Introduction to Supervised Machine Learning

Computational Communication Science II

May 6, 2025

Agenda 📚

1. Motivation
2. What is Machine Learning (ML)?
3. Families of ML
    - Supervised, Unsupervised, Reinforcement
4. Supervised Learning Deep Dive
    - Features (X), Labels (y), the model (f)
5. Supervised ML Tasks
    - Regression vs Classification
6. Classic Supervised Models
    - Linear, Logistic, k-NN, Trees, SVM, Neural Nets
7. Recap & Next Week
8. Practice/Questions

Why should we care about ML?

- Social media platforms use ML to decide **which posts you see**
- Streaming services use ML to recommend **what to watch next**
- ML can analyze **millions of messages**, fast — good for research
- Knowing the basics helps you **ask sharper research questions**
- It also helps you **spot algorithmic bias** in today's media world

Dog or Muffin?

Dog or Muffin?

Dog or Muffin?

Dog or Muffin?

How were you able to decide this so quickly?

Feature	Chihuahua Face 🐶	Blueberry Muffin 🧁
Shape	Snout, ears, round head	Smooth dome top, straight-sided paper cup
Texture	Fur: soft strands pointing in one direction	Crumbly top with embedded blueberries
Distinction	Eyes, nose, mouth, whiskers	Blueberry spots, sugar crystals, paper pleats
Color Patterns	Fur markings (patches or gradients)	Golden-brown batter with purple/blue dots
Context Clues	Often on grass or floor, with collar/harness	On a plate, in a muffin tin, with crumbs

So what is Machine Learning?

Machine learning means **teaching a computer with examples**, instead of specifying every rule by hand.

Analogy: 🍎 vs. 🍊

Imagine showing a child lots of labeled pictures of apples and oranges.

Eventually, the child can recognize a **new apple** and say “That is an apple!” — even if they’ve never seen *that same apple* before.

That’s how machine learning works: **learn from examples, then generalize**.

Families of Machine Learning

Family	Core idea	Example	Today?
Supervised	Learn from examples **with answers**	Classifying tweets as toxic or not	**Yes**
Unsupervised	Find patterns when **no answers are given**	Grouping similar news articles	No
Reinforcement	Learn by **trial & error with rewards**	Training a chatbot to have a conversation	No

🔍 Zoom in on Supervised ML

We teach the computer using *examples with correct answers*.

Supervised ML in 3 steps

Context: A post on X.

1. We give the computer:
    - **One or more Inputs** (X) — e.g., post length, number of emojis, time of day
    - **One output Label** (y) — e.g., whether a post got over 100 likes 👍
2. It learns a rule or pattern:
   $f : X \to y$ (e.g., short evening posts with emojis $\to$ more than 100 likes)
3. Later, we give it a **new input** (X), and it predicts y based on what it learned.

Features/Inputs/X

Features are the **things we know before** making a prediction. They are the **inputs** the model uses to learn patterns that help predict y.

Example features:

- 📝 **Post length** — number of words or characters
- 😀 **Number of emojis** — expressive signals
- 🕑 **Time of day** — when was it posted?
- 🔗 **Number of hashtags or links**
- 👥 **Author's follower count**

These are all examples of **X** — measurable inputs we feed into the model to predict some y (like likes or shares).

Labels/Outputs/Y

Labels are the **answers** the model learns to predict. They are what we already know for some part of the data, and want to guess for the remaining data.

Example labels:

- ❤️ **Number of likes**
- 🔥 **Goes viral: Yes or No**
- 😡 **Toxic or not toxic**
- 🎥 **Topic of a YouTube comment** (Politics, Music, Gaming...)

These are the y values — the "ground truth" we use to teach the model.

Once trained, the model tries to predict y for **new examples**.

🧠 Model/f

A **model** is a **mathematical recipe** that learns how to connect **inputs (X)** to **outputs (y)** — like a smart guesser.

Example models:

- **Linear Regression**: Learns a **straight-line rule** to predict a number *(e.g., more emojis → more likes?)*
- **Logistic Regression**: Learns to predict **categories** using a curve that separates them *(e.g., more emojis → will a post go viral or not?)*

Think of the model as the part that **learns from the data** first, and then **makes predictions** on new stuff.

🦾 Training

The model sees A LOT of (X, y) examples — like:

> post length = 80, emojis = 3 → 125 likes
> post length = 120, emojis = 3 → 105 likes
> post length = 80, emojis = 1 → 97 likes

It **learns patterns** by adjusting itself to match the real y as closely as possible.

🔮 Prediction

Now we give the model **new input** X — like:

> post length = 150, emojis = 0

It uses what it learned to **predict** y, e.g.,

> it might predict: 72 likes

Check-in

1 Go to wooclap.com

2 Enter the event code in the top banner

Event code
RCJFMQ

Enable answers by SMS

Two Types of Supervised Learning Tasks

Regression

Predict a **number** — our answer y is continuous.

Example:

How many likes will this Instagram post get?

> *42 likes, 97 likes, 3 likes...*

Classification

Predict a **category** — our answer y is discrete.

Example:

Will this tweet go viral?

> *(Yes or No — a binary category)*

Or:

What type of news is this article?

> *(Politics, Sports, Entertainment...)*

Labels/Outputs/Y

- **Number of comments** → ?
- **Account gets blocked: Yes or No** → ?
- **Message is from the government or not** → ?
- **Topic of a TikTok comment** (Music, Politics, Sports...) → ?

Labels/Outputs/Y

- **Number of comments** → *Regression*
- **Account gets blocked: Yes or No** → *Classification*
- **Message is from the government or not** → *Classification*
- **Topic of a TikTok comment** (Music, Politics, Sports...) → *Classification*

Break

Supervised Model Families 🧠

We will learn different ways a computer can “think” when we teach it from examples.

Example task:
Predicting whether a forum post is about sports or not, or how long it takes to read.

