INSA ROUEN NORMANDIE

Machine Learning

Introduction

ITI 4 · Benoit Gaüzère

Course organization

- **12 lectures** × 1 h 30
- **12 practical sessions** × 1 h 30
- **Mini-project:** two 3-hour sessions in weeks 13 and 14 *(to be confirmed)*
- Slides, notebooks and project deliverables in **English**

Teaching coordination: **Benoit Gaüzère**

Teaching team : Maxime Alaarabiou, Gilles Gasso

Moodle

Course roadmap

Weeks	Theme
1–3	Problem formulation, evaluation, validation and leakage
4–7	Linear models, k-NN, SVM and kernels
8–10	Trees, ensembles, boosting and MLP
11–12	Dimensionality reduction and clustering
13–14	Reproducible mini-project

Each new model will be added to a common experimental workflow.

Recommended starting points

- Chloé-Agathe Azencott, *Introduction au Machine Learning*, 3rd ed., Dunod, 2025 — concise theory and exercises.
- Aurélien Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*, 3rd ed., O'Reilly, 2022 — applied Python workflows.
- Sebastian Raschka, Yuxi Liu and Vahid Mirjalili, *Machine Learning with PyTorch and Scikit-Learn*, Packt, 2022 — implementation and intuition.

All three cover most of the course from complementary viewpoints.

Deeper references

- Trevor Hastie, Robert Tibshirani and Jerome Friedman, *The Elements of Statistical Learning*, 2nd ed., Springer, 2009.
- Christopher M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006.
- Bernhard Schölkopf and Alexander J. Smola, *Learning with Kernels*, MIT Press, 2001/2018.

These are reference books: use selected chapters rather than reading them linearly.

Learning objectives

By the end of this lecture, you should be able to:

- distinguish the main machine-learning settings;
- formulate a learning problem using **samples, features, targets and predictions**;
- explain what **generalization** means;
- explain why training performance is not enough;
- describe the role of a train/test split;
- understand the basic idea of a first model: **k-nearest neighbours**.

Today’s content

1. From a real question to data
2. Main learning settings
3. Models, losses and empirical risk
4. Generalization and overfitting
5. A first evaluation protocol
6. A first algorithm: k-nearest neighbours

Warm-up — is this machine learning?

For each system, decide: **ML, not ML, or not enough information?**

1. A program sorts numbers using quicksort.
2. A mail server detects spam from previously labelled messages.
3. A thermostat switches heating on below 19 °C.
4. A program groups customers from their purchase histories.
5. A camera detects defective products on a production line.

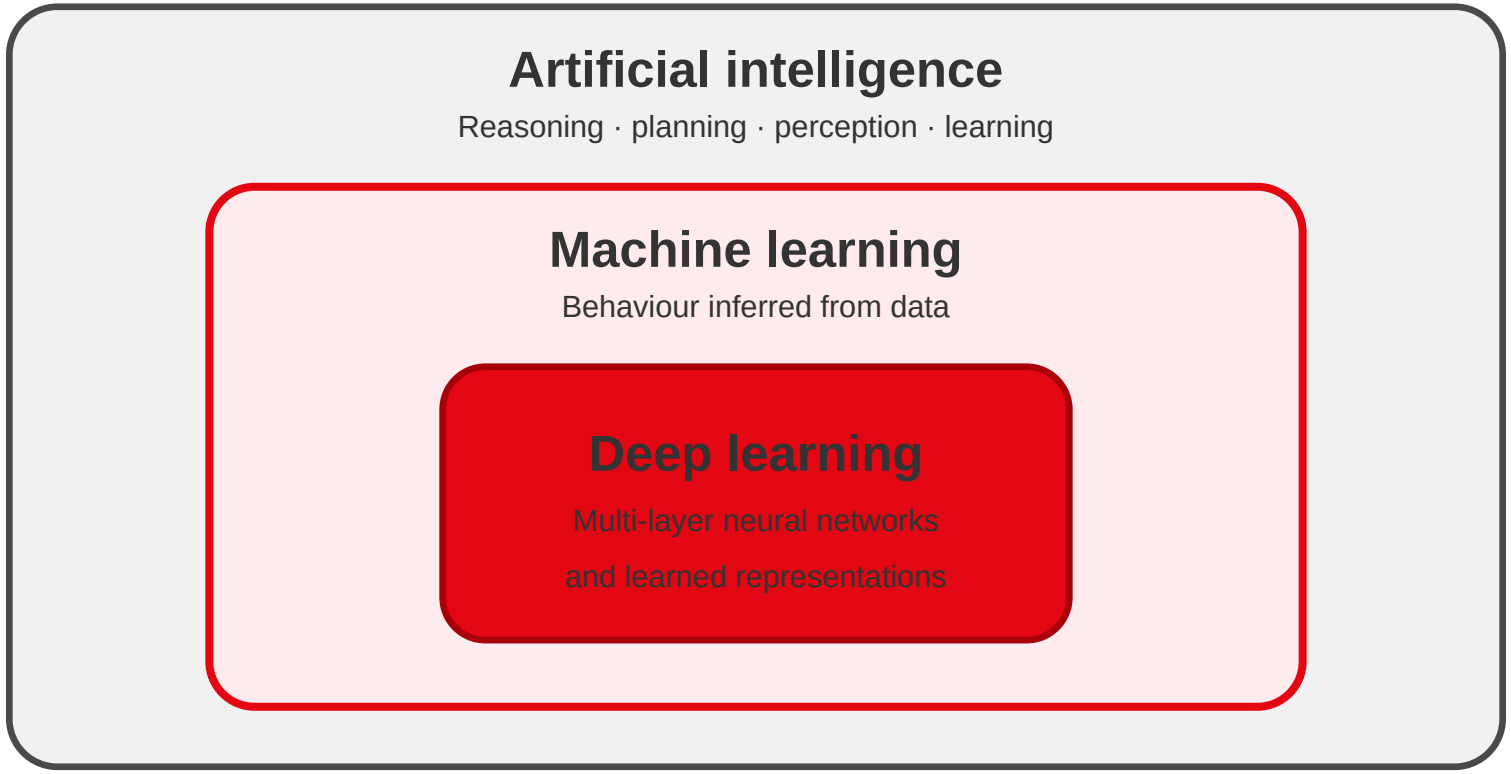
> What evidence would change your answer?

Warm-up — discussion

1. **Quicksort:** not ML; its behaviour follows explicitly programmed rules.
2. **Spam detection:** ML if its decision rule is estimated from labelled messages; it could also be a hand-written rule system.
3. **Thermostat:** not ML as stated; the threshold is fixed by design.
4. **Customer groups:** unsupervised ML if groups are inferred from the observations.
5. **Defect detection:** not enough information; either learned or conventional vision is possible.

The final question asks for evidence about **how the behaviour was obtained**: training data, an objective, learned parameters and evaluation on new cases.

