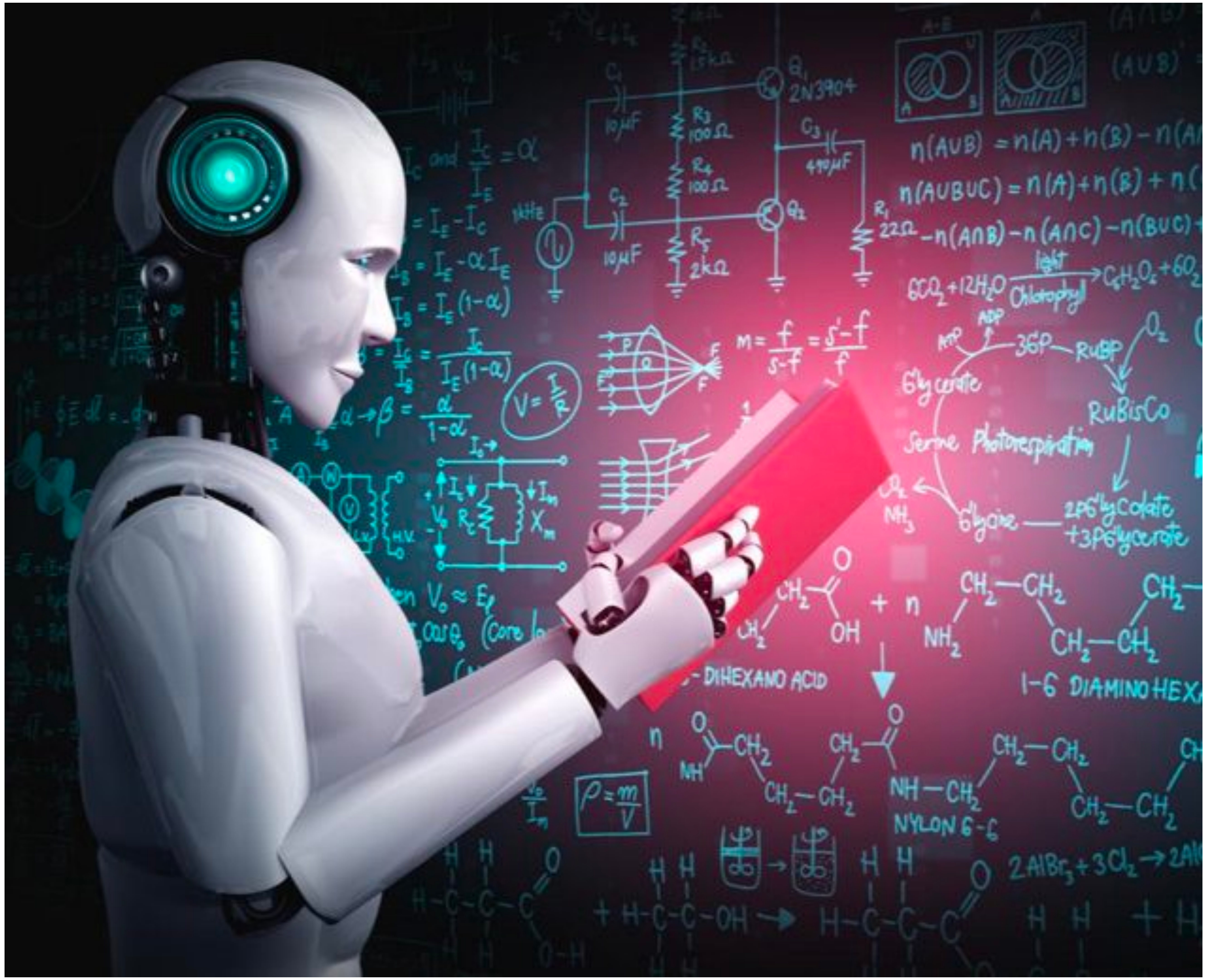
Introduction To Computer Science

Adji Bousso Dieng

PRINCETON
UNIVERSITY

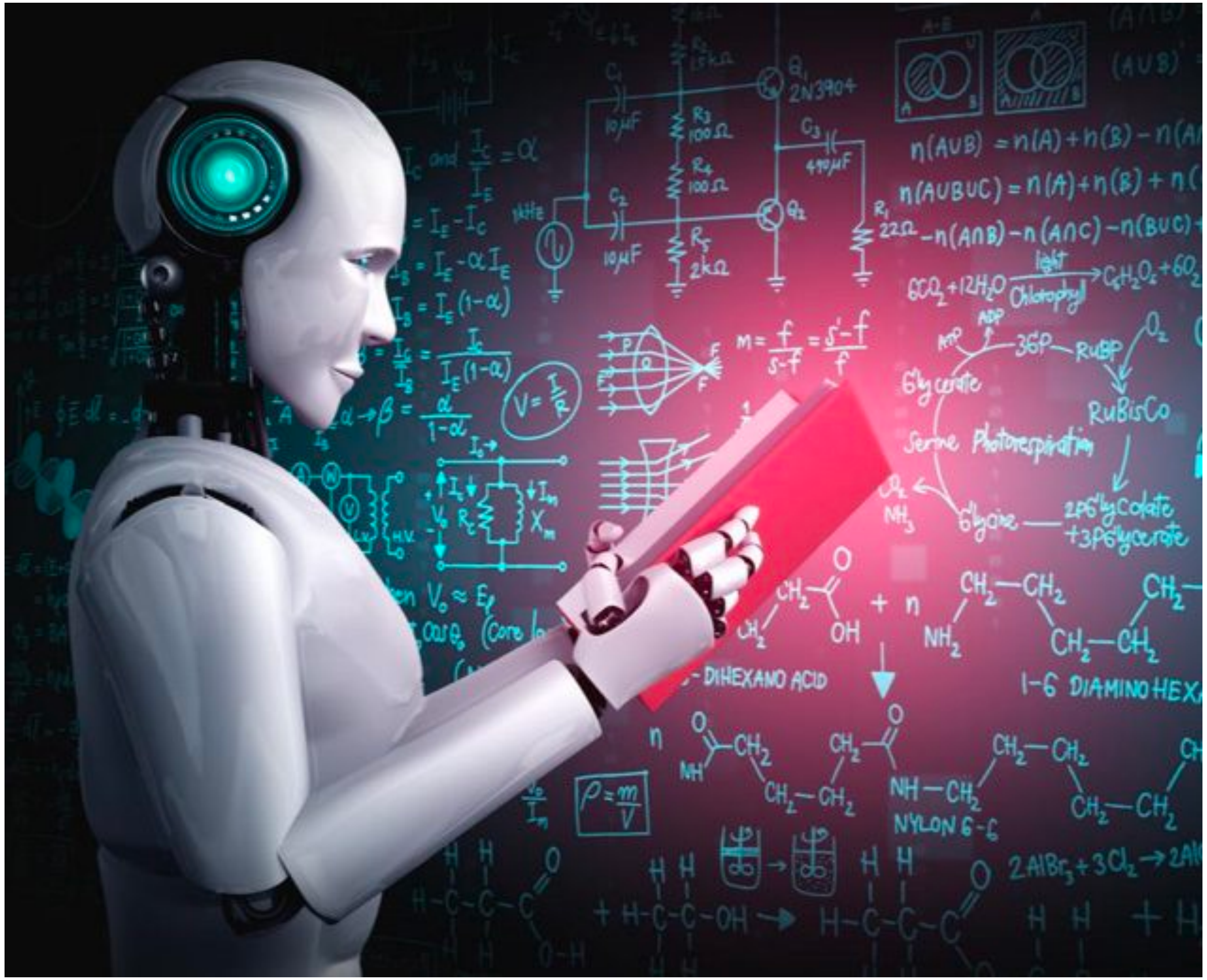
Introduction to Machine Learning

People have different views of AI…



Some think about robots that will be as intelligent or even more intelligent than humans…

People have different views of AI…

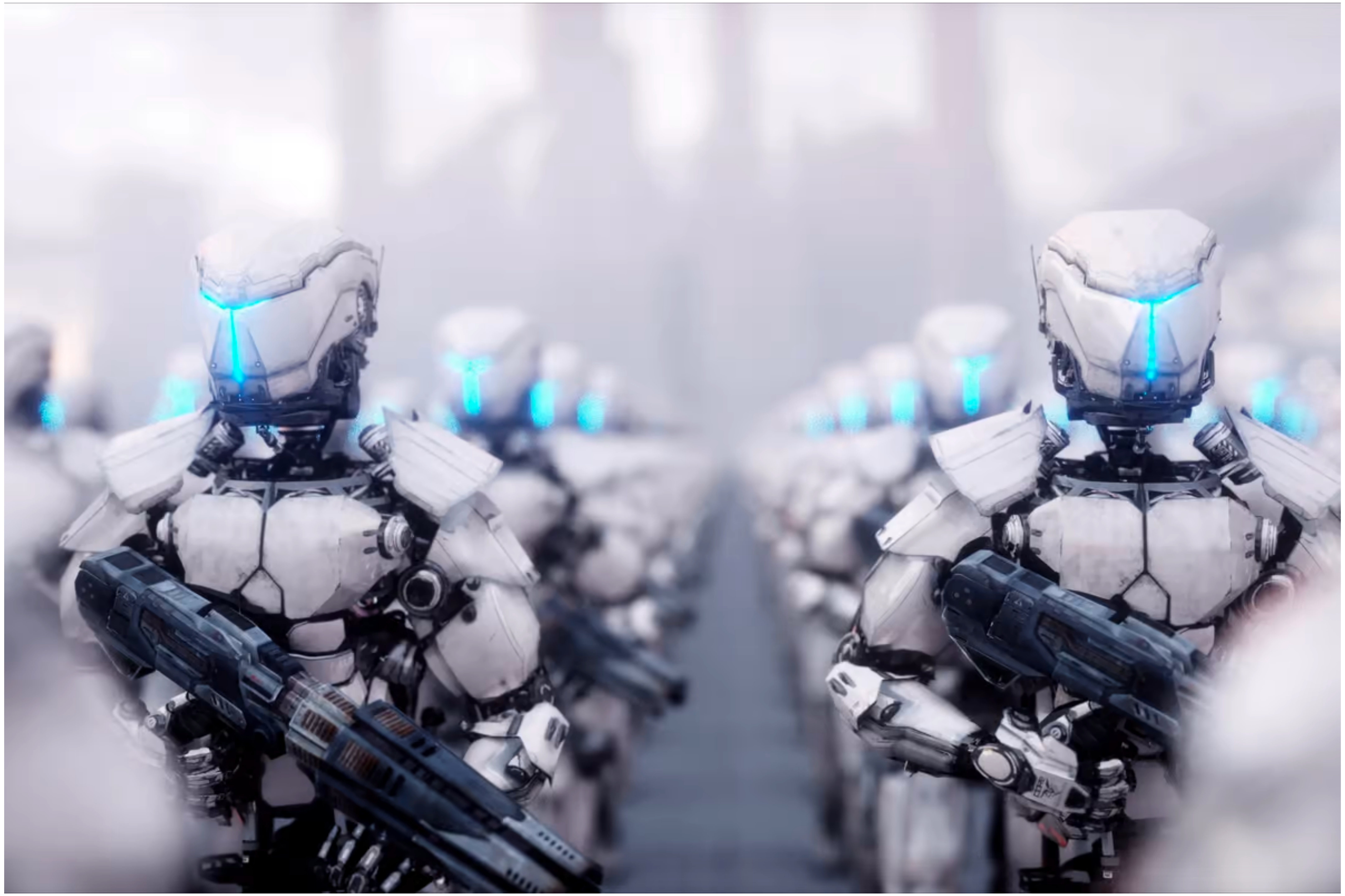


Some think about robots that will be as intelligent or even more intelligent than humans…

To help us tackle problems

Or take our jobs…

People have different views of AI…



Others think about doomsday scenarios, where robots will take over the world and humanity will go extinct…

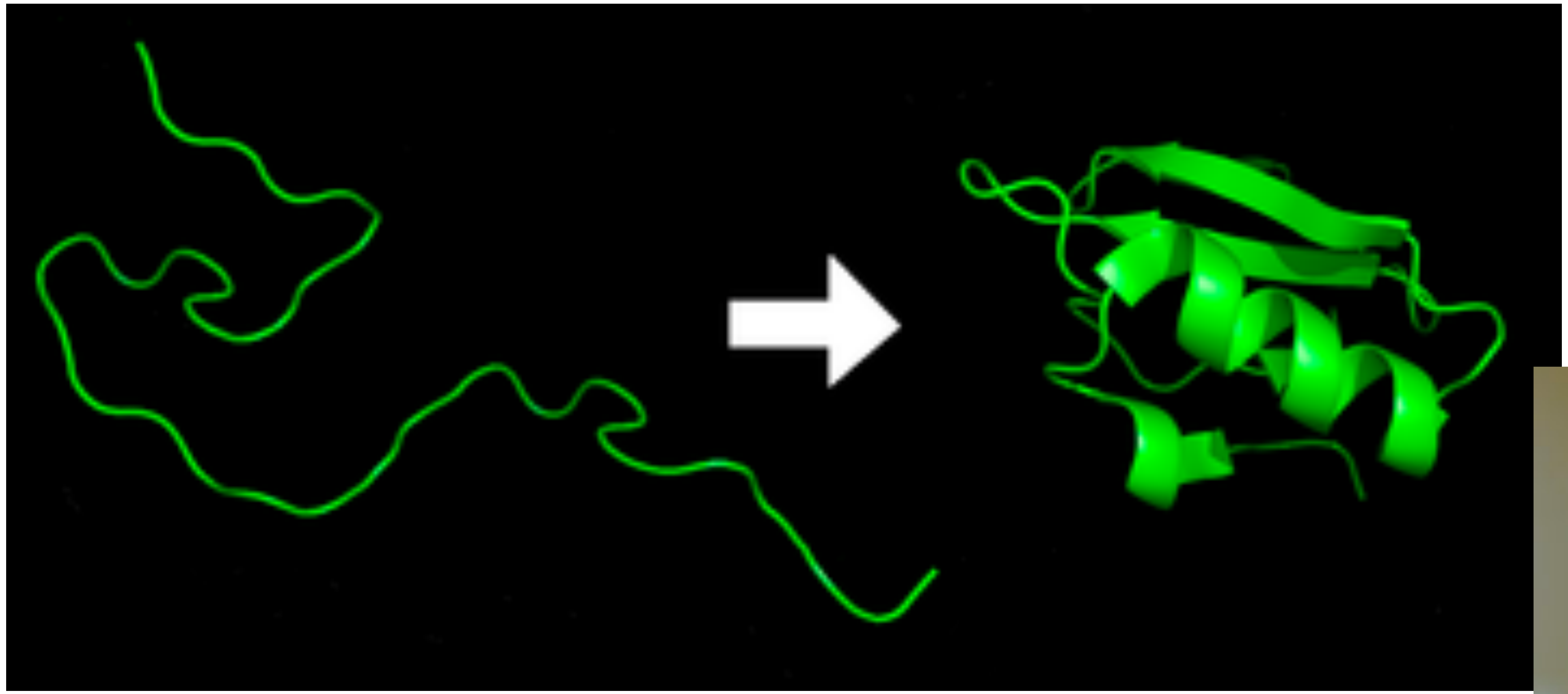
Exciting developments in machine learning today

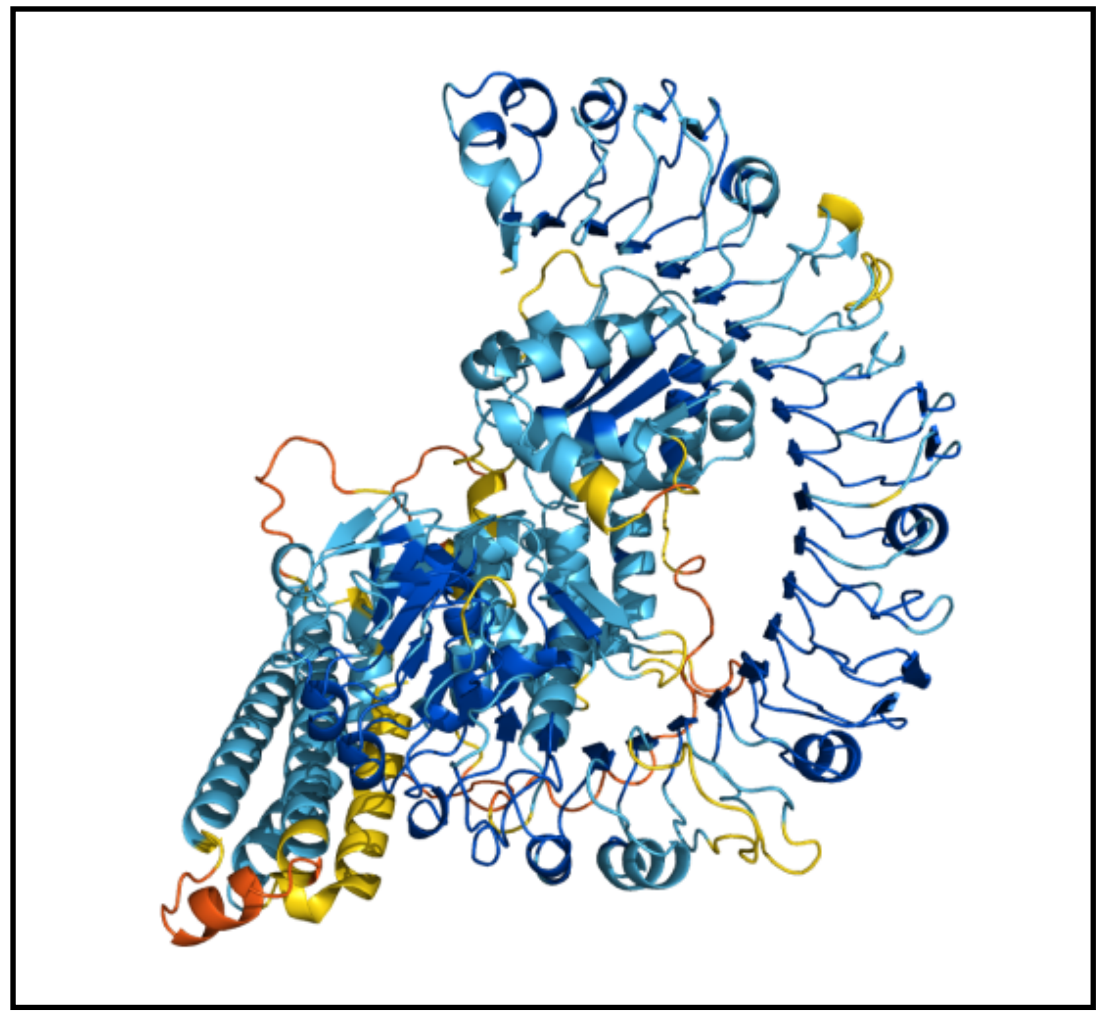
However, there are exciting things AI/ML have enabled and the purpose with this lecture is to show you what AI/ML is about and what you can do with it.

What are some exciting developments you've heard of in AI/ML?

It can play games and even beat human world champions!

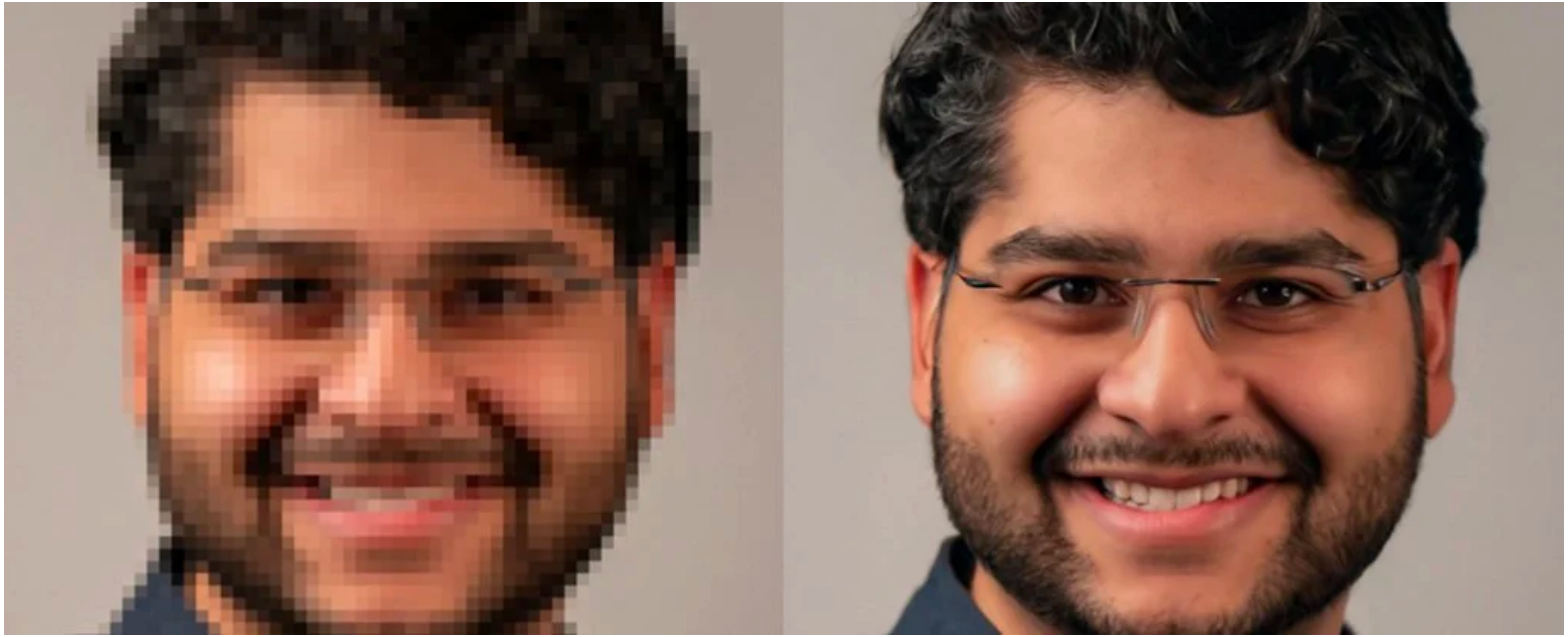
It can summarise documents for you…





AlphaFold

It can help speed up the development of new drugs…



It can help deblur images…this is called image superresolution…

It can help generate new images…based on a prompt…

A young boy is playing basketball.

Two dogs play in the grass.

A dog swims in the water.

A little girl in a pink shirt is swinging.

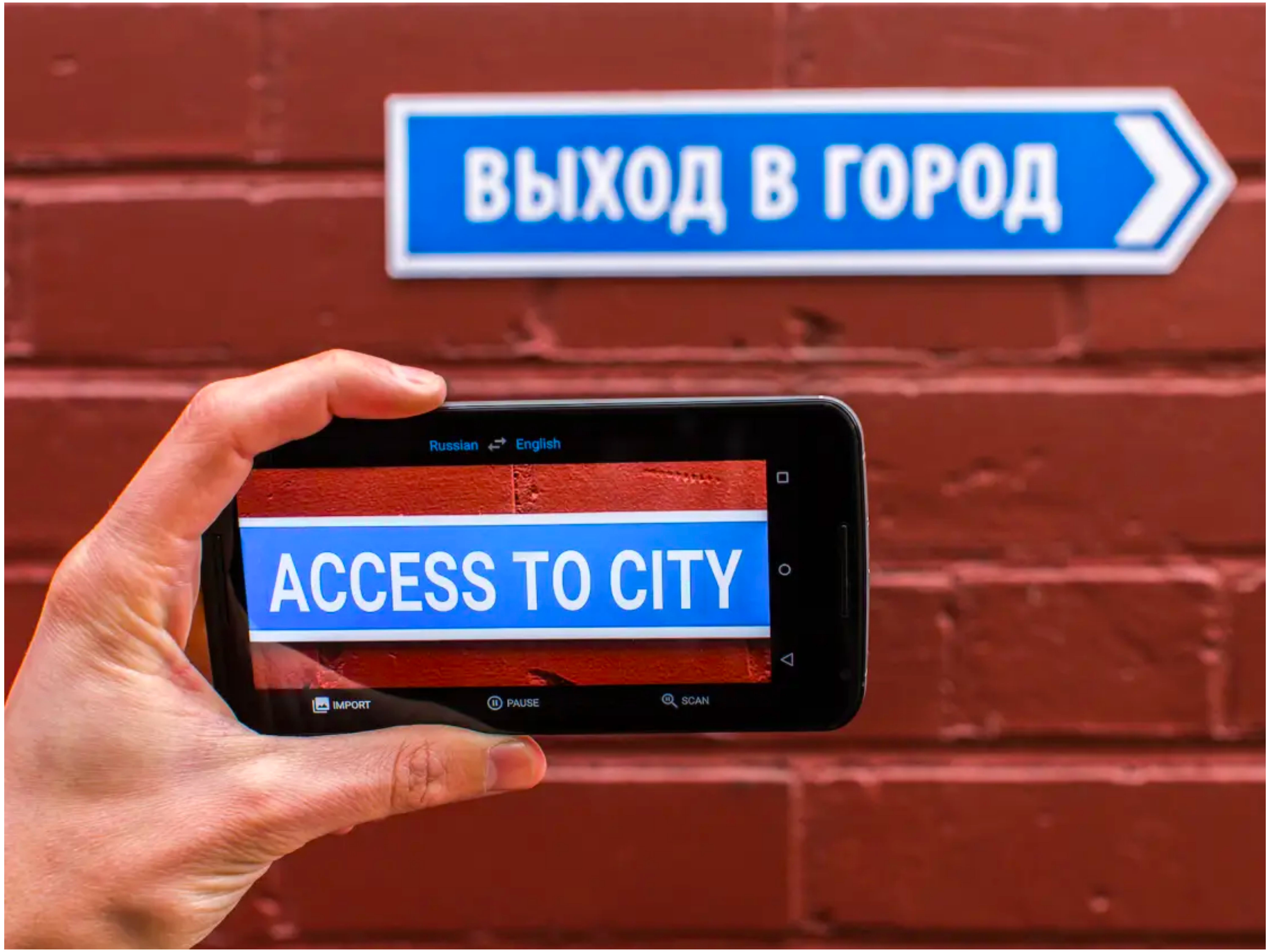
A group of people walking down a street.