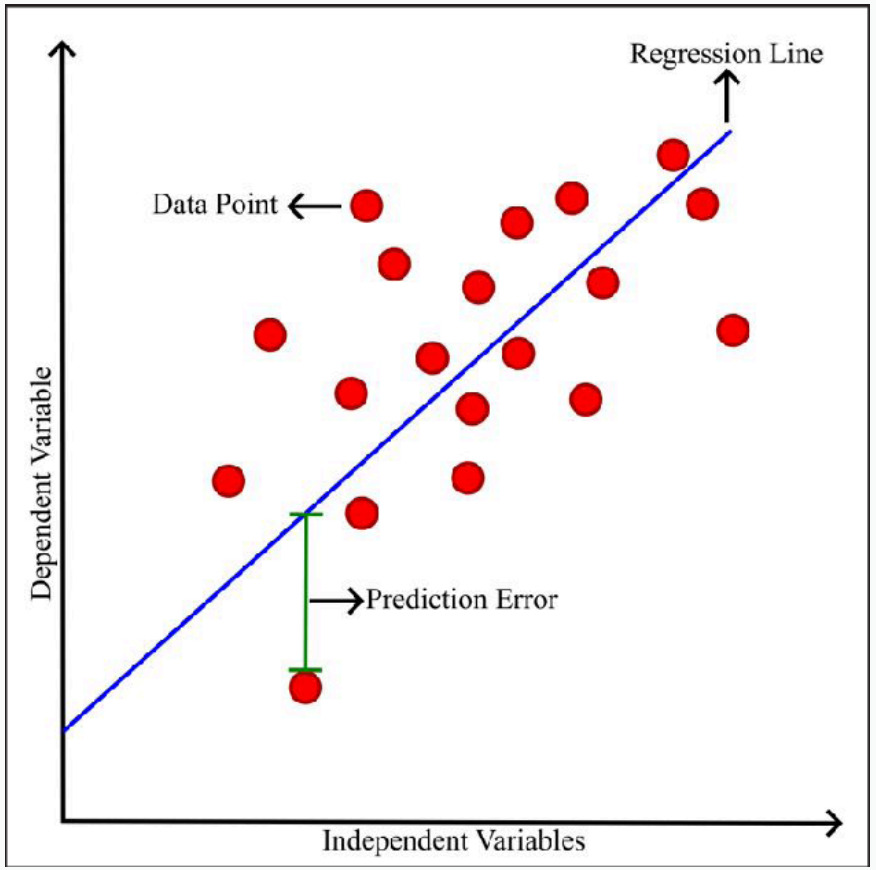
Linear Regression

(Think: drawing a best-fit line)

- **What it does**: Draws a straight line through your data points.
- **Example**: Plot post-length vs. reading time → line shows “longer post = more minutes.”

Linear Regression

(Think: drawing a best-fit line)



Linear Regression

(Think: drawing a best-fit line)

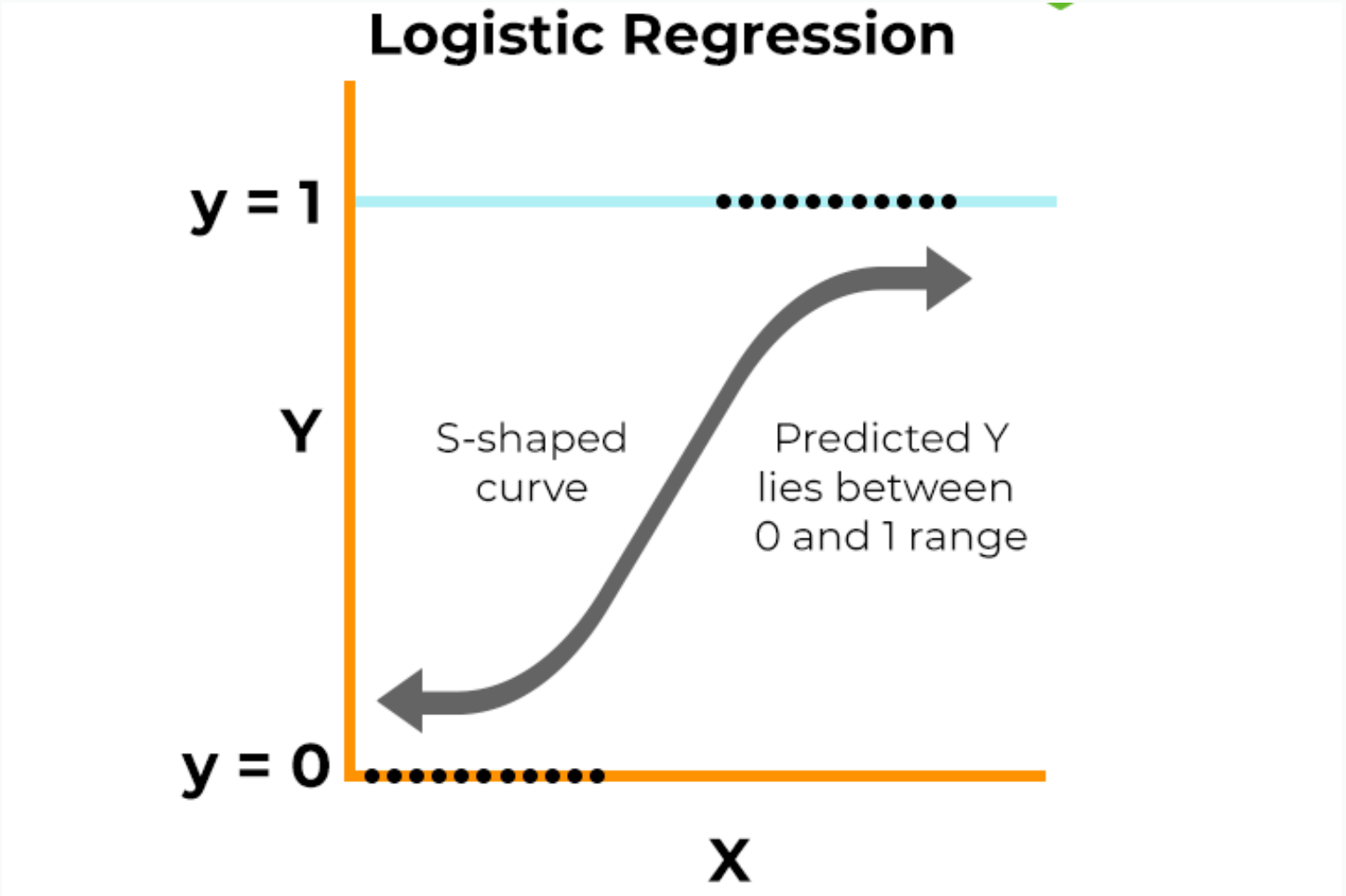
- **What it does**: Draws a straight line through your data points.
- **Example**: Plot post-length vs. reading time → line shows "longer post = more minutes."
- **Good for**: Predicting numbers when the pattern looks roughly straight.
- **Watch out**: Struggles if the real pattern is curved or wiggly.

Logistic Regression

(Think: an "S" curve that turns numbers into yes/no)

- **What it does**: Fits an S-shaped curve to estimate a probability (0–1).
- **Example**: Inputs → probability that "this post is about sports."

Logistic Regression



Logistic Regression

(Think: an "S" curve that turns numbers into yes/no)

- **What it does**: Fits an S-shaped curve to estimate a probability (0–1).
- **Example**: Inputs → probability "this post is about sports."
- **Good for**: Simple yes/no questions ("Will this post go viral?").
- **Watch out**: Name says "regression," but it's actually for classification.