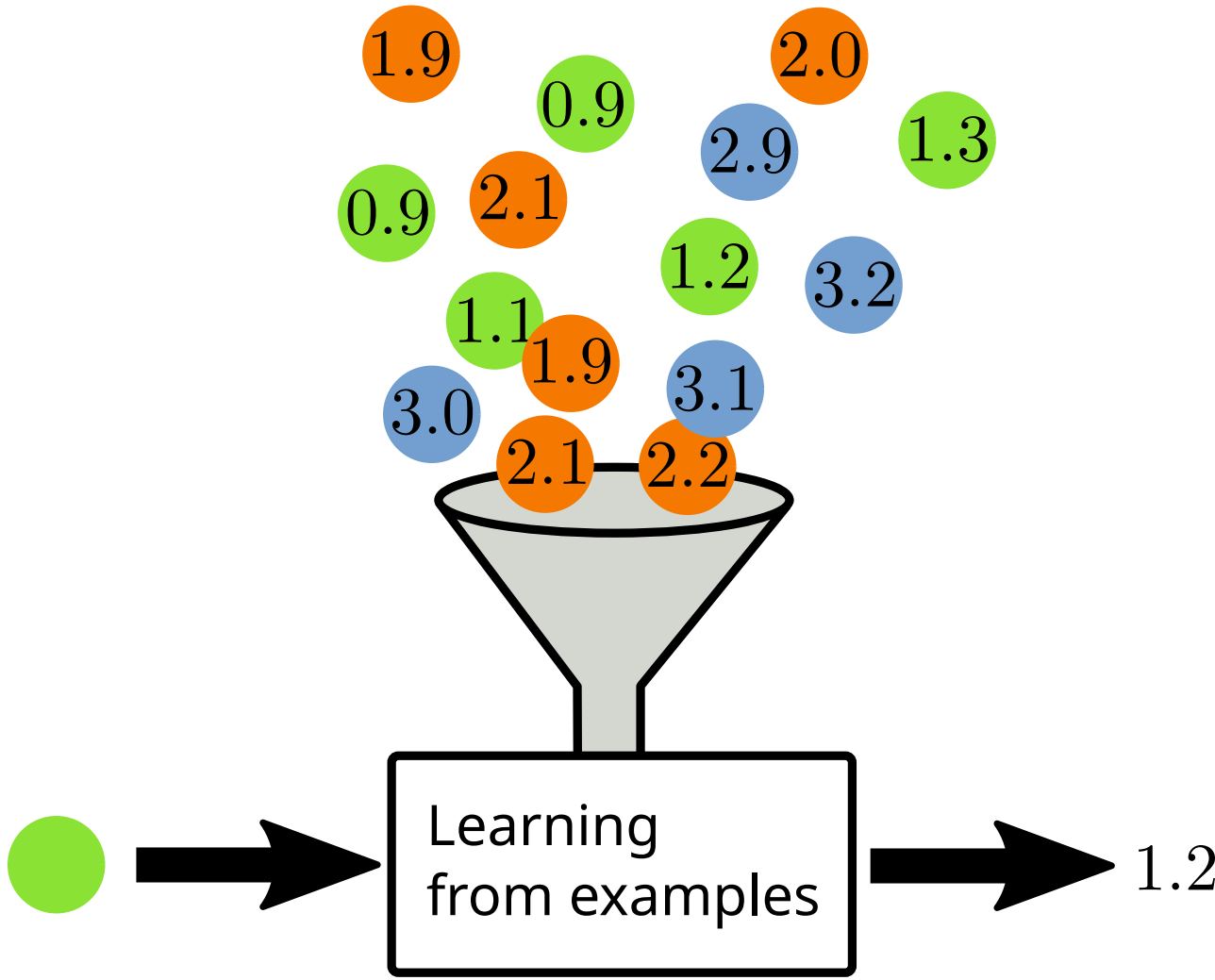
AI, machine learning and deep learning



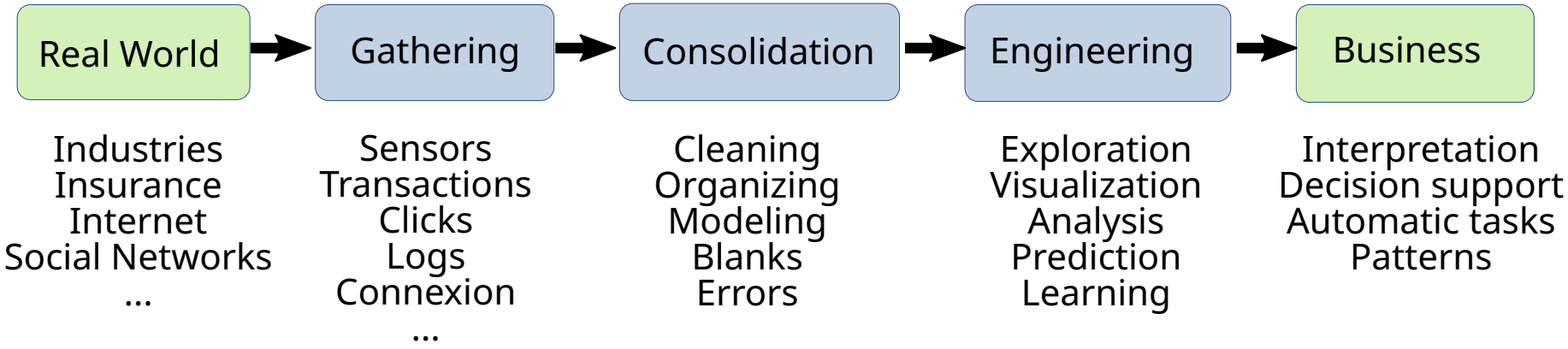
ML is one approach to AI. Deep learning is one family of ML methods.

Learning from examples

The learned rule is a **model**. It is inferred from a finite sample, not written case by case.



Machine learning is an experimental discipline



A model is only one component of a complete study.

1. From a real question to data

The basic representation

	Features — columns of X				Target y
	sepal length	sepal width	petal length	petal width	species
sample 1	5.1	3.5	1.4	0.2	setosa
sample 2	6.4	3.2	4.5	1.5	versicolor
sample 3	6.3	3.3	6.0	2.5	virginica

One row = one prediction situation

Here: predict the species from four measured features

n samples and p features form $X \in \mathbb{R}^{n \times p}$; supervised learning also uses targets y.

Vocabulary

Concept	Alternative names	Example: house-price prediction
Sample	observation, instance	one house
Feature	variable, attribute, input	area, district, age
Target	label, response, output	sale price
Model	estimator, predictor	fitted regression function
Prediction	estimated output	predicted sale price

Notation usually matters less than making each object explicit.

A learning problem begins before the algorithm

Ask, in this order:

1. **What decision or prediction is required?**
2. What is one sample?
3. What information is available **at prediction time**?
4. What is the target, if any?
5. What constitutes a serious error?
6. How will future data differ from the available data?

Choosing an algorithm comes later.

Formulation exercise — industrial diagnosis

A sensor records vibrations from a rotating machine. The objective is to warn operators before a bearing fails.

- What is one sample: one timestamp, one window, or one machine?
- What could the features be?
- What exactly is the target?
- Which information exists when the warning must be issued?
- How should machines and time periods be split between train and test?
- Which errors are most costly?

Industrial diagnosis — possible formulation

- **Sample:** a fixed-length signal window associated with one machine and one time interval.
- **Features:** spectral energy, peaks, envelope statistics or learned signal descriptors.
- **Target:** failure within a specified prediction horizon, defined before modelling.
- **Available information:** measurements recorded strictly before the warning time.
- **Split:** hold out complete machines and later time periods—not random overlapping windows.
- **Error costs:** a missed failure may be more costly than a false alarm, but excessive alarms make the system unusable.

