Learning from data

Guilherme Alves

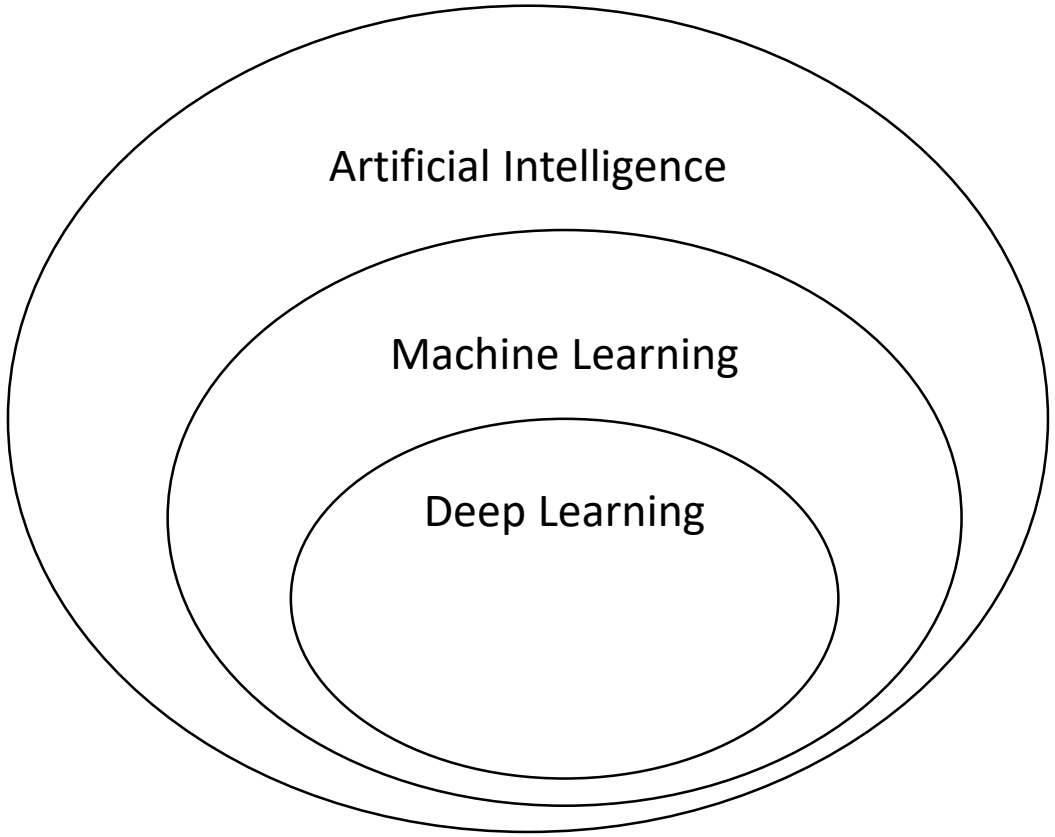
guilherme.alves-da-silva@univ-lorraine.fr

From AI “to” Machine Learning (ML)