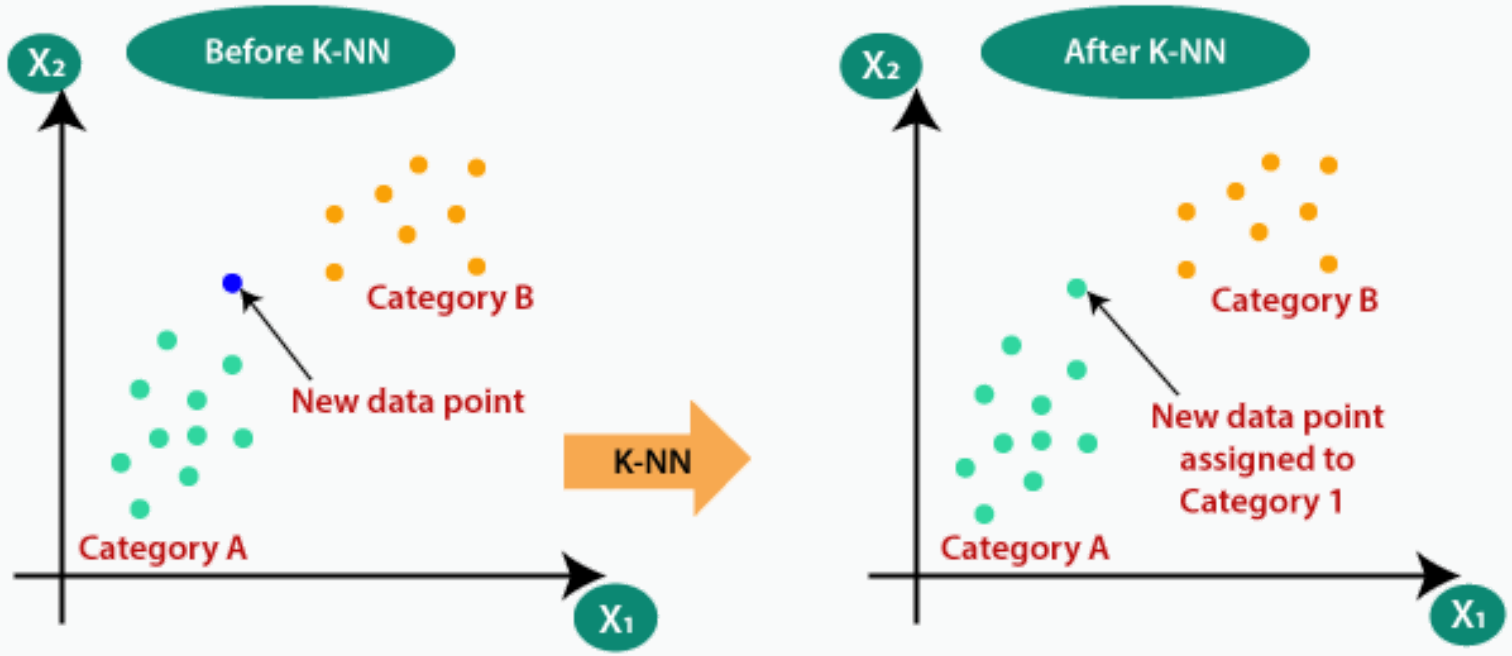
k-Nearest Neighbors (k-NN)

(Think: ask your closest friends)

- **What it does**: Finds the k most similar past examples and “borrows” their answer.
- **Example**: “These 5 posts look like mine → they were political, so mine is too.”

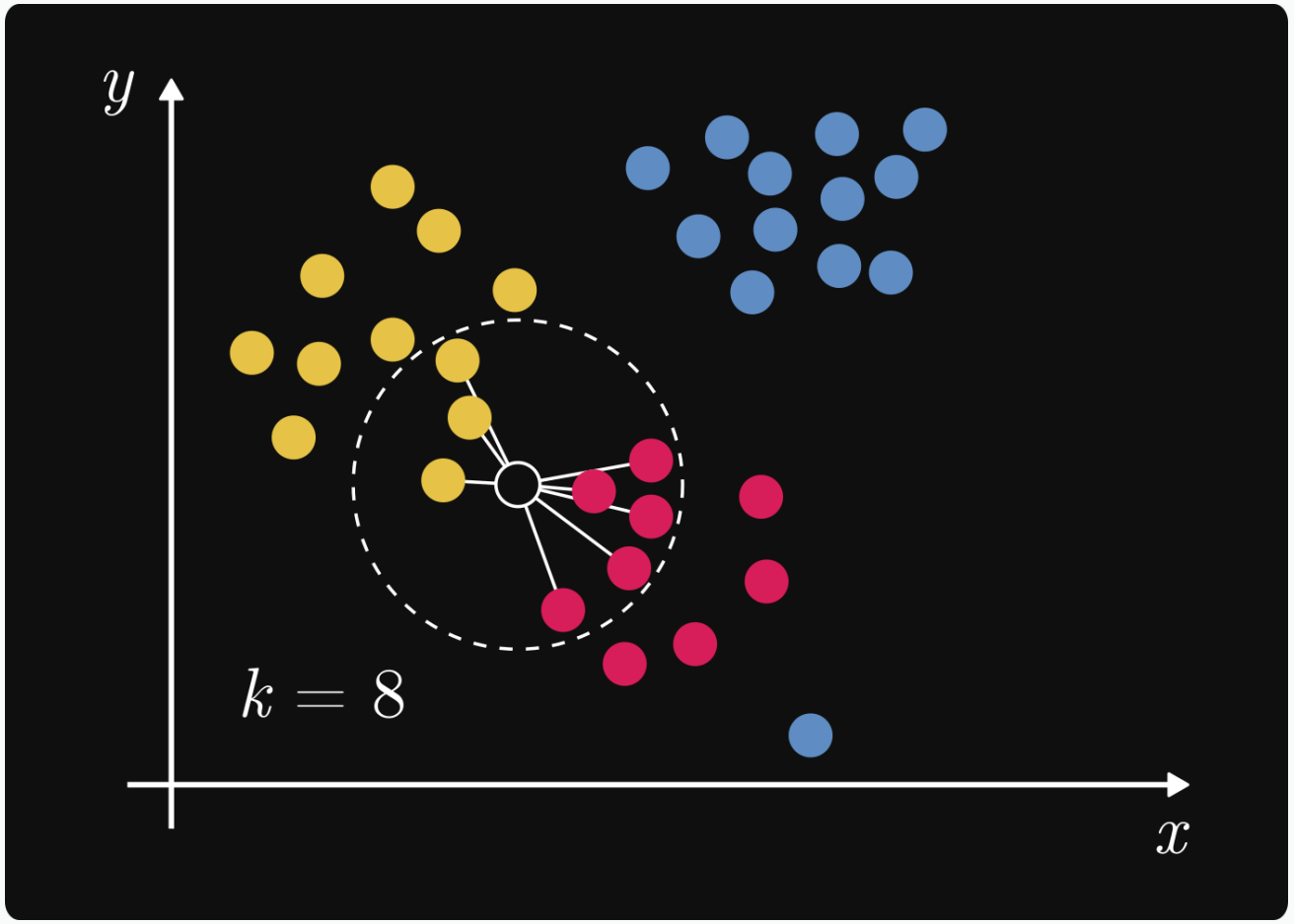
k-Nearest Neighbors (k-NN)

(Think: ask your closest friends)