A group of women dressed in formal attire.

Two children play in the water.

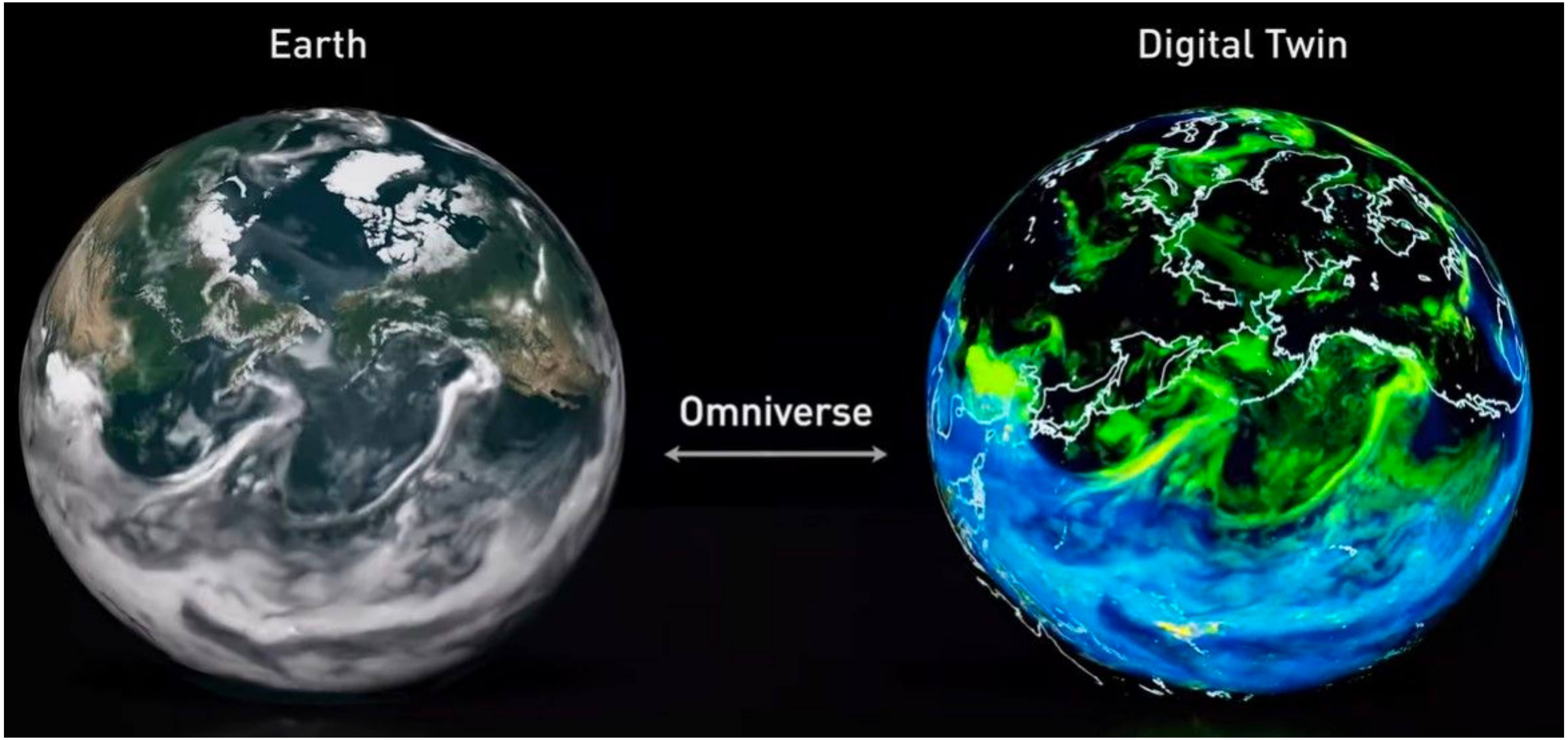
A dog jumps over a hurdle.

It can help caption/describe images, which is specially useful for the visually-impaired…

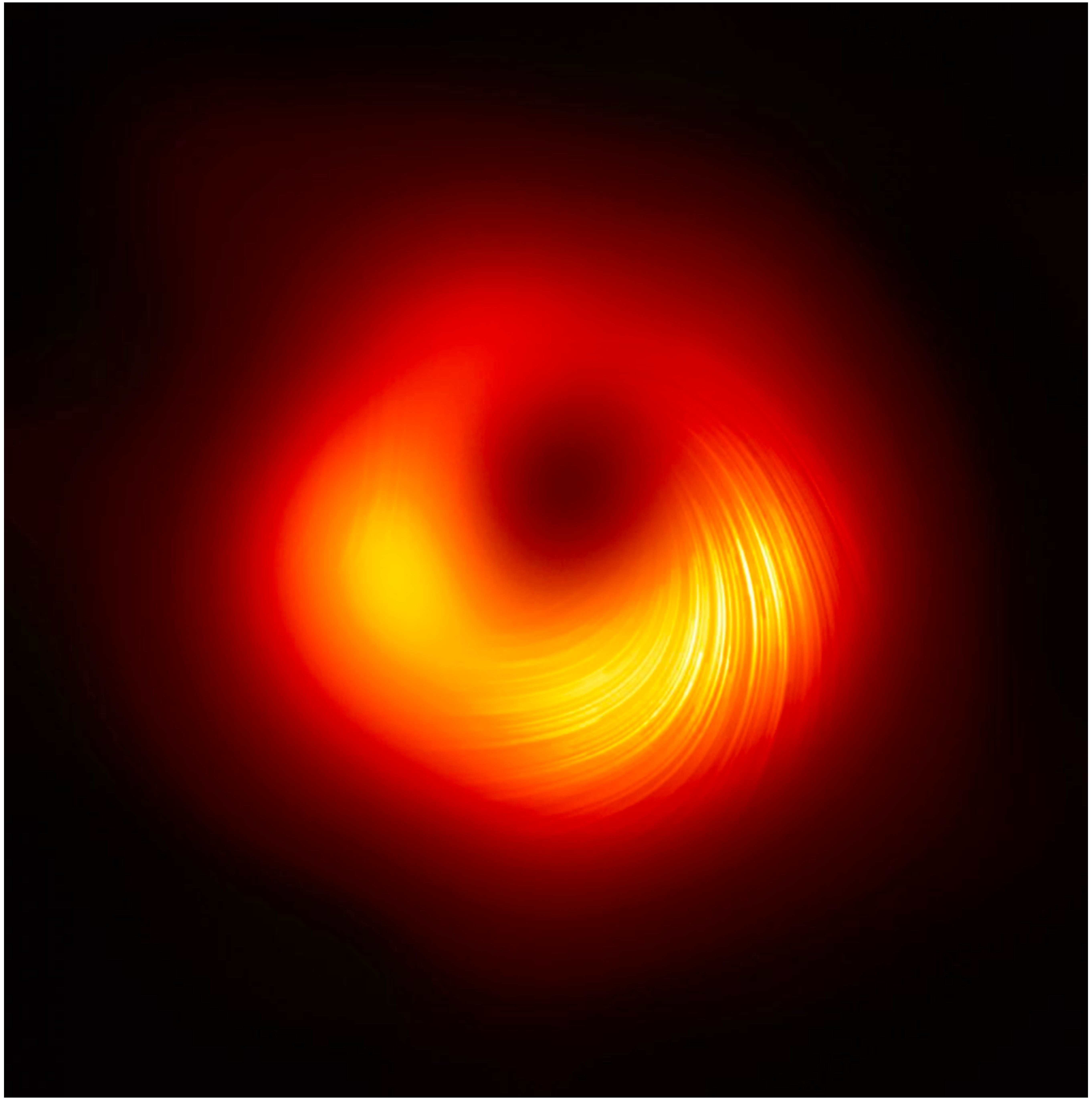


It can help translate between languages automatically…

It can help with automatic speech recognition…



It can help build digital twin of the earth for environmental simulations and planning…



It can even help visualise a black hole!

Hello ChatGPT! Say hi to the Princeton COS 126 class! What would you like to tell them about AI/ML?

Hello Princeton COS 126! 👋

If you take away one idea:

AI is not magic—it turns data, assumptions, and computation into behavior.

You are learning this at a time when:

- the foundations are still evolving
- big ideas are still being discovered
- the field is still wide open

This is rare. 🚀

It can answer your questions and hold a conversation with you!

AI/ML can help us tackle problems in several fields of study and domains

Roadmap

- What is machine learning? How does it relate to other fields?
- Machine learning paradigms
  - Supervised learning
  - Unsupervised learning
  - Reinforcement learning
- Perceptron
  - Example: Golden retriever vs. Doberman
  - Perceptron model
  - Training via update rule
  - Testing
- Multi-layer Perceptron (MLP)
- Multi-class Perceptron

Introduction to machine learning

ML data ⇔ related fields

Player	Minutes	Points	Rebounds	Assists
A	41	20	6	5
B	30	29	7	6
C	22	7	7	2
D	26	3	3	9

Tabular

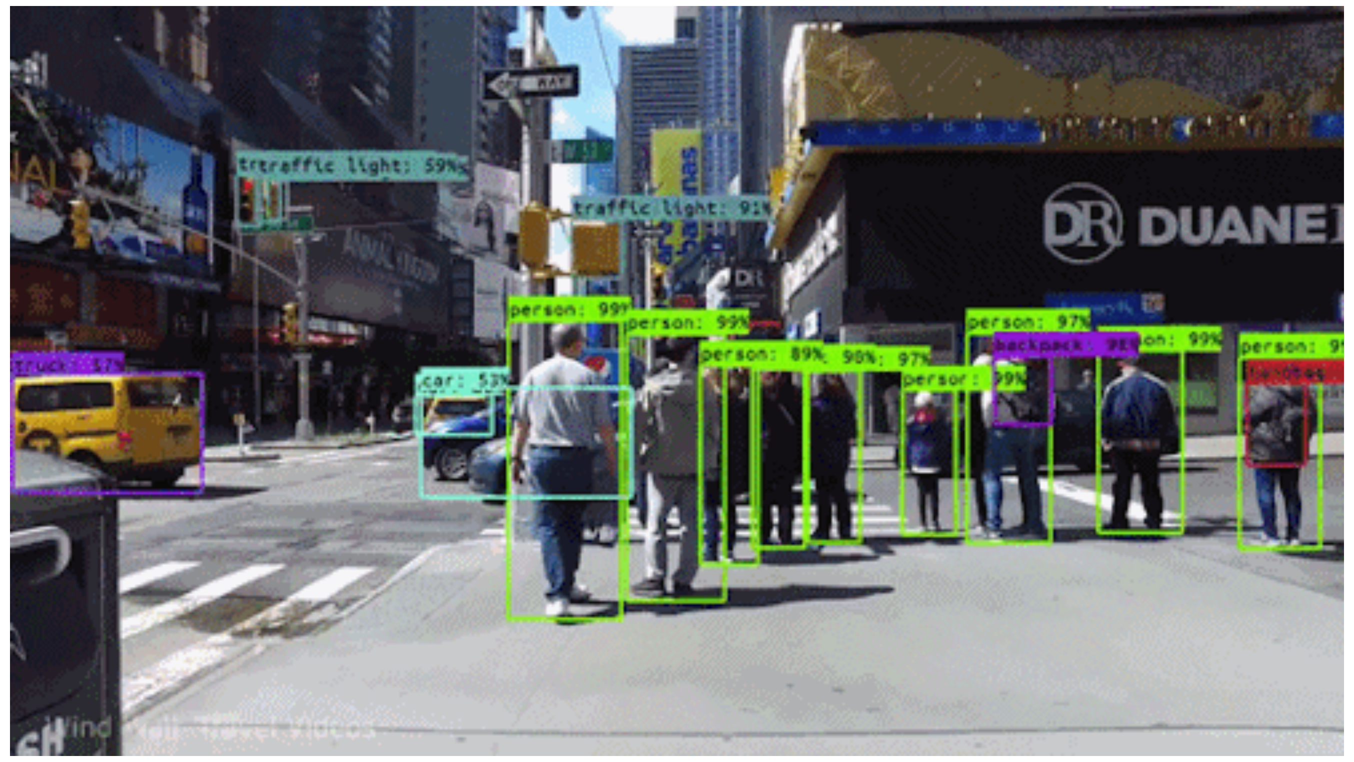
Visual

ML data ⇔ related fields

Player	Minutes	Points	Rebounds	Assists
A	41	20	6	5
B	30	29	7	6
C	22	7	7	2
D	26	3	3	9

Tabular

Machine learning



Visual

Computer vision

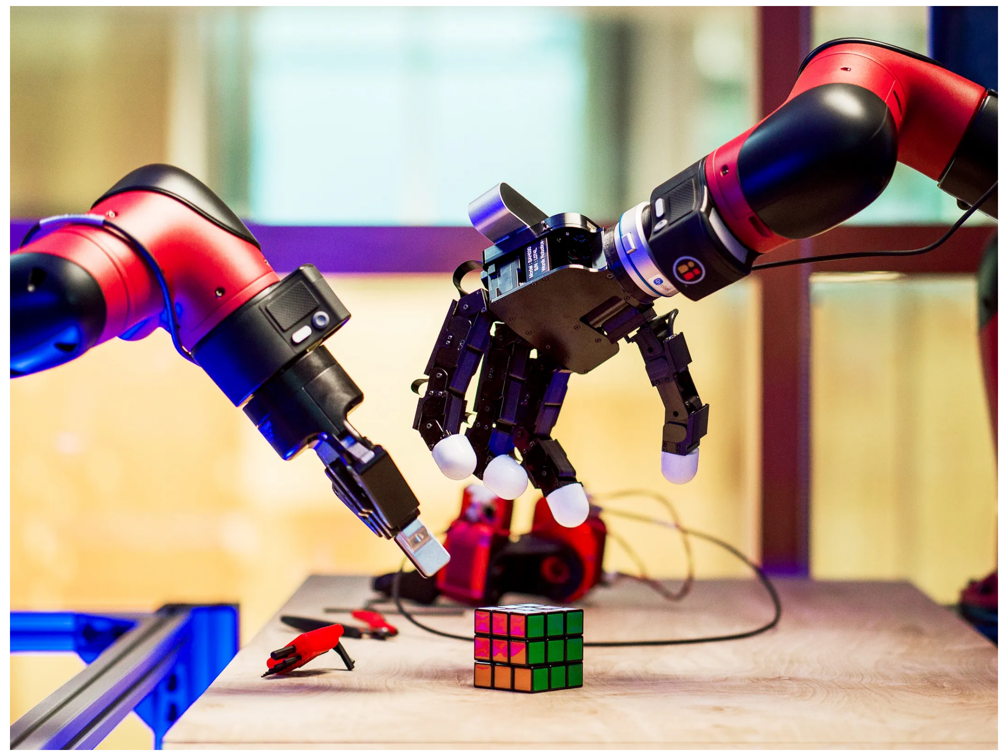
ES

Briefly explain chatGPT using a metaphor that would resonate with an audience of stock-market investors and avoiding AI jargon.

One way to explain chatGPT to stock-market investors is to compare it to a financial advisor who uses past performance data and market trends to make predictions about future investments. Just like a skilled financial advisor, chatGPT uses large amounts of training data to generate intelligent and informed responses to questions, providing valuable insights and suggestions to users.

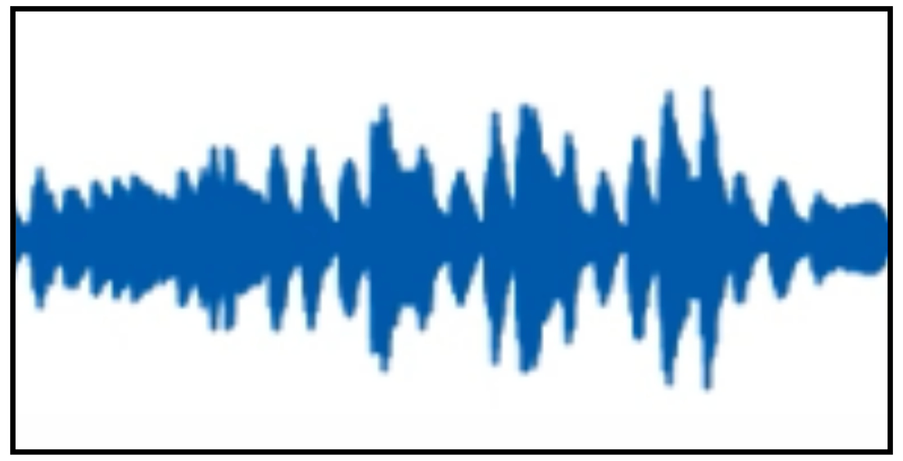
Text

Natural language processing (NLP)



Real manipulation

Robotics



Audio

Audio processing

What is machine learning?

Machine learning: Algorithms that improve automatically using data or repeated experience.

This was **not** true of the programs that you written so far in COS1 26
(e.g. your programs = explicit instructions for the computer).

Note: There are non-machine learning algorithms in computer vision, NLP, robotics, etc.

Machine learning paradigms

1. Supervised learning: learn to **predict** an output from an input.
2. Unsupervised learning: learn **patterns** from data.
3. Reinforcement learning: learn by **interacting** (i.e. **trial-and-error**) with an environment.

Supervised learning

Given (input, output) pairs, **predict** an output from its input.

$\text{input}_1 \rightarrow \text{output}_1$
$\text{input}_2 \rightarrow \text{output}_2$
$\text{input}_3 \rightarrow \text{output}_3$

Supervised learning

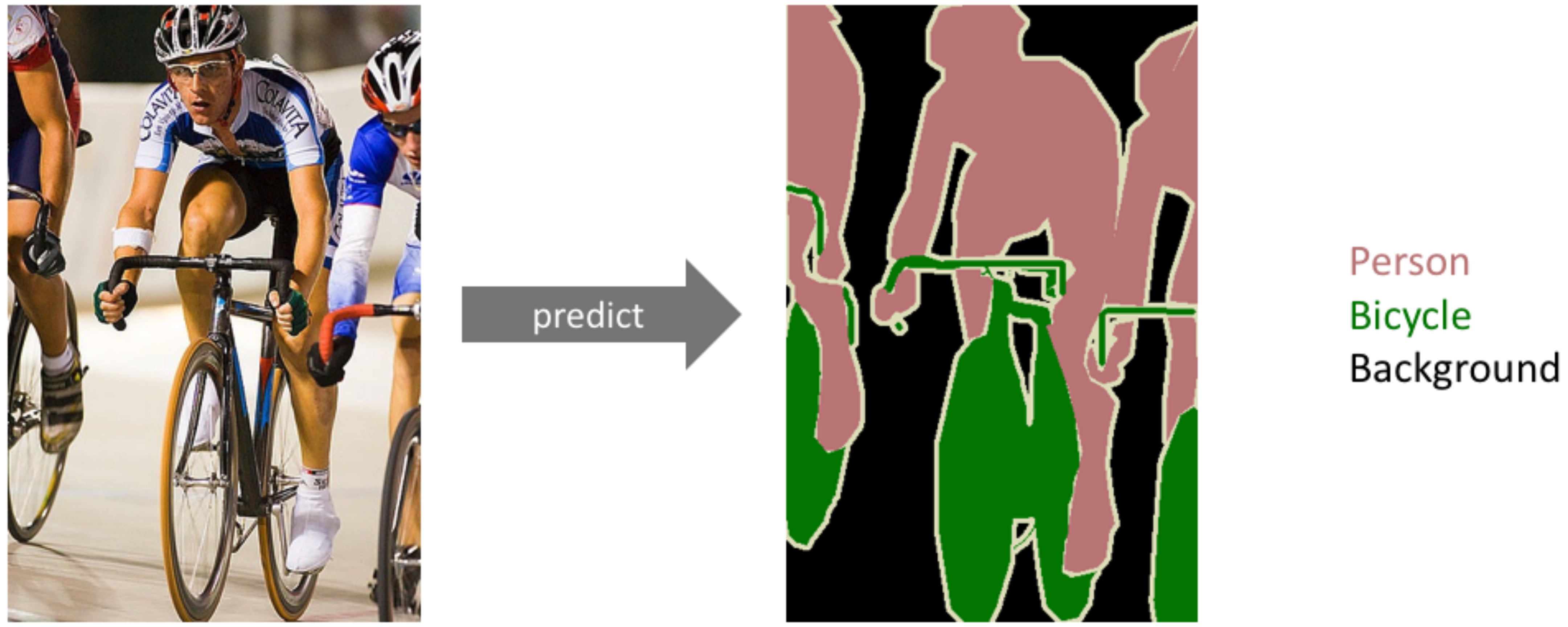
Medicine

- Input: medical information (i.e. age, body mass index [BMI], blood sugar levels, etc.)
- Output: blood pressure (in mm Hg)

Supervised learning

Computer vision: segmentation

- Input: an image
- Output: segmentation map of objects

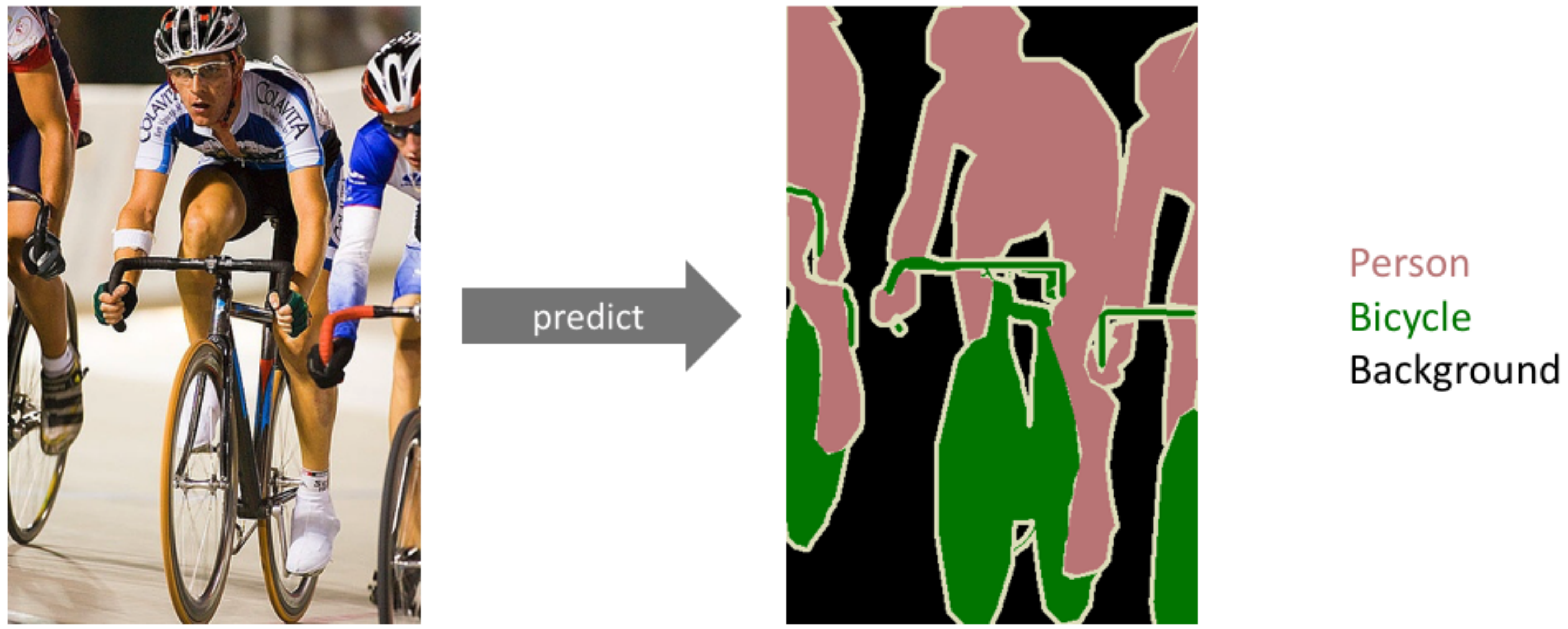


Pascal VOC dataset
http://host.robots.ox.ac.uk/pascal/VOC/voc2012/

Supervised learning

Computer vision: segmentation

- Input: an image
- Output: segmentation map of objects
- **Application: self-driving cars**



Pascal VOC dataset
http://host.robots.ox.ac.uk/pascal/VOC/voc2012/

https://github.com/meetps/pytorch-semseg

Supervised learning

NLP: Machine translation

- Input: a sentence in English
- Output: its translation in Spanish
- **Application: Google Translate**

Gender Bias in Machine Translation: https://towardsdatascience.com/gender-bias-in-machine-translation-819ddce2c452

"That river is dangerous to swim in." → "Es peligroso nadar en ese río."

"The baby is playing with some toys." → "El bebé juega con algunos juguetes."

Anki Tab-Delimited Bilingual Sentence Pairs
https://www.manythings.org/anki/

Supervised learning

Other applications

- Spam filtering: email text → {spam, not spam}
- Speech recognition: audio clip → text transcription
- Medical imaging: CT scan of COVID patients →
  severity of COVID symptoms
- …

If you can create a dataset of (input, output) pairs, you can use supervised learning.

Unsupervised learning

Given unlabelled data (i.e. no input-output pairs), learn **patterns** from data.