- Subfield of artificial intelligence
- Learning from data

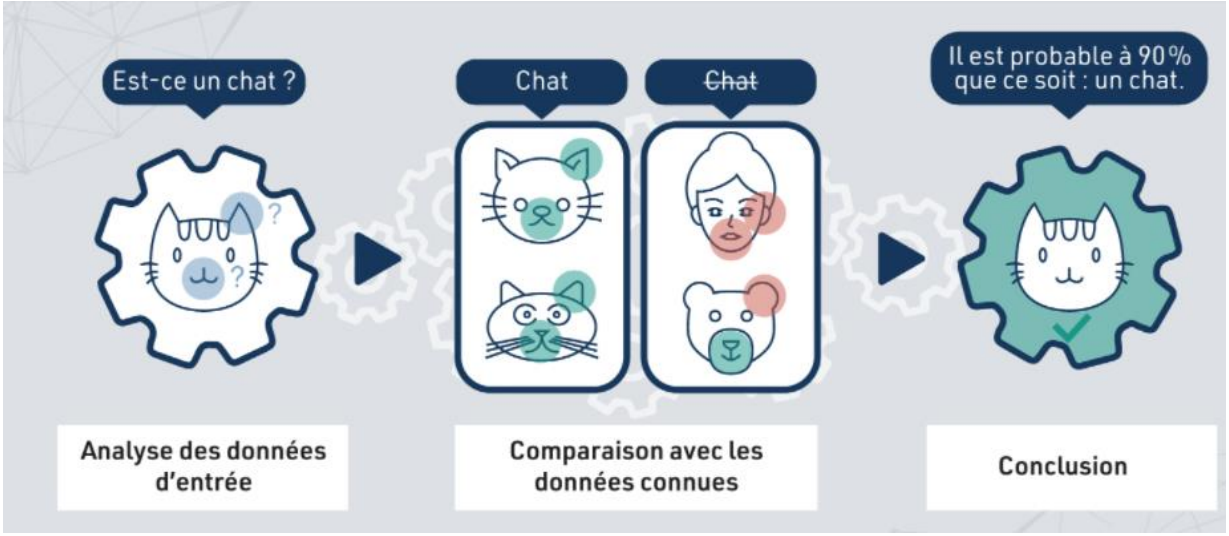
- “Input”: data
- “Output”: (in general) a ML model

- ML model: solves a problem (e.g. classification, regression, …)