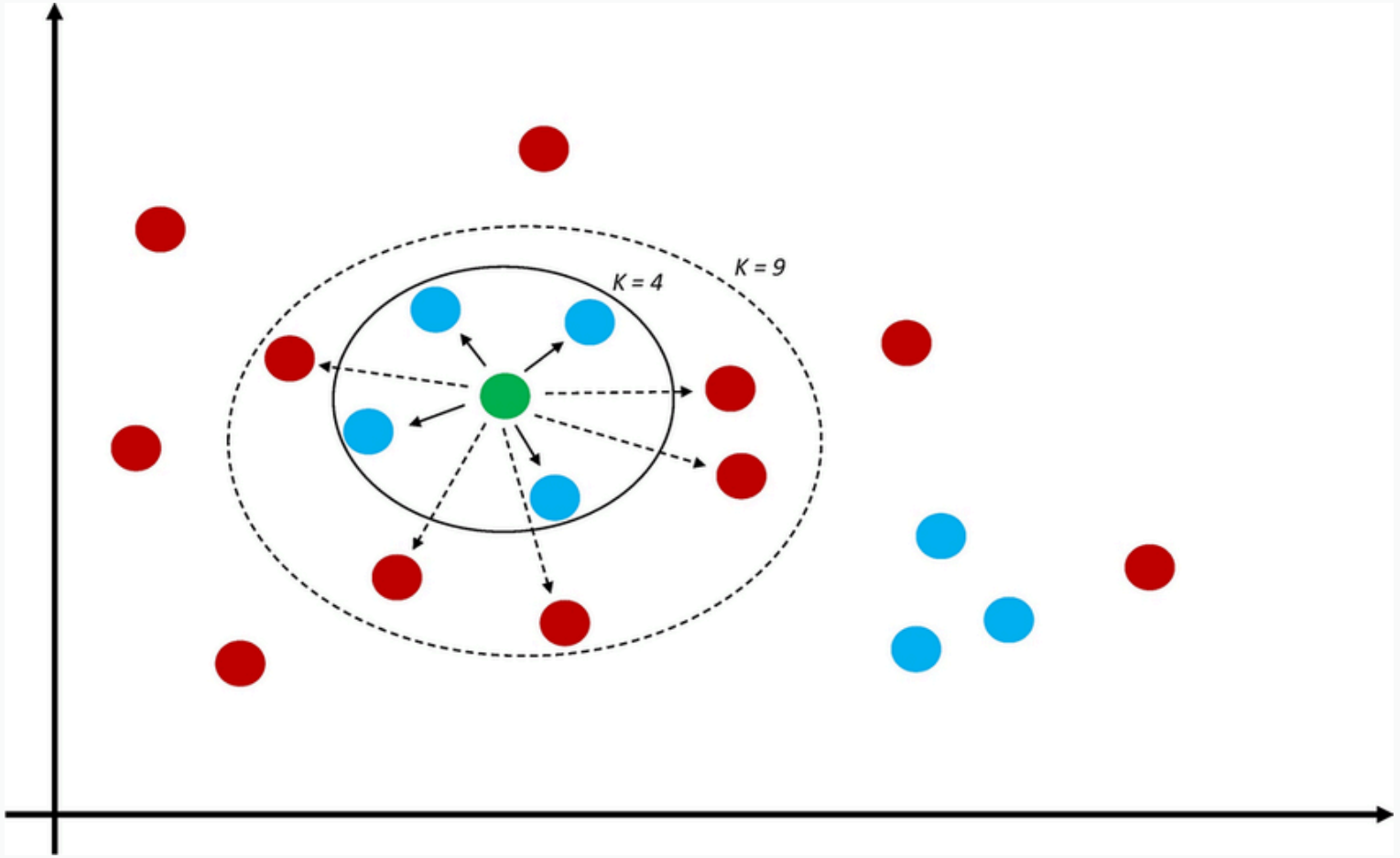
k-Nearest Neighbors (k-NN)

(Think: ask your closest friends)



k-Nearest Neighbors (k-NN)

(Think: ask your closest friends)



k-Nearest Neighbors (k-NN)

(Think: ask your closest friends)

- **What it does**: Finds the k most similar past examples and “borrows” their answer.
- **Example**: “These 5 posts look like mine → they were sporty, so mine is too.”
- **Good for**: Quick start – no training needed.
- **Watch out**: Slows down if you have tons of examples, sensitive to irrelevant details.

Decision Trees 🌳

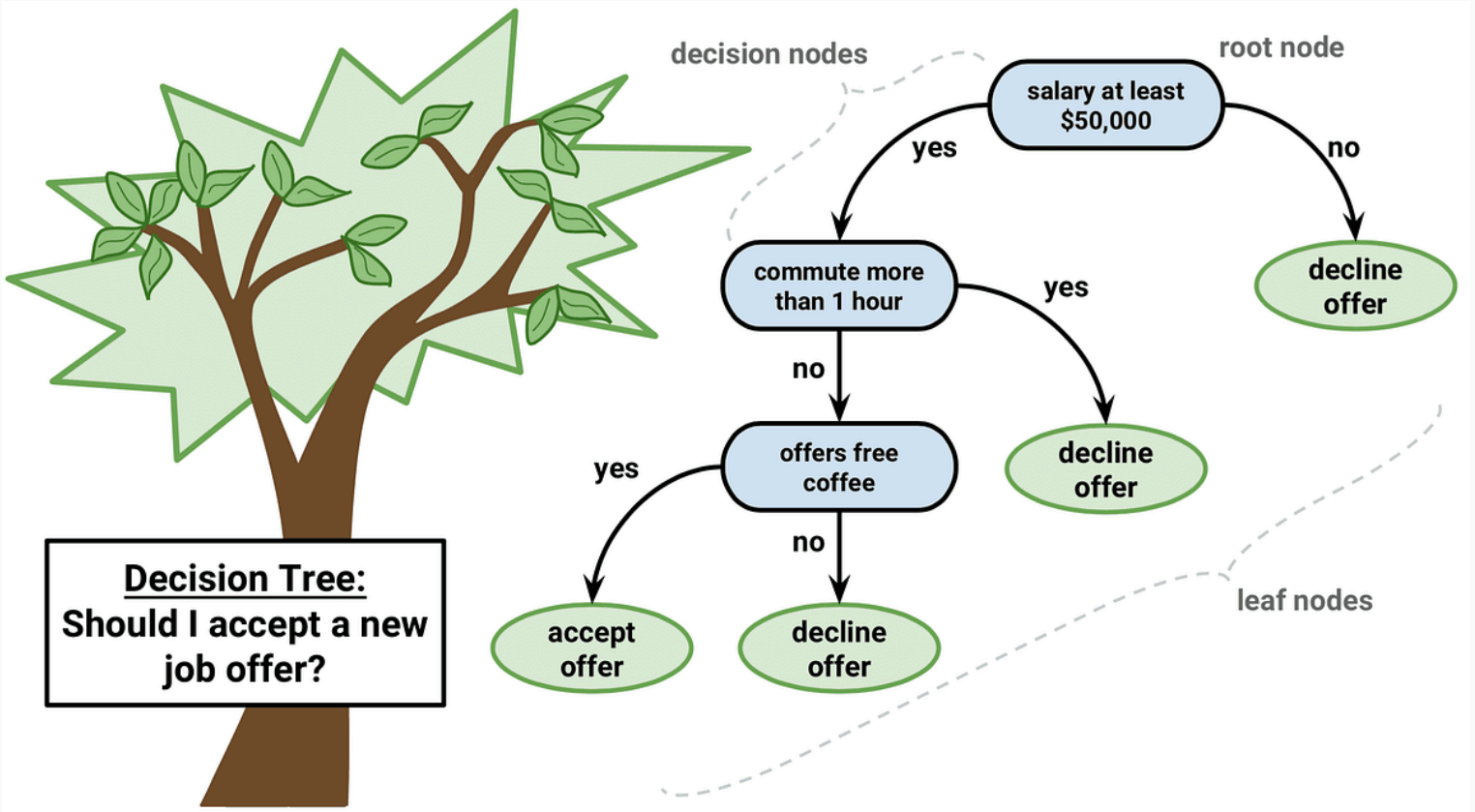
(Think: 20 questions)

- **What it does**: Splits data with yes/no questions in a flowchart.
- **Example**:

    1. “Does the X post mention ‘goal’?”
    2. If yes → probably sports.