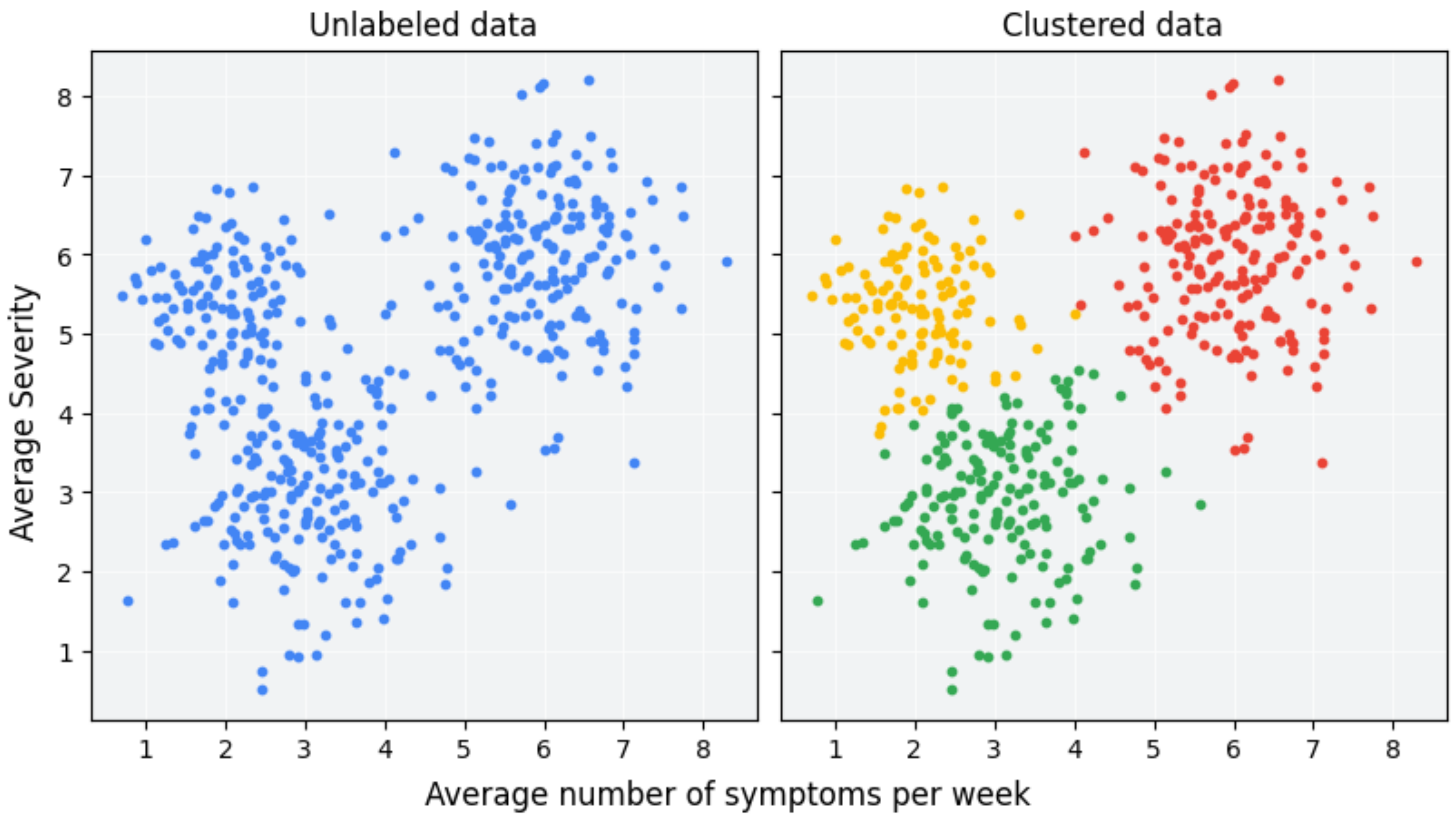
Examples:

- Clustering
- Dimensionality reduction
- Anomaly detection
- Generative modeling
- …

Unsupervised learning

Clustering

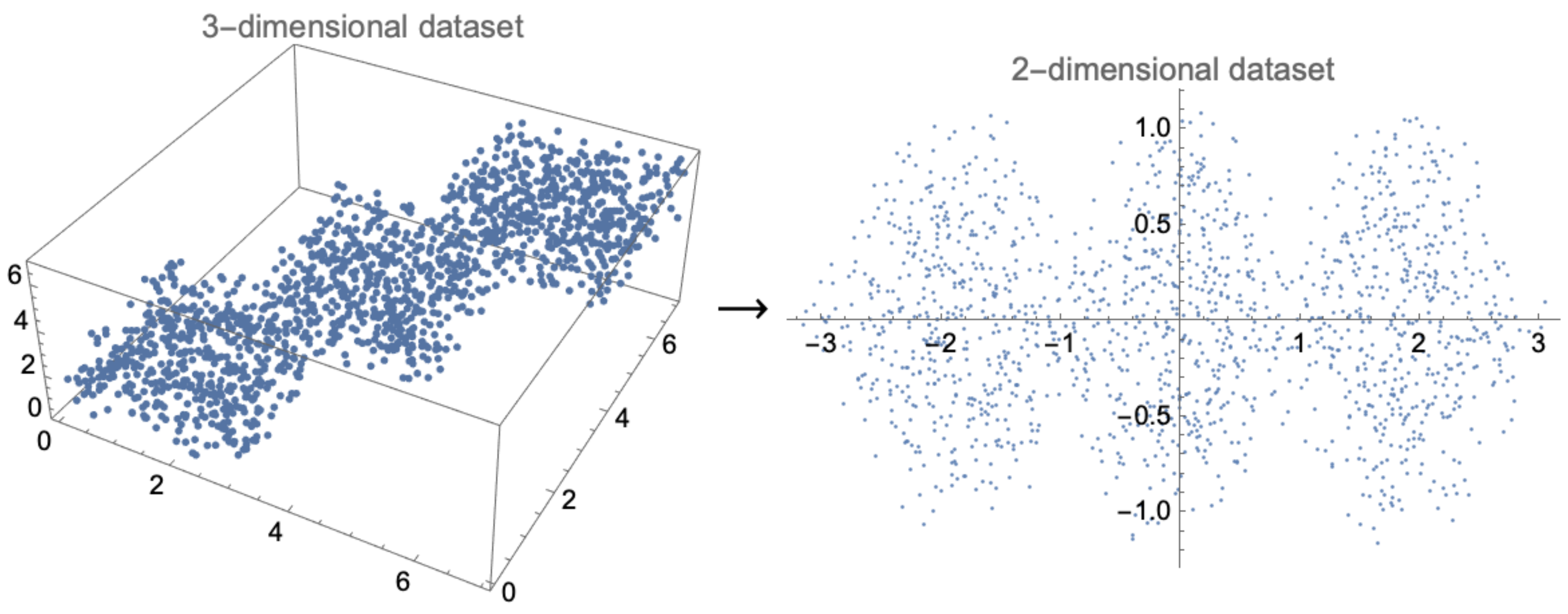
- Goal: Given a data point, categorize it into one of k clusters.
- Unlike supervised learning, there are no output labels!
  i.e. no (data point, cluster) pairs
- Application in retail marketing: What kinds of people have similar shopping habits?



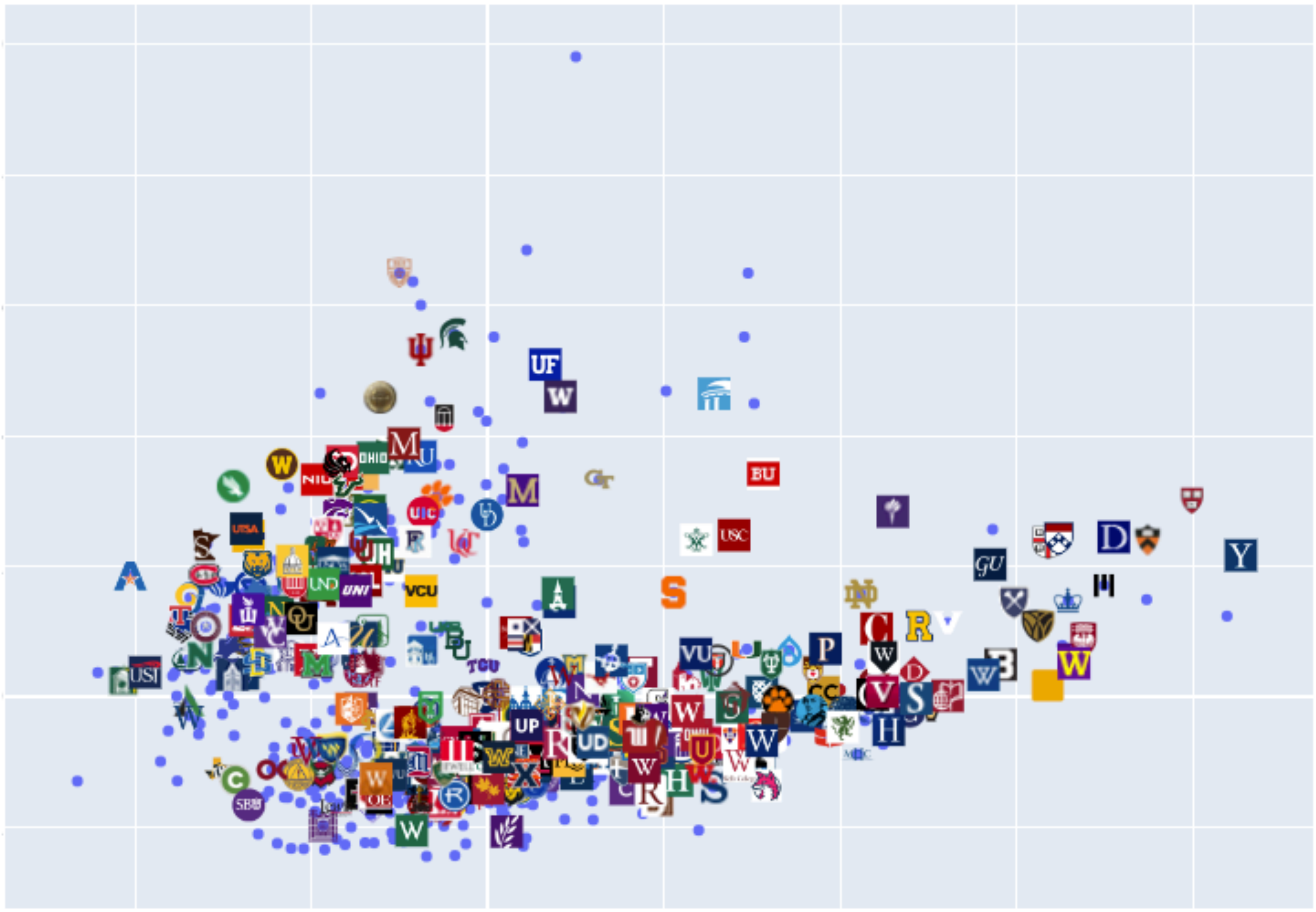
Unsupervised learning

Dimensionality reduction

- Goal: Reduce the number of variables in the data while preserving patterns in the data (e.g. distances between data points).
- Uses:
  - Compression
  - Visualization
  - …



PCA example of colleges dataset

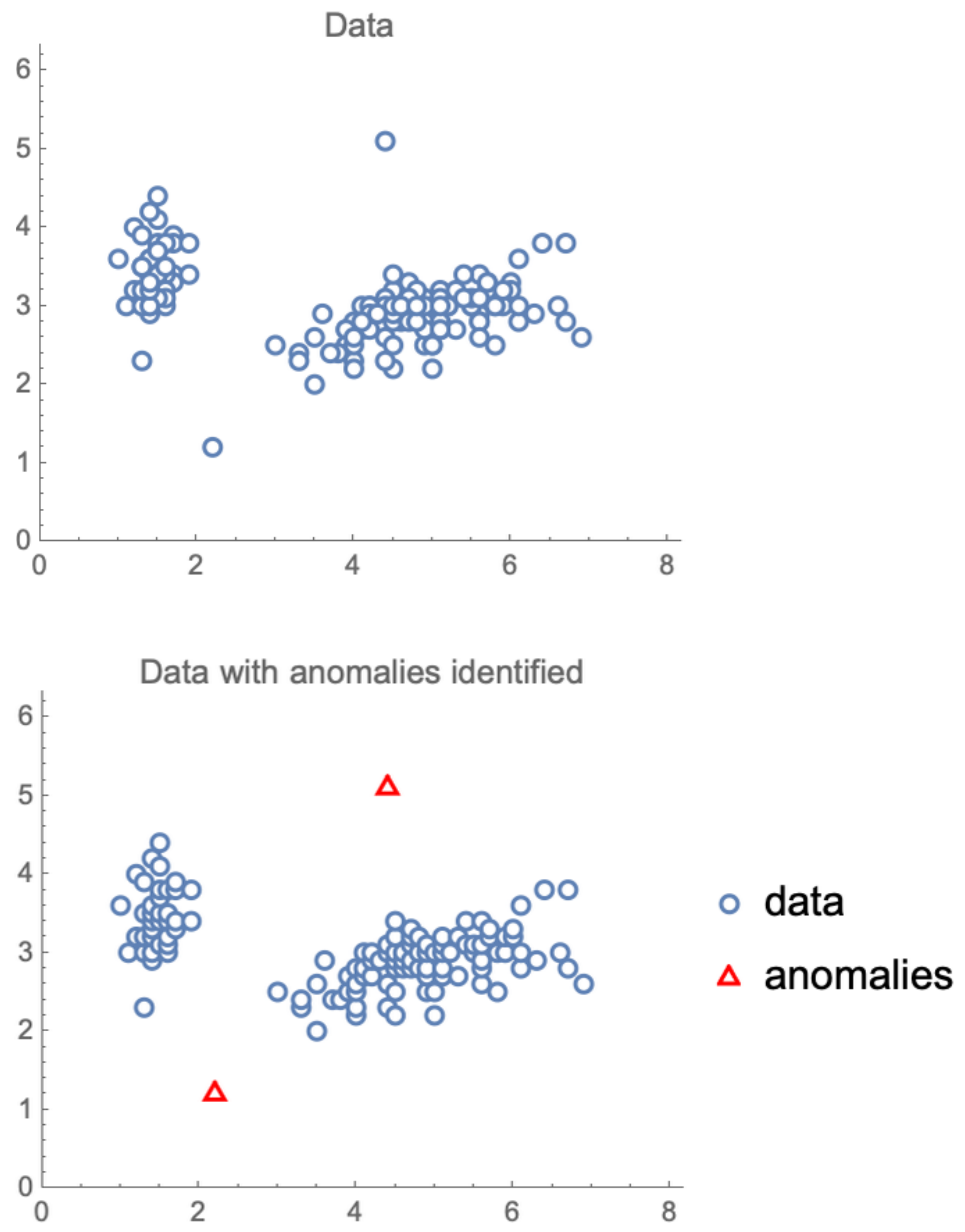


https://www.ruthfong.com/cos324/pca_colleges.html

Unsupervised learning

Anomaly detection

- Goal: Identify anomalies (i.e. outliers) in a dataset.
- Applications:
  - Credit card fraud
  - Medical diagnosis
  - …



Unsupervised learning

Generative model

- Goal: Generative synthetic examples that look realistic.
- Applications:
  - Computer vision: Given a dataset of human face images, generate realistic looking faces.
  - NLP: Given a dataset of Shakespeare's sonnets, generate a realistic sounding, Shakespearean sonnet.

Generated by StyleGAN

(Karras, et al., CVPR 2019)

"am of my faults thy sweet self dost deceive:
they look into the beauty of thy deeds
nativity, once in the chronicle of wasted time
not from the thing they see"

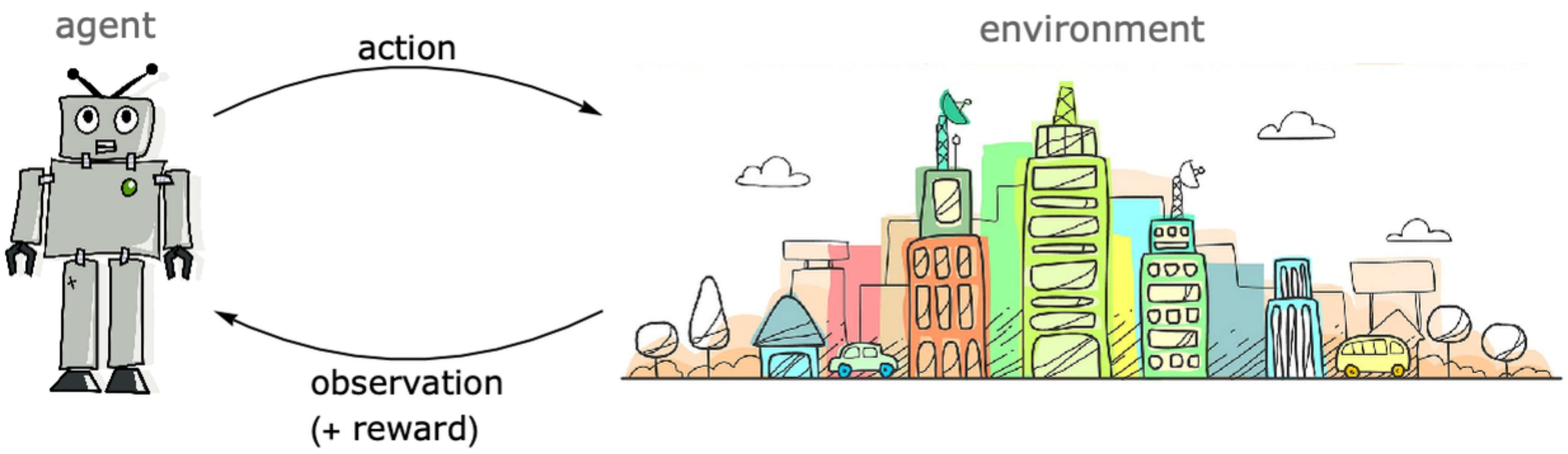
Generated by Markov Chains

https://rpubs.com/malcolmbarrett/shakespeare

Reinforcement learning

Learn by **interacting** (i.e. **trial-and-error**) with an *environment.*

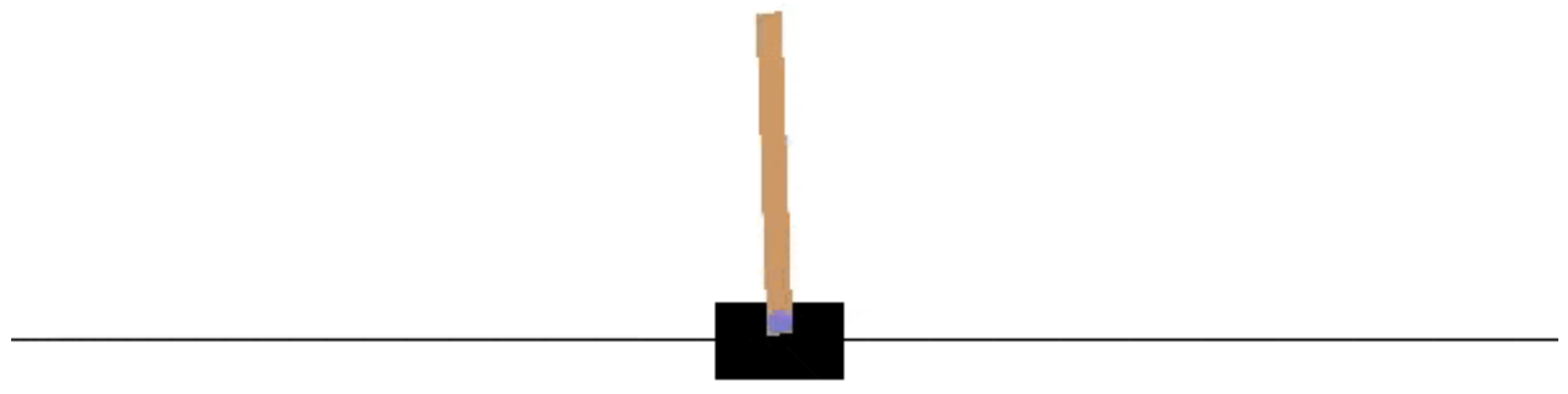
- An *agent* interacts with an *environment* by performing *actions* and then gathers *observations* (and sometimes *rewards*) that occur in response to its actions
- Interactions generate data (i.e. no fixed dataset).
- Goal: Learn *agent* to maximize *reward*.



Reinforcement learning

CartPole

- Goal: Balance a pole on a cart by moving it left and right.
- Actions: {left, right}
- Reward: +1, for each step the pole is upright.
- Learn over many attempts (i.e. episodes).

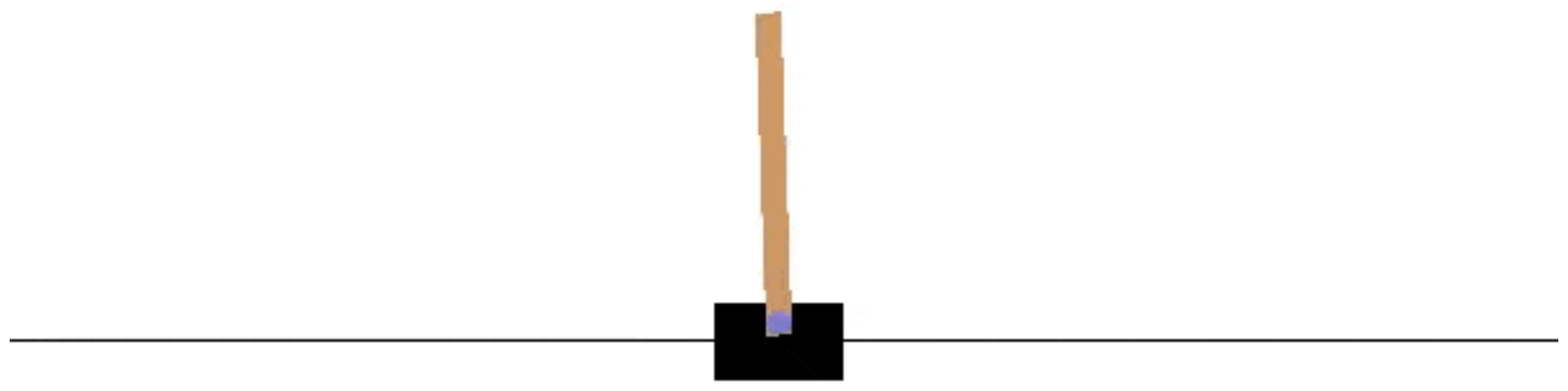


https://gymnasium.farama.org/environments/classic_control/cart_pole/

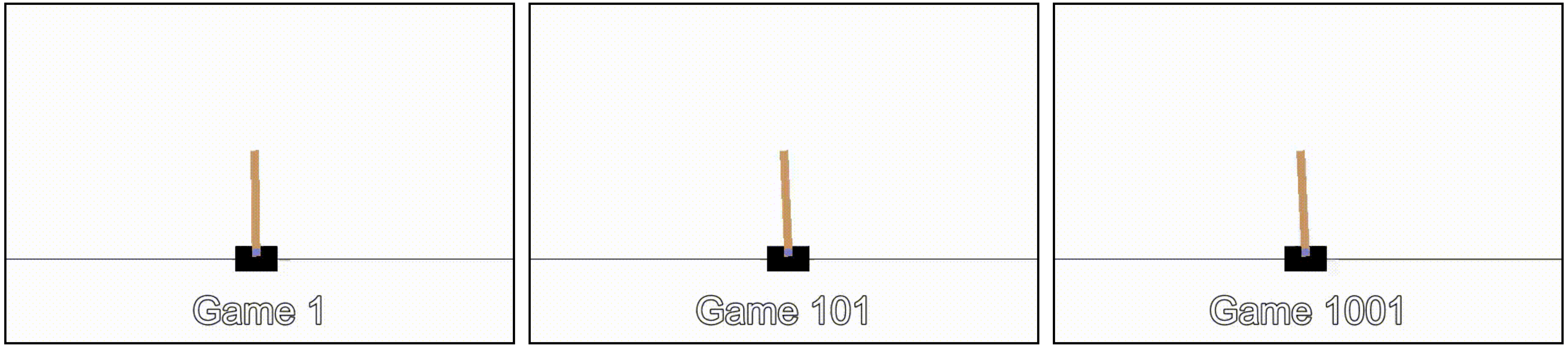
Reinforcement learning

CartPole

- Goal: Balance a pole on a cart by moving it left and right.
- Actions: {left, right}
- Reward: +1, for each step the pole is upright.
- Learn over many attempts (i.e. episodes).



