# Journée SFR FED 4226
# Neuro-Imagerie Fonctionnelle

## Du Machine Learning au Deep Learning, concepts de base et illustrations

Au programme :

- 9h-10h30 : Données, représentations, décisions - JY Ramel (LIFAT Tours)
- 11h-12h30 : Apprentissage automatique, modèles et évaluation - T. Brouard (LIFAT Tours)
- 12h30-13h30 : Pause Repas
- 13h30-15h : Le Deep Learning - R. Raveaux (LIFAT)
- 15h30-16h00 : Illustration 1 : Deep Learning en tractographie - Laurent Petit (IMN, CNRS, CEA, Université de Bordeaux)
- 16h00-16h30 : Illustration 2 : Exemples d'exploitation de DeepLabCut - Pierre-Olivier Fernagut (Laboratoire de Neurosciences Expérimentales et Cliniques - INSERM U-1084, Université de Poitiers)
- 16h30-17h : Echanges, Débriefing, ...

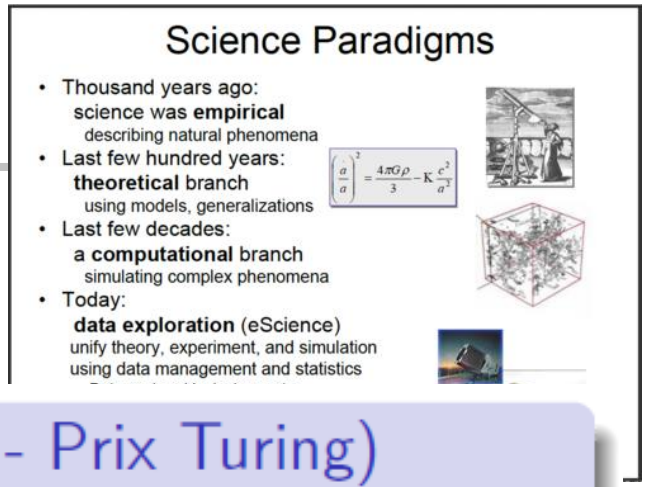
# Très brève introduction

Objectifs

- Faire découvrir les principaux aspects de l'IA, notamment le machine learning, et ses applications
- Donner les clés pour comprendre ces outils, et apprécier leur utilité, ou leur faiblesses, selon les usages
- Comprendre et illustrer les évolutions récentes des méthodes

- Peu de temps pour échanger sur les généralités…

# IA et Science des données




- *Jim Gray, Microsoft, 2005*

## Data driven science : le 4e paradigme (Jim Gray - Prix Turing)

Extrait : "A l'heure actuelle, la science vit une révolution qui conduit à nouveau paradigme selon lequel 'la science est dans les données', autrement dit la connaissance émerge du traitement des données [...] **Le traitement de données et la gestion de connaissances représentent ainsi le quatrième pilier de la science après la théorie, l'expérimentation et la simulation.** L'extraction de connaissances à partir de grands volumes de données (en particulier quand le nombre de données est bien plus grand que la taille de l'échantillon) , l'apprentissage statistique, l'agrégation de données hétérogènes, la visualisation et la navigation dans de grands espaces de données et de connaissances sont autant d'instruments qui permettent d'observer des phénomènes, de valider des hypothèses, d'élaborer de nouveaux modèles ou de prendre des décisions en situation critique"

# Très brève introduction

## L'I.A. ne sait pas (encore ?) tout faire. . .

- Une seule finalité : les algorithmes sont dédiés à un objectif "simple"
- Interprétabilité faible, car déterminer comment la machine est arrivée à telle décision est parfois très difficile, le comprendre également
- Biais dans les données d'apprentissage, dû au coût de récolte, de préparation et d'apprentissage des exemples

## L'I.A. commence à inquiéter. . .

- 2007 Charte sur l'éthique des robots
- 2009 Conférence organisée par l'AAAI 7 centrée sur les questions d'éthique, et la potentielle limitation de la recherche qui pourrait conduire à la perte de l'emprise humaine sur les machines
- 2015 : Prise de conscience du potentiel danger des avancées en deep learning : et si la machine dépassait l'homme ? (S. Hawking / E. Musk / B. Gates) puis tenue de la conférence d'Asilomar (2017)
- 2017 : En France, la C.N.I.L. organise un débat public et publie des recommandations pour la construction d'un modèle éthique d'I.A.

# Partie 1 : Données, représentations, décisions

Jean-Yves Ramel

ramel@univ-tours.fr

Laboratoire d'Informatique Fondamentale
et Appliquée de Tours

# IA ➔ Apprentissage automatique

## Définition

- Branche de l'IA qui concerne le développement d'algorithmes permettant de rendre une machine (un agent) capable d'accomplir des tâches complexes sans avoir été explicitement programmée dans ce but.

## Exemple

- Comment écrire un programme qui reconnaisse les caractères manuscrits ?