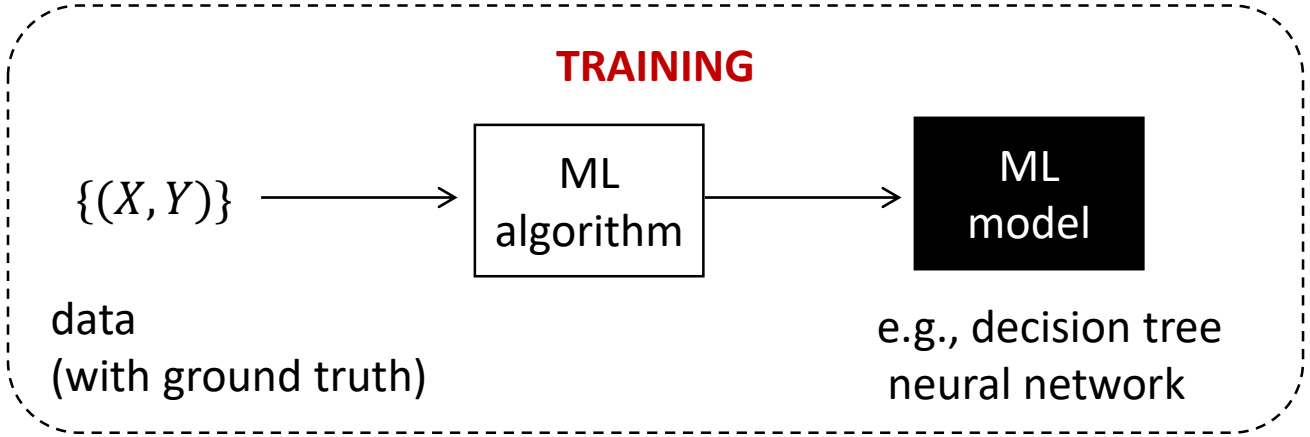


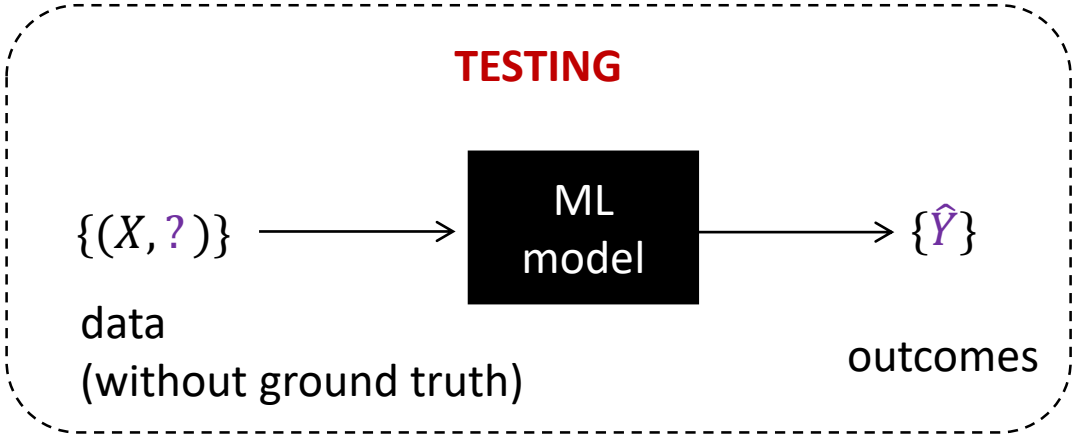
Supervised learning

- Image classification: cat detection



Supervised learning





EVALUATION

		Actual outcome	
		$Y = 1$	$Y = 0$
Pred	$\hat{Y} = 1$	TP (true positive)	FP (false positive)
	$\hat{Y} = 0$	FN (false negative)	TN (true negative)

Classification

- **Identify** a set of **objects** with measurable characteristics into different **classes**
  - Measurable characteristics: attributes, features

- **Example**: Iris plant

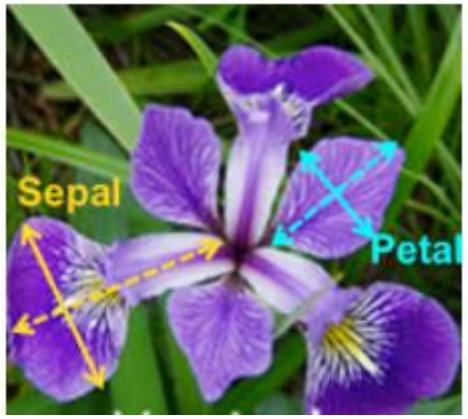
Iris Virginica

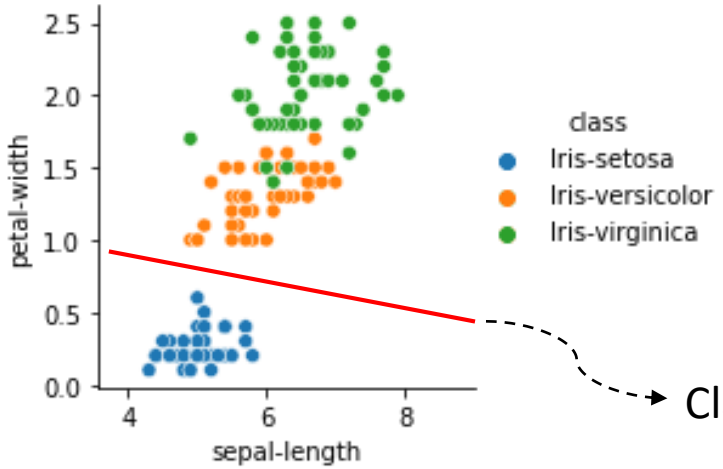
Iris Setosa

Iris Versicolor

Classification: Iris dataset

- Features:





Classifier

		X		Y
SepalLength	**SepalWidth**	**PetalLength**	**PetalWidth**	**Species (CLASS)**
5.2	3.4	1.4	0.2	Iris-setosa
7.2	3.0	5.8	1.6	Iris-virginica
5.1	2.5	3.0	1.1	Iris-versicolor
6.7	2.5	5.8	1.8	Iris-virginica
5.0	3.4	1.6	0.4	Iris-setosa
5.2	3.5	1.5	0.2	Iris-setosa
6.2	2.9	4.3	1.3	Iris-versicolor
5.6	3.0	4.1	1.3	Iris-versicolor
6.4	3.2	4.5	1.5	Iris-versicolor
5.5	4.2	1.4	0.2	Iris-setosa