https://gymnasium.farama.org/environments/classic_control/cart_pole/



https://blog.ml6.eu/importance-of-data-visualization-for-ml-936239c355d0

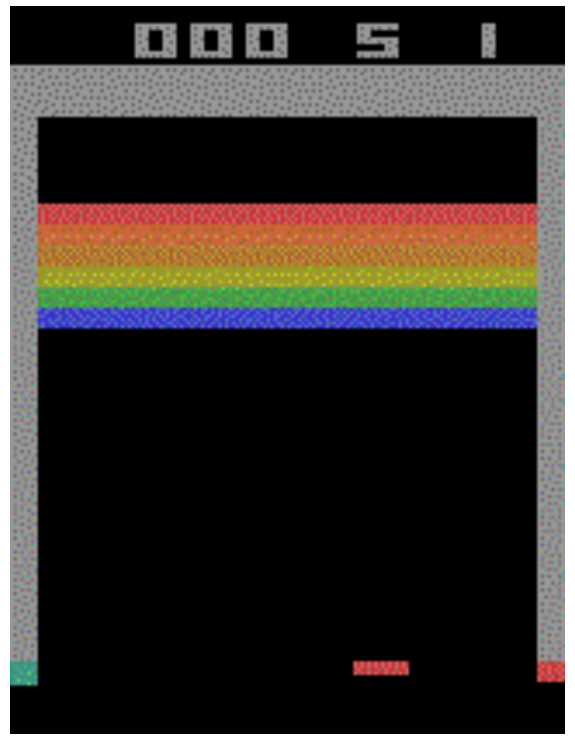
Reinforcement learning

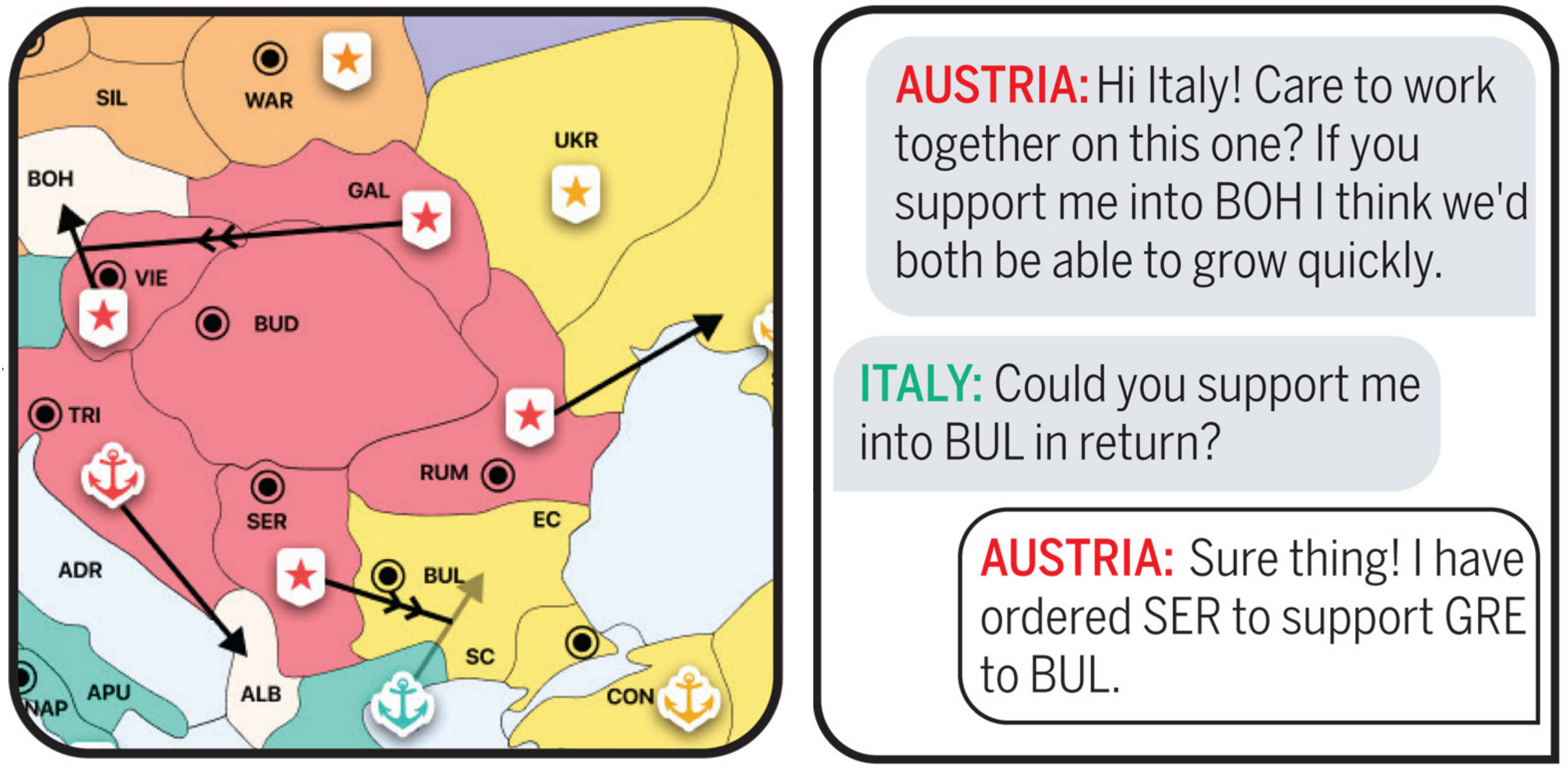
Application: Games

Reinforcement learning

Application: Games

- Atari: DeepMind (2015)
- Go (strategy): AlphaGo from DeepMind (2016)
- Poker (multi-player): Libratus and Pluribus from CMU (2017, 2019)
- Diplomacy (cooperative): Cicero from Meta AI (2022)





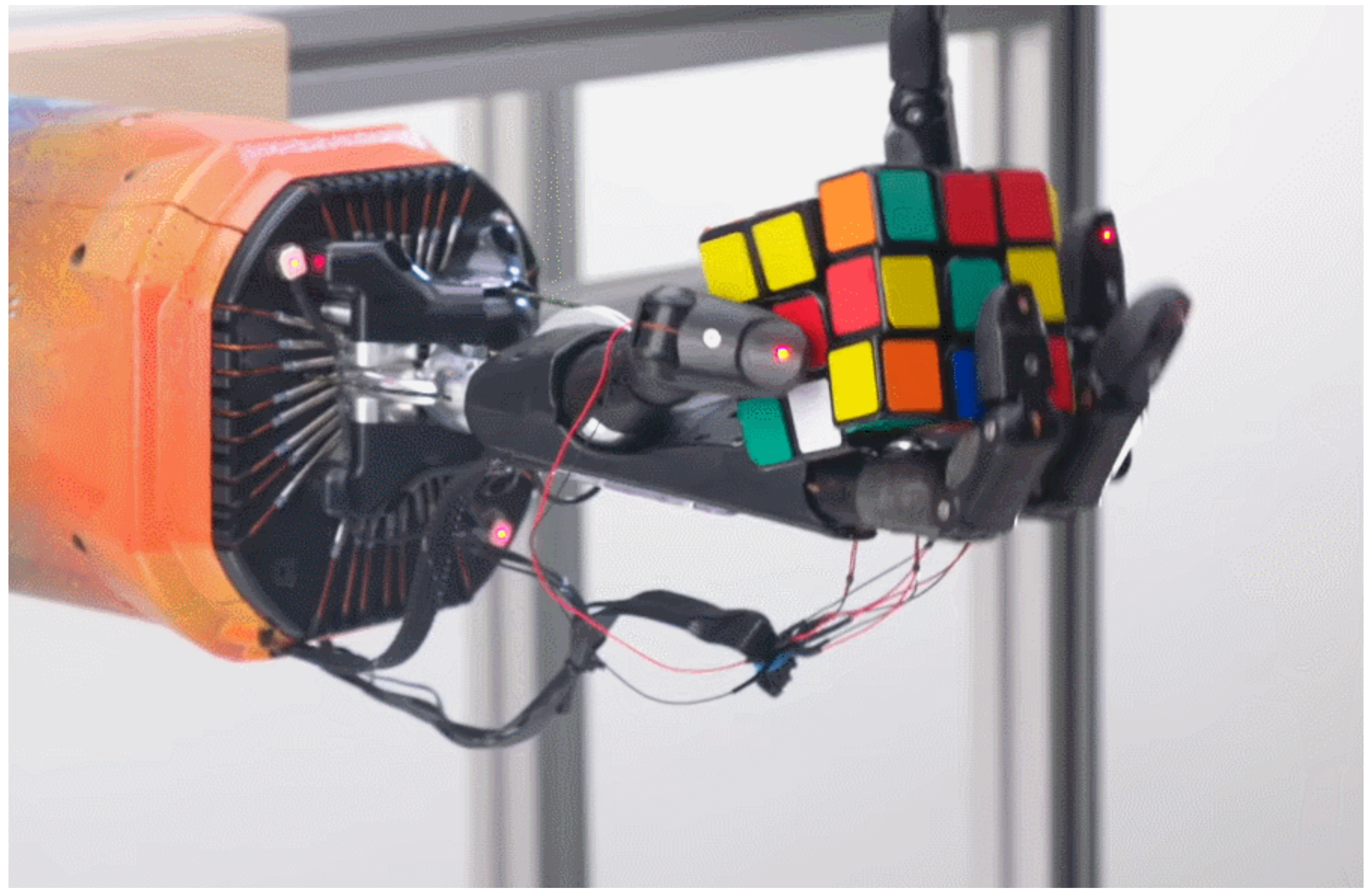
Reinforcement learning

Application: Robotics

Reinforcement learning

Application: Robotics

- Solving a Rubik's cube

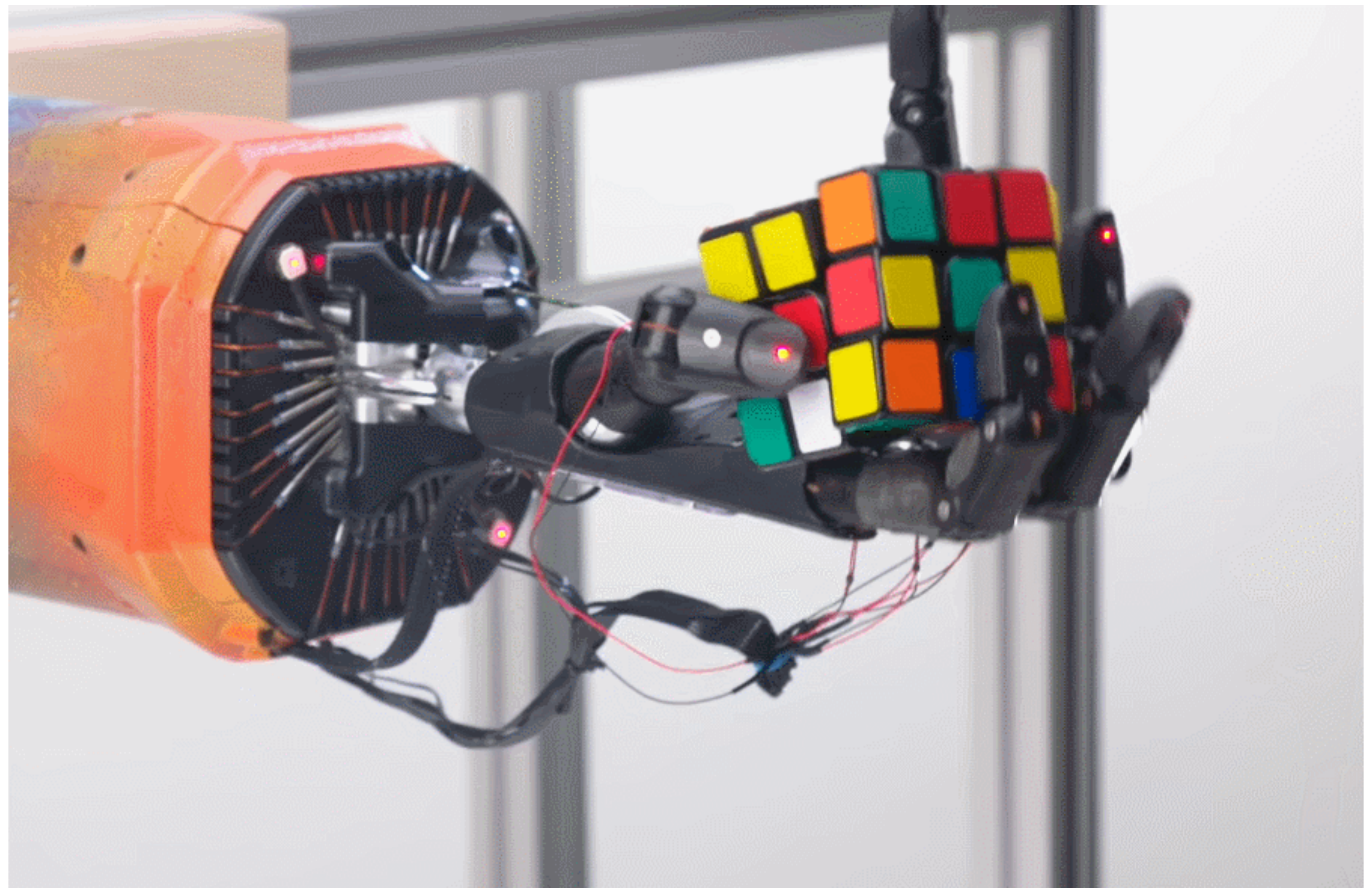


OpenAI's ADR (2019)

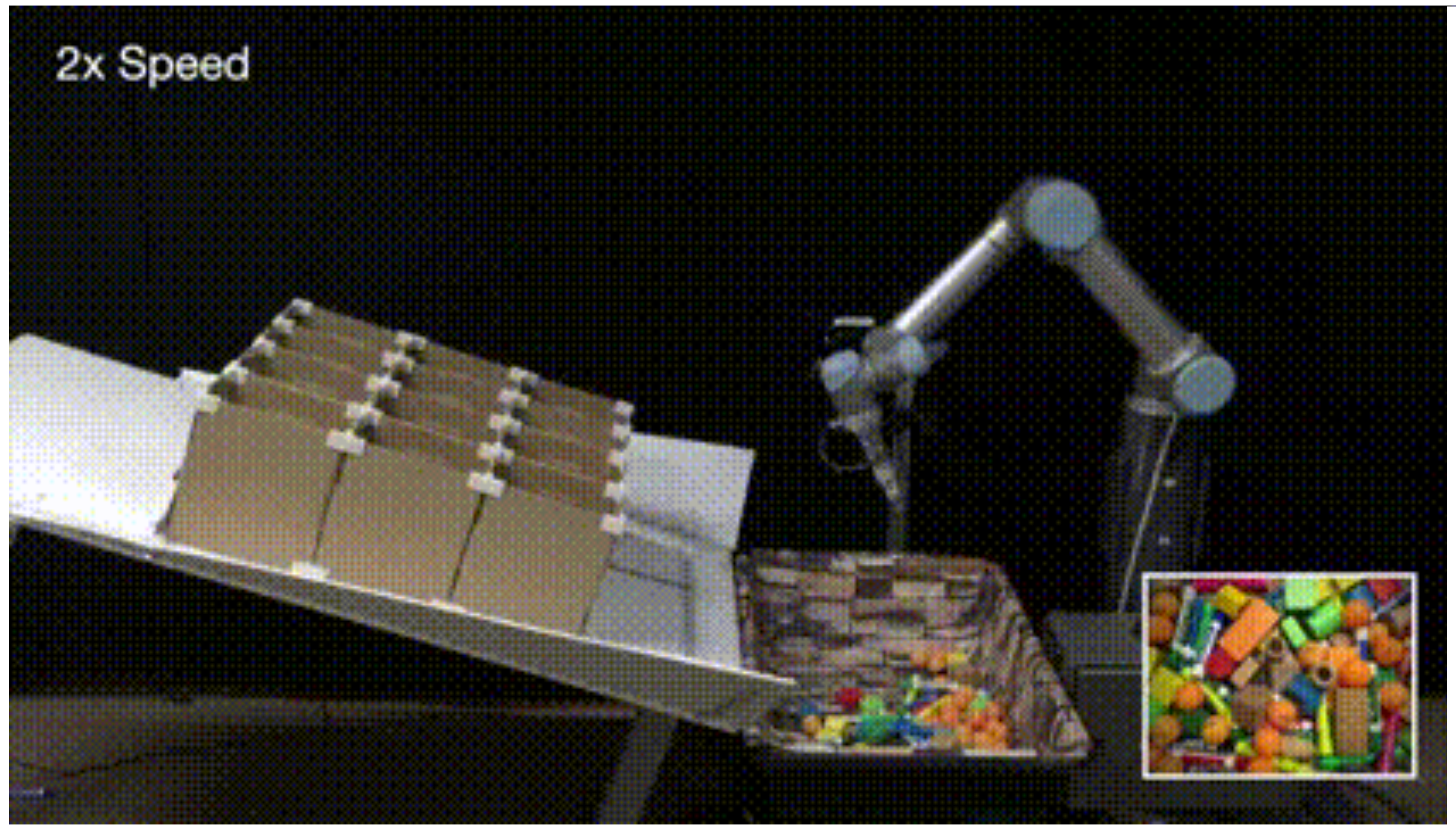
Reinforcement learning

Application: Robotics

- Solving a Rubik's cube
- Manipulating objects (e.g. picking up, throwing, etc.)



OpenAI's ADR (2019)

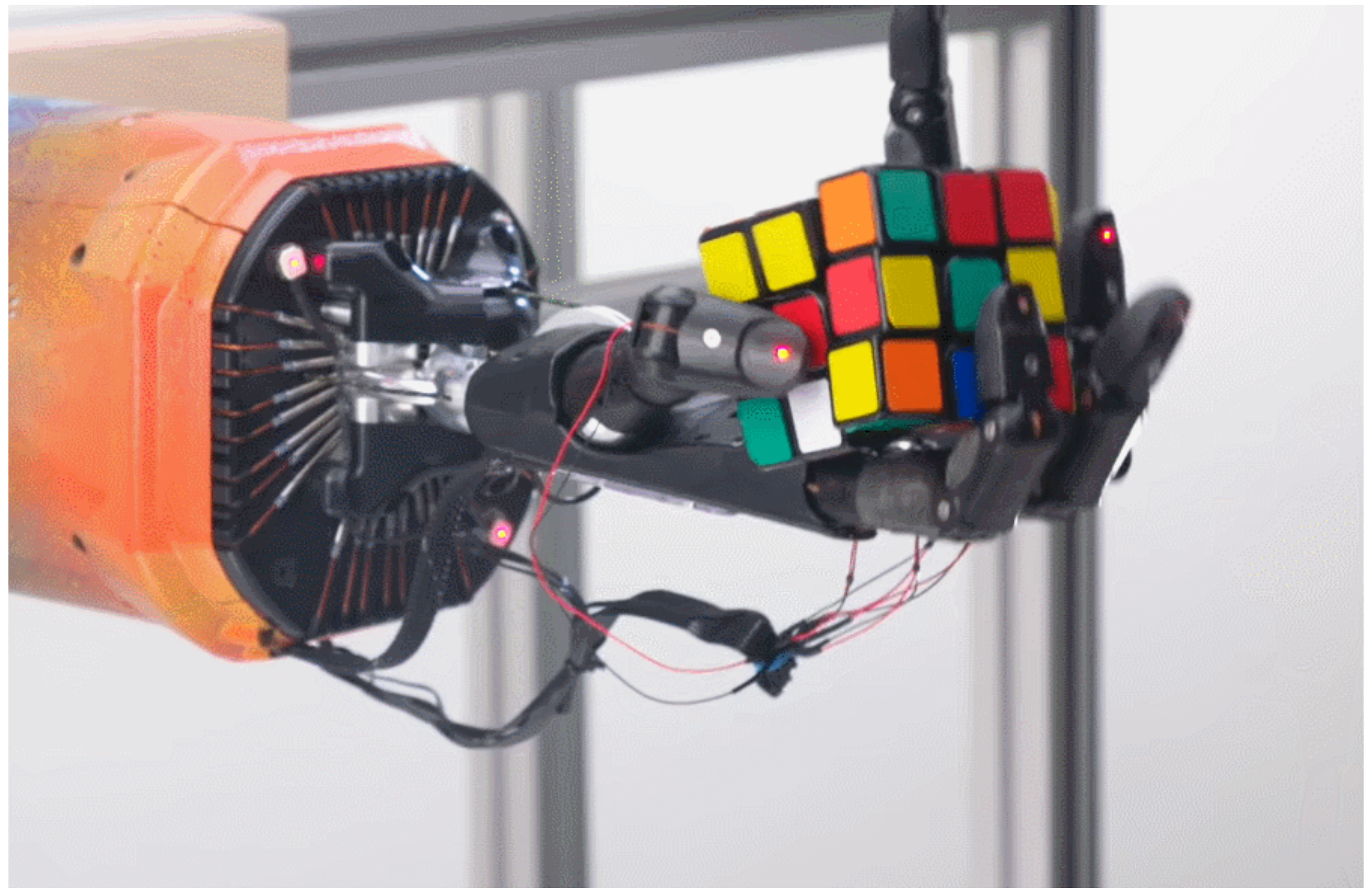


Princeton+Google's TossingBot (2019)

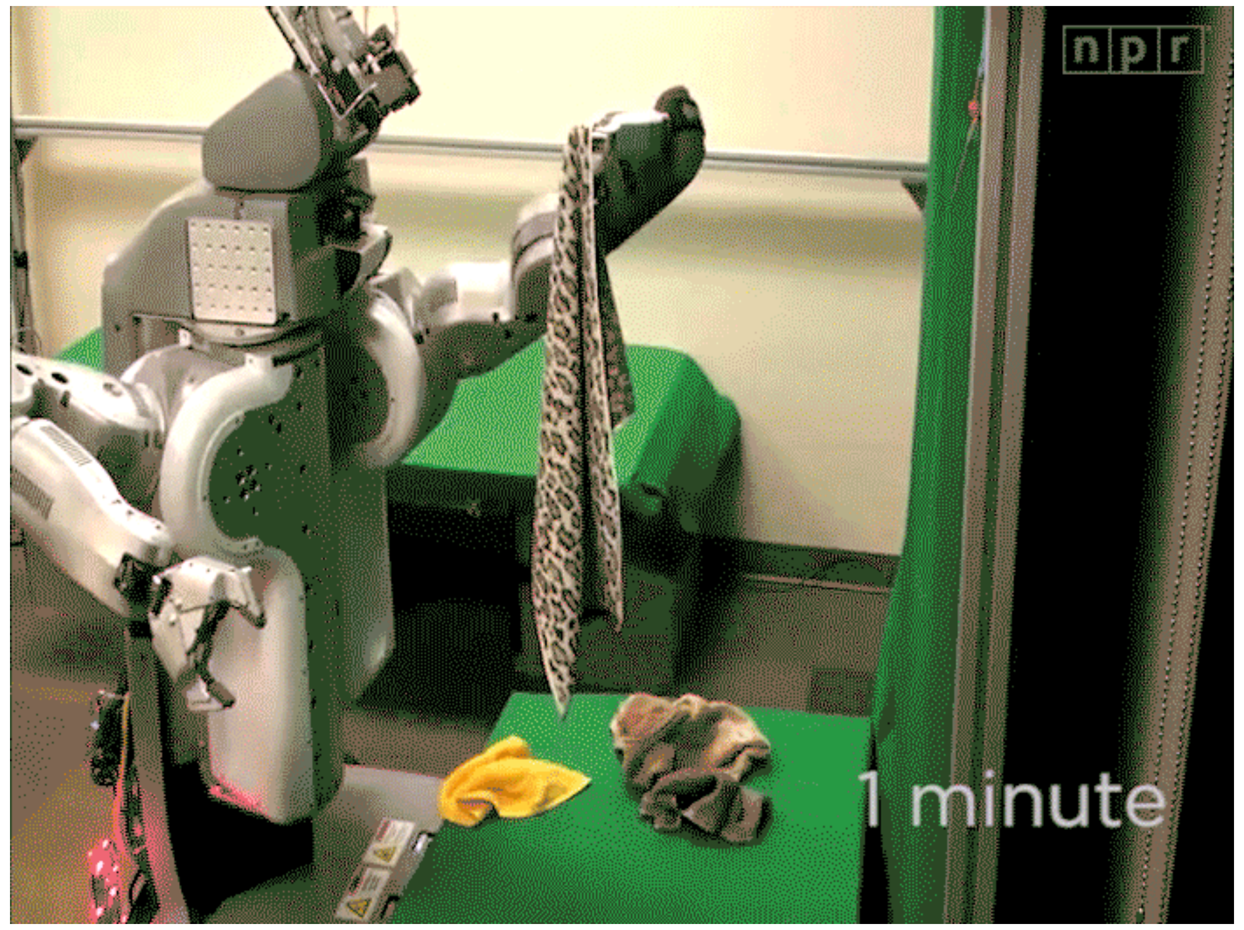
Reinforcement learning

Application: Robotics

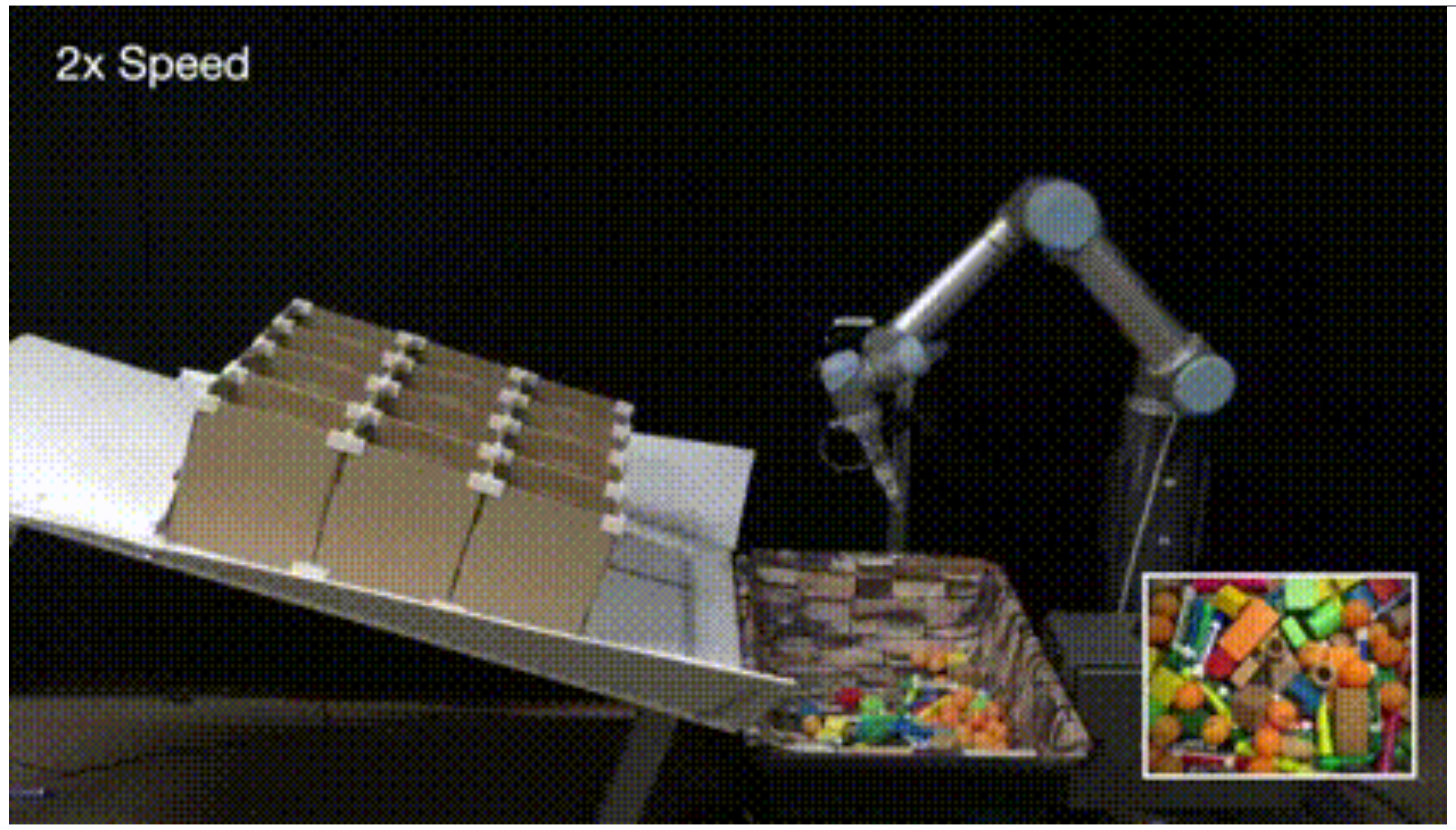
- Solving a Rubik's cube
- Manipulating objects (e.g. picking up, throwing, etc.)
- Laundry folding