Different operational choices produce different, equally plausible learning problems.

Case study — Iris classification

- **Sample:** one iris flower
- **Features:** sepal length/width and petal length/width
- **Target:** setosa, versicolor or virginica
- **Task:** three-class supervised classification
- **Prediction-time constraint:** the four measurements are known; the species is not

Question: would a model trained on this curated dataset necessarily work on photographs of wild irises?

Iris Versicolor | Iris Setosa | Iris Virginica

The data-generating process matters

Rows in a table are not necessarily independent.

- several measurements may come from one patient;
- overlapping windows may come from one signal;
- several conformers may represent one molecule;
- observations may be ordered in time;
- duplicates may be hidden under different identifiers.

The split must reflect the intended future use.

2. Main learning settings

A map of common ML tasks

Setting	Information available	Typical objective
Supervised	Inputs X and targets y	Predict y for new X
Unsupervised	Inputs X only	Discover or represent structure
Semi-supervised	Few labelled and many unlabelled samples	Improve prediction using both
Anomaly detection	Mostly normal, sometimes labelled data	Identify unusual observations

These settings may coexist in one application.

Supervised learning

We observe pairs

$$\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$$

and learn a function

$$f : \mathcal{X} \to \mathcal{Y}$$

that should predict the target of a **new** sample.

The nature of $\mathcal{Y}$ determines the task.

Classification

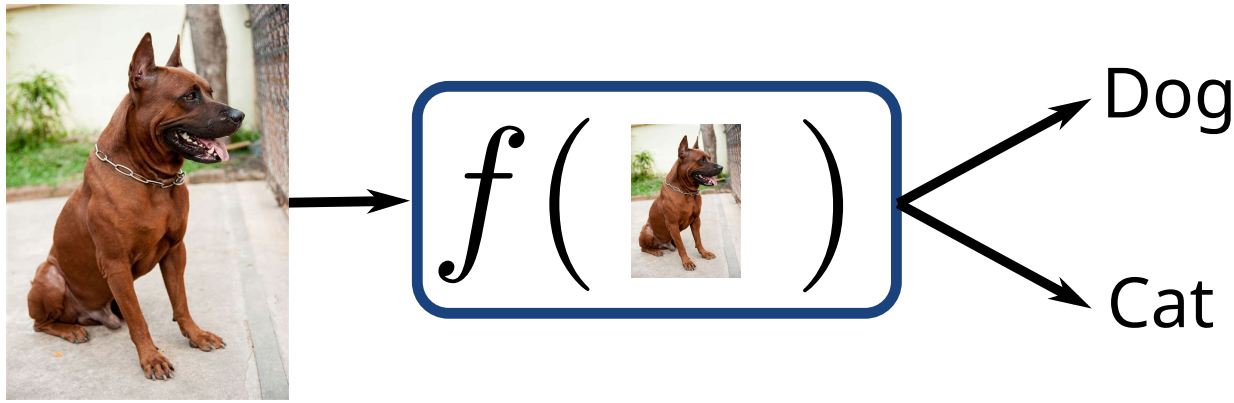
The target belongs to a finite set of classes:

$$y \in \{1, \ldots, K\}$$

Examples:

- spam / not spam;
- normal / defective product;
- species recognition;
- type of activity from a sensor signal.

The output may be a class, a score or an estimated probability.



Regression

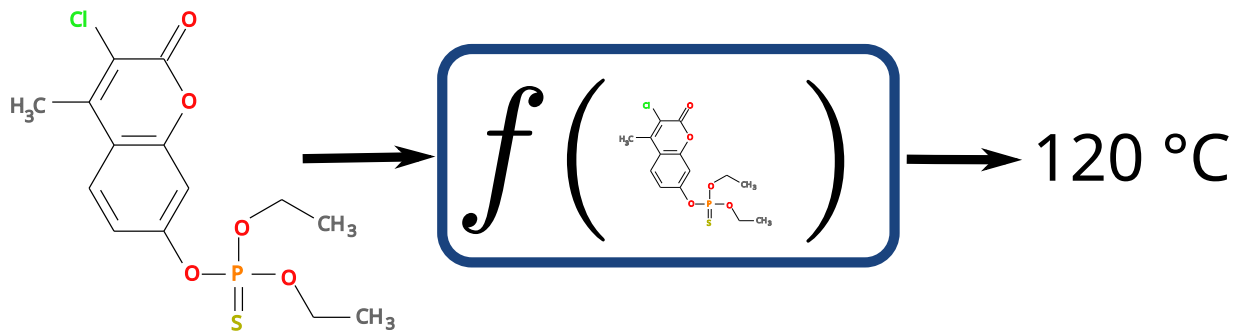
The target is numerical:

$$y \in \mathbb{R} \quad \text{or} \quad \mathbb{R}^q$$

Examples:

- energy consumption;
- molecular solubility;
- remaining useful life;
- travel time.

A numerical label does not automatically imply that squared error is the right objective.

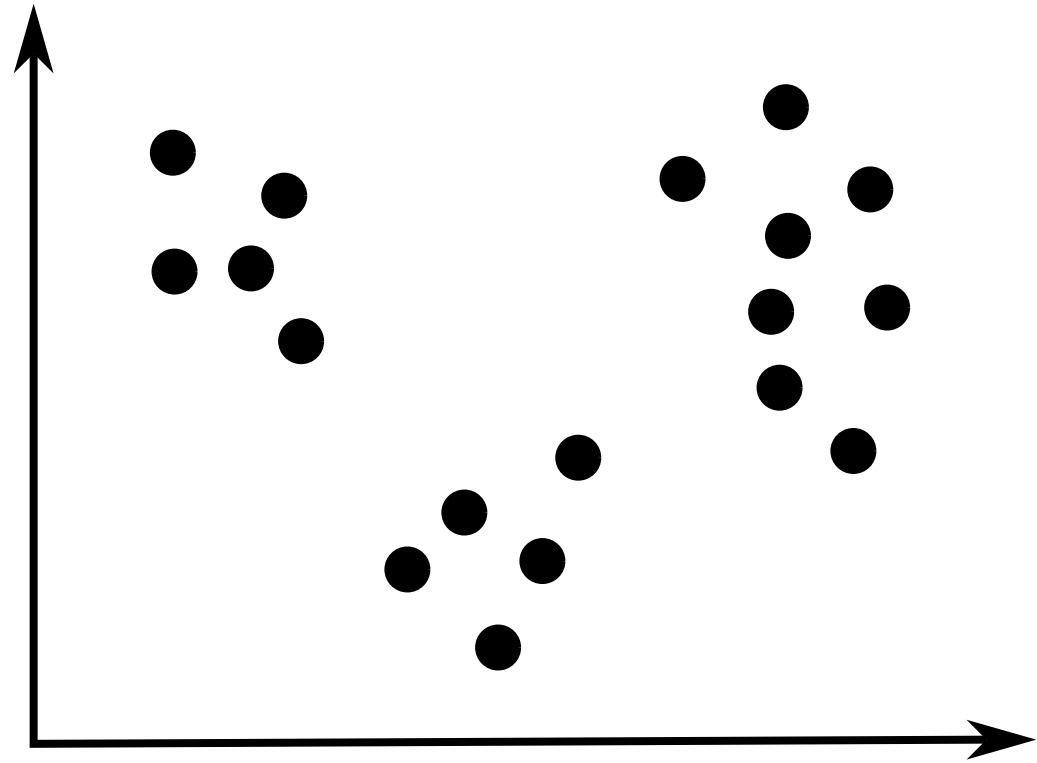


Unsupervised learning

No target y is provided. We only observe X.

