INSTITUT NATIONAL DES SCIENCES APPLIQUÉES TOULOUSE

# Machine Learning 2 – 5A
# Handling time series

Team: E. Alata, D. Dragomirescu, P. Leleux, V. Migliore

Université Fédérale Toulouse Midi-Pyrénées

# From factorial methods to inferential methods
## Previously...

Traditional machine learning workflow

Data description/preprocessing methods :

- Qualitative data,
- Quantitative data,
- Time series/Sequential data.

# From factorial methods to inferential methods
## Motivation

Traditional machine learning workflow

These geometric representations of data allow the use of the **vectorial space** notions :

- define distances between individuals/variables,
- weight the influence of individuals/variables,
- identify groups (aggregation/clusters),
- identify dependency relations/links between individuals/variables.

# From factorial methods to inferential methods
## Motivation

Traditional machine learning workflow

These geometric representations of data allow the use of the **vectorial space** notions :

- define distances between individuals/variables,
- weight the influence of individuals/variables,
- identify groups (aggregation/clusters),
- identify dependency relations/links between individuals/variables.

Inferential methods

# Classification
## Supervised learning

There exist 2 types of classification :

- If we have a **training basis**, with labelled data

| X | $X^1 \dots X^i \dots X^m$ | Classe |
|---|---|---|
| 1<br>⋮<br>$i$<br>⋮<br>$n$ | Caractéristiques variables | S variables nominales |

**Supervised classification :**

With the training basis, we can choose or learn a decision model which explains the relations between input features and the output class.

# Classification
## Unupervised learning

- If we do not have a training basis, the **unsupervised classification non supervisée** should be preferred.

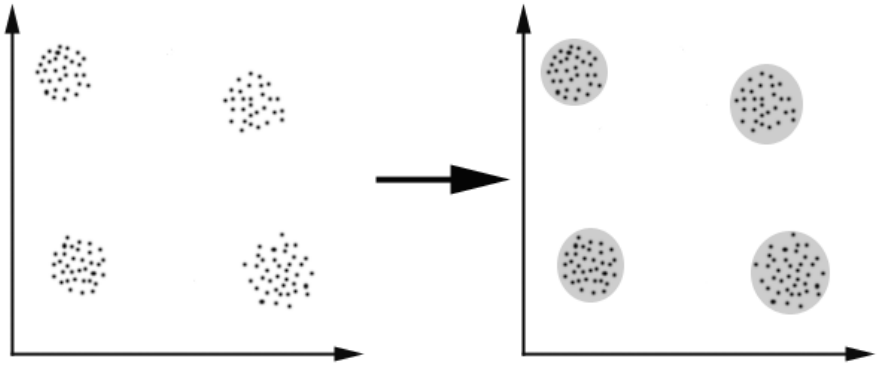


Figure – Example of unsupervised classification : divide the dataset in 4 classes using the distance between points.

### Unsupervised classification

Goal : create a partitioning (set of classes) of a dataset using similarity measures between the data points such that the data points belonging to a specific class are as similar as possible, and the data from di□erent classes are as dissimilar as possible.