OpenAI's ADR (2019)



Berkeley's BRETT (2015)



Princeton+Google's TossingBot (2019)

Poll

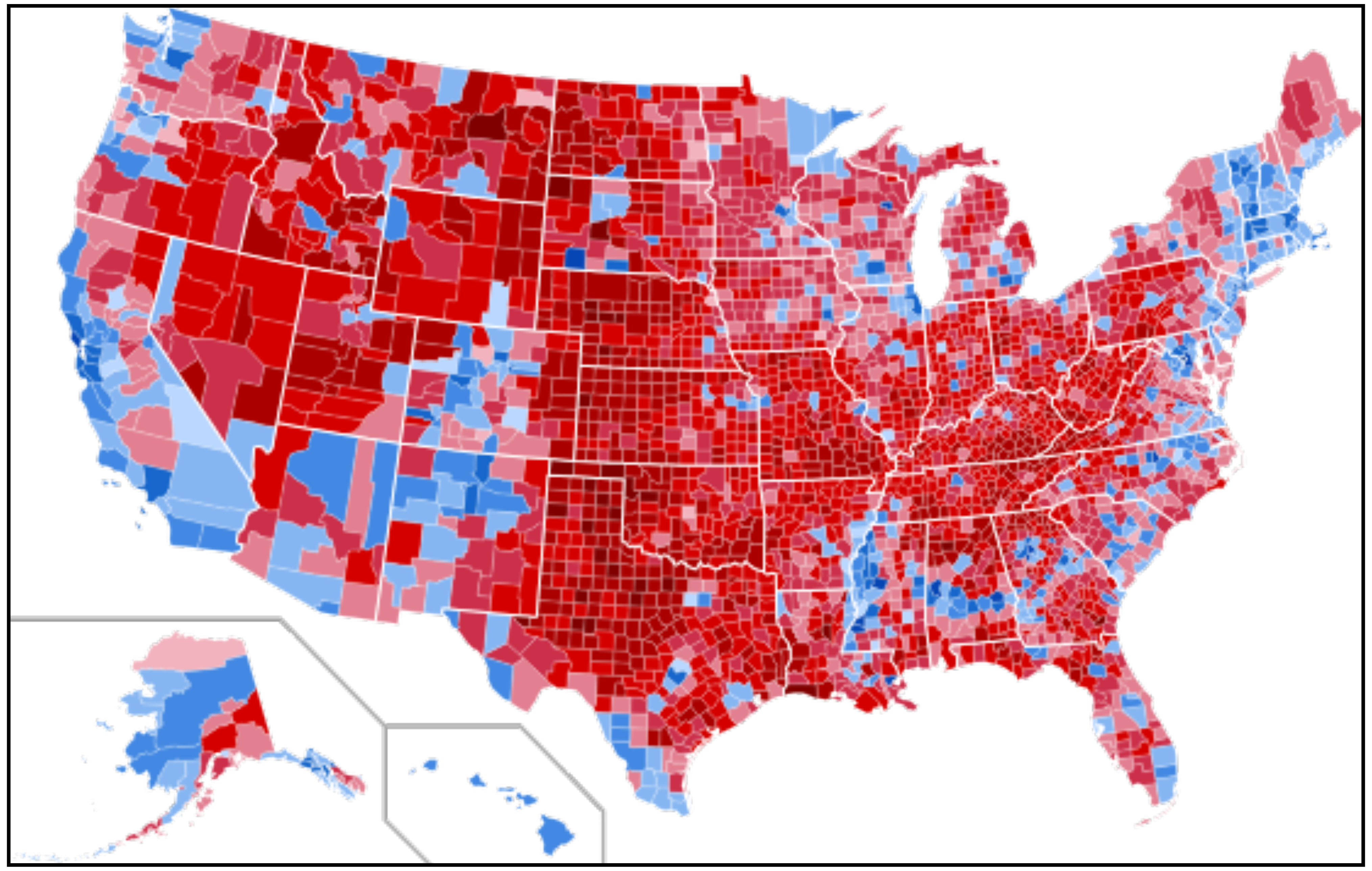
I would like to predict the outcome of the 2024 U.S. presidential election based on polling data leading up to the election and data from prior presidential elections.

What kind of machine learning problem is this?

a. Supervised learning

b. Unsupervised learning

c. Reinforcement learning

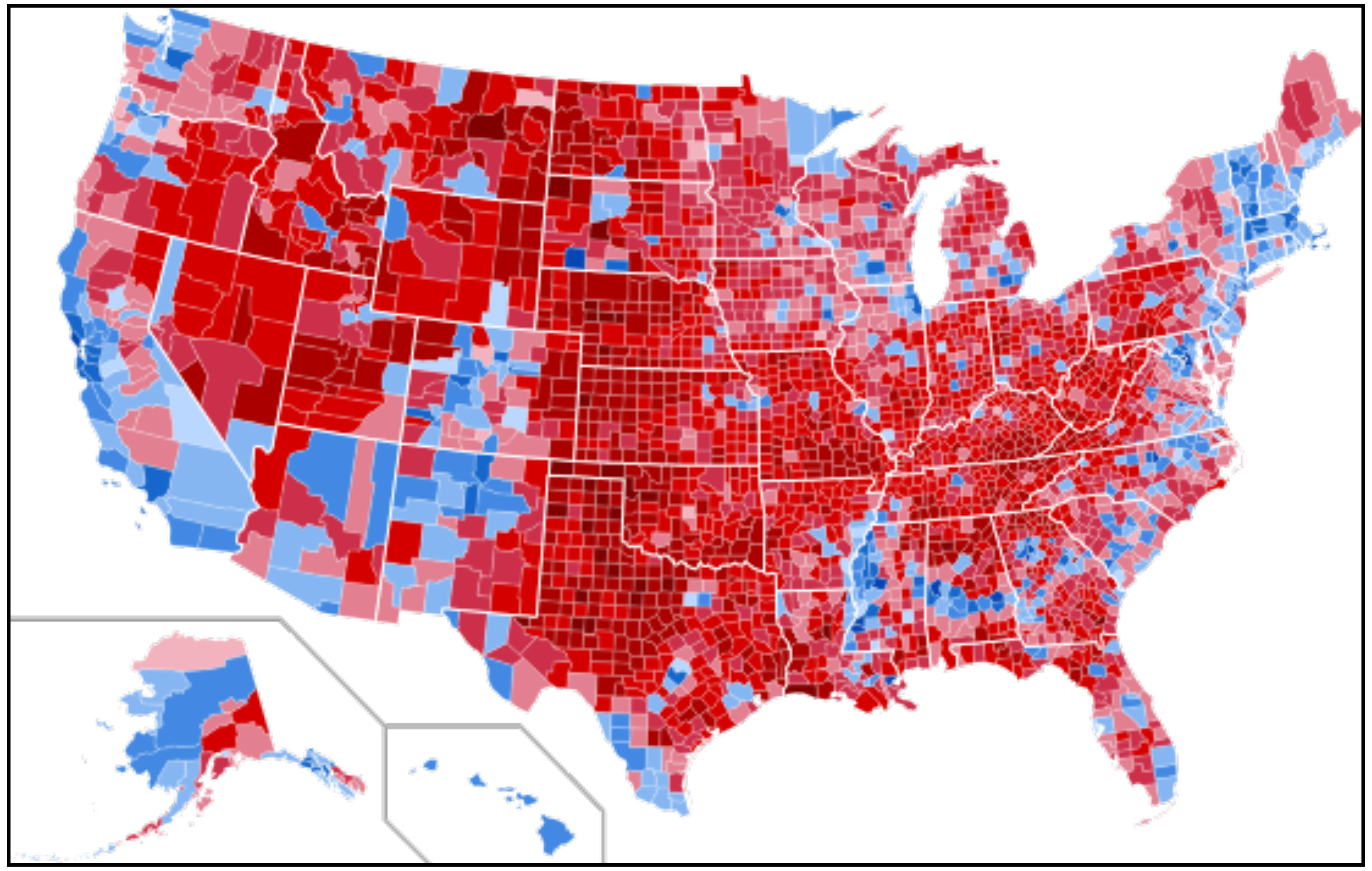


https://commons.wikimedia.org/wiki/File:2020_United_States_presidential_election_results_map_by_county.svg

Poll

I would like to predict the outcome of the 2024 U.S. presidential election based on polling data leading up to the election and data from prior presidential elections.

What kind of machine learning problem is this?



https://commons.wikimedia.org/wiki/File:2020_United_States_presidential_election_results_map_by_county.svg

Poll

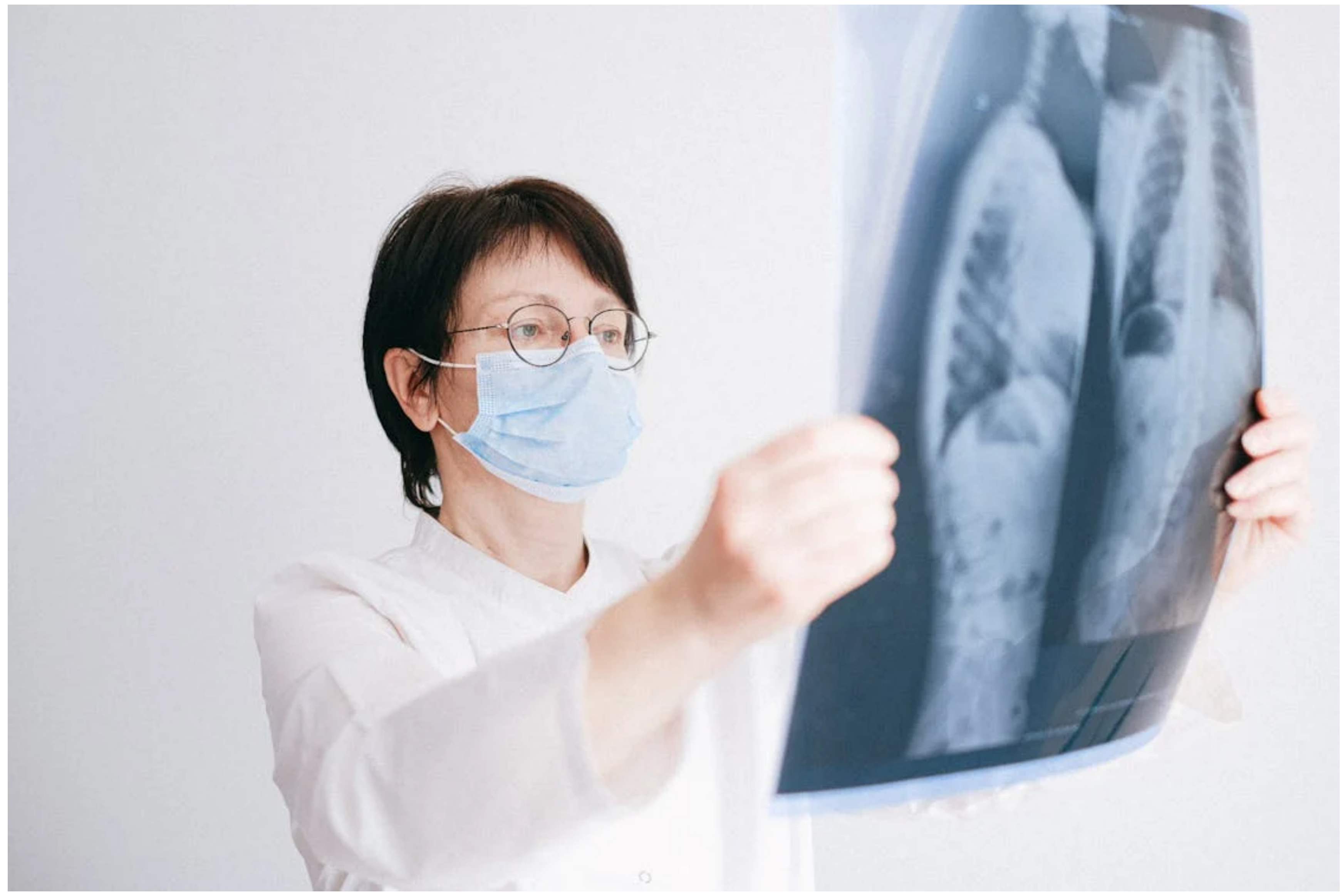
Given a dataset of lung scans of patients, I would like to predict if a new lung scan is abnormal.

What kind of machine learning problem is this?

a. Supervised learning

b. Unsupervised learning

c. Reinforcement learning

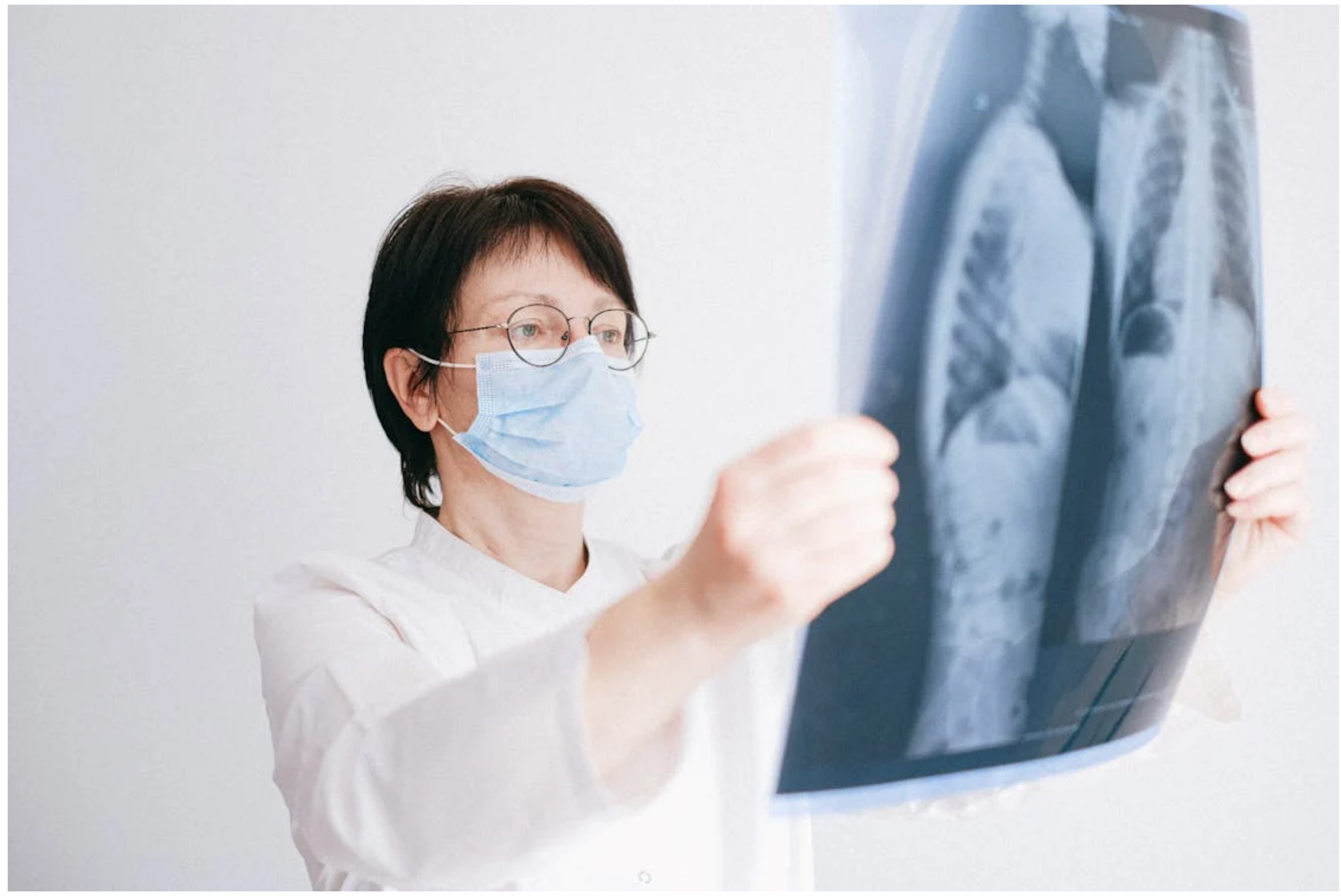


https://www.pexels.com/photo/doctor-looking-at-lung-scans-she-is-holding-4225926/

Poll

Given a dataset of lung scans of patients, I would like to predict if a new lung scan is abnormal.

What kind of machine learning problem is this?



https://www.pexels.com/photo/doctor-looking-at-lung-scans-she-is-holding-4225926/

Machine learning paradigms

1. Supervised learning: learn to **predict** an output from an input.
2. Unsupervised learning: learn **patterns** from data.
3. Reinforcement learning: learn by **interacting** (i.e. **trial-and-error**) with an environment.

Other paradigms:

- *Self-supervised learning*
- *Semi-supervised learning*
- *Transfer learning*
- *Online learning*
- *…*

Ingredients of ML

1. **Data.** Can be labelled, unlabelled, or from experience.
2. **Model.** A model maps from datapoint to a desired answer or output.
3. **Model parameters**. Internal parameters of model that are learnable.
4. **Training.** Given datapoints, find good model parameters.
5. **Testing.** Evaluate the performance of learned model on new, previously unseen data (i.e. test set).

Perceptron

Supervised learning

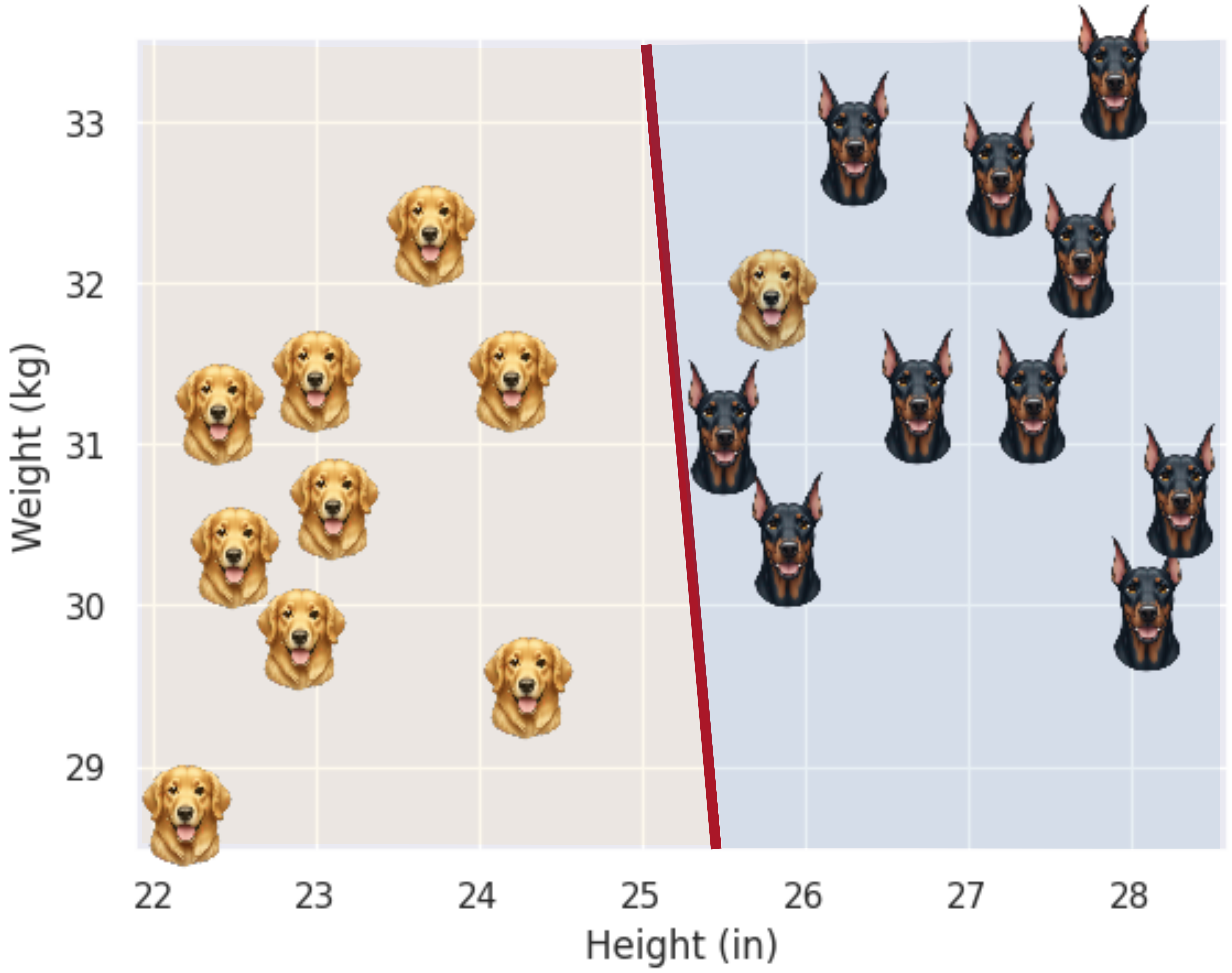
Given (input, output) pairs, **predict** an output from its input.

$\text{input}_1 \rightarrow \text{output}_1$
$\text{input}_2 \rightarrow \text{output}_2$
$\text{input}_3 \rightarrow \text{output}_3$

Basic problems described by output type:

- **Regression**: Predict a continuous value (e.g. number).
- **Classification**: Predict a discrete category (i.e. classes).
  - **Binary classification**: Predict one of **two** classes.
  - **Multi-class classification**: Predict one of **k** classes (k > 2).

Binary classification

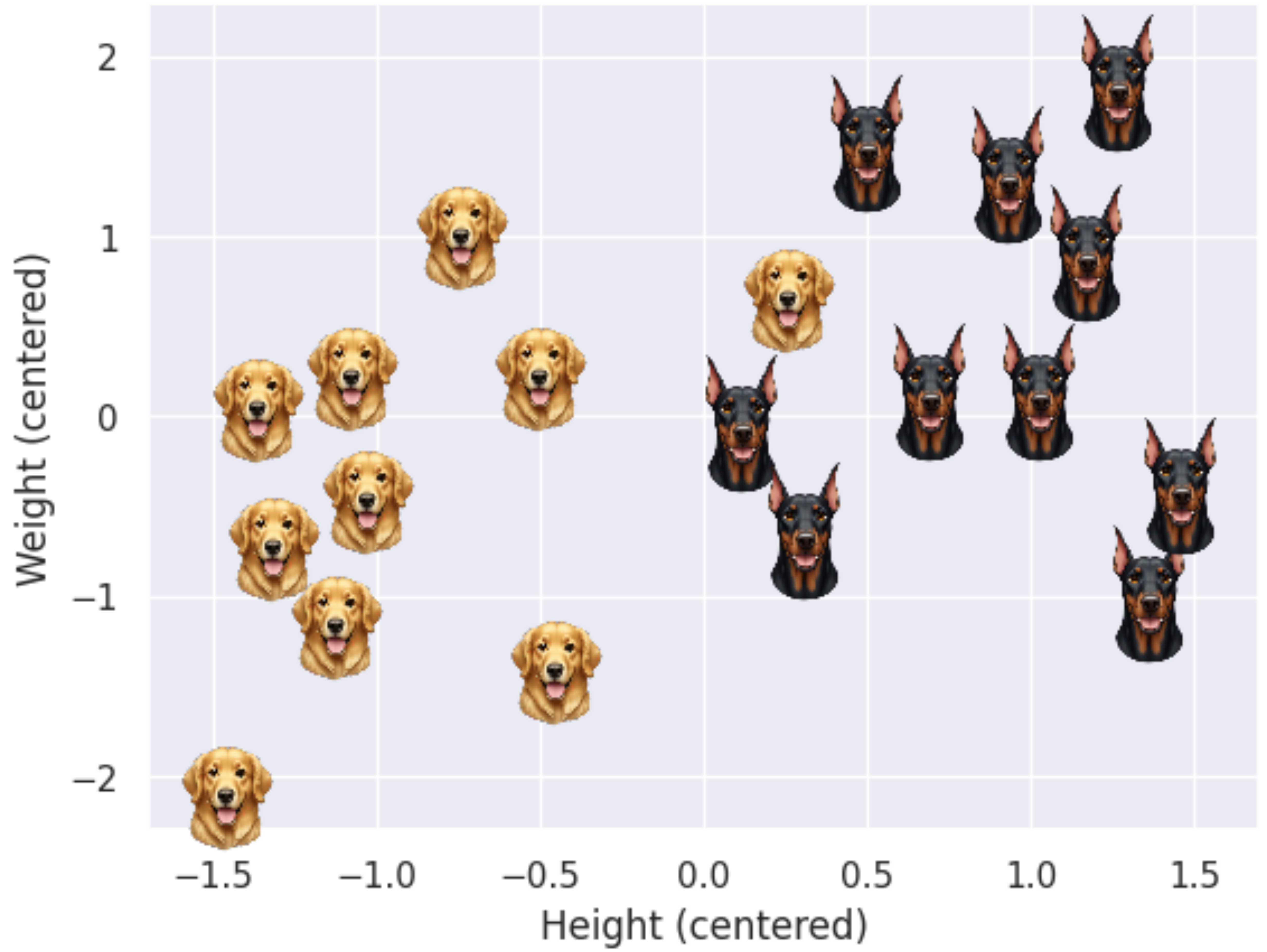


Data is nearly **linearly separable**.