Decision rules

- What is the simplest rule for classification?
- Set of rules: **knowledge base**

PetalWidth < 0,7 → **Class** = Iris-setosa

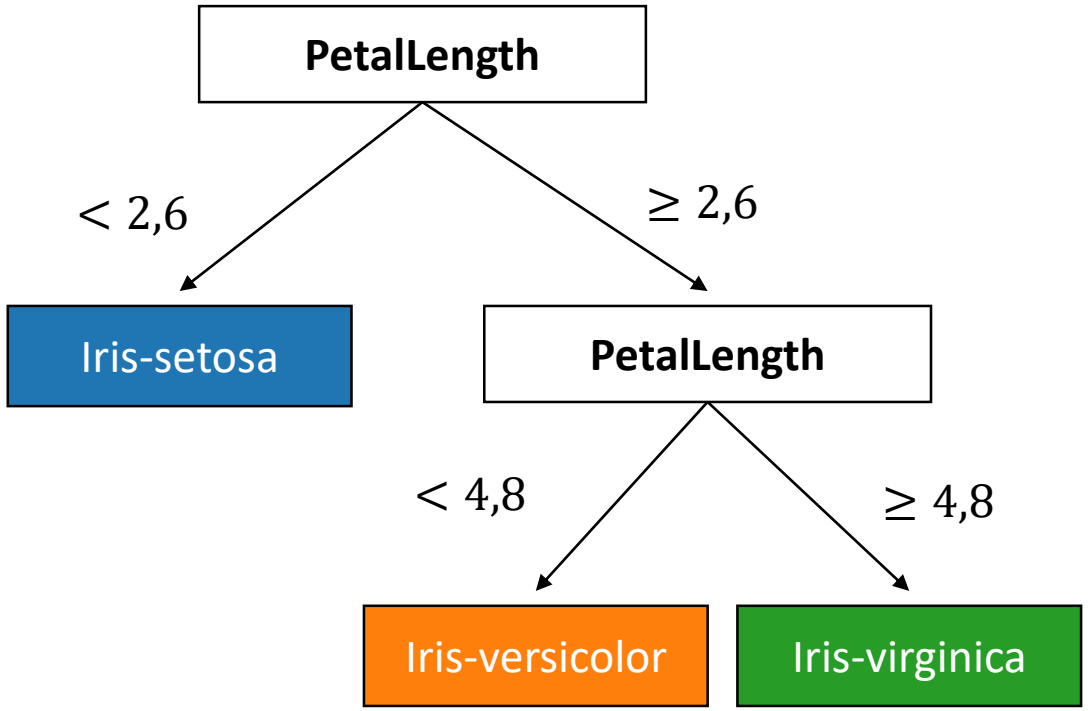
PetalWidth > 0,7 and **PetalWidth** ≤ 1,5 → **Class** = Iris-versicolor

SepalLength	SepalWidth	PetalLength	PetalWidth	Species (CLASS)
5.2	3.4	1.4	0.2	Iris-setosa
7.2	3.0	5.8	1.6	Iris-virginica
5.1	2.5	3.0	1.1	Iris-versicolor
6.7	2.5	5.8	1.8	Iris-virginica
5.0	3.4	1.6	0.4	Iris-setosa
5.2	3.5	1.5	0.2	Iris-setosa
6.2	2.9	4.3	1.3	Iris-versicolor
5.6	3.0	4.1	1.3	Iris-versicolor
6.4	3.2	4.5	1.5	Iris-versicolor
5.5	4.2	1.4	0.2	Iris-setosa

Decision tree: Example (Iris)

