

# A (brief) Introduction to Supervised Learning

DynamiTe Workshop, Florence, Sept. 24-28, 2018



Charles BOUVEYRON<sup>1</sup> & Etienne CÔME<sup>2</sup>

<sup>1</sup> Professor of Statistics, Université Côte d'Azur  
Chair Inria in "Data Science"  
charles.bouveyron@unice.fr - @cbouveyron

<sup>2</sup> Researcher in Statistical Learning, IFSTTAR  
etienne.come@ifsttar.fr - @comeetie

"Ce qui est simple est toujours faux.  
Ce qui ne l'est pas est inutilisable."

Paul Valéry

Material available at:

<https://github.com/DynamiteStaff/R-workshops>

*A friendly advice: clone the repository now (200 Mo to download)!*

# Outline

---

1. Introduction
2. The supervised learning process
3. A selection of supervised learning methods
4. Deep learning
5. Tutorial: Classification of road patterns

## Introduction: Learning from data...

---

"Learning" is the central task of artificial intelligence (AI):

- the field is currently moving a lot (new problems, new solutions, ...),
- learning in some situations is still a challenging problem:
  - high-dimensional ( $p$  large),
  - big or as stream ( $n$  large),
  - heterogeneous (categorical, functional, networks, texts, ...).

There are important needs (lot of expectations also!) in many fields of Science:

- Medicine / Biology,
- Astrophysics,
- Digital Humanities,
- ...

# A motivating example: cytology

---

## Cytology:

- it is the study of cells in terms of structure, function and chemistry,
- for the diagnosis of disease (we focused on cervical cancer).

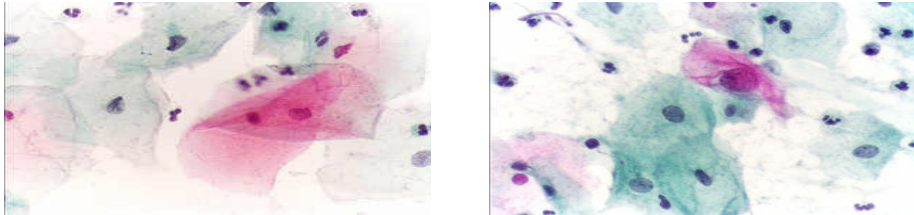


Figure: Normal (left) and abnormal (right) pap smears.

## Cervical cancer detection:

- it is an important public health field which is currently treated mostly manually,
- screening by human experts is complicated by the amount of cells (20 000/smear),
- and by the very small proportion of cancer cells (less than 1%).

# A motivating example: cytology

---

## Our data (BC Cancer Agency):

- 20 smears which contains between 4 000 and 10 000 cells,
- each nucleus is described by 111 features (morphological, photometric or texture features),
- only 0.52% of the cells are diseased cells.

## Classification is useful in this context:

- for building supervised classifiers which can select the most likely cancer cells,
- for helping experts in labeling the learning data through weakly-supervised classification,
- for selecting discriminative variables which can be used in a semi-automatic process.

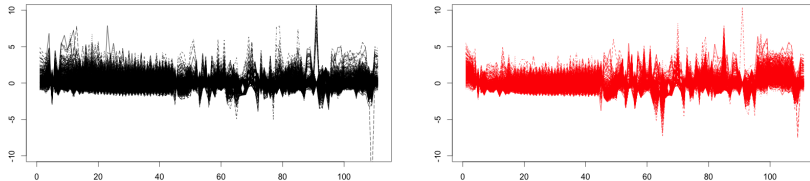


Figure: Control and (cervical) cancer data.

# Introduction: Learning from data...

One task, several families of approaches:

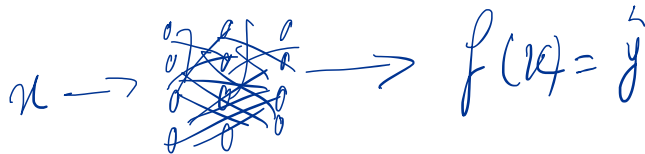
- Statistical learning



- Machine learning



- Deep learning



- ...



# Introduction: Learning from data...

---

Learning is a two-head problem:

Supervised

- data + labels
- X      y
- tasks :
  - classification
  - regression
  - ...