Binary classification

Predict dog species

- Input (centered*)
  - Height
  - Weight
- Output
  - Golden Retriever: $+1$
  - Doberman Pinscher: -1



* Input centered to have mean = 0 and standard deviation = 1.

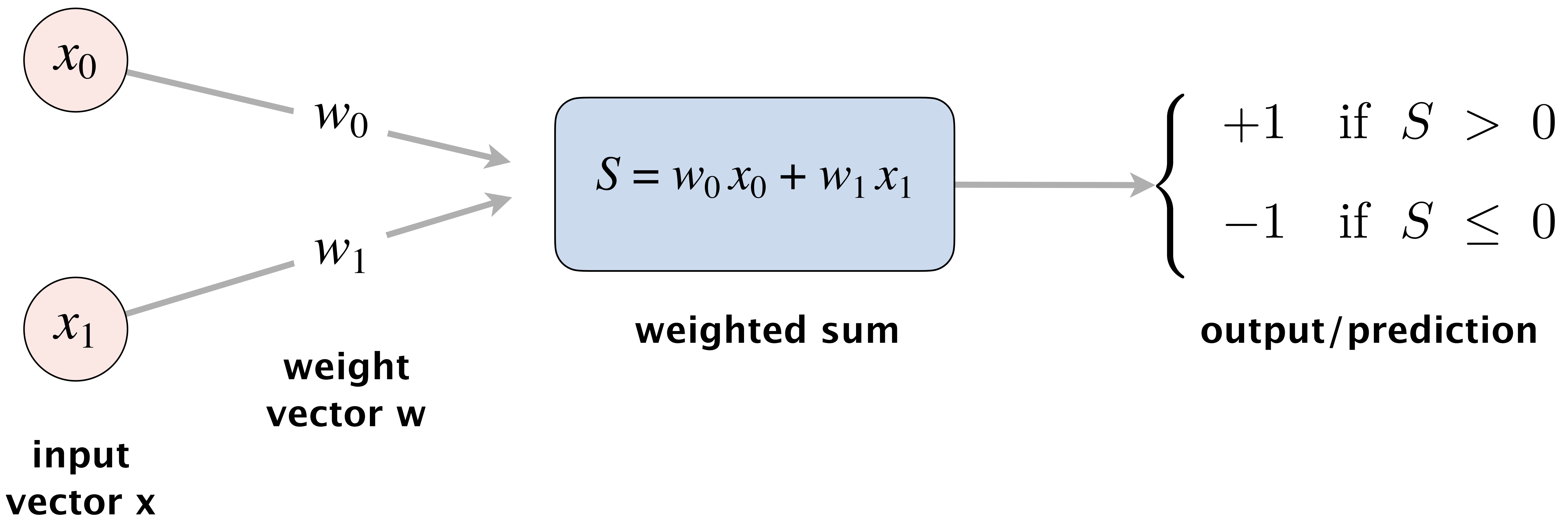
Perceptron

Algorithm that takes several inputs and produces a single, binary output.

- Inputs: x_0, x_1 (i.e. height, weight)

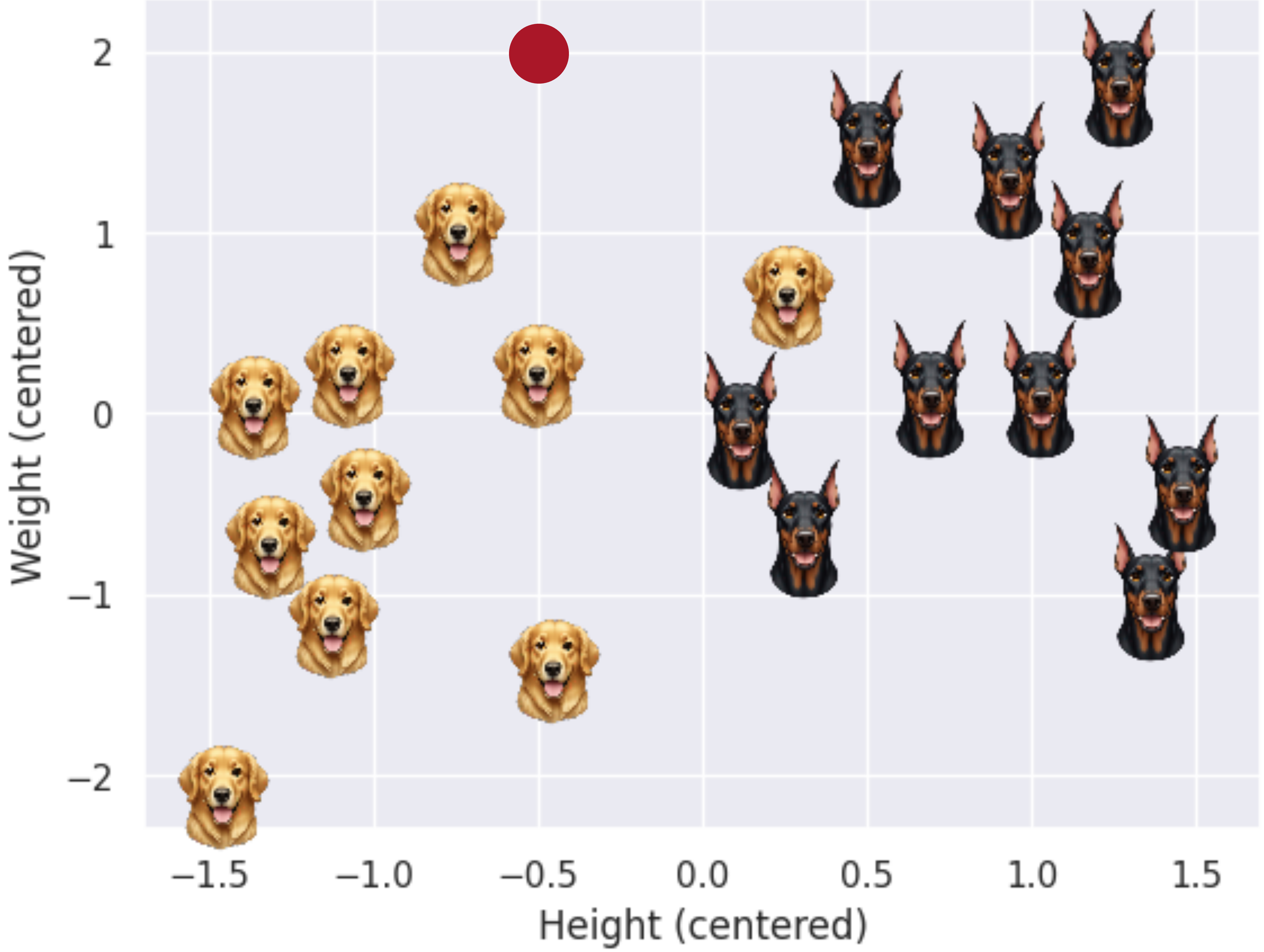
- Compute weighted sum
  $S = w_0 x_0 + w_1 x_1$

- Output
  $\text{sign}(S) = +1$ if $S > 0$
  $= -1$ if $S \leq 0$



Poll

- Inputs: $x_0 = -0.5,\ x_1 = 2$
- Weights: $w_0 = -2,\ w_1 = -0.1$
- Compute weighted sum
  $S = w_0 x_0 + w_1 x_1$
- Output
  $\text{sign}(S) = +1 \text{ if } S > 0$
  $\qquad\qquad = -1 \text{ if } S \leq 0$



Golden retriever: +1, Doberman -1

What is the weighted sum S?

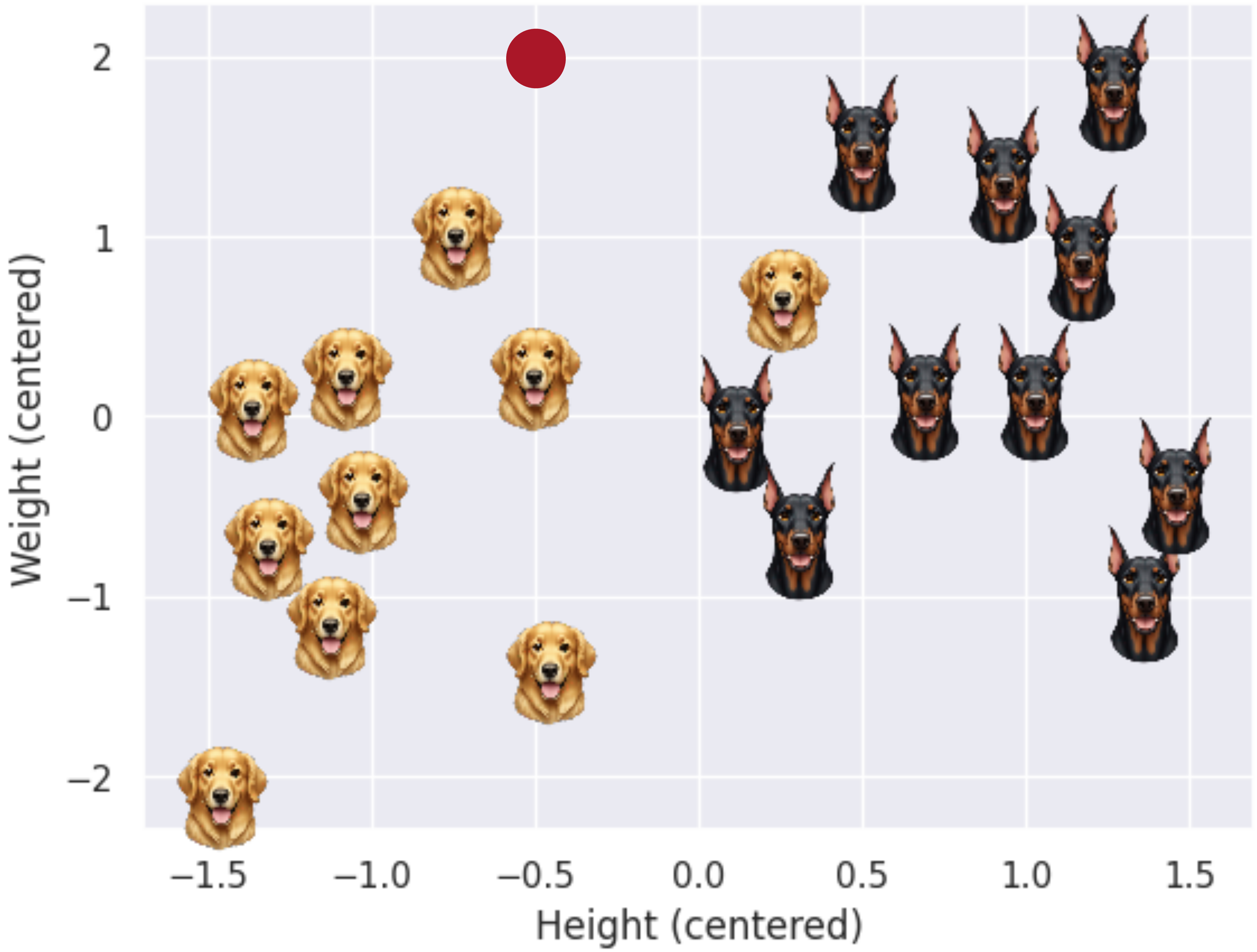
A. -1.2 C. 0.8

B. -0.2 D. 1.0

Poll

- Inputs: $x_0 = -0.5,\ x_1 = 2$
- Weights: $w_0 = -2,\ w_1 = -0.1$
- Compute weighted sum
  $S = w_0 x_0 + w_1 x_1$
- Output
  $\mathrm{sign}(S) = +1$ if $S > 0$
  $= -1$ if $S \leq 0$

What is the weighted sum S?



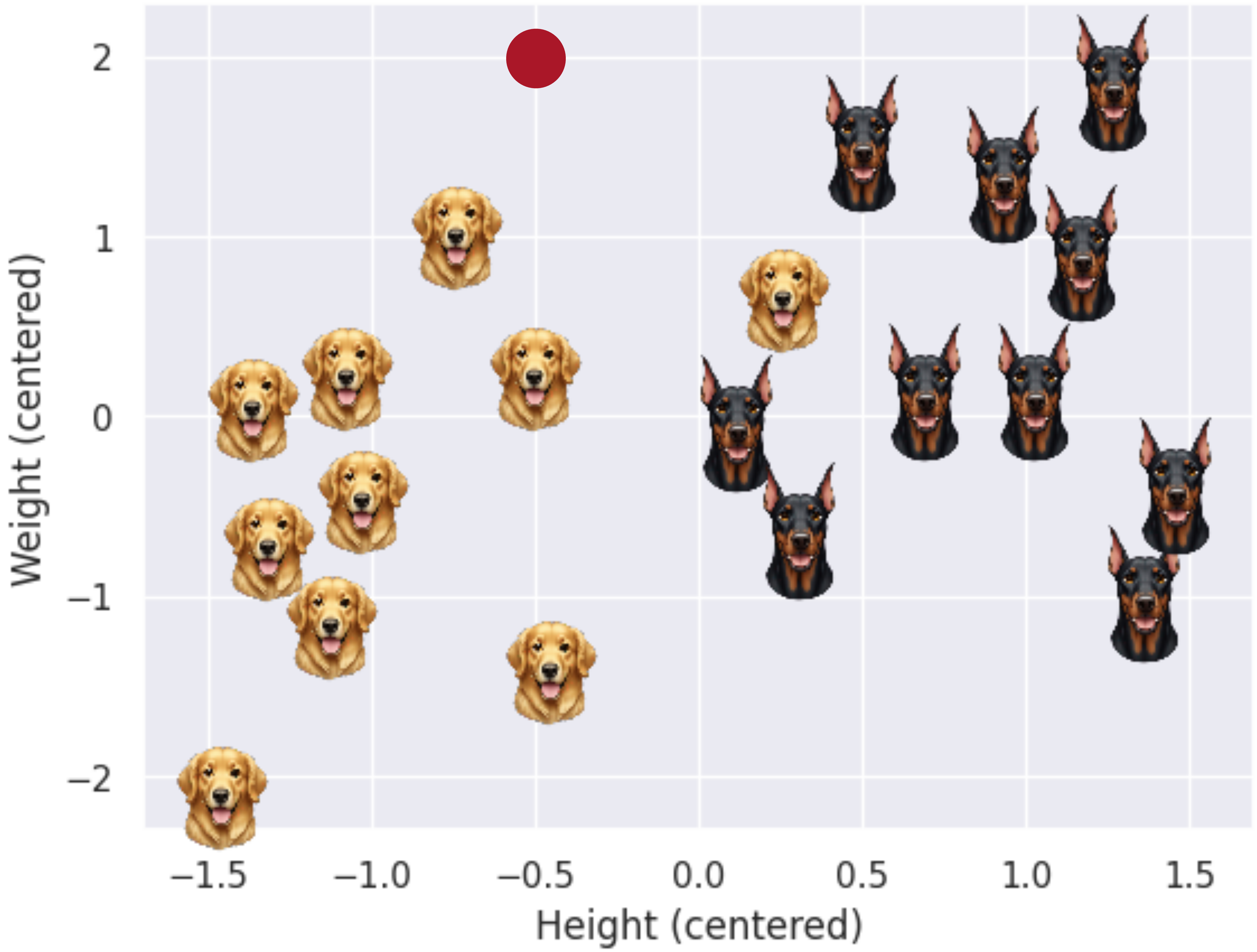
Golden retriever: +1, Doberman -1

Perceptron

- Inputs: $x_0 = -0.5,\ x_1 = 2$
- Weights: $w_0 = -2,\ w_1 = -0.1$
- Compute weighted sum
  $S = w_0 x_0 + w_1 x_1$
- Output
  $\text{sign}(S) = +1$ if $S > 0$
  $= -1$ if $S \leq 0$

$$S = w_0 x_0 + w_1 x_1$$
$$= (-2) \cdot (-0.5) + (-0.1) \cdot 2$$
$$= 1 + (-0.2) = 0.8$$

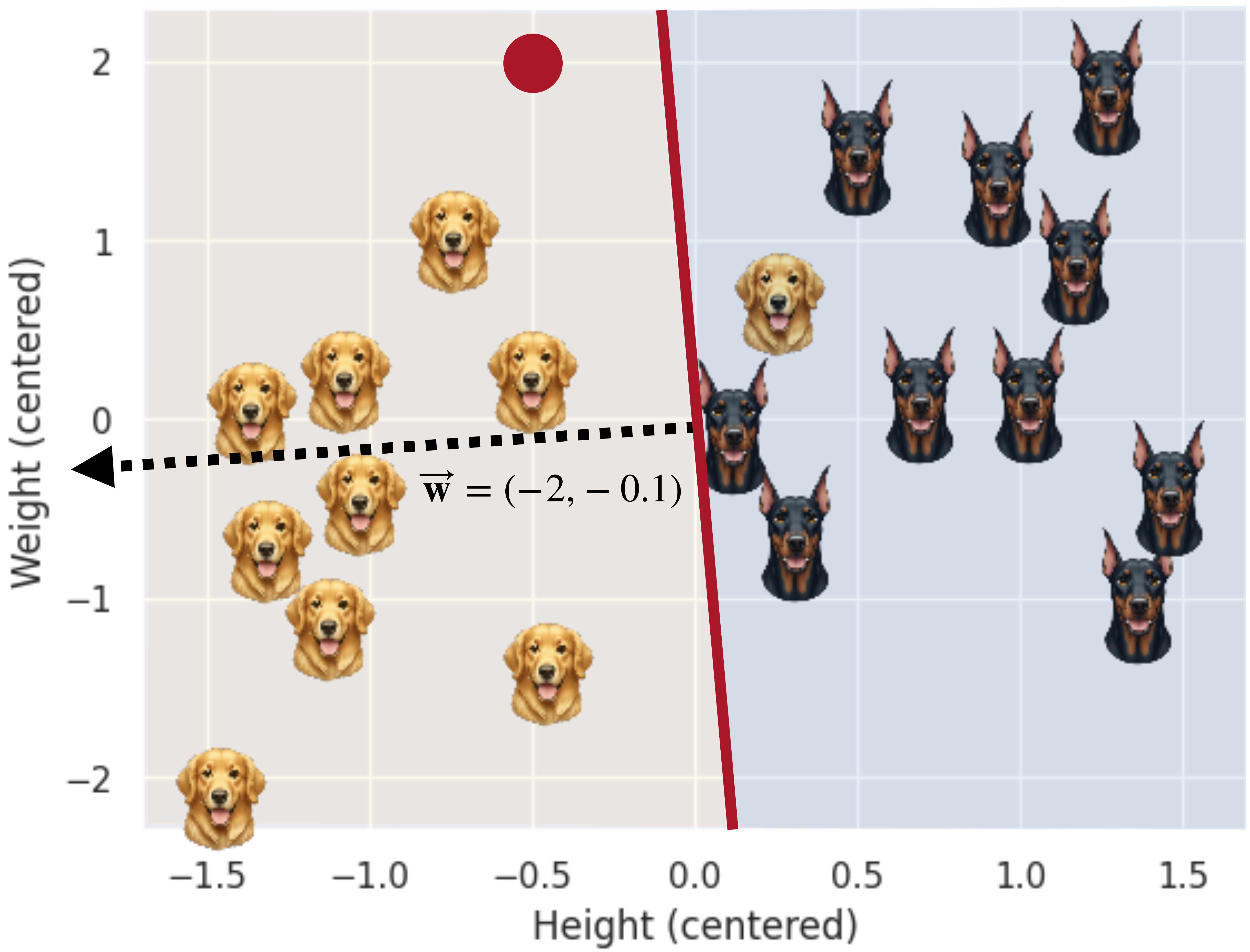
$\text{sign}(S) = \text{sign}(0.8) = +1$



Golden retriever: +1, Doberman -1

Perceptron: Geometric Perspective

- Inputs: $x_0 = -0.5,\ x_1 = 2$
- Weights: $w_0 = -2,\ w_1 = -0.1$
- Compute weighted sum
  $S = w_0 x_0 + w_1 x_1 = 0.8$
- Output
  sign$(S) = +1$
- Weight vector $\vec{\mathbf{w}}$ is **perpendicular** to linear decision boundary.
- Datapoints on the "side" with weight vector $\vec{\mathbf{w}}$ are predicted as $+1$; those on the "opposite site" as -1.



Golden retriever: +1, Doberman -1

Ingredients of ML

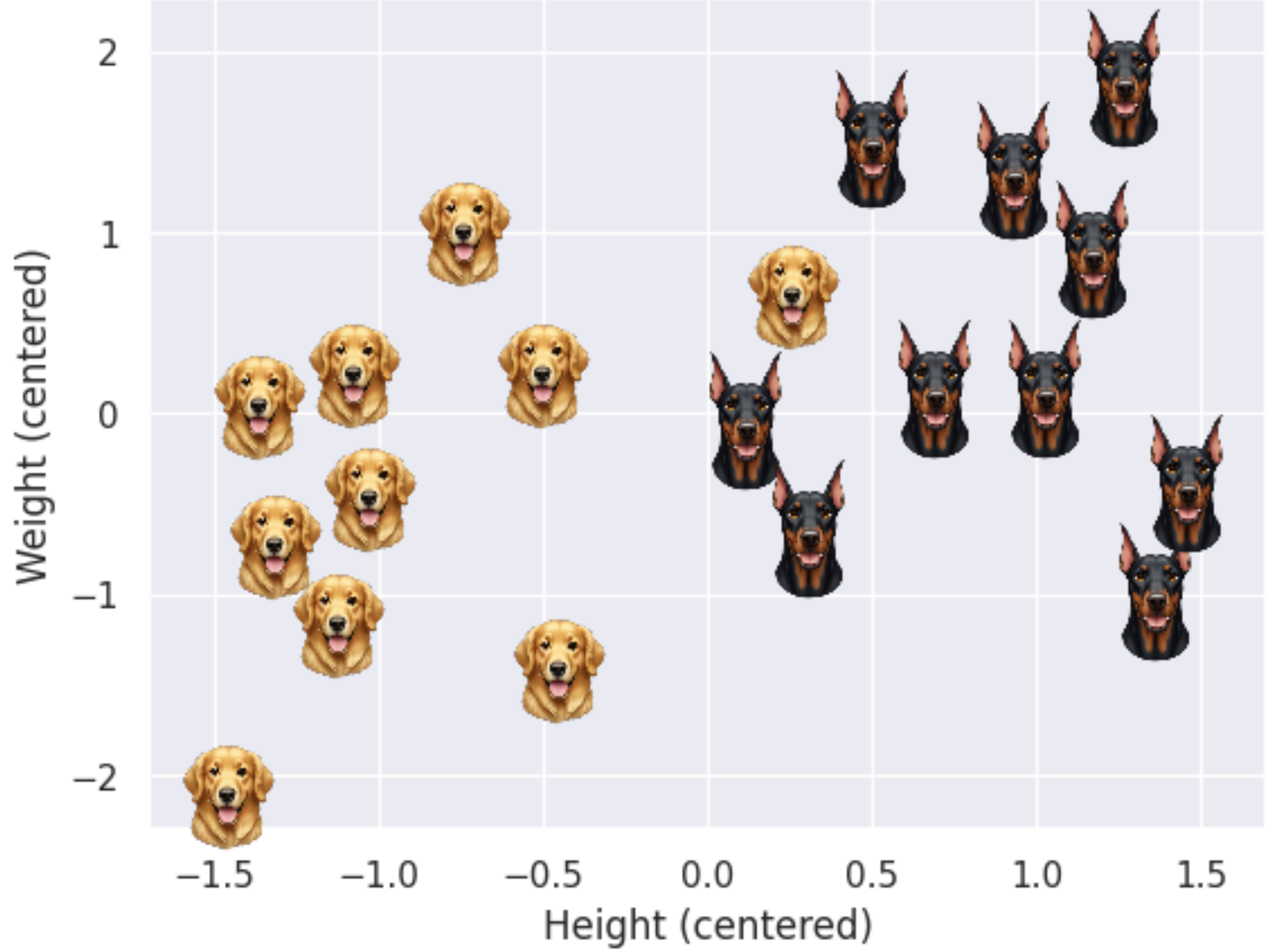
1. **Data.** Height and weight (x_0, x_1) and label $y \in \{+1, -1\}$ for Golden Retriever or Doberman.
2. **Model.** Perceptron model.
3. **Model parameters**. Weight vector $\vec{\mathbf{w}} = (w_0, w_1)$.
4. **Training.** Given datapoints, find good model parameters.
5. **Testing.** Evaluate the performance of learned model on new, previously unseen data (i.e. test set).