# IA ➔ Apprentissage automatique

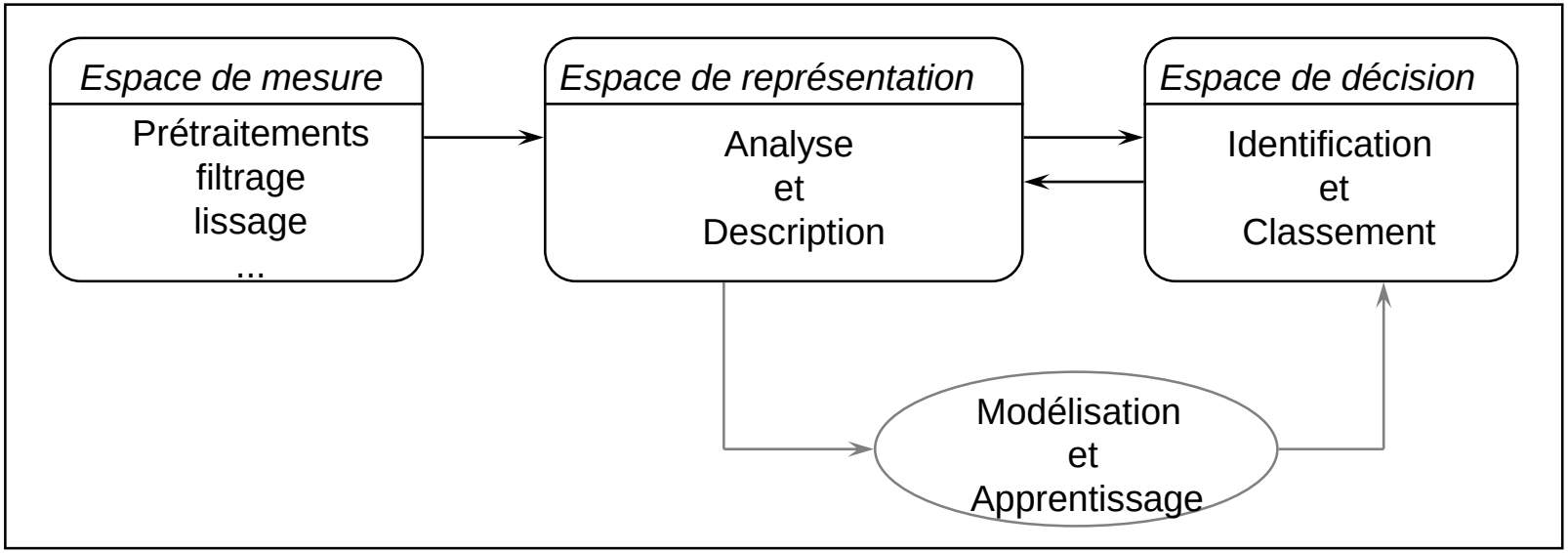
## Définition

- Branche de l'IA qui concerne le développement d'algorithmes permettant de rendre une machine (un agent) capable d'accomplir des tâches complexes sans avoir été explicitement programmée dans ce but.

## Exemple

- Comment écrire un programme qui reconnaisse les caractères manuscrits ?
  - Entrer des règles manuellement (difficile et peu fiable)
  - Meilleur méthode : écrire un algorithme (générique) qui produit automatiquement un programme de reconnaissance de caractères à partir d'un grand nombre d'exemples.

# Modélisation d'un système d'IA orientée « données »




## Espace de mesure

- Représentation du monde réel
- Obtention des données à l'aide d'une méthode de perception : **le capteur**
- Un certain nombre de (pré)traitements peut être effectué dans l'espace de mesure (au plus proche du capteur – **Edge intelligence** [Plastiras2018])

## Espace de représentation

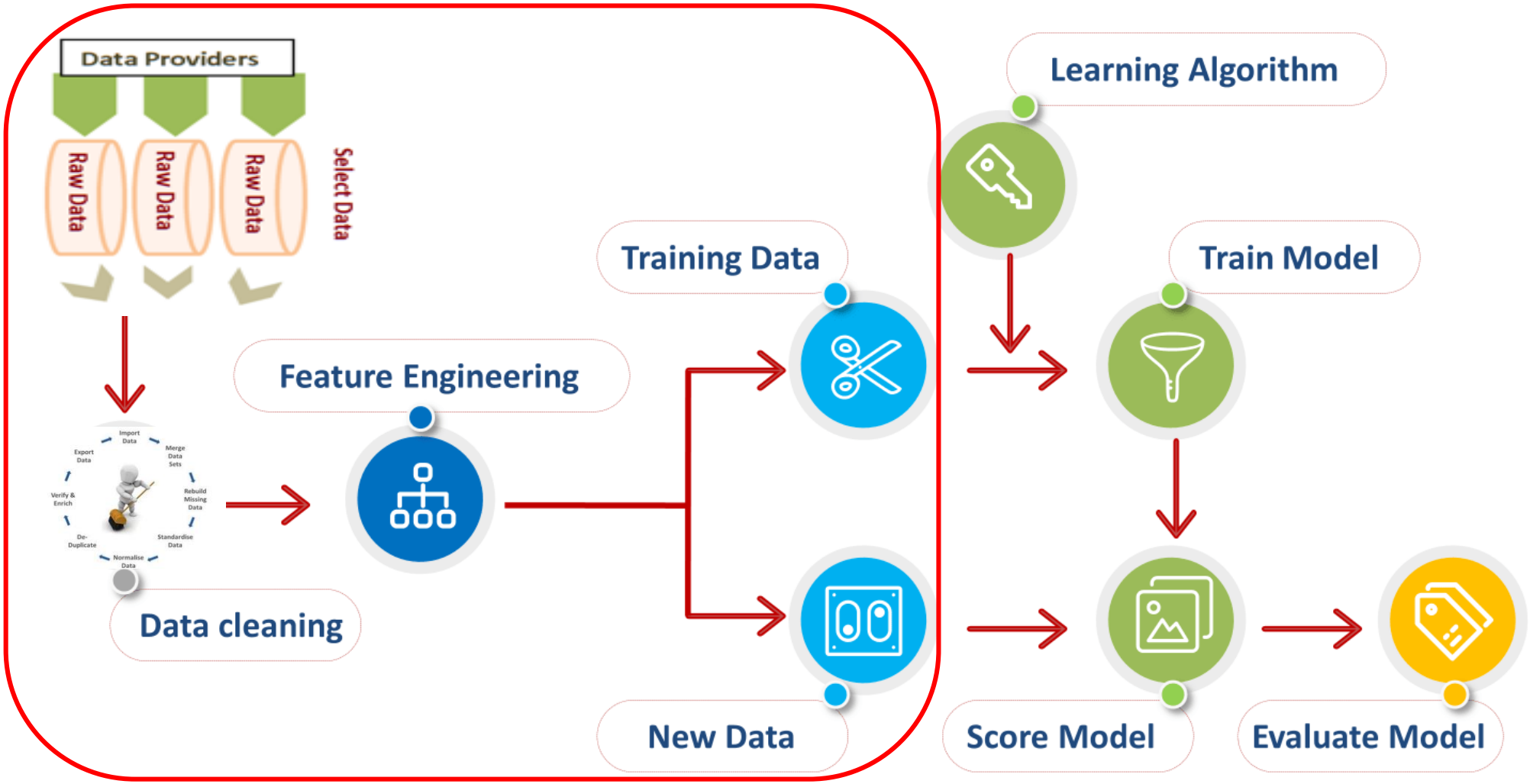
- Espace dans lequel seront fait les analyses pour produire les décisions
- Par transformation : espace de représentation → espace de décision