- **clustering:** form groups of similar samples;
- **dimensionality reduction:** compress or visualize the data;
- **density estimation:** model where observations occur;
- **representation learning:** construct useful features.

There is often no single ground truth against which to evaluate the result.



Clustering

The objective is to find groups using similarities in X.

Applications include:

- exploratory segmentation;
- discovering operating regimes;
- grouping documents or molecules;
- summarizing a large dataset.

A clustering is a learnt description, not a fact hidden in every dataset.

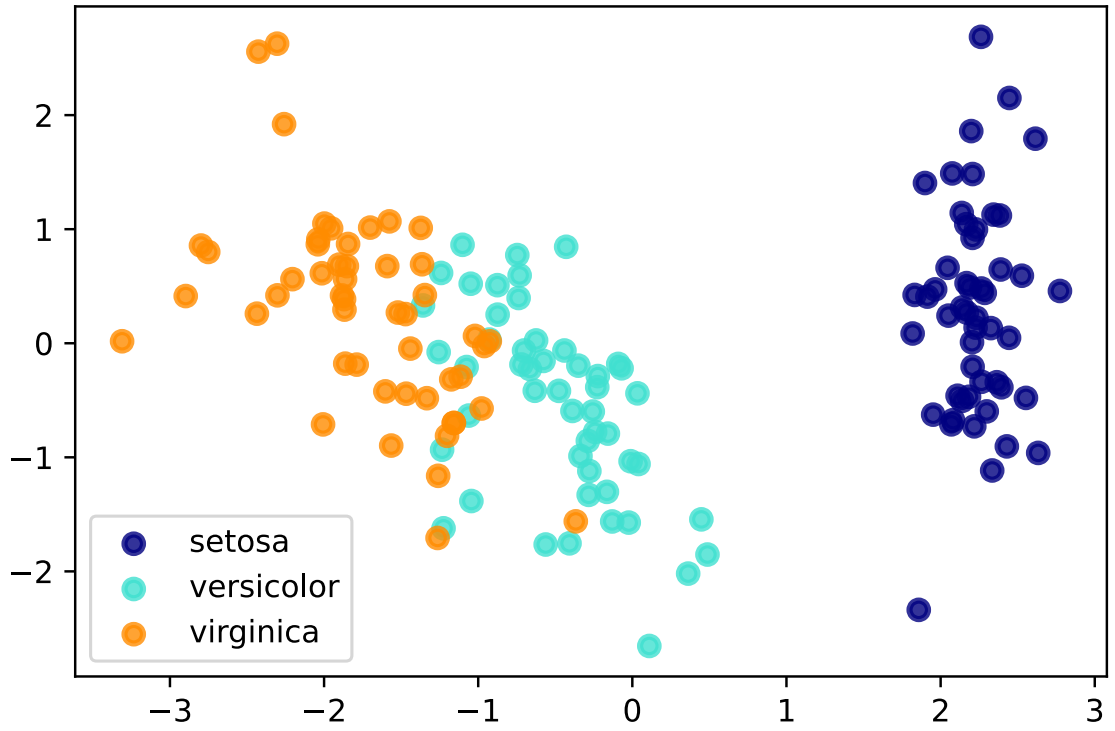
Dimensionality reduction

Map high-dimensional data to a lower-dimensional representation:

$$\mathbf{x} \in \mathbb{R}^p \longrightarrow \mathbf{z} \in \mathbb{R}^d, \quad d \ll p$$

Purposes: visualization, compression, denoising or preprocessing.

A two-dimensional plot is an approximate visualization of the original geometry.



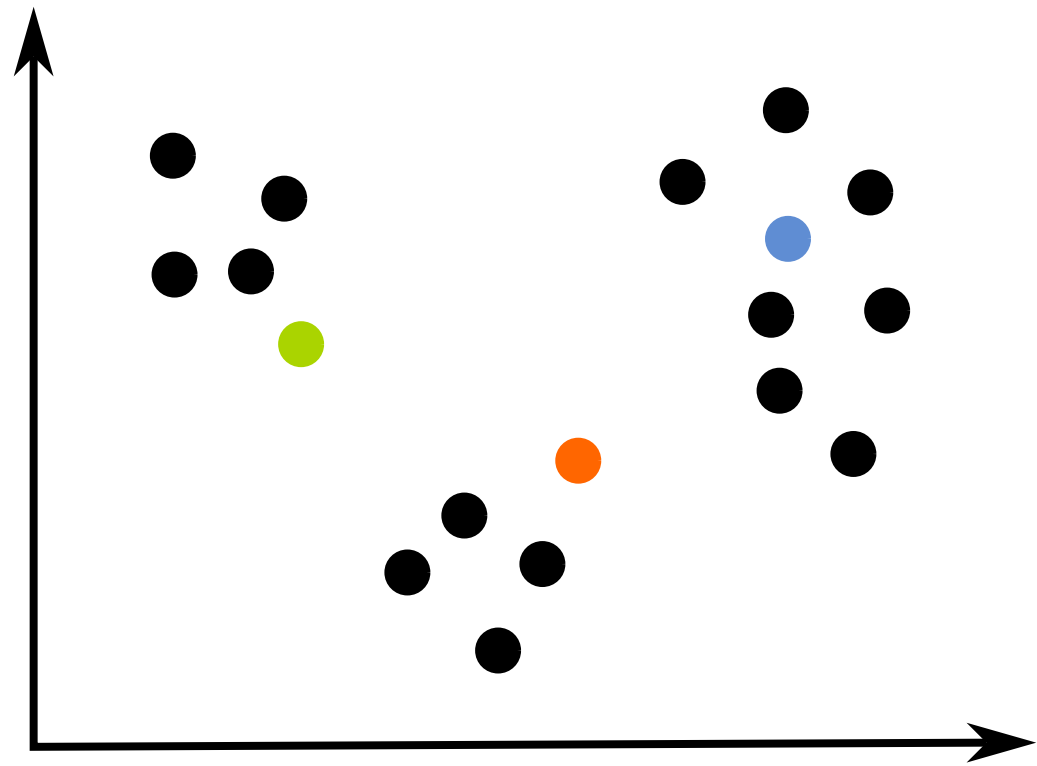
Semi-supervised learning

Suppose labels are expensive but unlabelled inputs are abundant:

$$\{(\mathbf{x}_i, y_i)\}_{i=1}^{n_l} \quad + \quad \{\mathbf{x}_j\}_{j=1}^{n_u}, \qquad n_u \gg n_l$$

The aim is to exploit structure in the unlabelled data to improve a supervised task.

This requires assumptions: unlabelled data are not automatically useful.