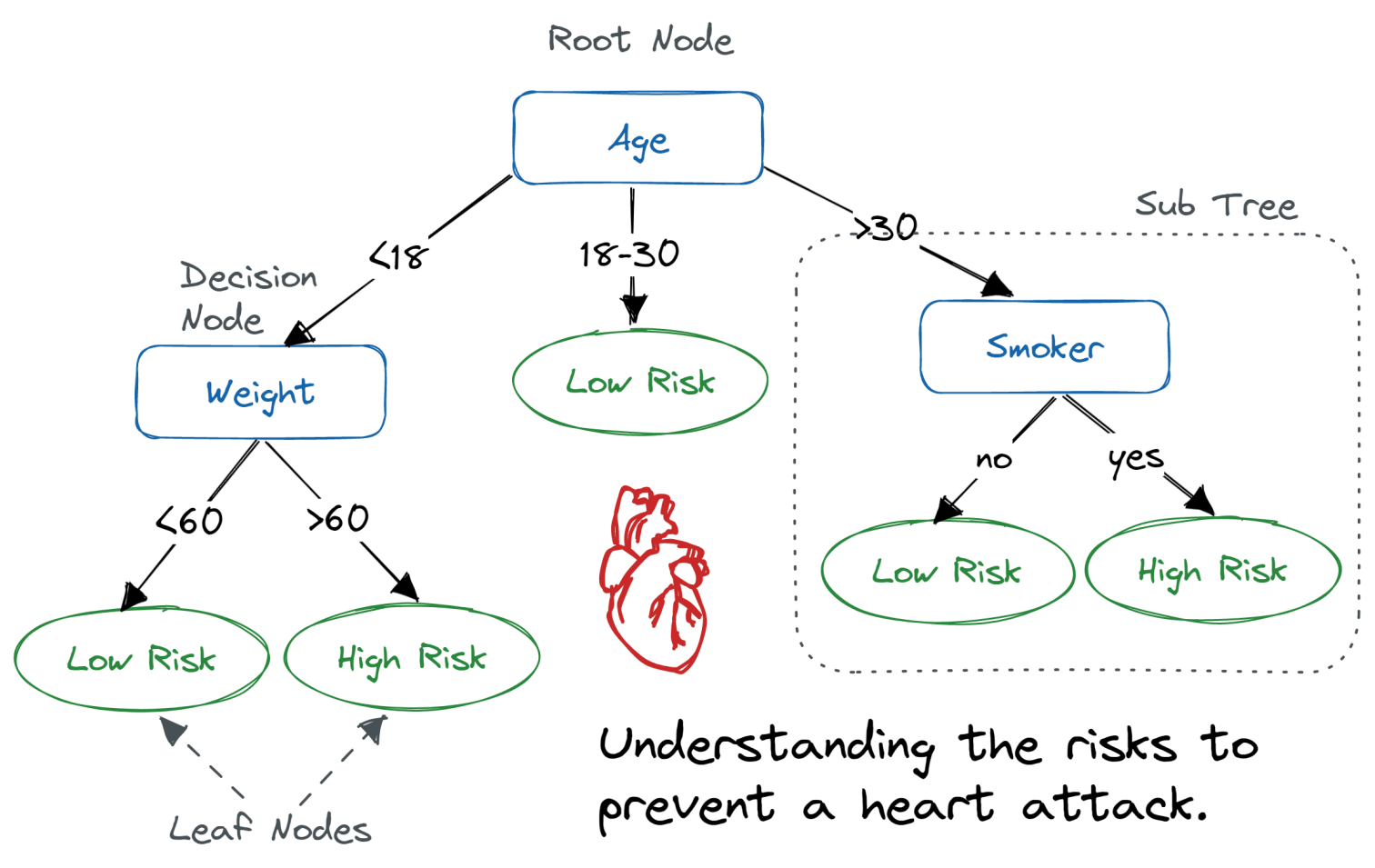
Decision Trees 🌳

(Think: 20 questions)



Decision Trees 🌳

(Think: 20 questions)



Decision Trees 🌳

(Think: 20 questions)

- **What it does**: Splits data with yes/no questions in a flowchart.
- **Example**:

    1. “Does the post mention ‘goal’?”
    2. Yes → probably sports.

- **Good for**: Clear rules you can draw and explain.
- **Watch out**: One tree can overlearn odd patterns (we will discuss *overfitting* next week).

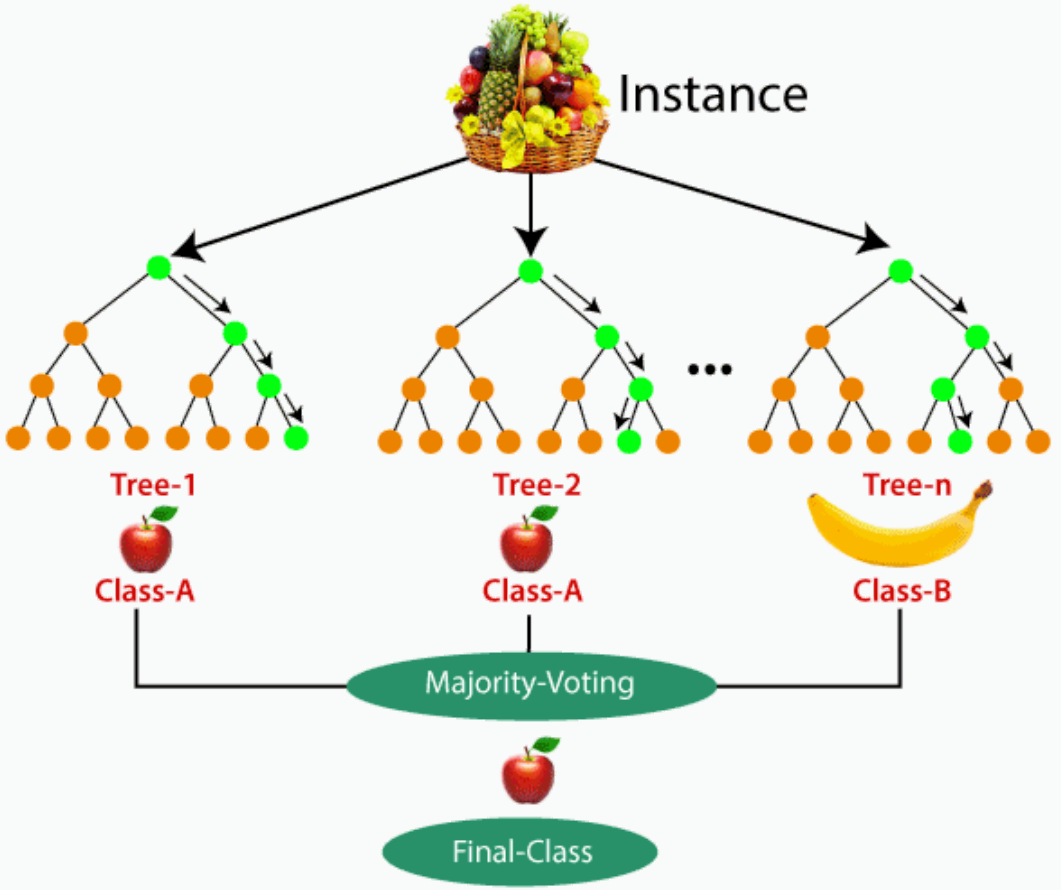
Random Forest 🌲🌲🌲

(Think: many small decision trees voting)

- **What it does**: Builds lots of decision trees on different slices of data, then averages their answers.
- **Example**: 100s of decision trees each vote “apple” or “banana” → majority wins.