Unsupervised

- data only X
- ↳ try to discover  
some patterns
- tasks :
  - + clustering
  - + visualization

# Introduction: Learning from data...

---

Methods are specific to each task:

Supervised

+ classification:  
+ KNN  
+ SVM  
+ log. reg  
+ xx DA

+ Regression: + EM  
+ ridge  
+ Lasso

Unsupervised

- Clustering : + k-means  
+ CAH  
+ EM algo.  
  
- Visu : + PCA  
+ t-SNE  
---

# Introduction: Supervised learning

Supervised learning is also a field with different sub-tasks:

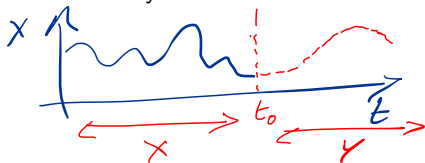
- classification:

$x \in \mathcal{X} \rightarrow$  explanatory data  
 $y \in \{1 \dots k\} \rightarrow$  target variable

- regression:

$x \in \mathcal{X}$   
 $y \in \mathbb{R} \text{ (or } \mathbb{R}^d)$

- time series analysis:



}  $(x, y)$   
 $\hookrightarrow f(x) = y$

- ...

# Outline

---

1. Introduction
2. The supervised learning process
3. A selection of supervised learning methods
4. Deep learning
5. Tutorial: Classification of road patterns

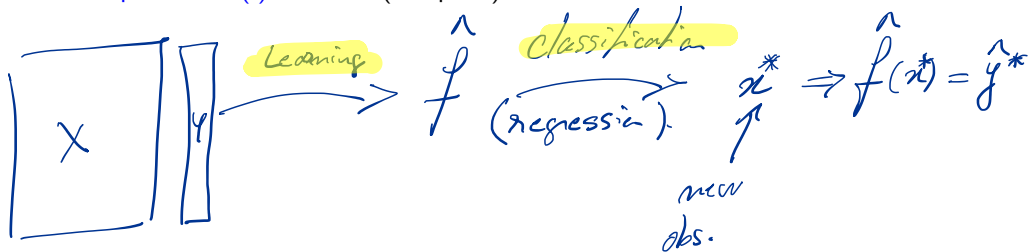
# The supervised learning process

The material: a set of (complete) data

$$(X, Y) = \left\{ (x_1, y_1), \dots, (x_n, y_n) \right\}$$

↑                      ↖  
image/  
vector                      a label (from an expert)

The goal: learn a predictor  $f(\cdot)$  from the (complete) data



## Measuring the learning performance

One comfortable thing of working in the supervised context is:

- to be able to measure the performance of the learned predictor,

(i) Classification error:

$$E_{\mathcal{H}} = \frac{1}{n} \sum_{i=1}^n \mathbb{1}\{y_i \neq \hat{y}_i\} \in [0,1] \leq 1 - \frac{1}{K}$$

(ii) Confusion table

	pred. 0	2
True 1	50	2
2	4	8

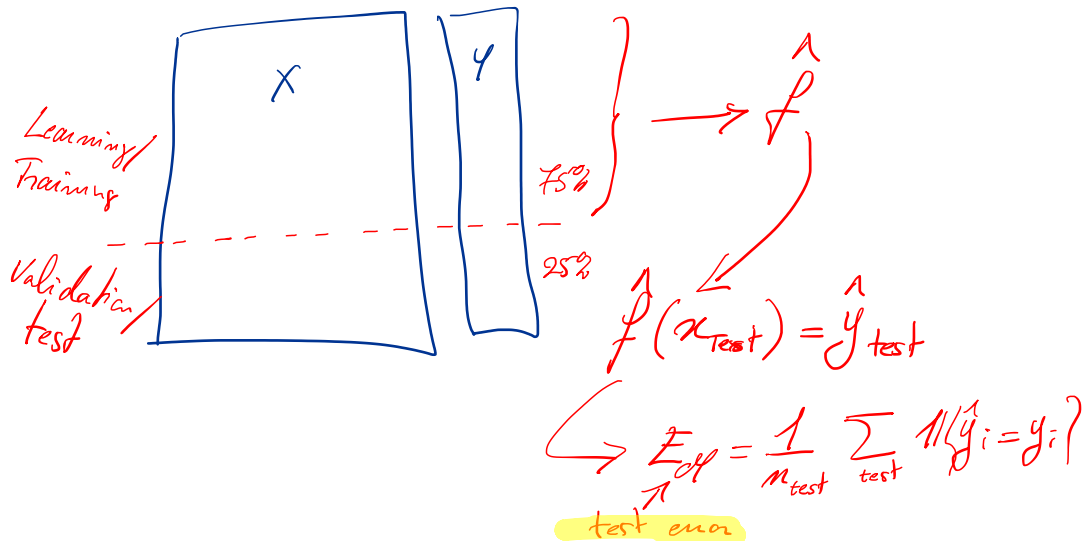
- compare several predictors and pick the most efficient one.