## Espace de décision

- doit aussi être réfléchi et **modéliser**

# Modélisation possible d'un système d'IA

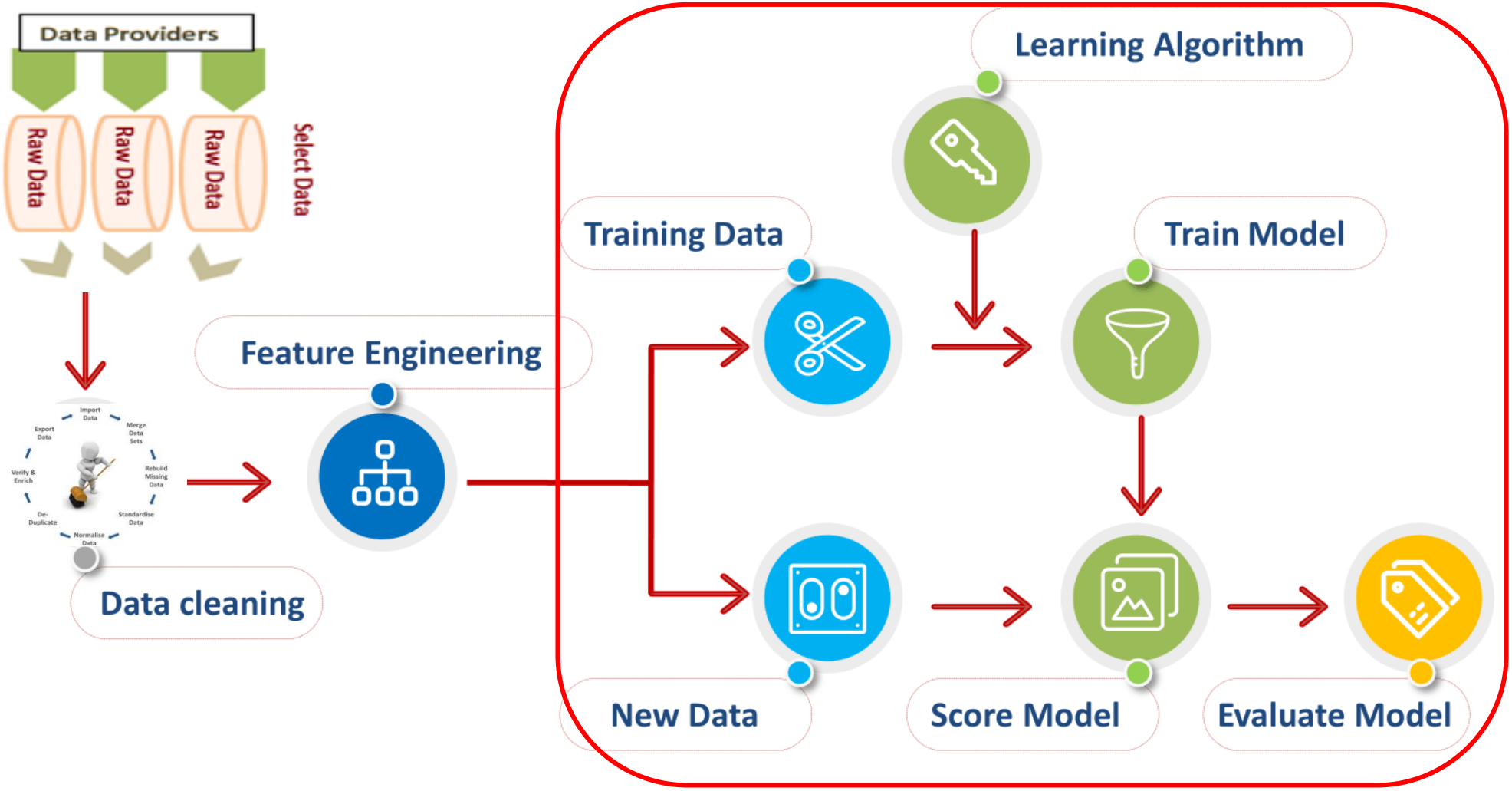
## Partie I : Les données




*Adapted from [Meduim2019]*

# Modélisation possible d'un système d'IA

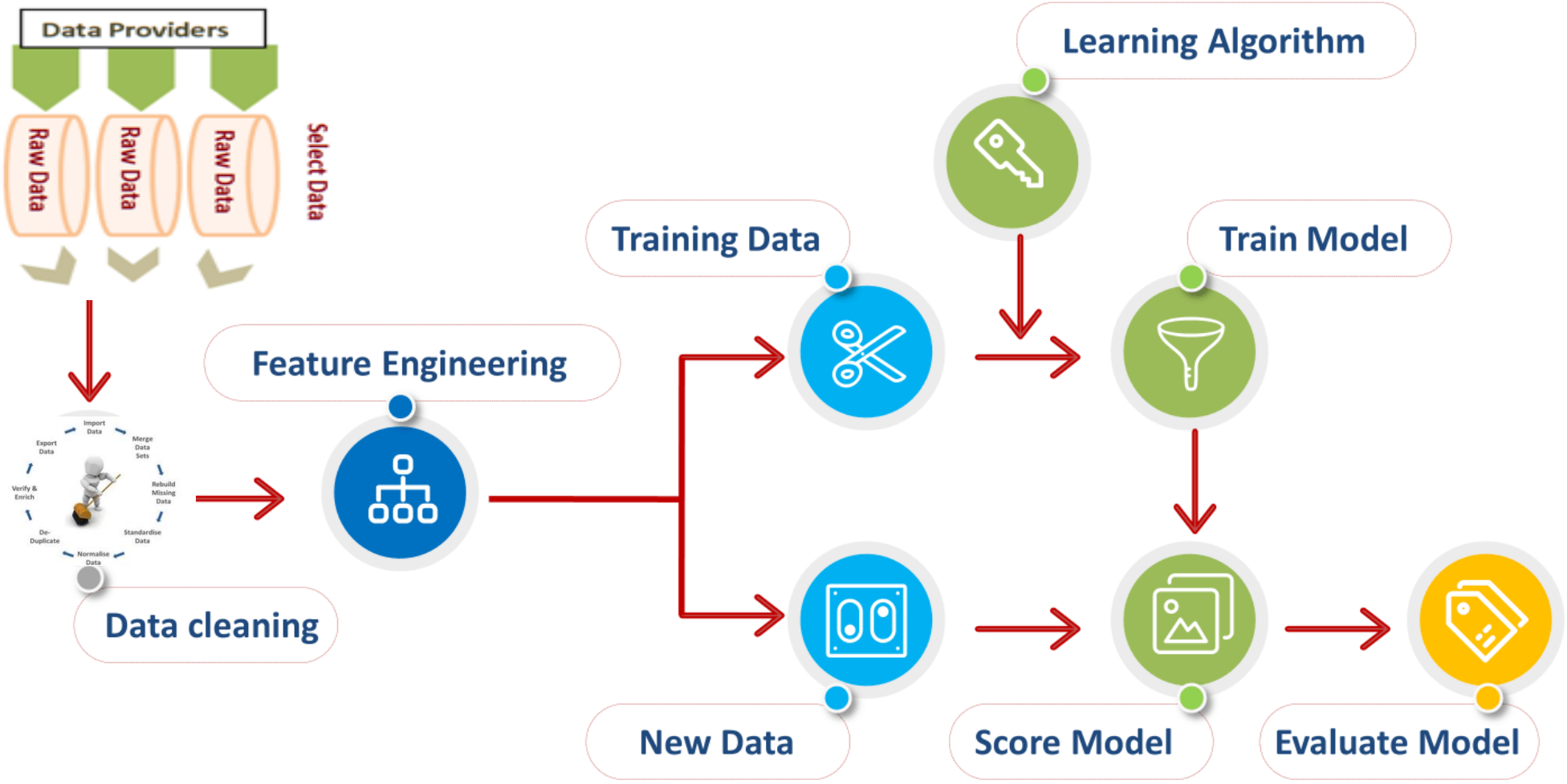
**Partie II : les modèles et leurs évaluations**