Random Forest 🌲🌲🌲

(Think: many small decision trees voting)



Random Forest 🌲🌲🌲

(Think: many small decision trees voting)

- **What it does**: Builds lots of decision trees on different slices of data, then averages their answers.
- **Example**: 100s of decision trees each vote “apple” or “banana” → majority wins.
- **Good for**: Stronger accuracy than a single tree.
- **Watch out**: Harder to explain all the little trees.

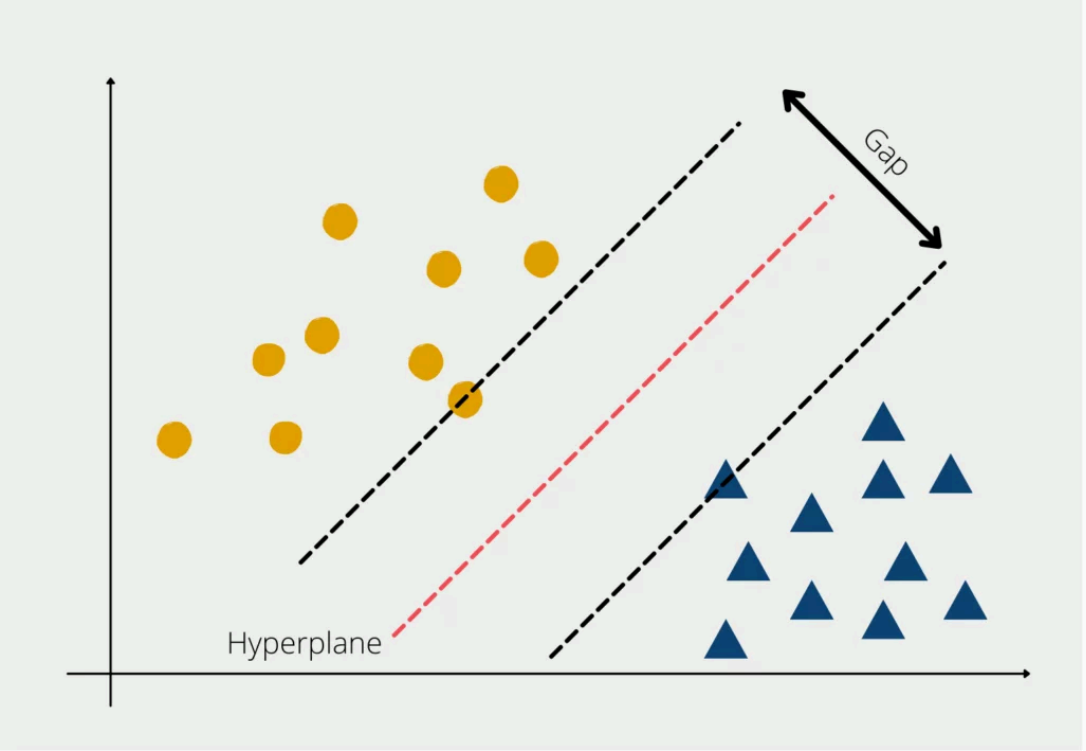
Support Vector Machine (SVM)

(Think: drawing the widest fence between groups)

- **What it does**: Finds the line (or hyperplane) that leaves the biggest gap between categories.
- **Example**: Draw a line that best separates sports from non-sports posts in feature space.

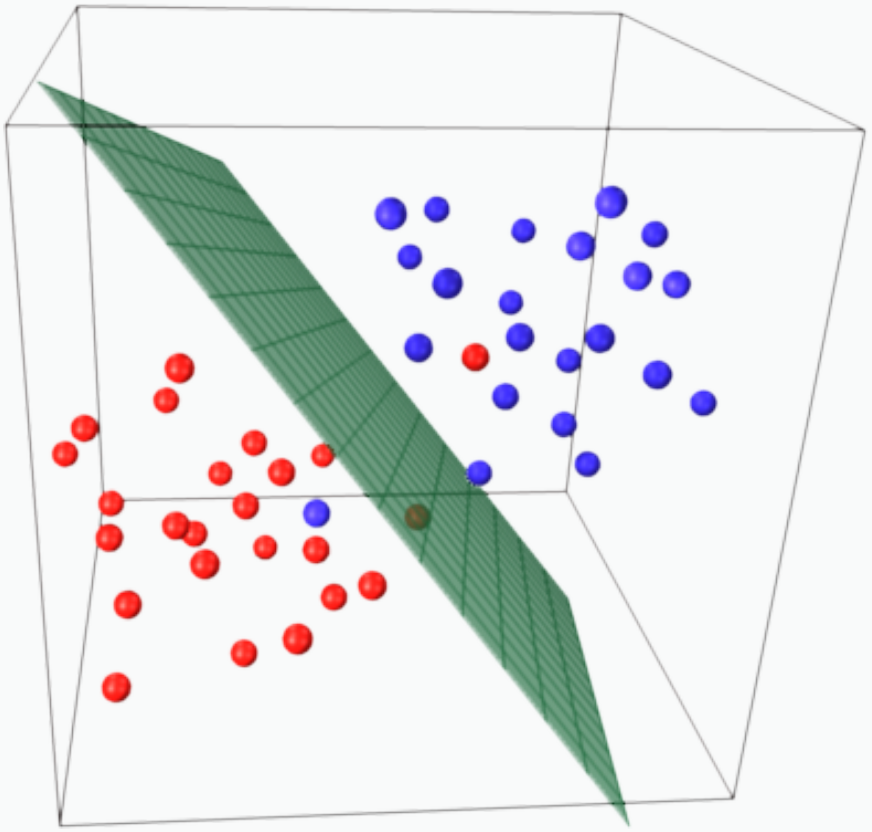
Support Vector Machine (SVM)

(Think: drawing the widest fence between groups)



Support Vector Machine (SVM)