Perceptron: Training

Initialize all weights to 0.

For each training example with label $y \in \{+1, -1\}$:

- Compute predicted output $\hat{y} = \text{sign}(S)$
- Update weights if incorrect (i.e. $\hat{y} \neq y$)

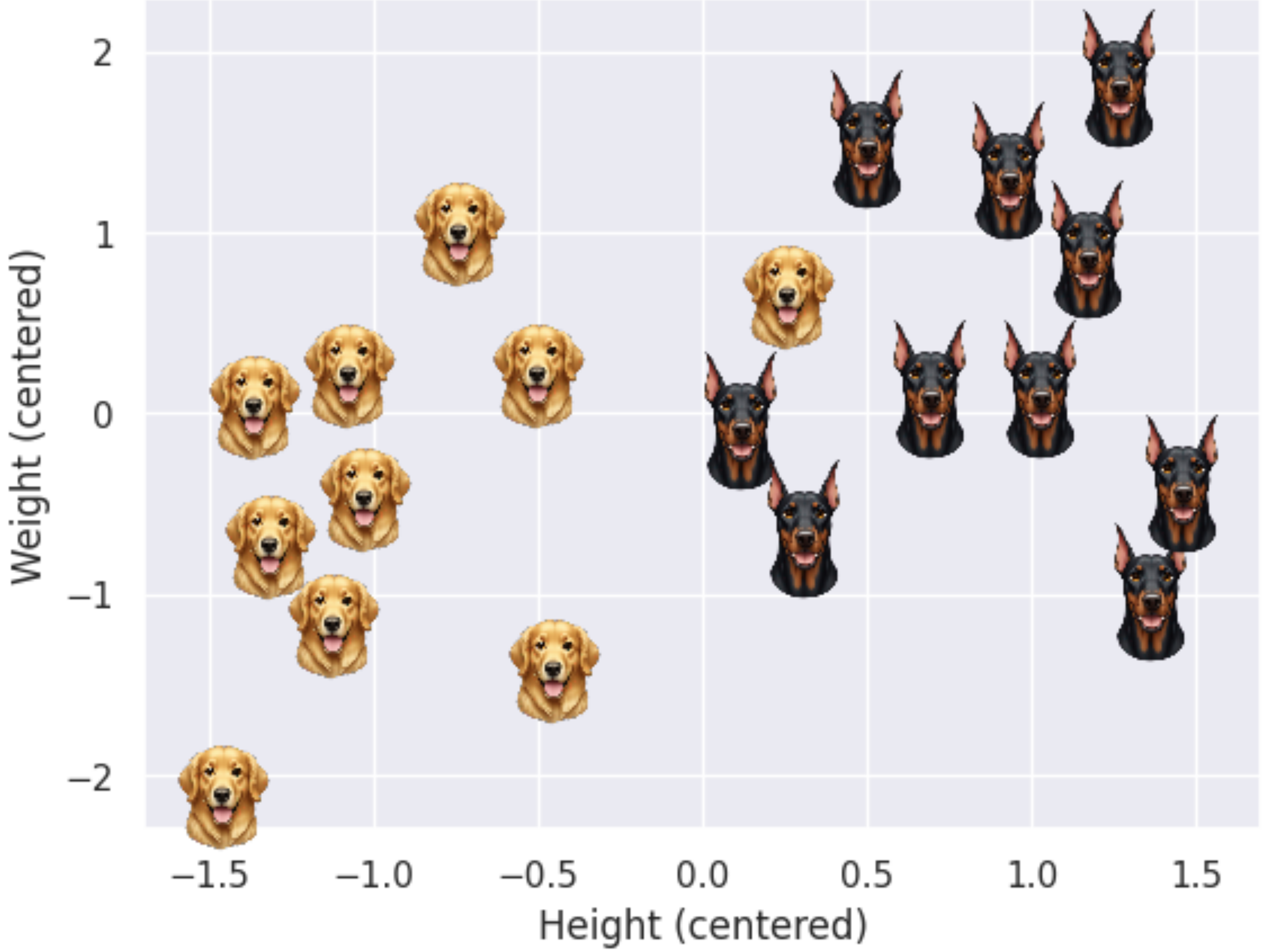


Golden retriever: +1, Doberman -1

Perceptron: Update Rule

For each training example with label $y \in \{+1, -1\}$:

- Compute predicted output
  $S = w_0x_0 + w_1x_1$
  $\hat{y} = \text{sign}(S) = +1$ if $S > 0$
  $= -1$ if $S \leq 0$
- If correct ($y = \hat{y}$), do nothing.
- If false positive ($y = -1, \hat{y} = +1$):
  $w_j' = w_j - x_j$
- If false negative ($y = +1, \hat{y} = -1$):
  $w_j' = w_j + x_j$



Golden retriever: +1, Doberman -1

Poll

- Inputs: $x_0 = 0.3,\ x_1 = 0.6,\ y = +1$
- Weights: $w_0 = -2,\ w_1 = -0.1$
- Compute predicted output

$$S = w_0 x_0 + w_1 x_1$$

$$\hat{y} = \text{sign}(S) = +1 \text{ if } S > 0$$

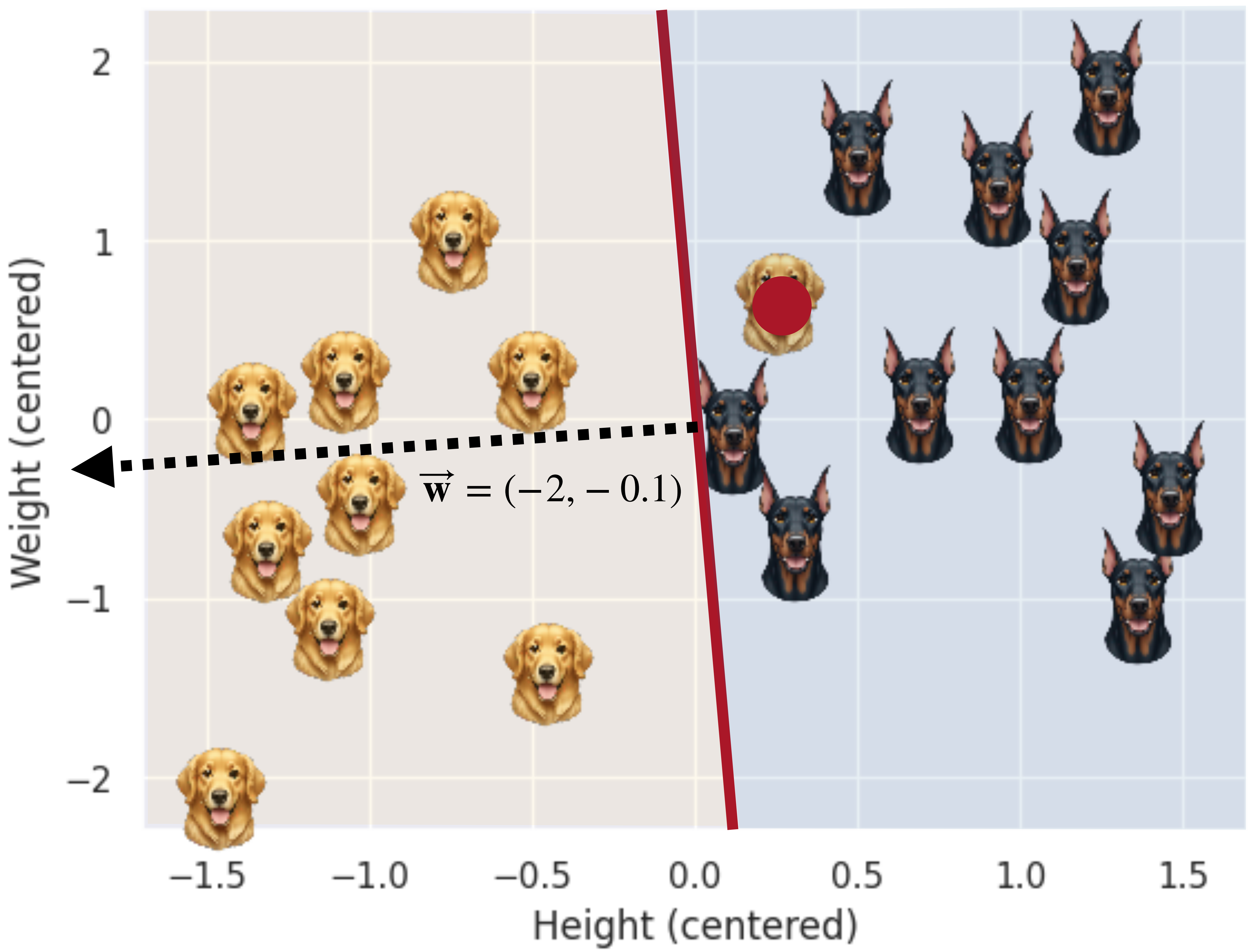
$$= -1 \text{ if } S \leq 0$$

Which scenario is true?

A. Correct prediction ($y = \hat{y}$)

B. False positive ($y = -1,\ \hat{y} = +1$)

C. False negative ($y = +1,\ \hat{y} = -1$)



Golden retriever: +1, Doberman -1

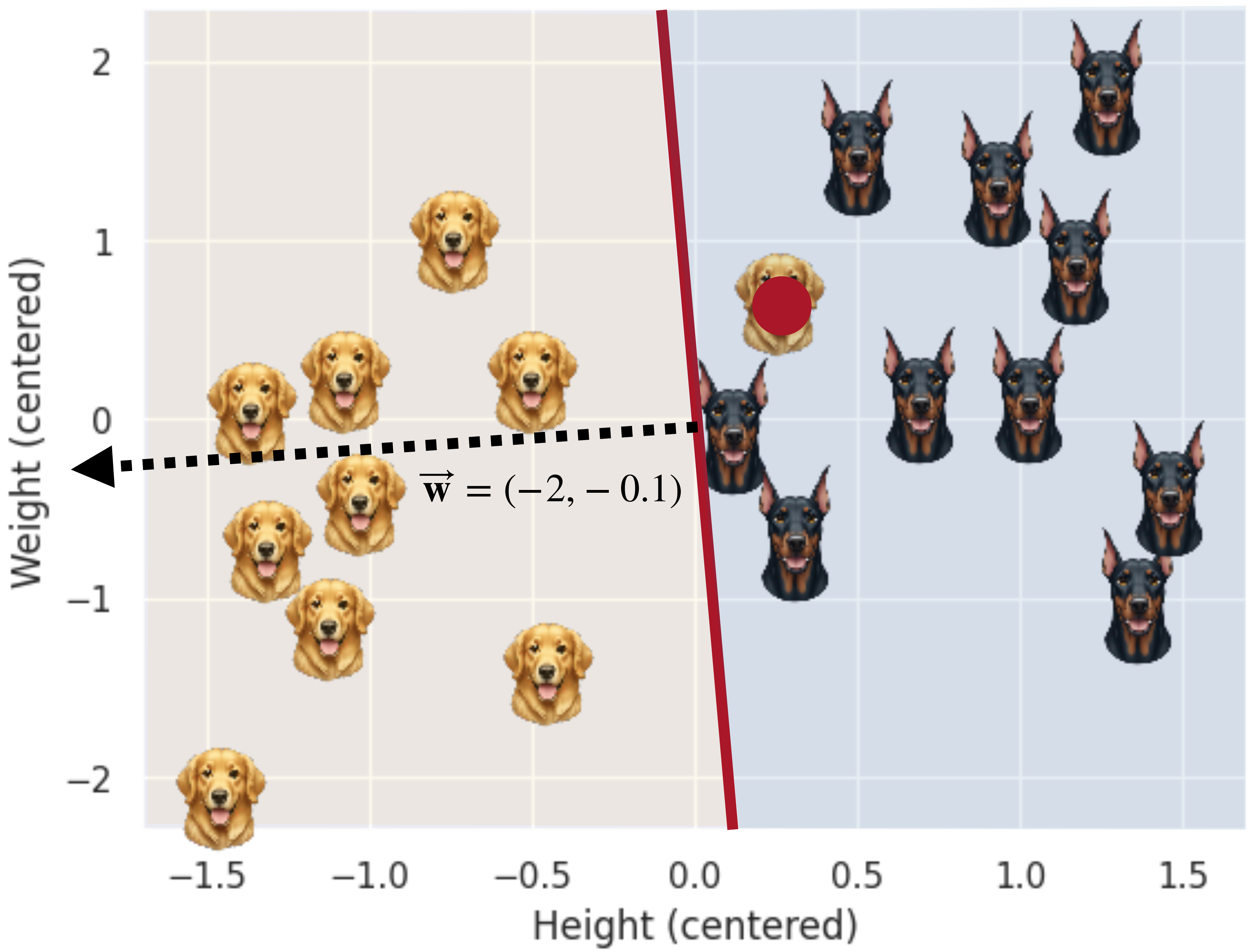
Poll

- Inputs: $x_0 = 0.3,\ x_1 = 0.6,\ y = +1$
- Weights: $w_0 = -2,\ w_1 = -0.1$

$$S = w_0 x_0 + w_1 x_1$$
$$= (-2) \cdot 0.3 + (-0.1) \cdot 0.6$$
$$= -0.6 + (-0.06) = -0.66$$

$$\hat{y} = \text{sign}(S) = -1$$

Which scenario is true?



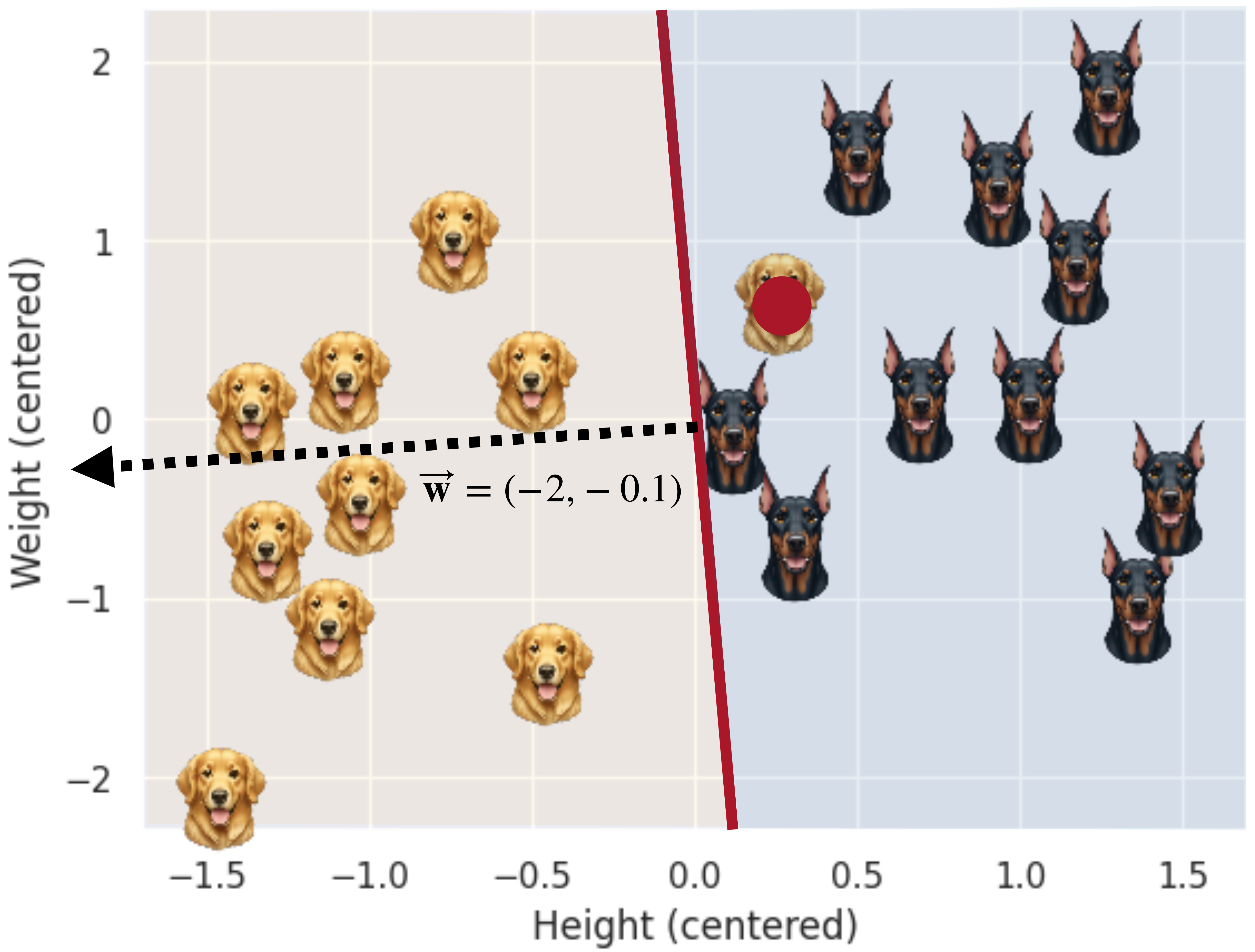
Golden retriever: +1, Doberman -1

Perceptron: Update Rule

- Inputs: $x_0 = 0.3,\ x_1 = 0.6,\ y = +1$
- Weights: $w_0 = -2,\ w_1 = -0.1$

$S = w_0 x_0 + w_1 x_1 = -0.66$

$\hat{y} = \text{sign}(S) = -1$



Update rules:

- If correct ($y = \hat{y}$), do nothing.
- If false positive ($y = -1,\ \hat{y} = +1$):
  $w_j' = w_j - x_j$
- If false negative ($y = +1,\ \hat{y} = -1$):
  $w_j' = w_j + x_j$

Perceptron: Update Rule

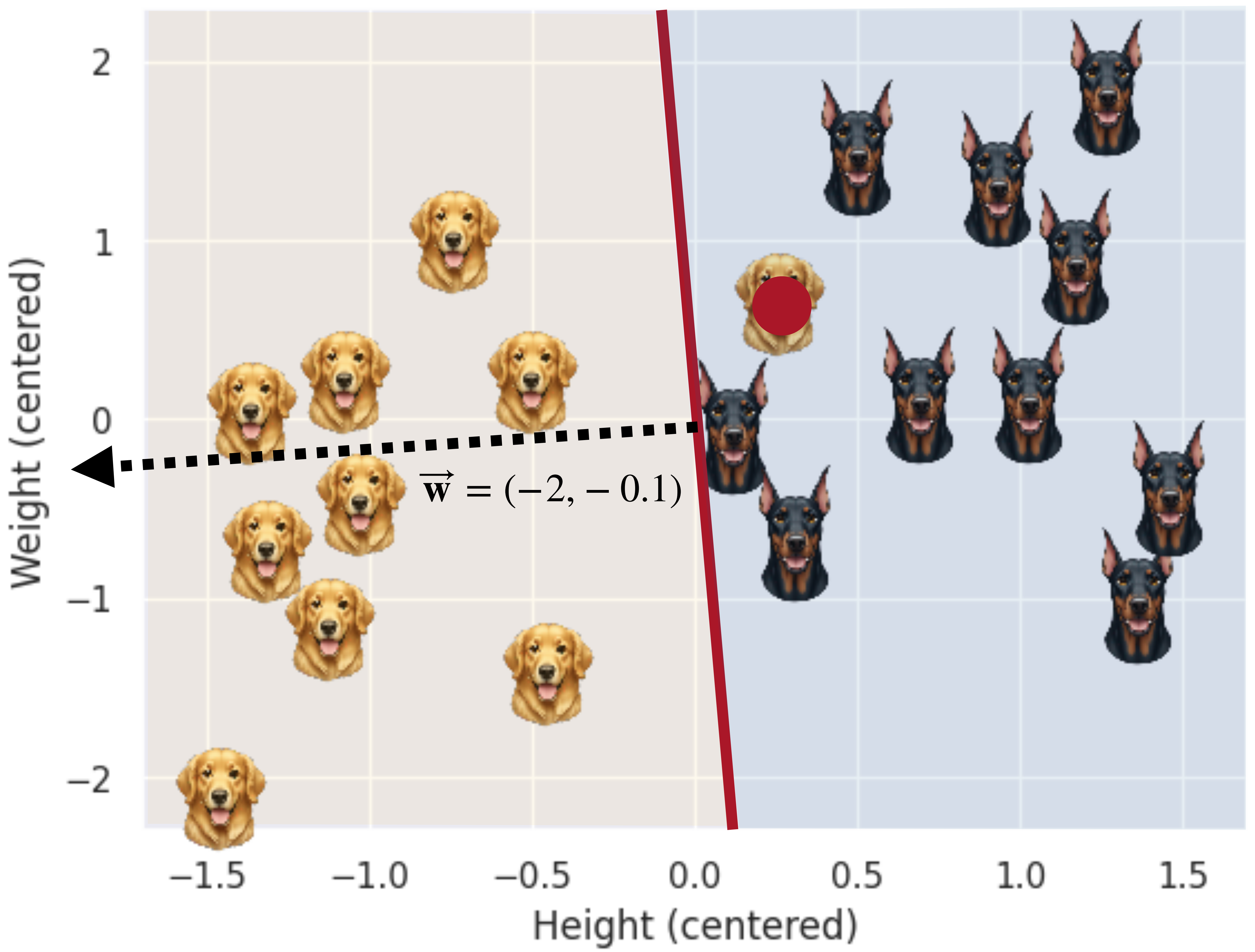
- Inputs: $x_0 = 0.3,\ x_1 = 0.6,\ y = +1$
- Weights: $w_0 = -2,\ w_1 = -0.1$

$S = w_0 x_0 + w_1 x_1 = -0.66$

$\hat{y} = \text{sign}(S) = -1$

Update rules:

- If correct ($y = \hat{y}$), do nothing.
- If false positive ($y = -1,\ \hat{y} = +1$):
  $w_j' = w_j - x_j$
- **If false negative ($y = +1,\ \hat{y} = -1$):**
  $w_j' = w_j + x_j$



Perceptron: Update Rule

- Inputs: $x_0 = 0.3,\ x_1 = 0.6,\ y = +1$
- Weights: $w_0 = -2,\ w_1 = -0.1$

$S = w_0 x_0 + w_1 x_1 = -0.66$

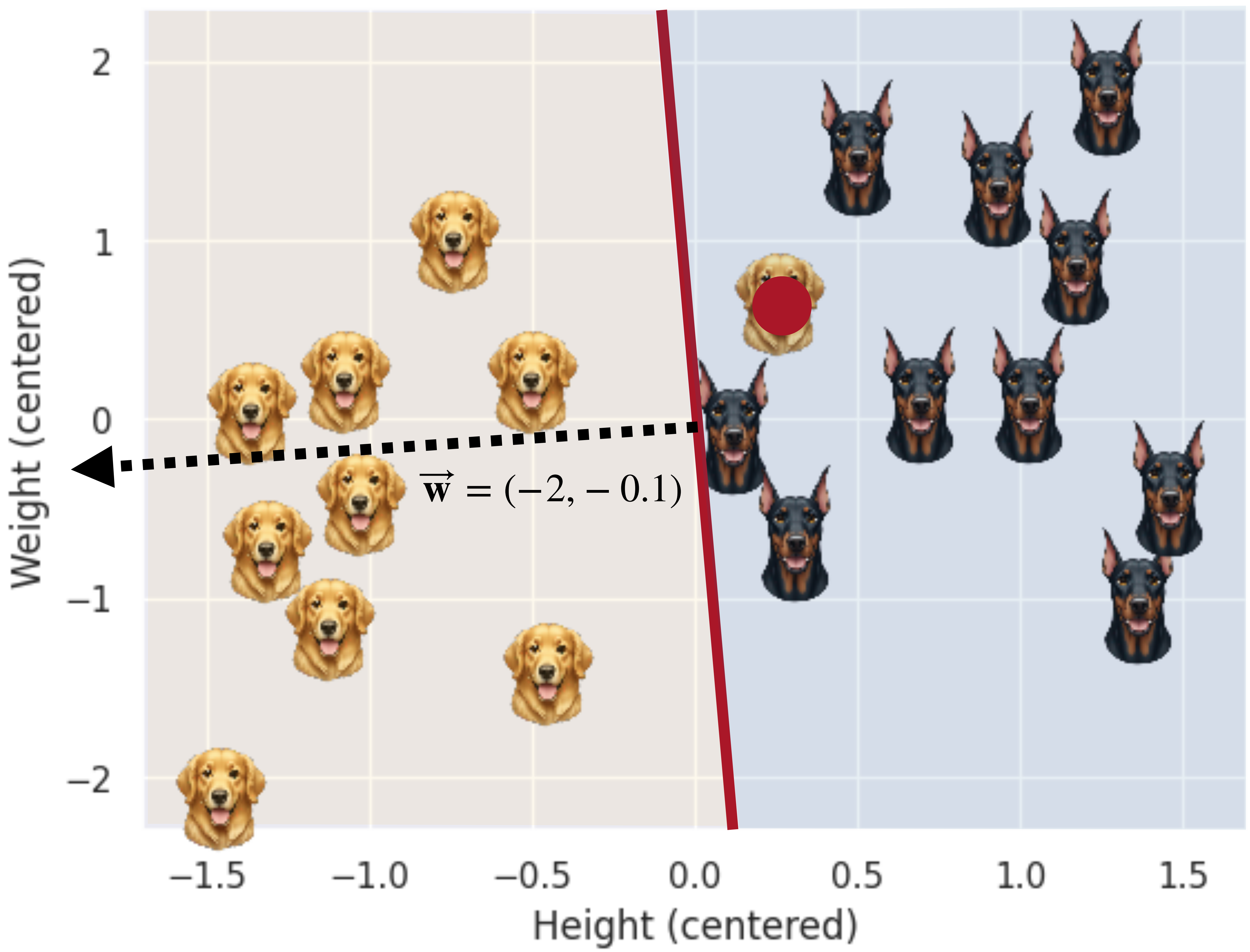
$\hat{y} = \text{sign}(S) = -1$

If false negative ($y = +1,\ \hat{y} = -1$):