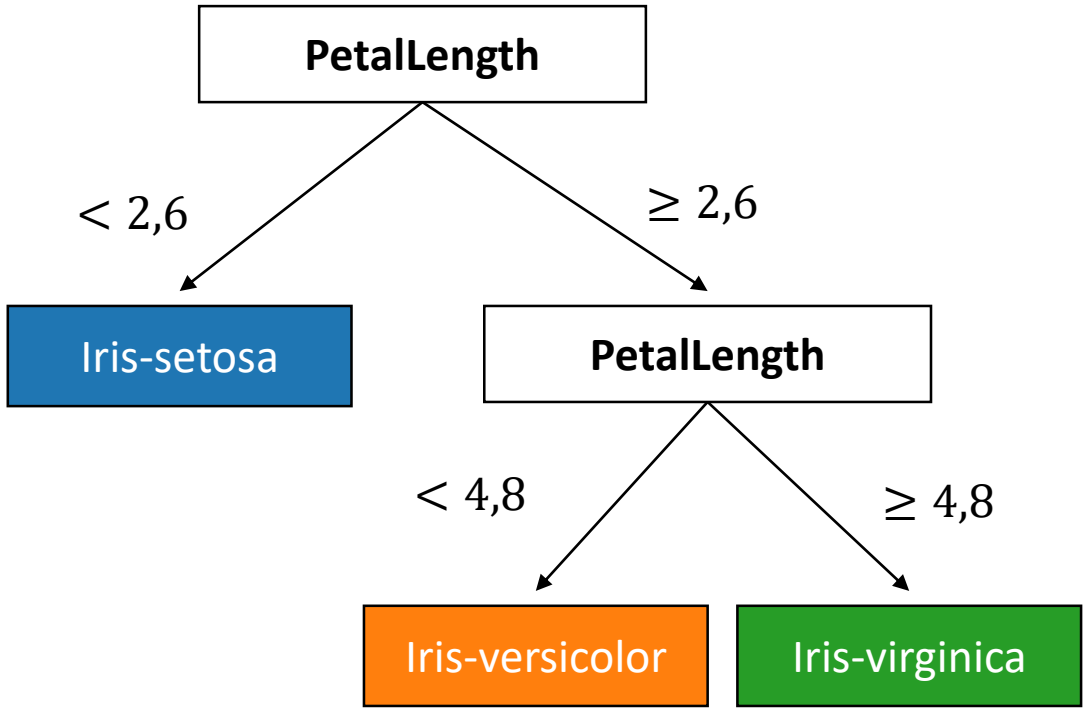
What would a **decision tree <u>look like</u>** in the case of the Iris dataset?

SepalLength	SepalWidth	PetalLength	PetalWidth	Species (CLASS)
5.2	3.4	1.4	0.2	Iris-setosa
7.2	3.0	5.8	1.6	Iris-virginica
5.1	2.5	3.0	1.1	Iris-versicolor
6.7	2.5	5.8	1.8	Iris-virginica
5.0	3.4	1.6	0.4	Iris-setosa
5.2	3.5	1.5	0.2	Iris-setosa
6.2	2.9	4.3	1.3	Iris-versicolor
5.6	3.0	4.1	1.3	Iris-versicolor
6.4	3.2	4.5	1.5	Iris-versicolor
5.5	4.2	1.4	0.2	Iris-setosa