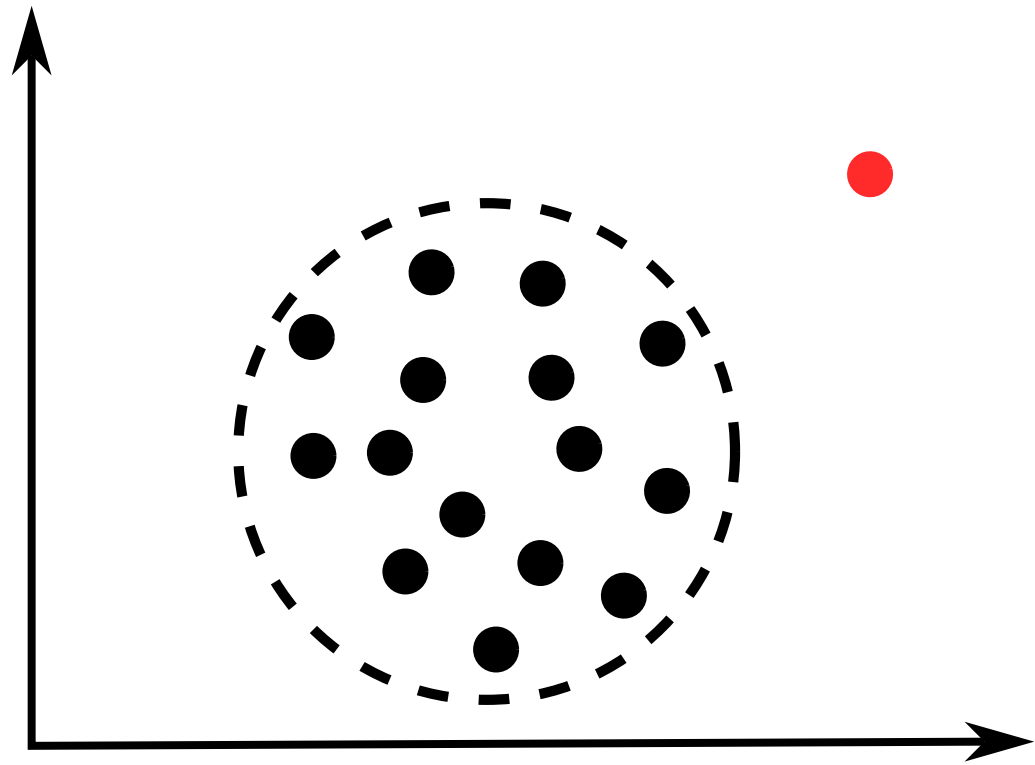
Anomaly detection

Goal: find observations that are unusual relative to a reference population.

Examples:

- a rare industrial fault;
- fraudulent activity;
- an abnormal physiological signal;
- an out-of-distribution input.

Challenges: anomalies are rare, heterogeneous and often poorly labelled. Accuracy can be meaningless here.



Identify the setting

For each objective, name the learning setting and its output:

1. Predict the concentration of a compound from a spectrum.
2. Visualize 2,000-dimensional image descriptors in 2D.
3. Group machines by their vibration profiles.
4. Detect unusual network traffic when only normal traffic is available.
5. Classify images with 500 labelled and 50,000 unlabelled examples.

3. What does it mean to learn?

A model is a parameterized function

We choose a family of functions $\{f_\theta\}$ and estimate its parameters from data.

Examples:

- linear model: $f_\theta(\mathbf{x}) = \theta^\top \mathbf{x} + b$;
- decision tree: parameters encode splitting rules;
- nearest neighbours: training examples define local predictions;
- neural network: parameters are weights across layers.

The model family introduces **assumptions** about the data.

Errors must be quantified

A **loss function** measures the cost of a prediction:

$$\mathcal{L}\big(f_\theta(\mathbf{x}_i), y_i\big)$$

Examples:

- squared error for regression;
- absolute error for robust regression;
- zero-one error for classification;
- log loss for probabilistic classification.

The loss is a modelling choice: it defines what “good” means during learning.

Empirical risk minimization

The average training loss is the empirical risk:

$$\widehat{R}_{\text{train}}(f_\theta) = \frac{1}{n}\sum_{i=1}^{n} \mathcal{L}\big(f_\theta(\mathbf{x}_i), y_i\big)$$

Learning often means finding parameters that make this quantity small.

But the real objective is not to predict the training examples.

The real objective: generalization

> A model generalizes when it performs well on relevant observations that were not used to fit it.

Generalization depends on:

- the amount and quality of data;
- whether future data follow a comparable process;
- the model assumptions and complexity;
- preprocessing and hyperparameter choices;
- the evaluation protocol.

Training error is optimistic

The parameters were selected to reduce training error.

Therefore:

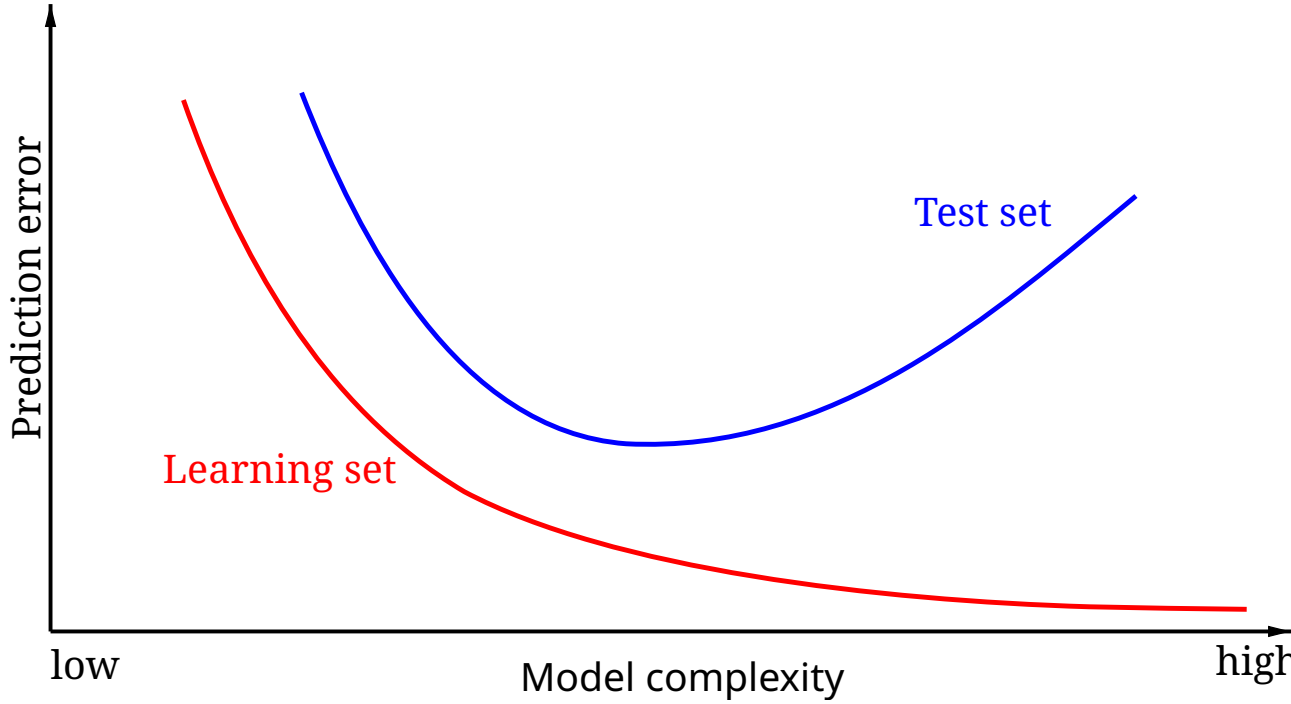
$$\widehat{R}_{\text{train}} \quad \text{is not an unbiased estimate of future error.}$$

Extreme example: a lookup table can memorize every training label and still fail on every new input.