*Adapted from [Meduim2019]*

# Modélisation possible d'un système d'IA

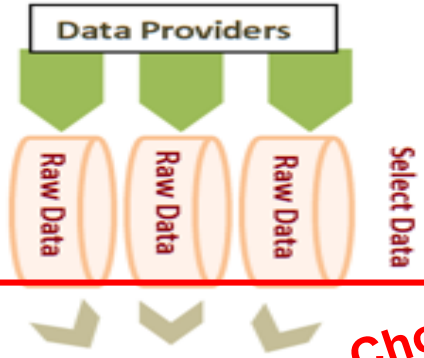
**Ou se trouve l'intelligence humaine ???**




*Adapted from [Meduim2019]*

# Zoom sur les données et leur préparation

Choix des données….




Choix des caractèristiques et modèles….

Annotations….


Data cleaning

Feature Engineering

Training Data

New Data

# Zoom sur les données…

**Les données** peuvent être vues comme une collection d'objets décrits par des attributs

- Un attribut est une propriété, une caractéristique de l'objet
- Un ensemble d'attributs décrit un objet

Attributes

Objects

| *Tid* | Refund | Marital Status | Taxable Income | Cheat |
|---|---|---|---|---|
| 1 | Yes | Single | 125K | No |
| 2 | No | Married | 100K | No |
| 3 | No | Single | 70K | No |
| 4 | Yes | Married | 120K | No |
| 5 | No | Divorced | 95K | Yes |
| 6 | No | Married | 60K | No |
| 7 | Yes | Divorced | 220K | No |
| 8 | No | Single | 85K | Yes |
| 9 | No | Married | 75K | No |
| 10 | No | Single | 90K | Yes |

Attribut - valeur

- La valeur d'un attribut est un nombre ou un symbole
- Ne pas confondre attribut et valeur

Types des valeurs

- Quantitative (numérique, exprime une quantité)
  - Discrète (ex : nb d'étudiants) ou continue (longueur, …)
  - Echelle proportionnelle (chiffre d'affaires, taille), ou échelle d'intervalle (QI)
- Qualititative (ou symbolique)
  - Variable ordinale (classement à un concours, échelle de satisfaction client, …)
  - Variable nominale (couleur de yeux, diplôme obtenu, sexe, …)

Les modalités d'une variable sont l'ensemble des valeurs qu'elle peut prendre
Ex : les modalité de notes sont 0*;* 1*;* 2*;* *;* 20 - les modalités de couleur sont bleu, vert, noir,...

# Des données aux bases d'apprentissage

## Individus, objets VS variables, descripteurs

**Acquisition des données ➔ Préparation ➔ Espace de représentation**

**Dans l'espace de représentation (vectoriel), on parle de :**

- **Population** : groupe ou ensemble d'individus que l'on analyse
- **Recensement** : étude de tous les individus d'une population donnée
- **Sondage** : étude d'une partie seulement d'une population appelée échantillon
- **Individus :** les enregistrements (ensemble d'attributs) deviennent des individus ou objets

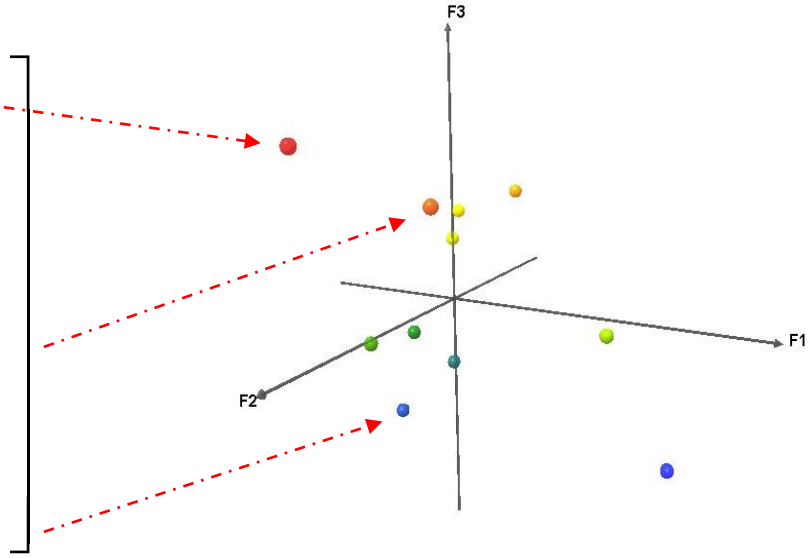
- **Variables, descripteurs, caractéristiques :** les attributs, comme déjà indiqué, possiblement de différents types :
  - Quantitatives : nombres sur lesquels les opérations usuelles (somme, moyenne,...) ont un sens, peuvent être discrètes
  - Qualitatives : appartenance a une catégorie donnée, peuvent être nominales ou ordinales