$$\mathcal{H}_1 \longrightarrow E_{\mathcal{H}_1} = 0.08 \pm 0.01$$

$$\mathcal{H}_2 \longrightarrow E_{\mathcal{H}_2} = 0.07 \pm 0.005$$

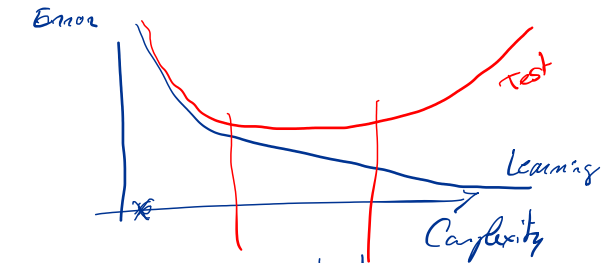
# A minimal setup for supervised learning

The minimal setup for building a supervised predictor  $f()$  from data is as follows:

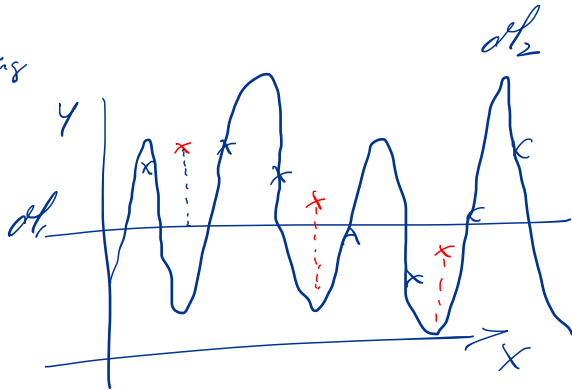


# Why such a minimal setup?

The goal is to avoid **over-fitting** when choosing the model or the model parameters:



	Learning	Test
$E_{d_1}$	10	$\approx 10$
$E_{d_2}$	0	$+\infty$

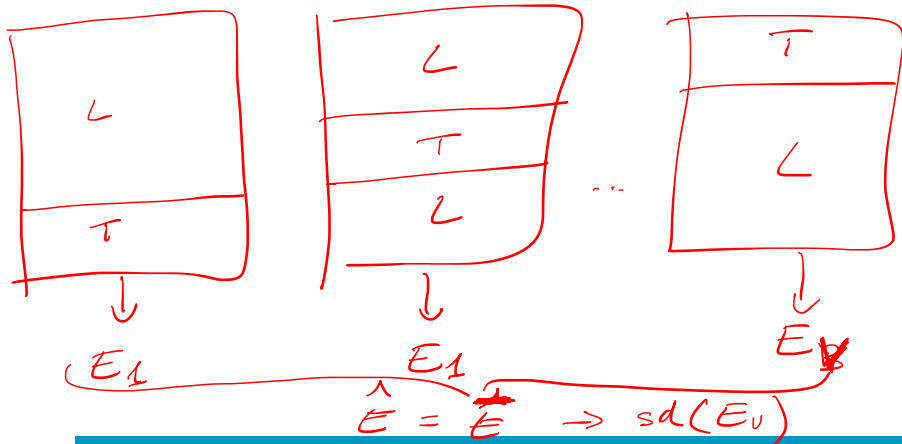




# An advanced setup for supervised learning

## Resampling techniques:

- there are several methods (leave-one-out, V-fold cross-validation, bootstrap) depending on the context (sample size, computing time, ...),
- V-fold cross-validation:



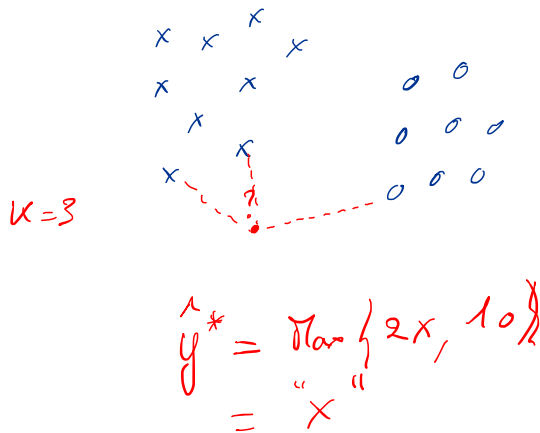
# Outline

---

1. Introduction
2. The supervised learning process
3. A selection of supervised learning methods
4. Deep learning
5. Tutorial: Classification of road patterns

## K-Nearest Neighbors (KNN)

K-nearest neighbors (KNN) is probably the **most simple classification method** (not really a learning method in fact):



simply a vote  
to find the class  
of  $x^*$

## K-Nearest Neighbors (KNN)

---

Pros / cons:

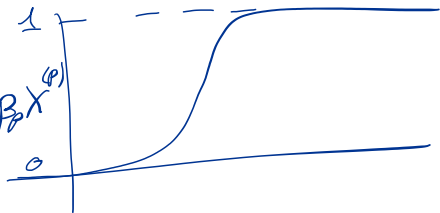
- ⊕ simple / cost very low
- ⊖ you need to keep all learning data for classifying.
- ⊖ sensitive to the choice of  $k$

Within R: function `knn()` in the `class` package or `knn3()` in the `caret` package.

## Logistic regression

The logistic regression turns the classification problem into a regression one thanks to the [logistic function](#):

$$\log \left( \frac{P(Y=1|X,\theta)}{P(Y=0|X,\theta)} \right) = \beta_0 + \beta_1 X^{(1)} + \dots + \beta_p X^{(p)}$$



$$\theta = \{ \beta_0, \beta_1, \dots, \beta_p \}$$

⇒ Learning = estimating  $\theta$  by Max. Likelihood

## Logistic regression

---

Pros / cons:

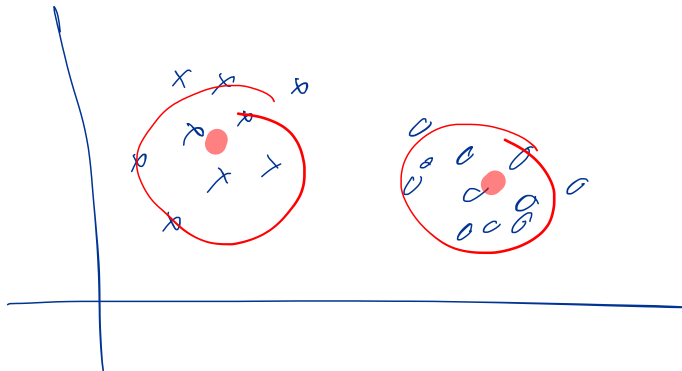
- ⊕ a statistical foundation  $\rightarrow P(Y=k|X, \theta)$
- ⊕ very stable, no assumptions on the class dist.
- ⊖ pb in high-dim
- ⊖ binary pb
- ⊖ linear classification boundaries.

Within R: function `glm()` in the base package.

# Linear Discriminant Analysis (LDA)

LDA (Fisher, 1936) is a **generative classification method** (as most of the "xxDA" methods):

$$p(X|Y=k) = \mathcal{N}(x_i; \mu_k, \Sigma_k)$$



# Linear Discriminant Analysis (LDA)

Classification (MAP) rule for a new observation  $x$ :

$$y = \operatorname{argmin}_k \{ \underbrace{\mu_k^t \Sigma^{-1} \mu_k - 2 \mu_k^t \Sigma^{-1} x}_{\text{red bracket}} - 2 \log(\pi_k) + C^{st} \}.$$

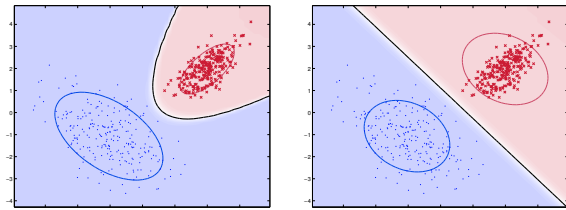


Fig. Decision boundaries for QDA (left) and LDA (right).