Successful memorization is not necessarily successful learning.

Underfitting and overfitting

- **Underfitting:** the model cannot capture useful structure.
- **Overfitting:** the model captures accidental details of the training sample.
- The best complexity depends on data quantity, noise and the task.



Thought experiment

Two classifiers are evaluated on the training set:

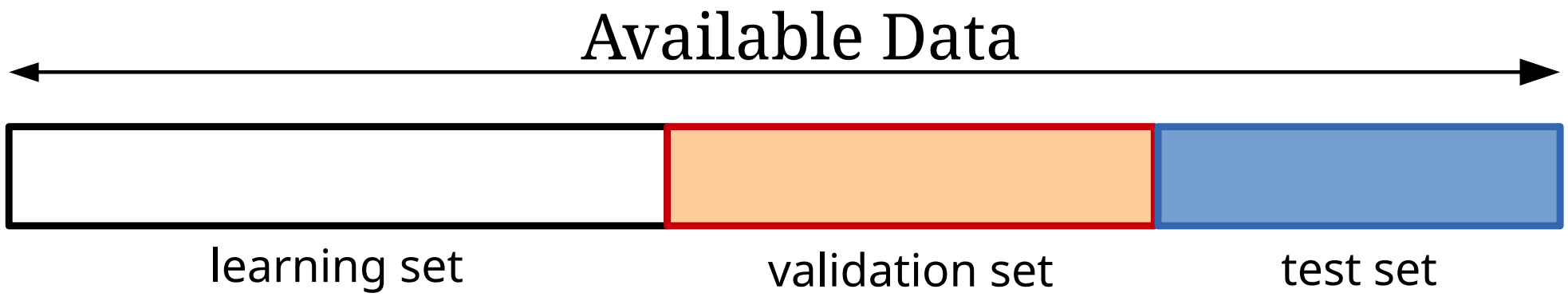
- Model A: 82% accuracy
- Model B: 100% accuracy

Which one should we deploy?

> What additional information is required before answering?

4. A first evaluation protocol

Separate learning from evaluation



- **Training set:** estimate model parameters.
- **Validation:** make modelling and hyperparameter choices.
- **Test set:** estimate final performance once choices are fixed.

Today we begin with train/test. The next lecture develops validation and cross-validation.

The test set represents the future task

A useful test set must be:

- unseen during training and model selection;
- drawn according to the intended use;
- separated by group or time when required;
- large enough to provide a meaningful estimate;
- evaluated with an appropriate metric.

Random splitting is a tool, not a universal rule.

Data leakage

Leakage occurs when training uses information that would not genuinely be available for future predictions.

Examples:

- scaling using the whole dataset before splitting;
- selecting variables using all labels;
- near-duplicates in train and test;
- using measurements recorded after the event;
- windows from the same signal in both sets.

Leakage produces impressive but misleading results.

A metric answers a question

“Performance” is not a single number.

- Is every class equally important?
- Are false positives and false negatives equally costly?
- Is a large regression error much worse than a small one?
- Do predicted probabilities need to be reliable?
- Is average performance sufficient, or do subgroups matter?

We will study metrics and uncertainty in the next block.

A baseline is mandatory

Before a sophisticated model, compare against something simple:

- majority-class or random prediction;
- mean/median prediction in regression;
- a simple linear model;
- a domain-specific rule already in use.

If a complex system barely improves on a baseline, its added cost may not be justified.

5. A first learning algorithm: k-NN

k-nearest neighbours

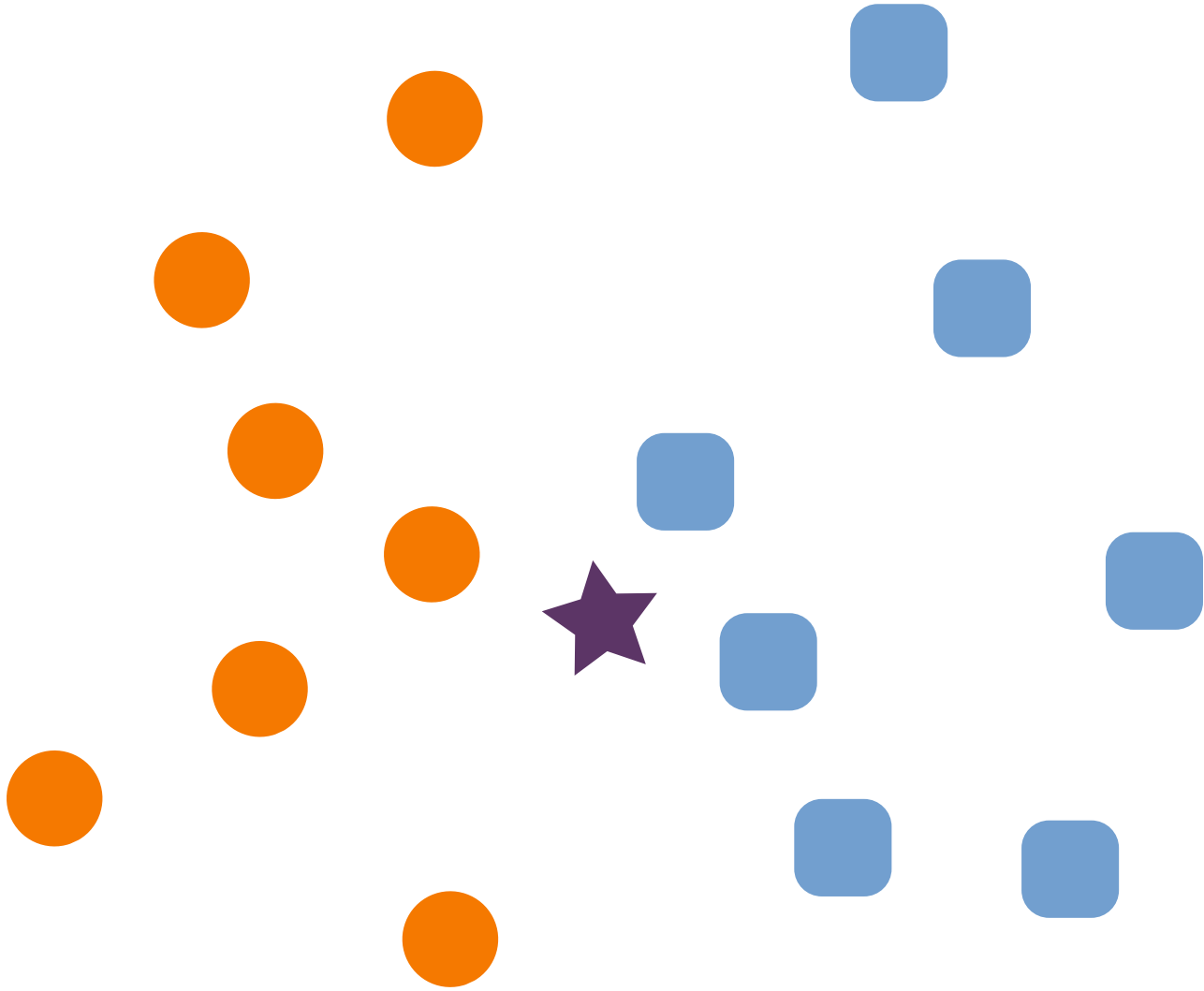
To predict a new input $\mathbf{x}$:

1. measure its distance to training samples;
2. select the k nearest neighbours;
3. aggregate their targets.