$w_j' = w_j + x_j$

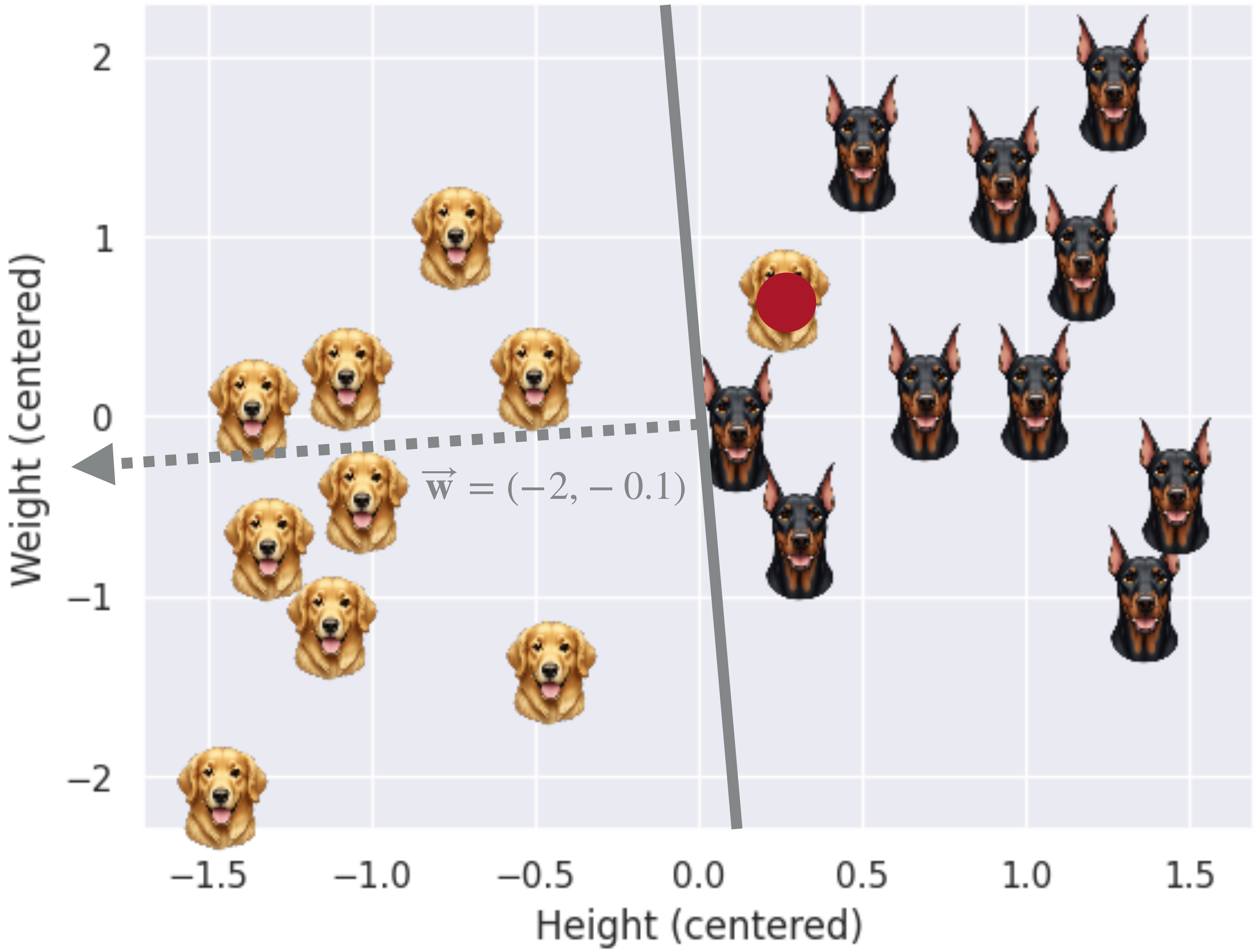
$w_0' = w_0 + x_0 = -2 + 0.3 = -1.7$

$w_1' = w_1 + x_1 = -0.1 + 0.6 = 0.5$



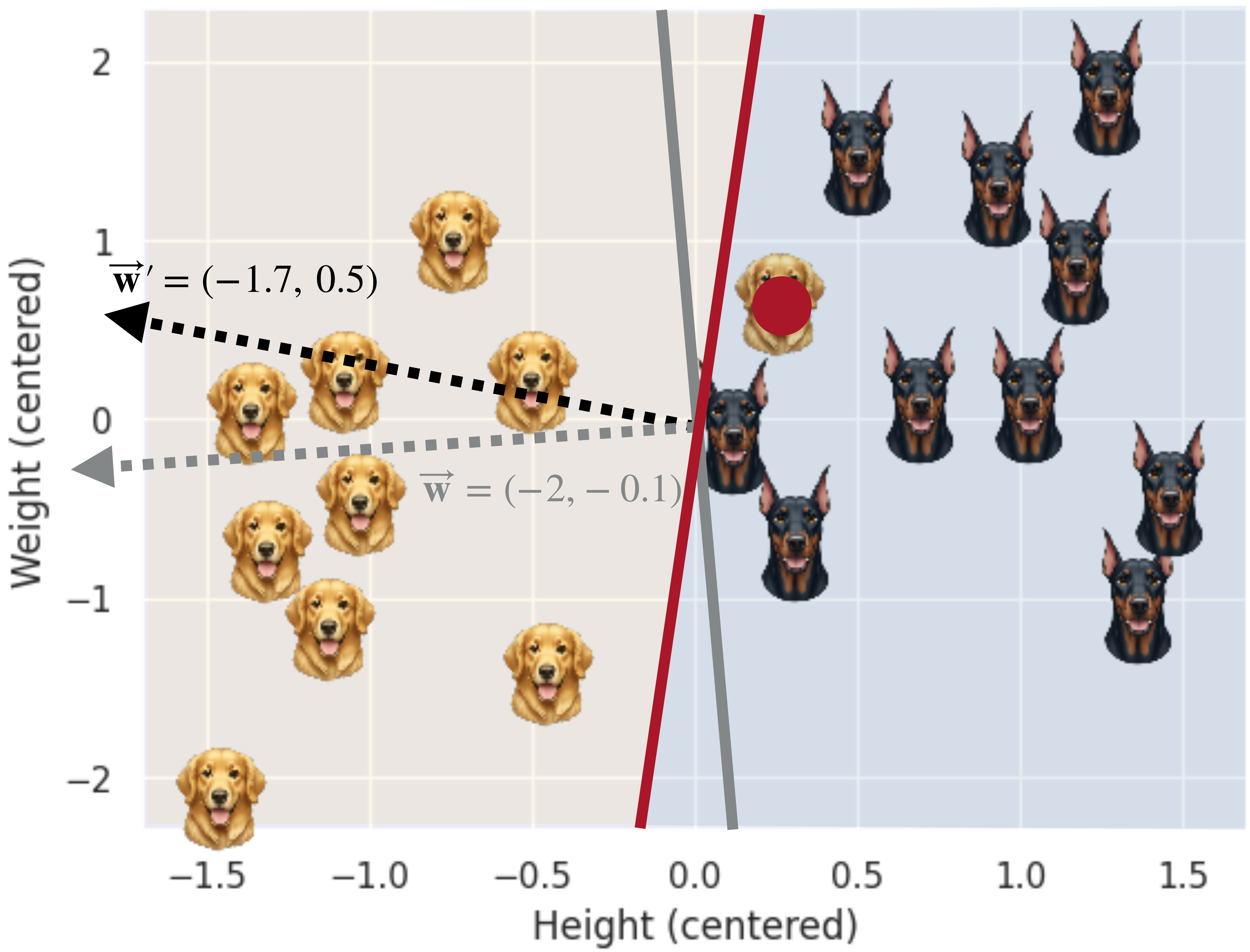
Perceptron: Update Rule

- Inputs: $x_0 = 0.3,\ x_1 = 0.6,\ y = +1$
- Old weights: $w_0 = -2,\ w_1 = -0.1$



Perceptron: Update Rule

- Inputs: $x_0 = 0.3,\ x_1 = 0.6,\ y = +1$
- Old weights: $w_0 = -2,\ w_1 = -0.1$
- New weights: $w_0' = -1.7,\ w_1' = 0.5$



Perceptron: Update Rule

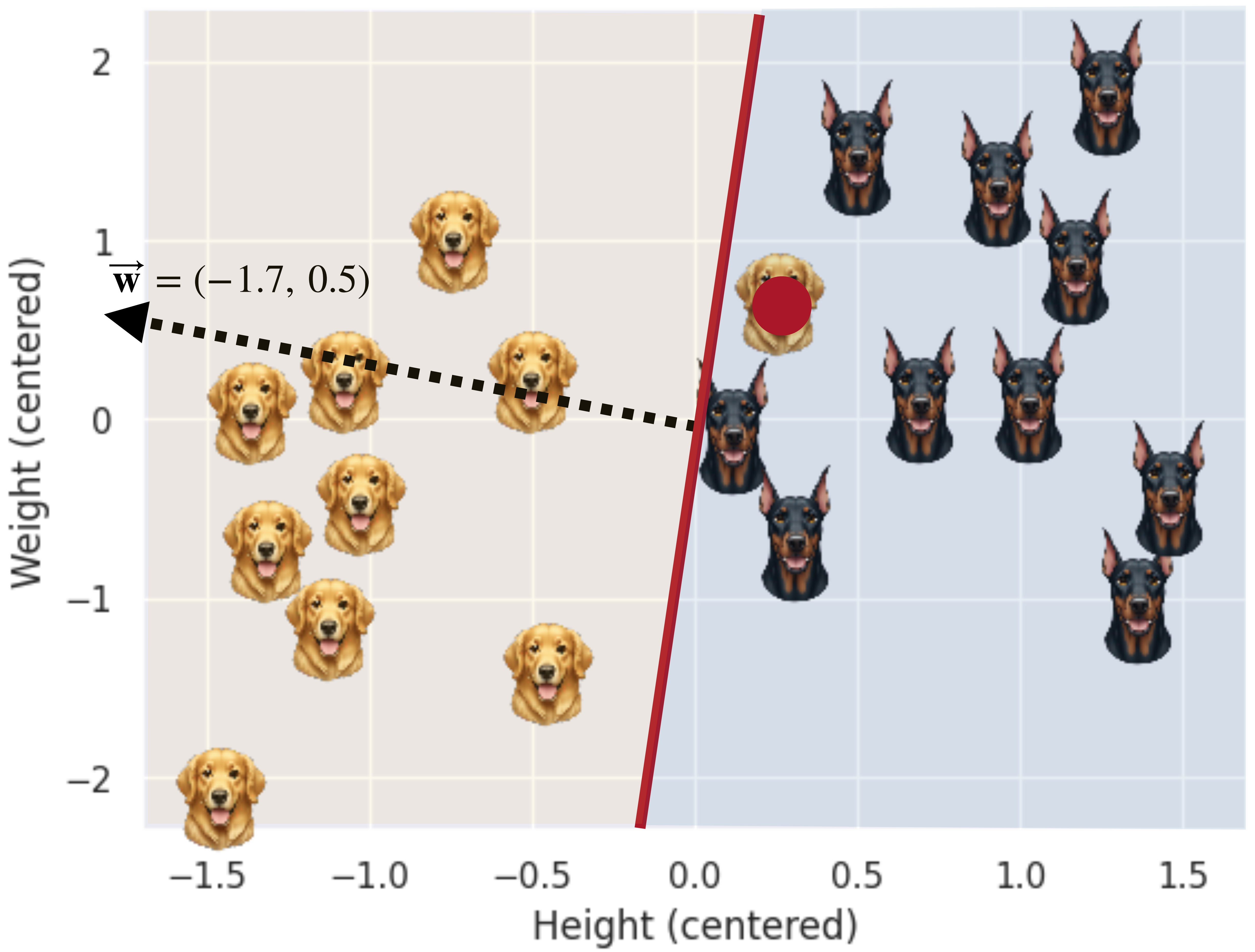
- Inputs: $x_0 = 0.3,\ x_1 = 0.6,\ y = +1$
- New weights: $w_0 = -1.7,\ w_1 = 0.5$

$S = w_0x_0 + w_1x_1 = -0.21$

$\hat{y} = \text{sign}(S) = -1$

Still false negative, but closer to boundary.

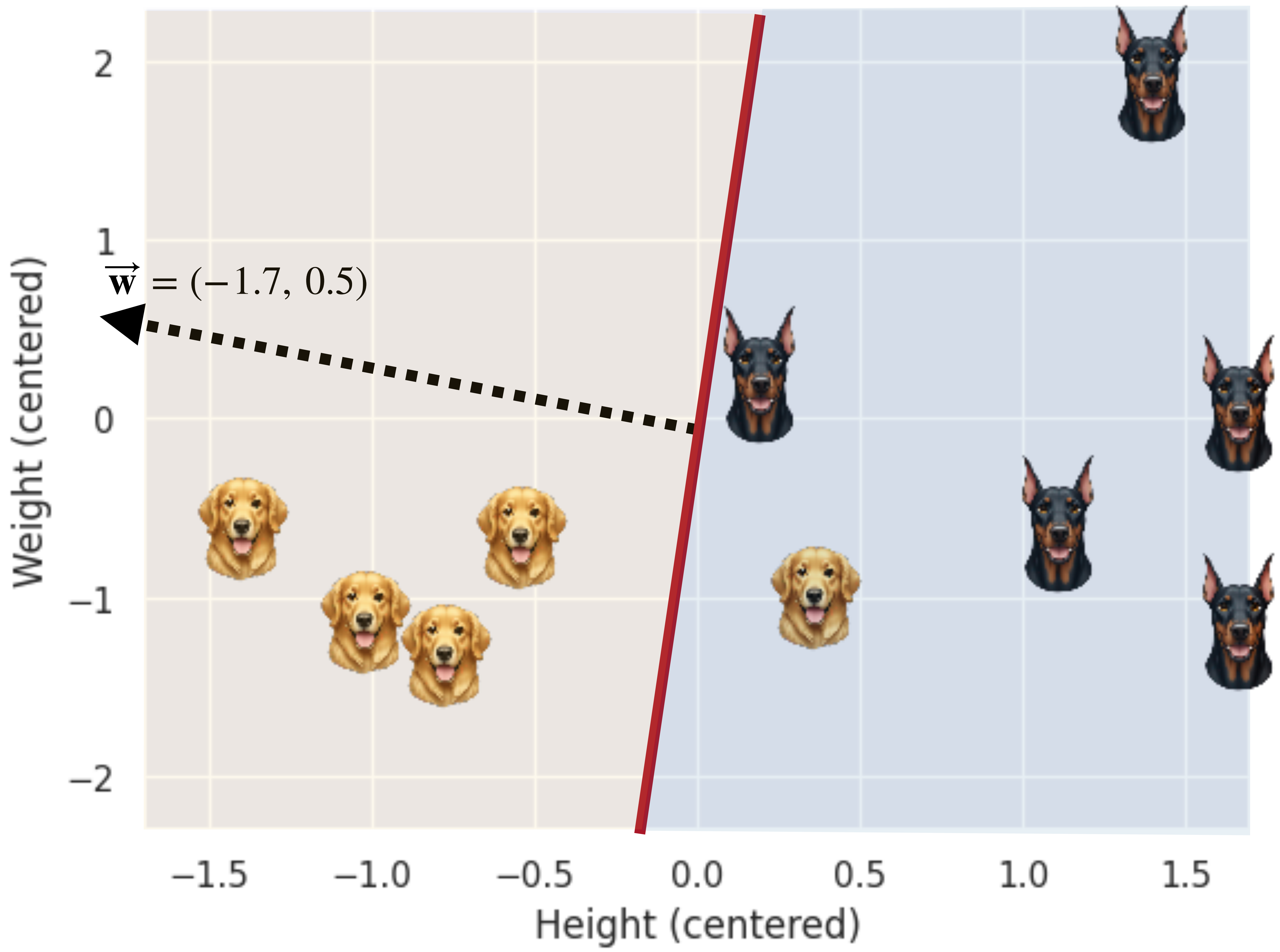
Intuition: Update rule moves decision boundary to be better for the current training example.



Perceptron: Testing

Evaluate performance using an evaluation metric on an unseen set of examples (e.g. test examples).

$$\text{Error rate} = \frac{\#\ \text{incorrect}}{\#\ \text{total}}$$



Poll

Evaluate performance using an evaluation metric on an unseen set of examples (e.g. test examples).

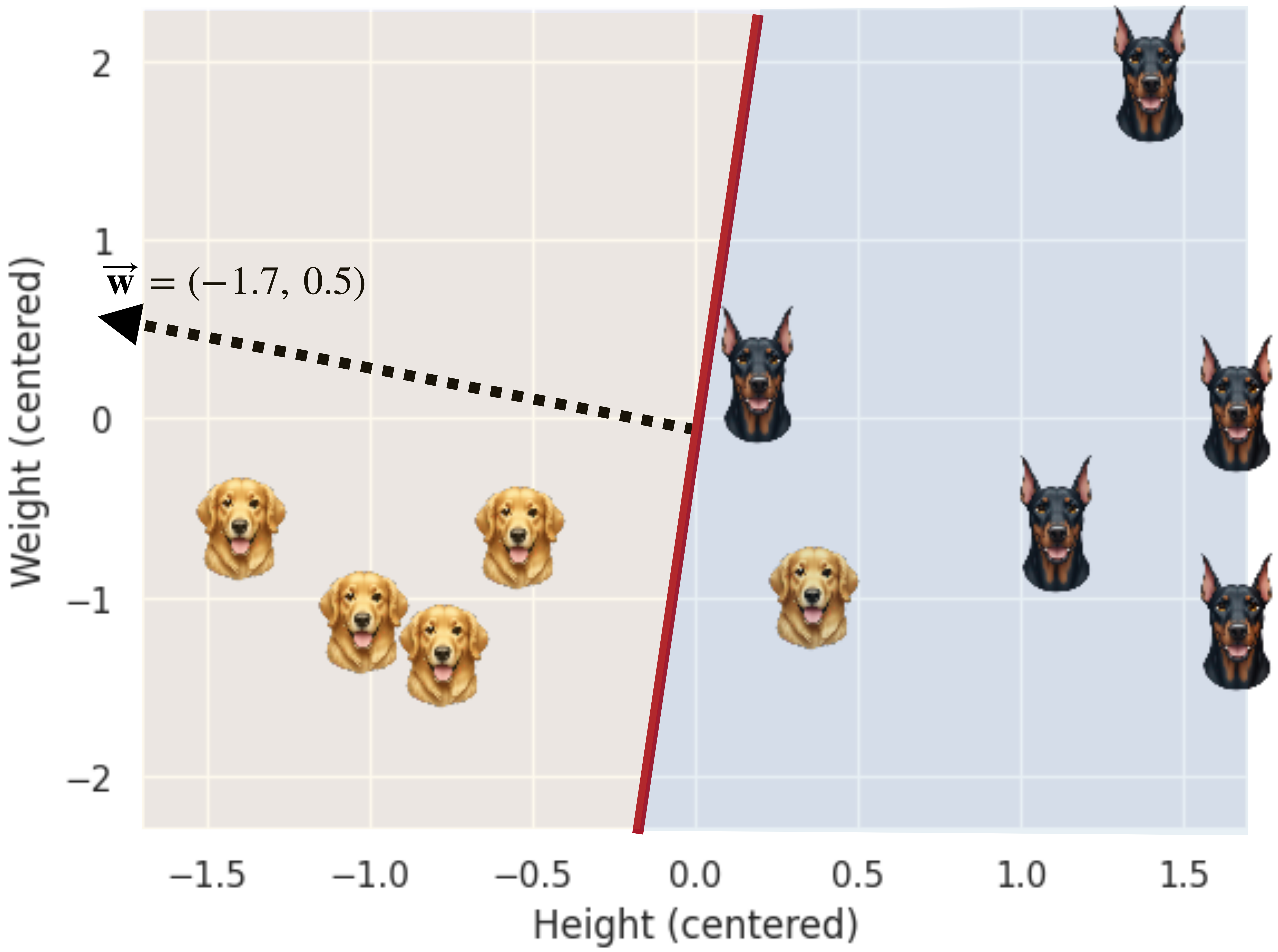
$$\text{Error rate} = \frac{\#\ \text{incorrect}}{\#\ \text{total}}$$

Consider the following unseen examples (5 Golden Retrievers, 5 Dobermans) and the given Perceptron model.

What is the error rate on test set?

A. $0/10$ C. $2/10$

B. $1/10$ D. $9/10$



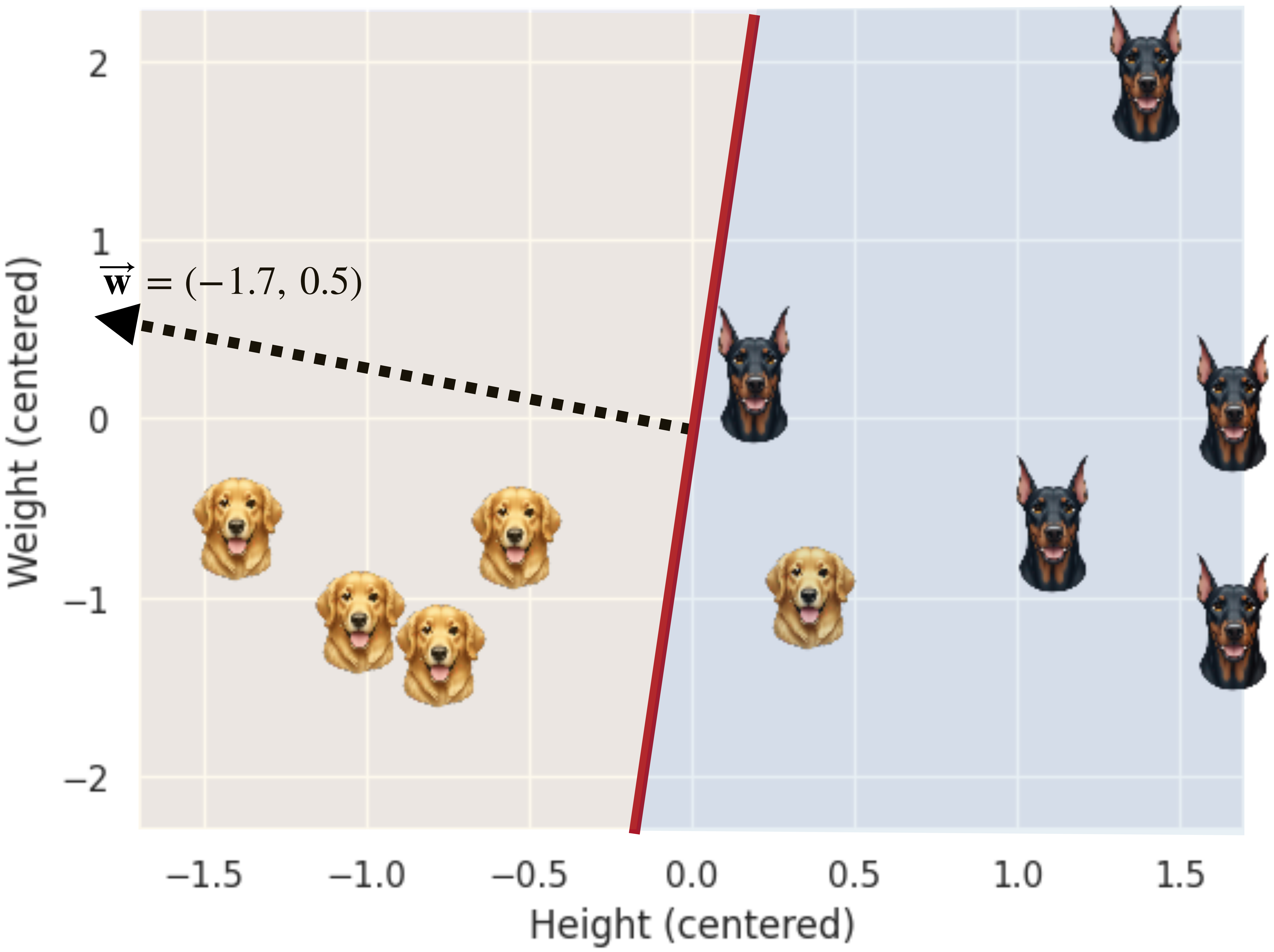
Poll

Evaluate performance using an evaluation metric on an unseen set of examples (e.g. test examples).

$$\text{Error rate} = \frac{\#\ \text{incorrect}}{\#\ \text{total}}$$

Consider the following unseen examples (5 Golden Retrievers, 5 Dobermans) and the given Perceptron model.

What is the error rate on test set?



Ingredients of ML

1. **Data.** Height and weight (x_0, x_1) and label $y \in \{+1, -1\}$ for Golden Retriever or Doberman.
2. **Model.** Perceptron model.
3. **Model parameters**. Weight vector $\vec{\mathbf{w}} = (w_0, w_1)$.
4. **Training.** Use the Perceptron update rule to train on training examples.
5. **Testing.** Compute error rate on new, previously unseen data (i.e. test set).

Readings

Required:

- Etienne Bernard, Wolfram. Introduction to Machine Learning: Machine Learning Paradigms. https://www.wolfram.com/language/introduction-machine-learning/machine-learning-paradigms/