# Tableaux, vecteurs, nuages de points

## Tableau de données


- Pour n individus et p variables, on a le tableau X
- X est une matrice rectangulaire a n lignes et p colonnes
- X est aussi un nuage de n points dans $R^p$

$$X = (x^1,\ldots,x^p) = \begin{bmatrix} x_1^1 & x_1^2 & \cdots & x_1^p \\ x_2^1 & x_2^2 & & \\ & & \ddots & \\ \vdots & & x_i^j & \vdots \\ & & & \ddots \\ x_n^1 & & \cdots & x_n^p \end{bmatrix}$$

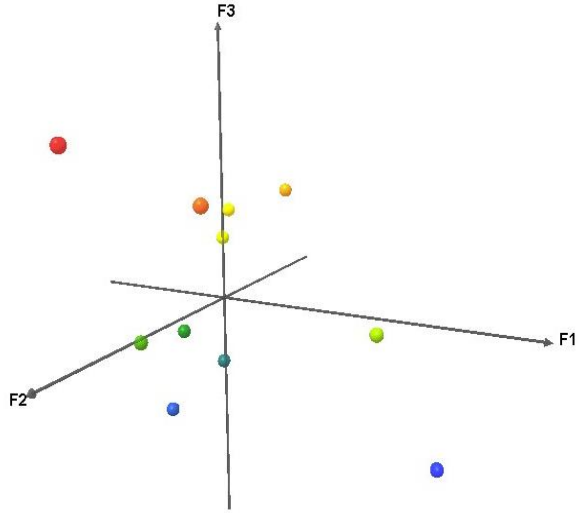


# Tableaux, vecteurs, nuages de points

## Vecteurs, individus et variables


- **Variable**
  Une colonne du tableau

$$x^j = \begin{bmatrix} x_1^j \\ x_2^j \\ \\ x_n^j \end{bmatrix}$$




- **Individu**
  Une ligne du tableau

$$e_i' = (x_i^1 \quad x_i^2 \quad \quad x_i^p)$$

- Les individus sont décrits par des variables (caractéristiques, descripteurs, …)

# Pondération, corrélation et Inertie

## La matrice des poids

- **Pourquoi**
  utile quand les individus n'ont pas la même importance

- **Comment**
  on associe aux individus un poids pi tel que

$$p_1 + p_2 + \ldots + p_n = 1$$

  et on représente ces poids dans la matrice diagonale de taille n

$$D = \begin{bmatrix} p_1 & & \ldots & 0 \\ & p_2 & & \\ \vdots & & \ddots & \vdots \\ 0 & & \ldots & p_n \end{bmatrix}$$

- **Cas uniforme**
  tous les individus ont le même poids $p_i = 1 / n$ et $D = I / n$

# Pondération, corrélation et Inertie

## Rappel Moyenne arithmétique

- **Définition**
  On note

$$\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$$

  ou pour des données pondérées

$$\bar{x} = \sum_{i=1}^{n} p_i x_i$$

- **Propriétés**
  la moyenne arithmétique est une mesure de tendance centrale qui dépend de toutes les observations et est sensible aux valeurs extrêmes. Elle est très utilisée a cause de ses bonnes propriétés mathématiques.

# Pondération, corrélation et Inertie

## Rappel Variance et écart-type

- **Définition**
  la variance de x est définie par

$$s_x^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2 \qquad ou \qquad s_x^2 = \sum_{i=1}^{n} p_i (x_i - \bar{x})^2$$

  L'écart type $s_x$ est la racine carrée de la variance.

- **Propriétés**
  La variance satisfait la formule suivante

$$s_x^2 = \frac{1}{n}\sum_{i=1}^{n} p_i x_i^2 - (\bar{x})^2$$

  La variance est « la moyenne des carres moins le carre de la moyenne ». L'ecart-type, qui a la même unité que x, est une mesure de dispersion.

# Pondération, corrélation et Inertie

## Rappel Mesure de liaison entre deux variables

- Définitions la covariance observée entre deux variables x et y est

$$s_{xy} = \sum_{i=1}^{n} p_i (x_i - \bar{x})(y_i - \bar{y}) = \sum_{i=1}^{n} p_i x_i y_i - \overline{xy}$$

et le coefficient de corrélation est donnée par

$$r_{xy} = \frac{s_{xy}}{s_x s_y} = \frac{\sum_{i=1}^{n} p_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n} p_i (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^{n} p_i (y_i - \bar{y})^2}}$$

# Pondération, corrélation et Inertie

## Inertie d'un nuage de points

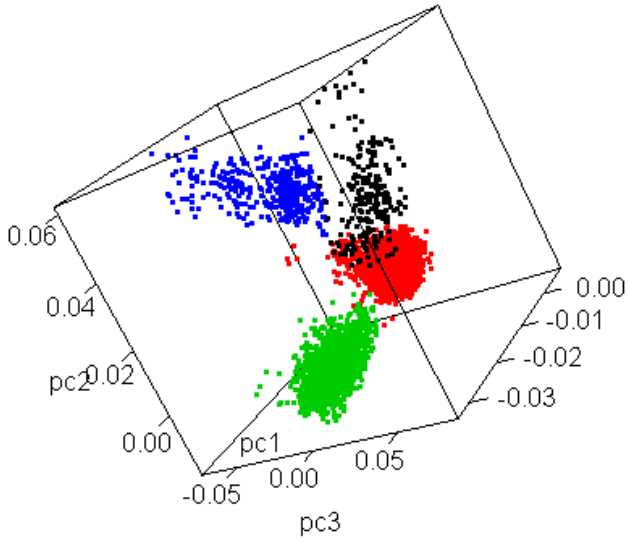
- **Définition**
  l'inertie en un point a du nuage de points est

$$I_a = \sum_{i=1}^{n} p_i \left\| e_i - a \right\|_M^2 = \sum_{i=1}^{n} p_i (e_i - a)' M (e_i - a)$$

- L'inertie totale est aussi donnée par la trace de la matrice V (somme de ses éléments diagonaux).