Classification uses a vote; regression usually uses an average.

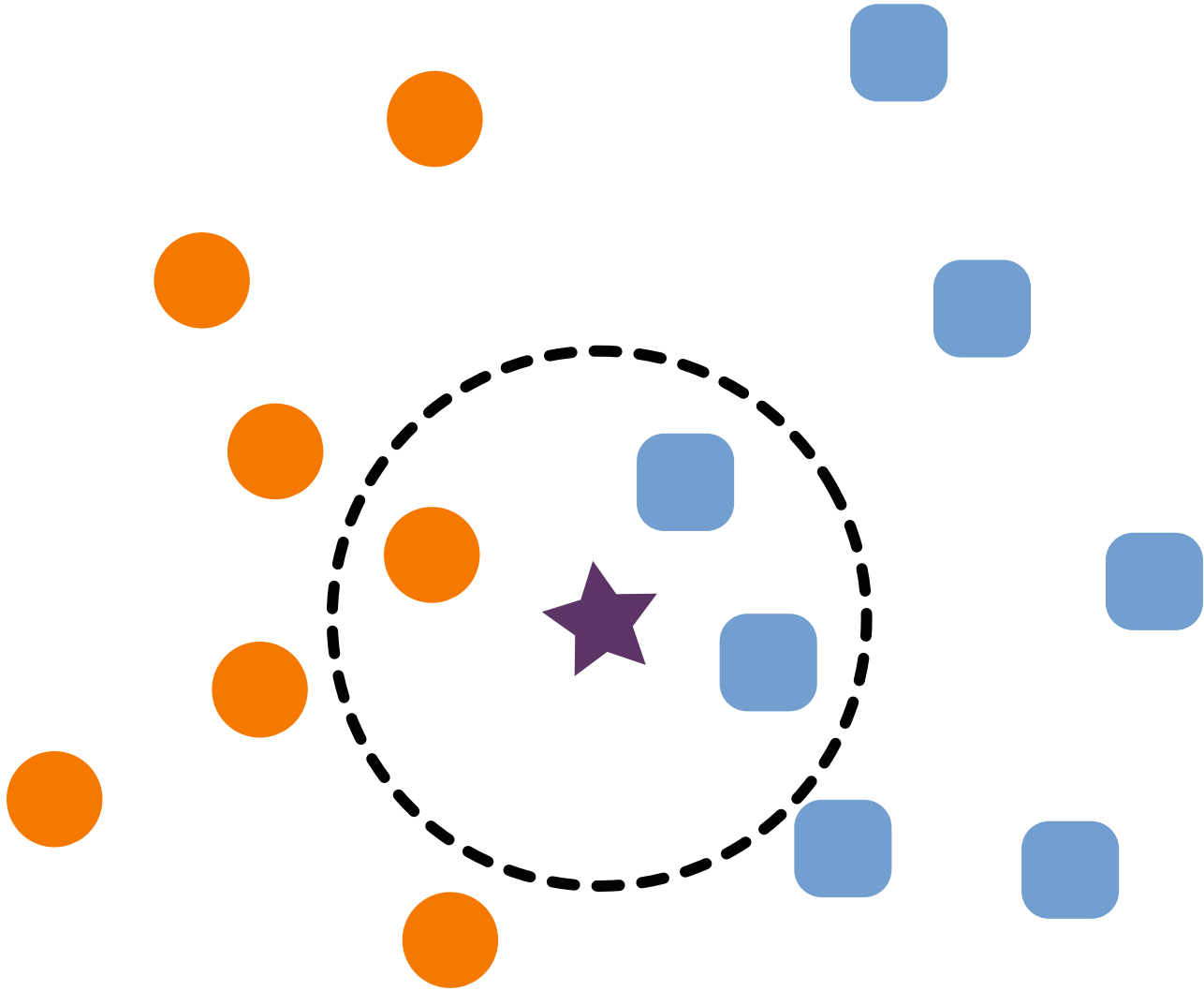
Step 1 — locate the query

The training set is the model’s memory. No explicit decision boundary has been fitted yet.



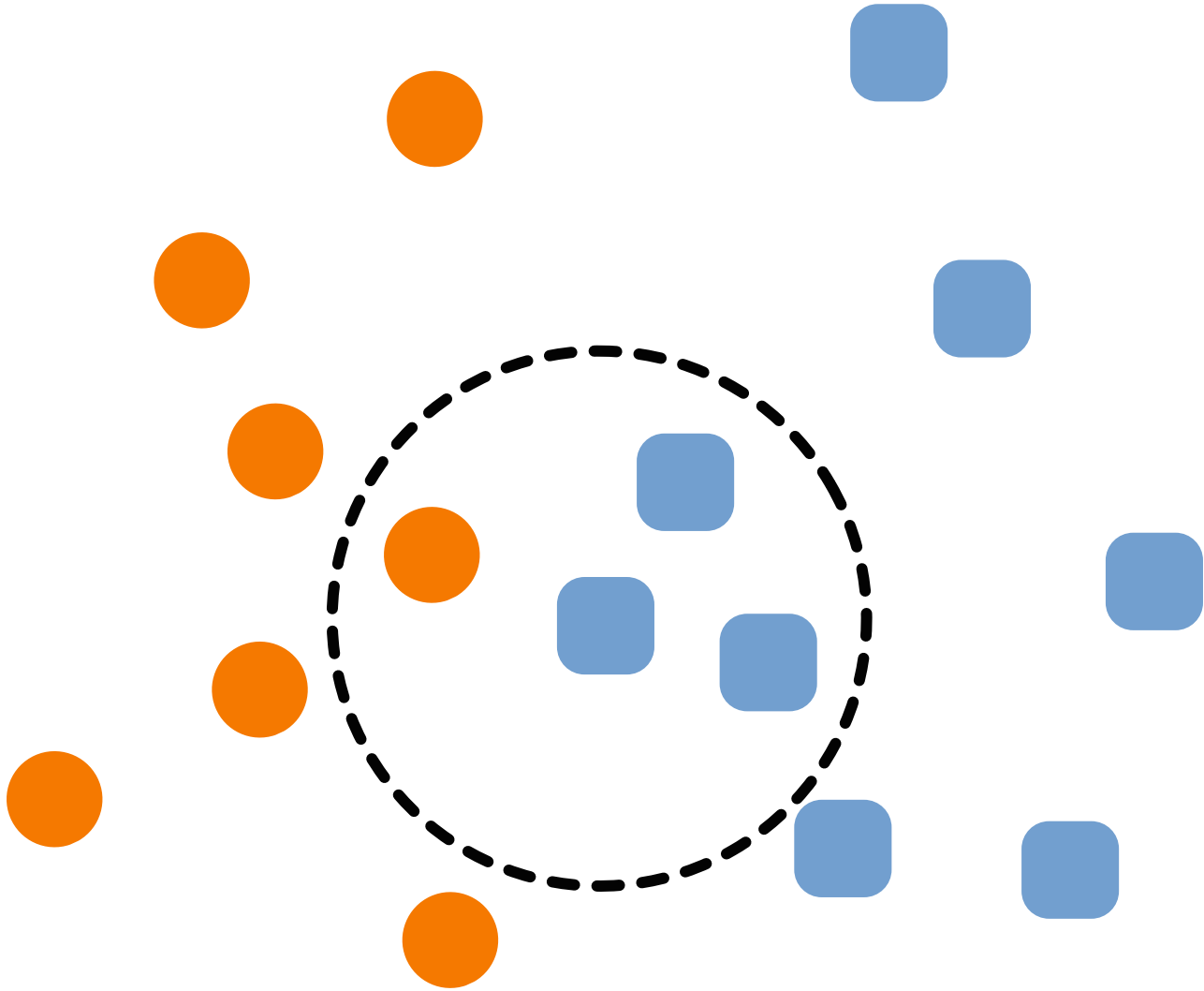
Step 2 — choose a neighbourhood

The value of k controls how local the decision is.