Pros / cons:

- ⊕ robust to non-normal data
- ⊕ easy to understand
- ⊕ probe as output
- ⊖ works for  $p < 50$

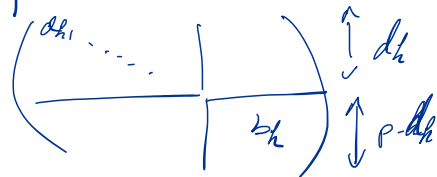
Within R: function `lda()` in the MASS package.



## High-Dimensional Discriminant Analysis (HDDA)

HDDA (Bouveyron et al., 2007) is a generative method designed for high-dimensional data:

$$p(x|y=h, \theta) = \mathcal{N}(\mu_h, \underbrace{\Sigma_h}_{p \times p})$$

$$\Sigma_h = \underbrace{Q_h^{\epsilon}}_{p \times d_h} \Delta_h Q_h$$


# High-Dimensional Discriminant Analysis (HDDA)

Classification (MAP) rule for a new observation  $x$ :

$$H_k(x) = \frac{1}{a_k} \|\mu_k - P_k(x)\|^2 + \frac{1}{b_k} \|x - P_k(x)\|^2 + d_k \log(a_k) + (p - d_k) \log(b_k) - 2 \log(\pi_k).$$

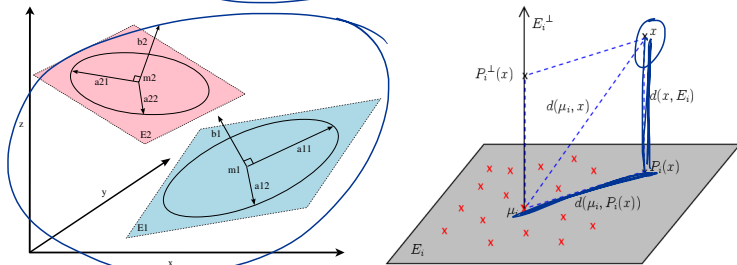


Fig. Modeling of the classes into low-dimensional subspaces.

Pros / cons:

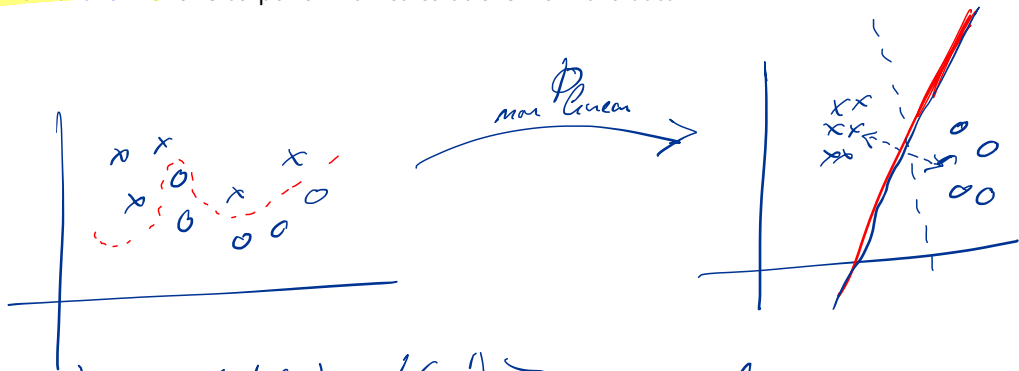
- ⊕ partitioning → works very well in HD space
- ⊖ normal assumptions
- ⊕ the interpretation of  $d_k$

Within R: function `hdda()` in the `HDclassif` package or the `caret` package.