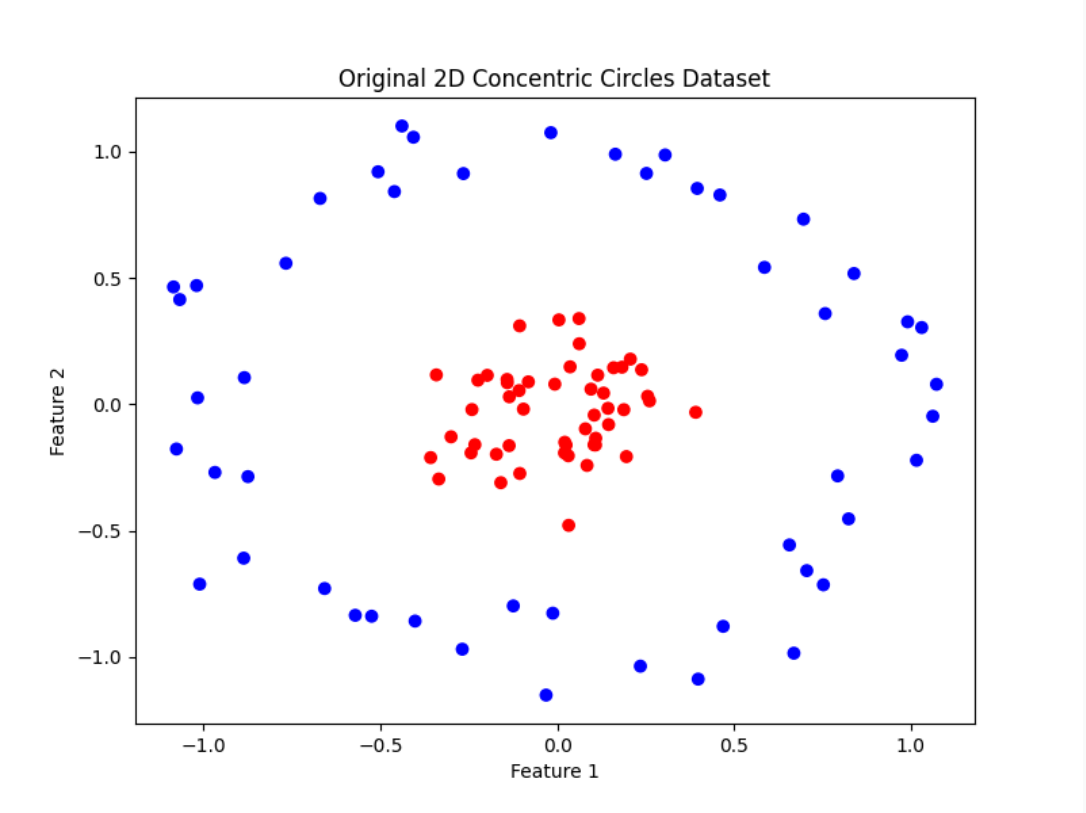
(Think: drawing the widest fence between groups)



Support Vector Machine (SVM)

(Think: drawing the widest fence between groups)

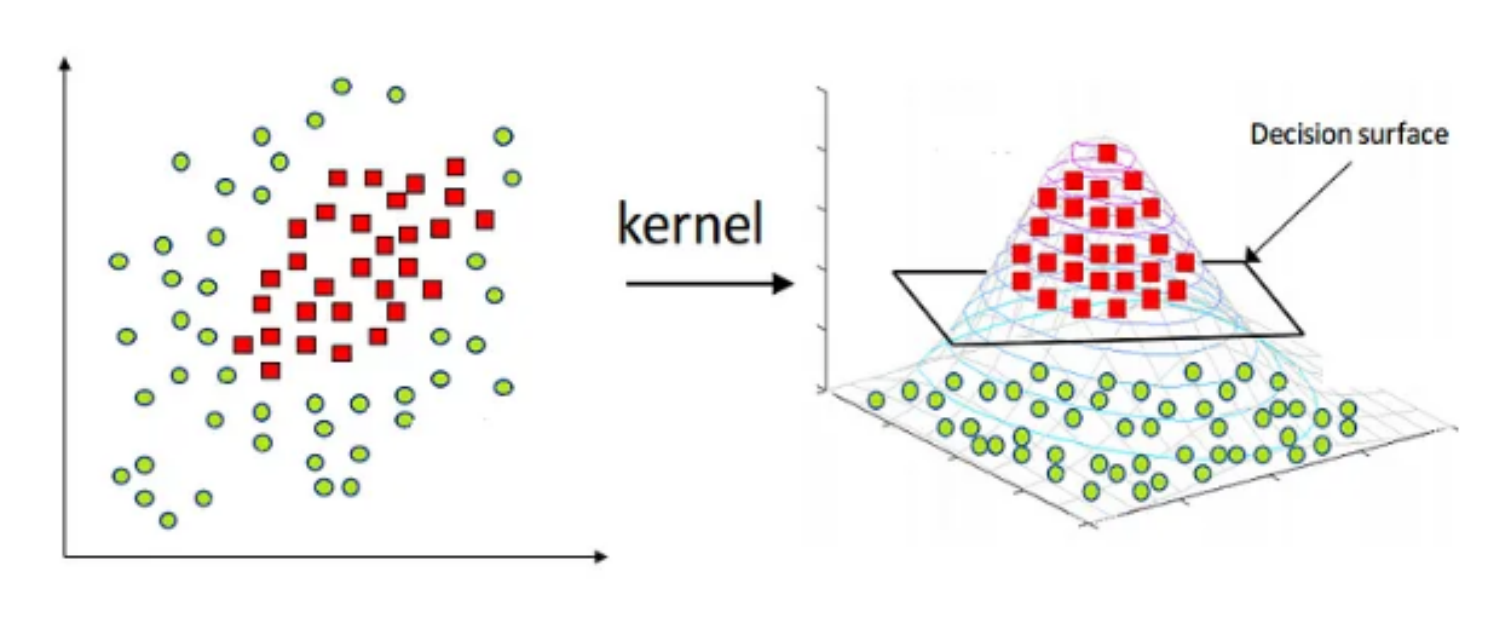
Can we use SVM here?



Support Vector Machine (SVM)

(Think: drawing the widest fence between groups)

Can we use SVM here? Yes.



Support Vector Machine (SVM)

(Think: drawing the widest fence between groups)

- **What it does**: Finds the line (or hyperplane) that leaves the biggest gap between categories.
- **Example**: Draw a line that best separates sports from non-sports posts in feature space.
- **Good for**: High-dimensional data (lots of features).
- **Watch out**: Choosing how to draw that fence (kernel) can be tricky.