Decision tree

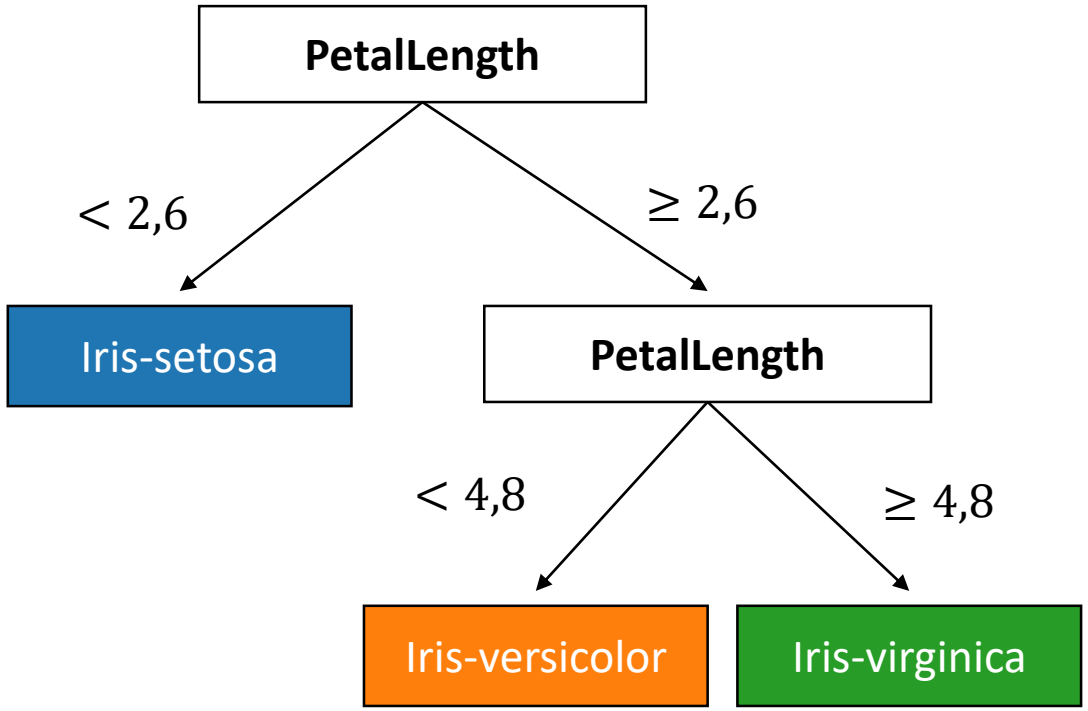
- Divide-and-conquer approach
- Represents a set of rules in a form of a **tree** to classify a set of objects
  - ***"Inverted"* tree**: root on top and leaves on the bottom
- Leaves indicate a **class label**
- Other notes: a division in the decision space
- **Classification rule**: a path between the root and a leaf



Leaf nodes

Decision tree

- How to build a tree?
  - Induction
  - Given a new node, how to select a feature?
  - Splitting criterion!
    - Classification error, Entropy, Gini, …
- Several algorithms
  - Ex.: J4.8 (Weka)



Evaluation: Iris

		Actual outcome	
		$Y =$ Iris-setosa	$Y \neq$ Iris−setosa
Prediction	$\hat{Y} =$ Iris-setosa	TP (true positive)	FP (false positive)
	$\hat{Y} \neq$ Iris-setosa	FN (false negative)	TN (true negative)

$$accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions} = \frac{TP+TN}{TP+TN+FP+FN}$$

$$precision = \frac{TP}{TP+FP}$$

$$recall = \frac{TP}{TP+FN}$$

$$F = 2 * \frac{precision * recall}{precision + recall}$$

Evaluation

- How to split a dataset into training and test sets?

SepalLength	SepalWidth	PetalLength	PetalWidth	Species (CLASS)
5.2	3.4	1.4	0.2	Iris-setosa
7.2	3.0	5.8	1.6	Iris-virginica
5.1	2.5	3.0	1.1	Iris-versicolor
6.7	2.5	5.8	1.8	Iris-virginica
5.0	3.4	1.6	0.4	Iris-setosa
5.2	3.5	1.5	0.2	Iris-setosa
6.2	2.9	4.3	1.3	Iris-versicolor
5.6	3.0	4.1	1.3	Iris-versicolor
6.4	3.2	4.5	1.5	Iris-versicolor
5.5	4.2	1.4	0.2	Iris-setosa

Test set

Training set

Evaluation

- **Cross-validation**: k-fold method
- Each object is taken into account
  - Training: $k - 1$ times
  - Testing: once
- Error, precision, accuracy, …
  - Averaging over all k iterations !