- **Autres relations**
  coefficient d agrégation est la moyenne des carres des distances entre les individus

$$D_k^2 = \frac{1}{n_k (n_k - 1)} \sum_{i=1}^{n_k} \sum_{j=1}^{n_k} d_k^2(x_i, x_j)$$



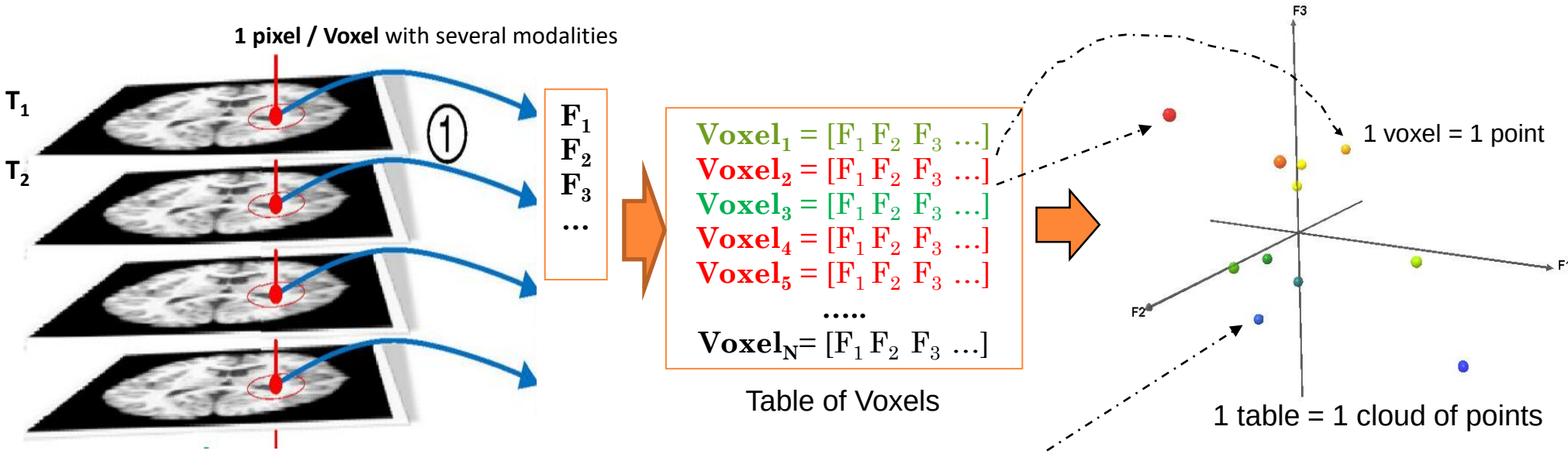

**L'inertie est une mesure fondamentale pour évaluer la qualité d'un espace de représentation (inertie totale, inter-classe, intra-classe)**

# Tableaux, vecteurs, nuages de points

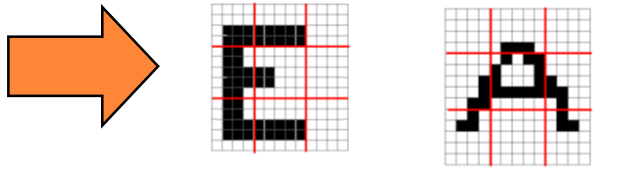
**Principe : 1 objet ⇨ p caractèristiques ⇨ 1 Vecteur = 1 Point ∈ $R^p$**


- 1 objet ⇨ p features ⇨ 1 Vector = 1 Point ∈ $R^p$
- 1 tableau = n objets (individus)
- Statistiques, analyse de données, Machine Learning deviennent opérationnels

## Adaptation à l'analyse d'images (segmentation)




## Classificatiion d'images (OCR)

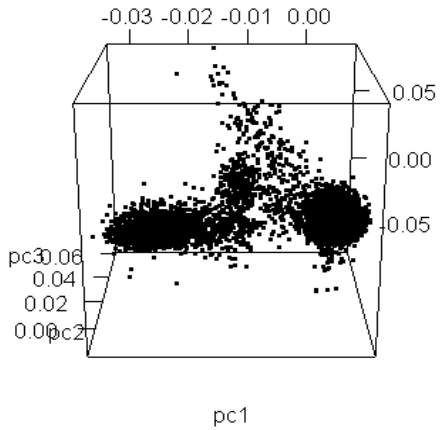


# Supervisé VS non supervisé ➔ Bases d'apprentissage

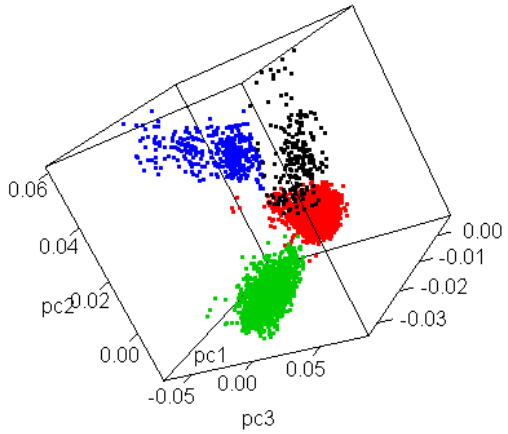
**Principe : 1 objet ⇨ p caractèristiques ⇨ 1 Vecteur = 1 Point ∈ $R^p$**

- Chaque individu/objet est décrit par des descripteurs numériques auxquels peuvent s'ajouter des **labels additionels ➔ base d'apprentissage**
- 1 tableau = n objets (individus)
- 1 object ⇨ p features ⇨ 1 Vector = 1 Point ∈ $R^p$ + additional labels
- **Apprentissage non-supervise ➔ apprentissage supervisé**

$\mathbf{Voxel_1} = [F_1\ F_2\ F_3\ \ldots]$
$\mathbf{Voxel_2} = [F_1\ F_2\ F_3\ \ldots]$
$\mathbf{Voxel_3} = [F_1\ F_2\ F_3\ \ldots]$
$\mathbf{Voxel_4} = [F_1\ F_2\ F_3\ \ldots]$
$\mathbf{Voxel_5} = [F_1\ F_2\ F_3\ \ldots]$
.....
$\mathbf{G} = [F_1\ F_2\ F_3\ \ldots]$




$\mathbf{Voxel_1} = [F_1\ F_2\ F_3\ \ldots]$ [Tumor]
$\mathbf{Voxel_2} = [F_1\ F_2\ F_3\ \ldots]$ [Normal]
$\mathbf{Voxel_3} = [F_1\ F_2\ F_3\ \ldots]$ [Tumor]
$\mathbf{Voxel_4} = [F_1\ F_2\ F_3\ \ldots]$ [Tumor]
$\mathbf{Voxel_5} = [F_1\ F_2\ F_3\ \ldots]$ [Normal]
$\mathbf{G_G} = [F_1\ F_2\ F_3\ \ldots]$ ↑
$\mathbf{G_R} = [F_1\ F_2\ F_3\ \ldots]$ **Label**