# Support Vector Machines (SVM)

The idea of SVM is:

- to project the data into a high-dimensional space in order to ease the classification task,
- and to use a linear classifier in the projection space (feature space),
- the "kernel trick" allows to perform all calculations from the data.



$$K(x, x') = \langle \phi(x), \phi(x') \rangle$$

$$\|\phi(x) - \phi(x')\|^2 = K(x, x) + K(x', x') - 2K(x, x')$$

Kernel

## Support Vector Machines (SVM)

---

The kernel trick: how to optimize into the feature space directly from the observed data points.

- ⊕ very efficient ---- if you find the right kernel!
- ⊖ high complexity for learning
- ⊖ lack of understanding

Within R: function `svm()` in the `e1071` package or function `svmradial()` in `caret`.

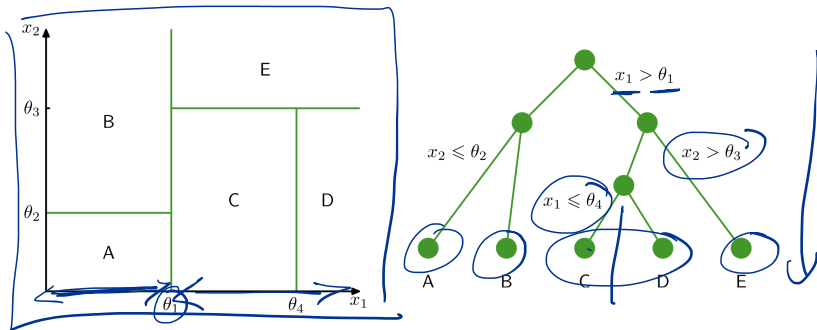
# Classification trees and random forest

The idea of classification (and regression) trees (CART) is to:

- choose a variable at each step that best splits the set of data in term of classification,
- according to some metric, usually **the Gini impurity index**:

$$I_G(\tau) = \sum_{k=1}^K p_{\tau k}(1 - p_{\tau k}),$$

where  $p_{\tau k} = \frac{1}{n} \sum_{i=1}^n \mathbf{1}\{x_i \in \tau\}$ .



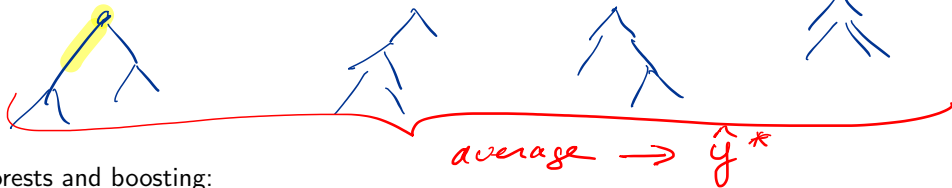
# Decision trees and random forest

Random forest:

- the aim is to robustify CART by a better exploration of the solution space,
- by sampling both on observations and variables to create  $B$  solutions,
- on which the solution is averaged.

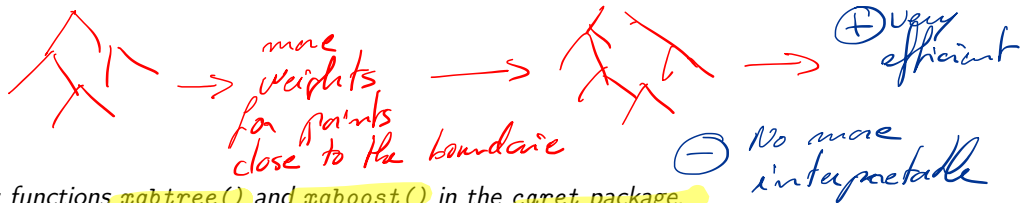
⊕ variable importance ⊕ more robust

⊖ less interpretable



Trees, forests and boosting:

- it is again possible to robustify CART and RF with boosting,
- the idea is to more importance to the observations which are difficult to classify.



Within R: functions `xgbtree()` and `xgboost()` in the `caret` package.

# Outline

---

1. Introduction
2. The supervised learning process
3. A selection of supervised learning methods
4. Deep learning
5. Tutorial: Classification of road patterns

# Outline

---

1. Introduction
2. The supervised learning process
3. A selection of supervised learning methods
4. Deep learning
5. Tutorial: Classification of road patterns