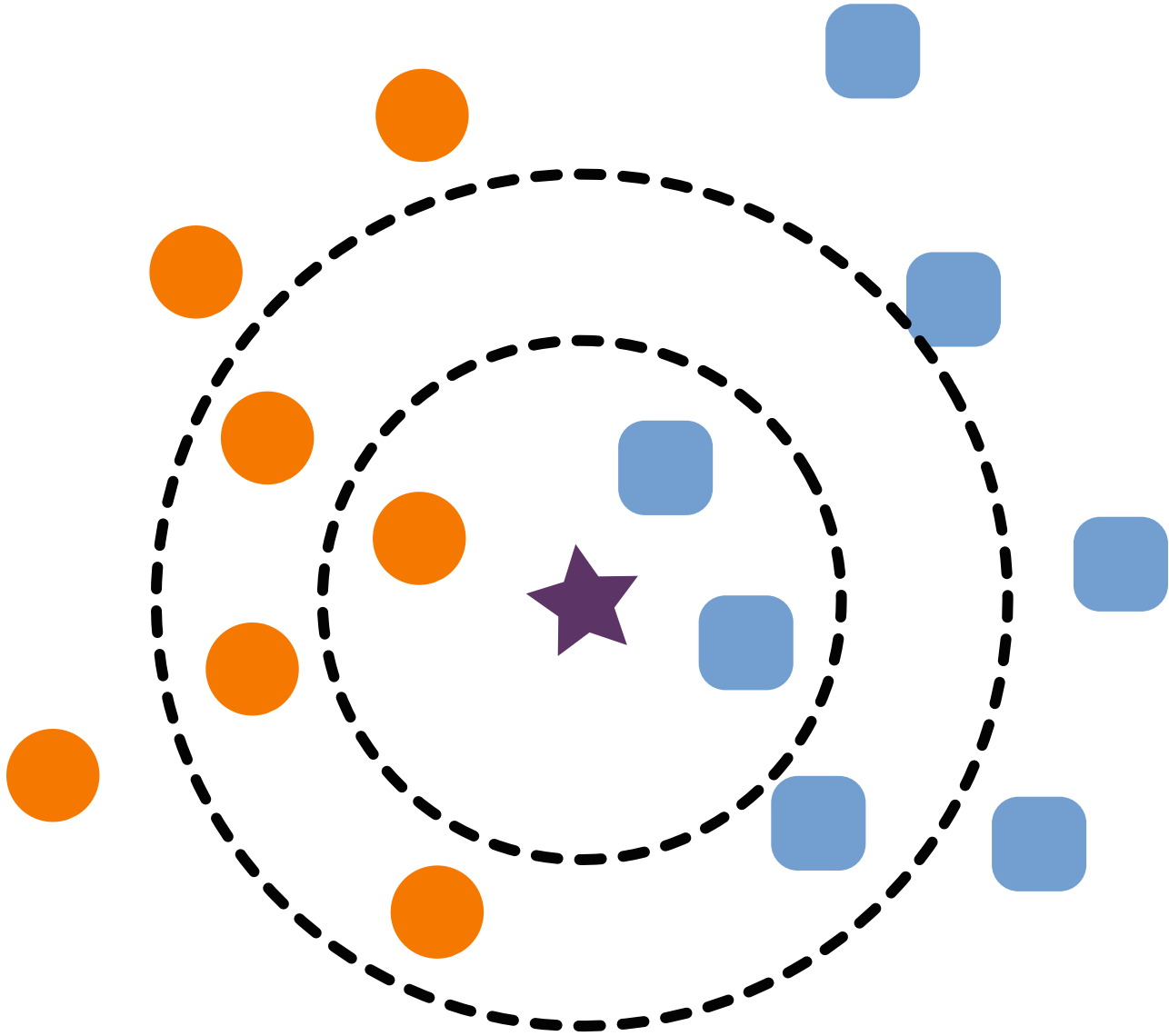
Step 3 — vote

The predicted class is the majority class among the selected neighbours.



The decision can change according to k

The predicted class is the majority class among the selected neighbours.



What can go wrong with k-NN?

- Features measured on large numerical scales dominate the distance.
- Irrelevant variables can hide useful neighbours.
- A small k is flexible and sensitive to noise.
- A large k produces smoother, more biased decisions.
- Prediction becomes costly for large training sets.
- Distances become less informative in high dimension.

k-NN makes the importance of **representation and preprocessing** immediately visible.

Models encode different assumptions

Model family	Main intuition
Linear model	One global linear relationship
k-NN	Nearby inputs should have similar outputs
Decision tree	Recursive axis-aligned rules
Kernel SVM	Large-margin separation in a feature space
Random forest / boosting	Combine many trees
MLP	Learn nonlinear compositions

There is no universally best model independent of the data and objective.

6. From a score to a trustworthy study

A complete ML study

1. Formulate the task and prediction-time constraints.
2. Inspect the data-generating process.
3. Define the split and evaluation metric.
4. Establish a baseline.
5. Build preprocessing and model together.
6. Tune choices without touching the final test set.
7. Evaluate uncertainty and analyse errors.
8. Report assumptions, limitations and reproducibility information.

The course will progressively implement this workflow.

Reproducibility begins now

Record at least:

- dataset version and filtering rules;
- target definition and feature availability;
- split strategy and random seeds;
- preprocessing learned from training data;
- model and hyperparameters;
- metrics and software versions;
- limitations and known failure cases.

A notebook is evidence of a protocol, not merely a sequence of cells that ran once.

Practical session 0 — preview

You will receive a small dataset and investigate:

- What is the target?
- What is available at prediction time?
- How should the data be split?
- Which metric is appropriate?
- How do train and test scores differ?
- How do scaling and k affect a k-NN classifier?

Five messages to remember

1. **Start with the question, not the algorithm.**
2. A dataset represents a sampling and measurement process.
3. Learning means aiming for performance on relevant unseen data.
4. Training performance alone cannot demonstrate generalization.
5. Evaluation design is part of the model, not an afterthought.

Exit ticket

Before leaving, write one sentence for each question:

1. What is generalization?
2. Give one example of data leakage.
3. Why can two reasonable train/test splits lead to very different conclusions?
4. Which assumption makes k-NN meaningful?

Next lecture

Learning and evaluating correctly

Pipelines · metrics · validation · hyperparameters · leakage · uncertainty