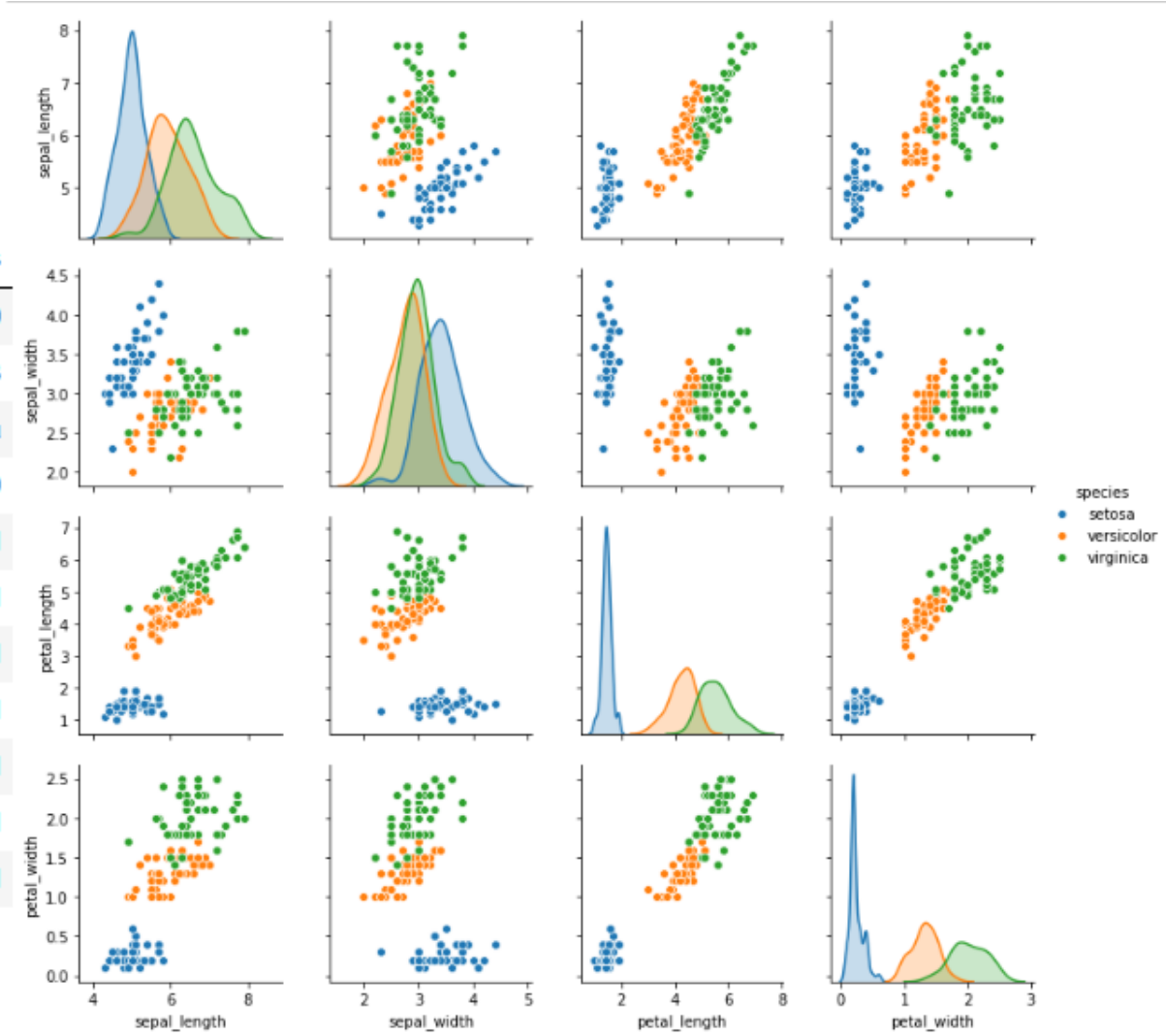


# 1er problème : Choix des descripteurs

**Bien choisir les descripteurs est crucial** (feature engineering)

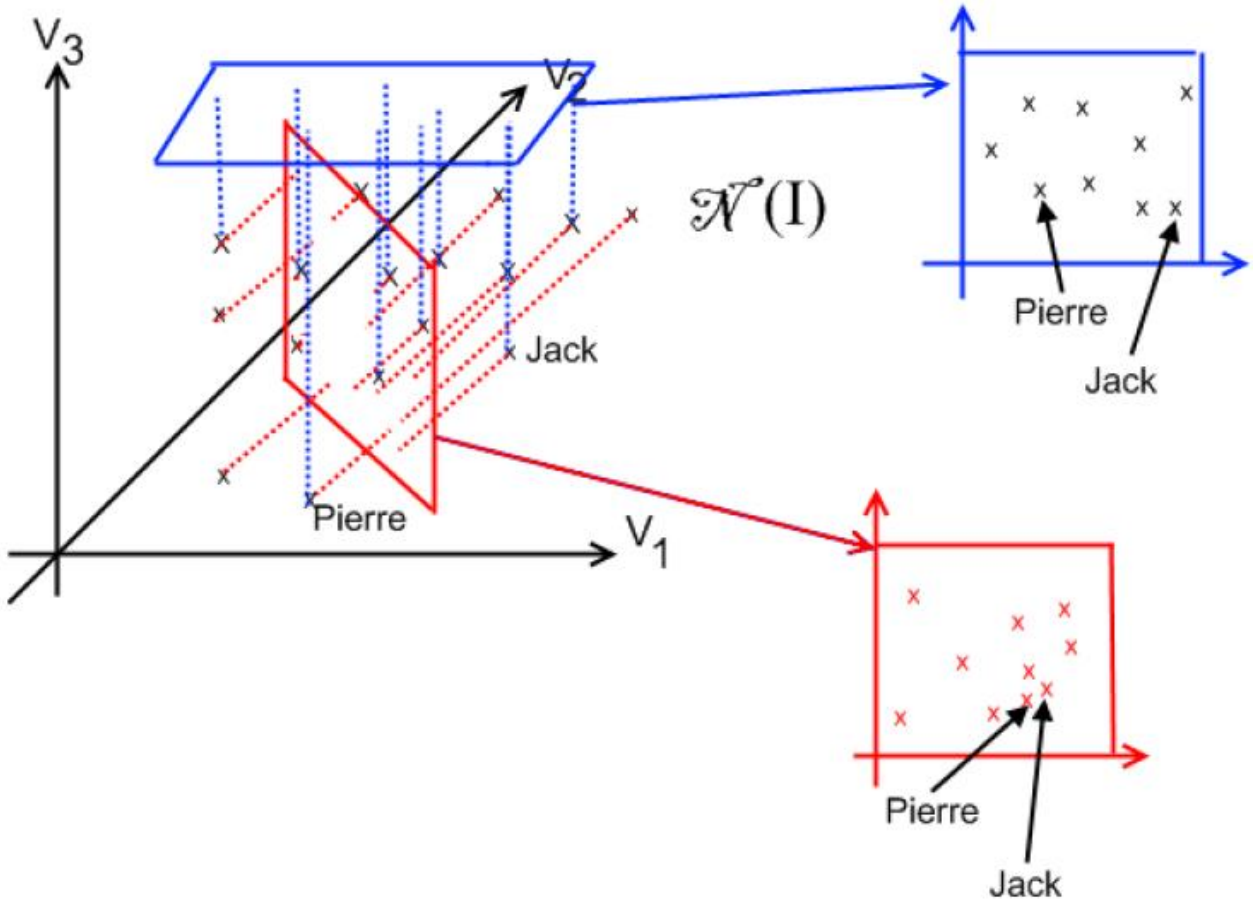
- Expertise, experimentation,…
- Stables
- Discriminantes