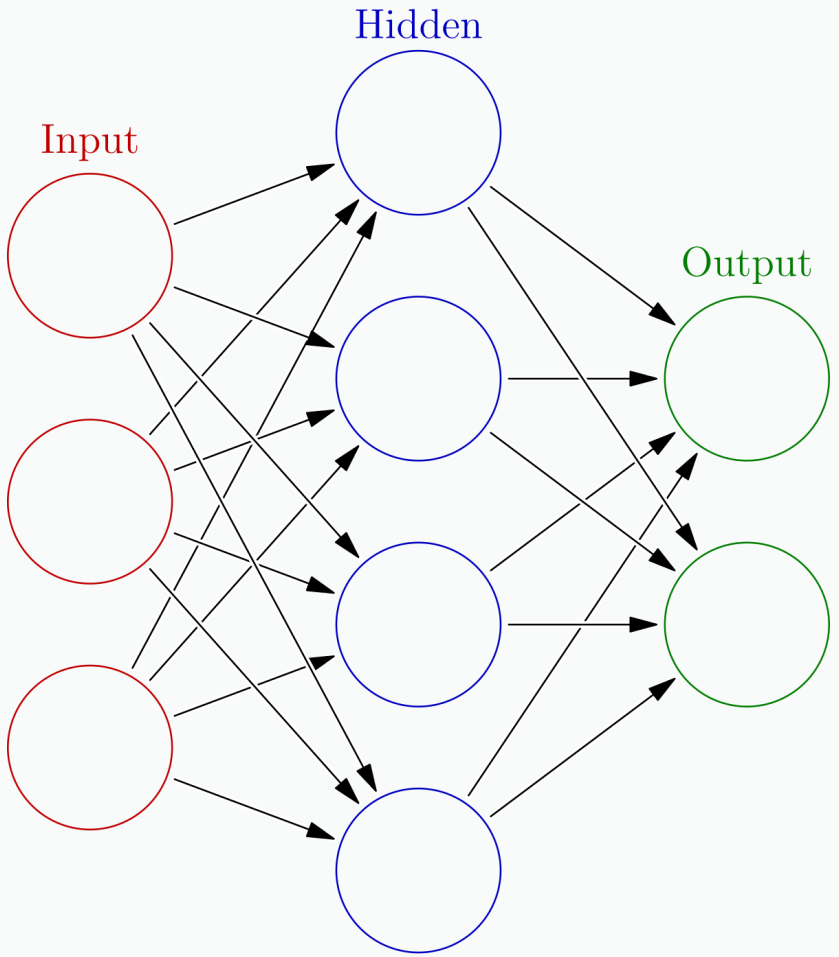
Neural Networks

(Think: layers of simple decision units)

- **What it does**: Stacks many “neurons” (tiny decision-makers) in layers to learn complex patterns.
- **Example**:
  - Layer 1 spots word rhythms,
  - Layer 2 spots emoji patterns,
  - Layer 3 combines both to guess topic.

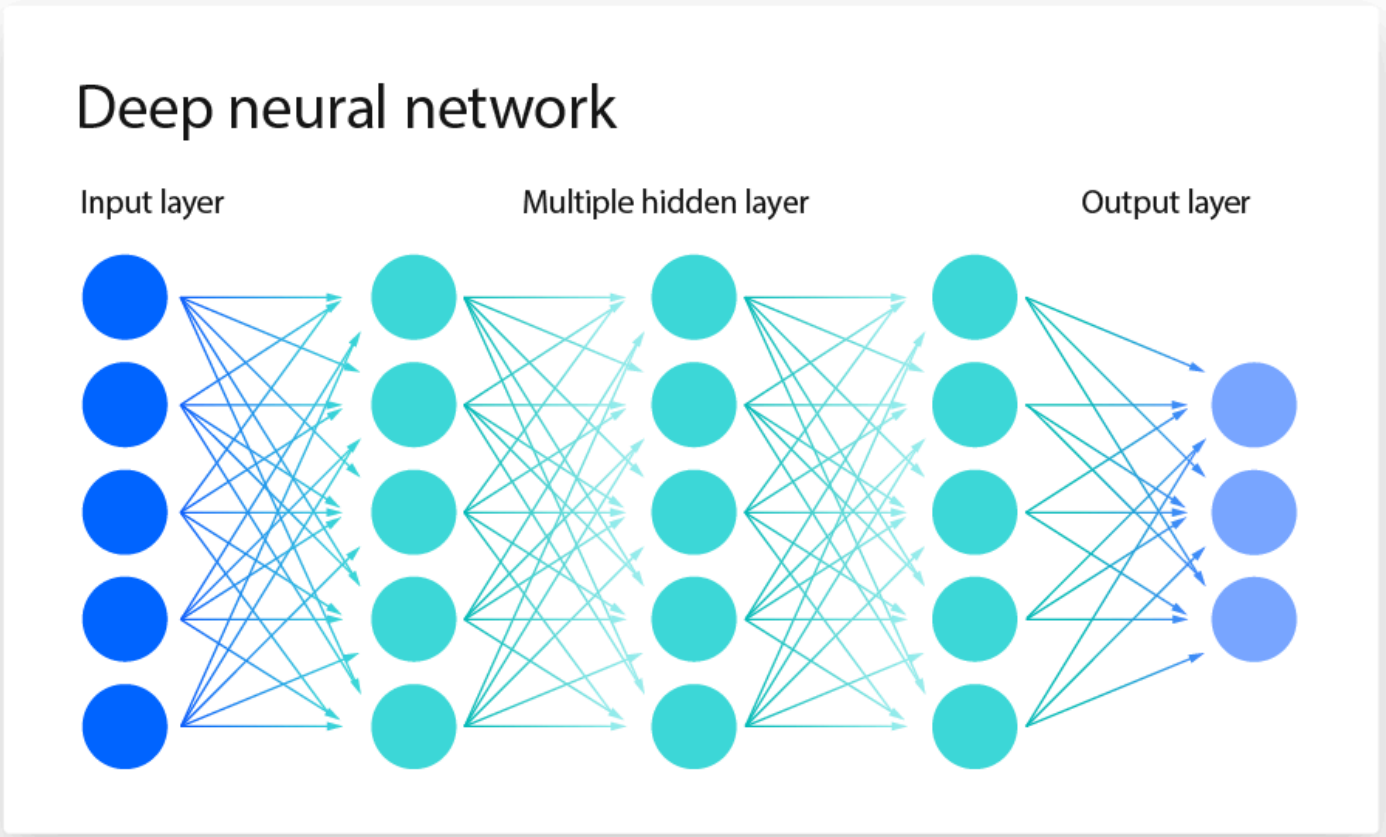
Neural Networks

(Think: layers of simple decision units)



Neural Networks

(Think: layers of simple decision units)



Neural Networks

(Think: layers of simple decision units)

- **What it does**: Stacks many “neurons” (tiny decision-makers) in layers to learn complex patterns.
- **Example**:
  - Layer 1 spots sentiment,
  - Layer 2 spots emoji patterns,
  - Layer 3 combines both to guess topic.
- **Good for**: Huge datasets—images, audio, text.
- **Watch out**: Needs a lot of data and can feel like a “black box.”

Quick Recap

Model type	When to choose it	Example goal
Linear Regression	Data looks straight	Predict reading time
Logistic Regression	Simple yes/no	Will it be a sports post?
k-NN	You want no “training” step	Quick topic guesses
Decision Tree	You need clear rules	Show your reasoning flow
Random Forest	You want strong accuracy	Best topic prediction
SVM	Many features, gap matters	Separate tricky cases
Neural Network	Huge, complex data	Spot deep patterns

Check-in

1 Go to wooclap.com

2 Enter the event code in the top banner

Event code
RCJFMQ

Enable answers by SMS

Next Week: Validation in Supervised ML

Lecture: Monday, May 12, 2025

Lab Session: Tuesday, May 13, 2025

Reading: The Search for Solid Ground in Text as Data: A Systematic Review of Validation Practices and Practical Recommendations for Validation

Objectives:

- Deepen your understanding of supervised machine learning.
- Learn how to validate classifiers to ensure correctness and generalizability.
- Practice different validation techniques (e.g., train/test split, cross-validation).

Thank You!