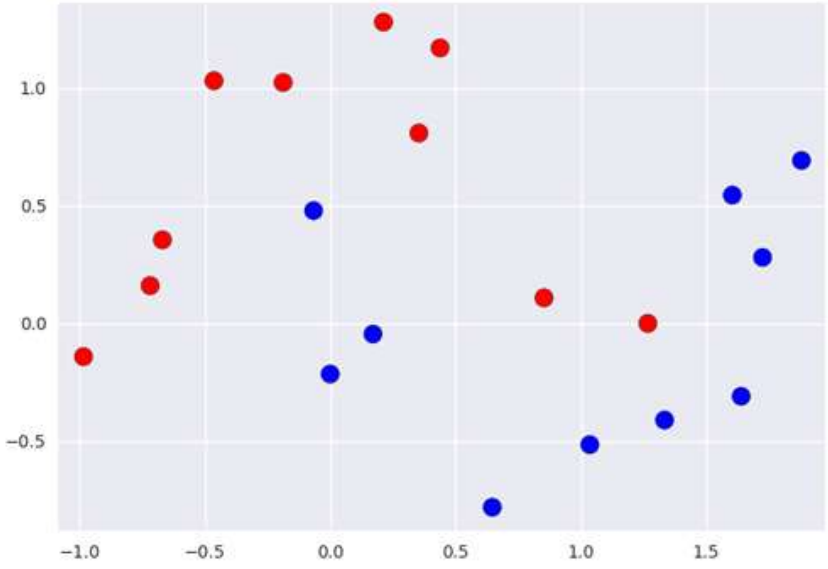
# Section 1

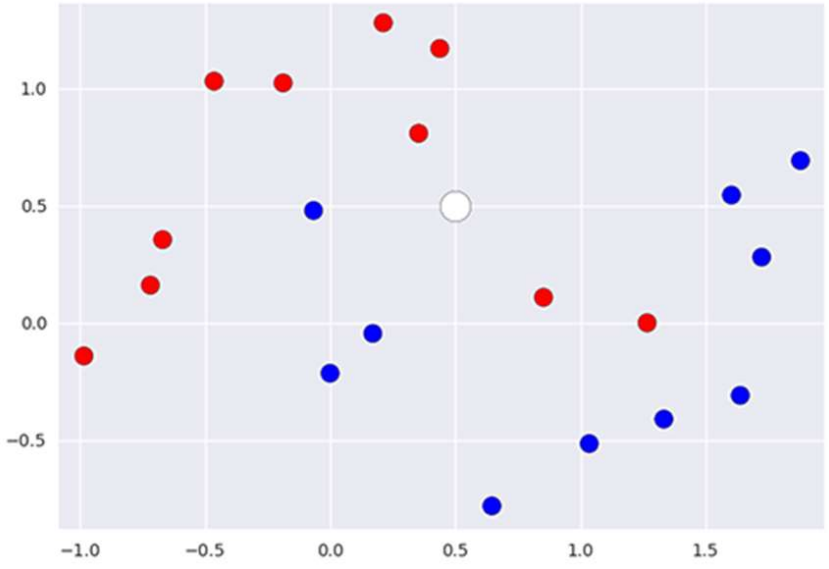
## k-Nearest Neighbors

# k-Nearest Neighbors (k-NN)
## The example problem

Let's consider the dataset $D = \left\{ \left( x^{(1)}, y_1 \right), \ldots, \left( x^{(m)}, y_m \right) \right\}$ with

- $x^{(i)} \in \mathbb{R}^n$ data points (represented in 2D),
- $y_i \in \{1, \ldots, K\}$ the output labeled in $K$ classes.



# k-Nearest Neighbors (k-NN)
## Principle

Let's consider the dataset $D = (X, y) = \left( x^{(1)}, y_1 \right), \ldots, \left( x^{(m)}, y_m \right)$ with

- $x^{(i)}$ $\mathbb{R}^2$ data points,
- $y_i$ $\{1, \ldots, K\}$ the output labeled in $K$ classes.

**Algorithm** k-NN algorithm

Input : new data point $x$
Compute distances to points in dataset
Find 5 nearest points
Determine the most represented class $\hat{k}$
Assign class $\hat{k}$ to new point $x$

# k-Nearest Neighbors (k-NN)
## The example problem : application

Results depend on the choice of $k$...

- k=1 : class *red*
- k=2 : class undetermined
- k=3 : class *red*
- k=4 : class undetermined
- **k=5 : class** *red*

# k-Nearest Neighbors (k-NN)
## The example problem : application

Results depend on the choice of $k$... but also on the measure of similarity !

- k=1 : class *red*
- k=2 : class undetermined
- k=3 : class *red*
- k=4 : class undetermined
- **k=5 : class** ***red***

# k-Nearest Neighbors (k-NN)
## Pros and cons

- Advantages :
  - Simplicity
  - No training phase
  - Adaptability to local patterns : k-NN can capture complex patterns in the data, even with non-linear boundaries (e.g. identification of written numbers, satellite images)
  - No sensitivity to outliers during Training
- Disadvantages :
  - Can be expensive : need to compute $(n^2)$ distances and store all data points,
  - Sensitive to non-pertinent and correlated features,
  - Sensitive to outliers during Testing,
  - Parameter : **distance**.

# k-Nearest Neighbors (k-NN)
## Mesure of similarity ?

Let's use a distance :

- Euclidian distance : $d(u,v) = \sum_{i=1}^{n} (u_i - v_i)^2$
- Manhattan distance : $d(u,v) = \sum_{i=1}^{n} |u_i - v_i|$
- Minkowski distance : $d_q(u,v) = \sqrt[q]{\sum_{i=1}^{n} |u_i - v_i|^q}$