| | sepal_length | sepal_width | petal_length | petal_width | species |
|---|---|---|---|---|---|
| count | 150.000000 | 150.000000 | 150.000000 | 150.000000 | 150 |
| unique | NaN | NaN | NaN | NaN | 3 |
| top | NaN | NaN | NaN | NaN | setosa |
| freq | NaN | NaN | NaN | NaN | 50 |
| mean | 5.843333 | 3.057333 | 3.758000 | 1.199333 | NaN |
| std | 0.828066 | 0.435866 | 1.765298 | 0.762238 | NaN |
| min | 4.300000 | 2.000000 | 1.000000 | 0.100000 | NaN |
| 25% | 5.100000 | 2.800000 | 1.600000 | 0.300000 | NaN |
| 50% | 5.800000 | 3.000000 | 4.350000 | 1.300000 | NaN |
| 75% | 6.400000 | 3.300000 | 5.100000 | 1.800000 | NaN |
| max | 7.900000 | 4.400000 | 6.900000 | 2.500000 | NaN |



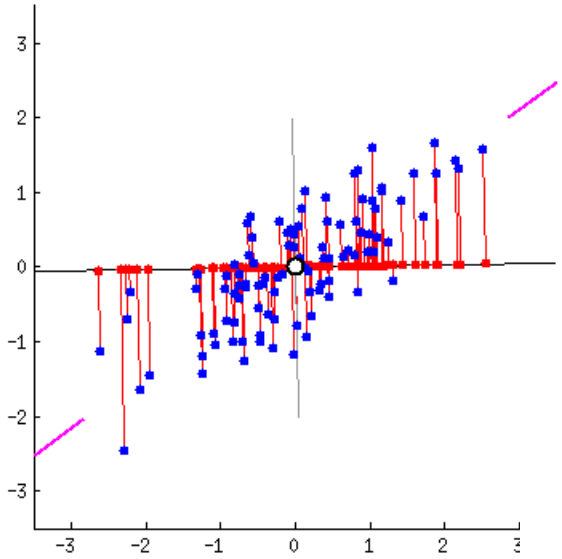
# Choix des descripteurs, une étape clé

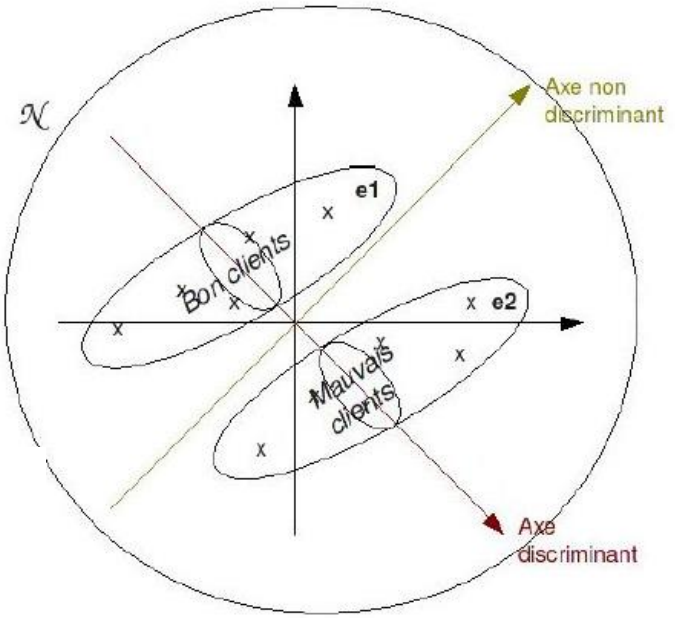
- Réduction de dimensionnalité → ACP [Hotelling1933]
- Espace de représentation et projections

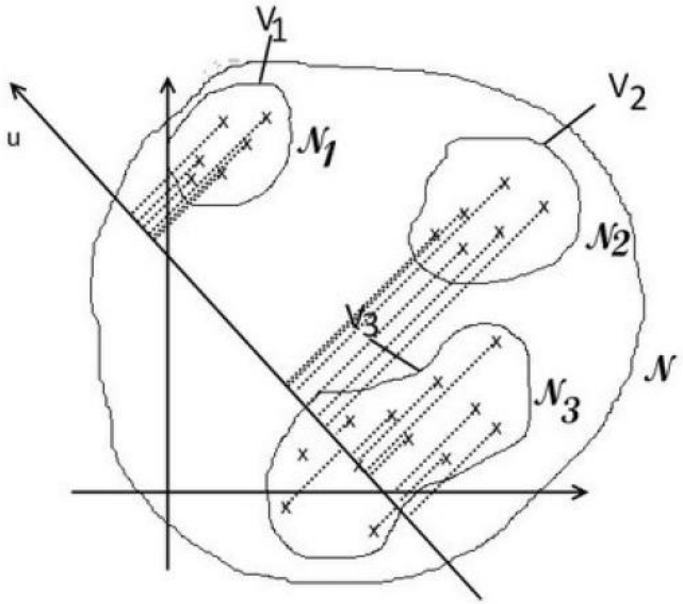


# Choix des descripteurs, une étape clé

- Génération de variables discriminantes (AFD [Saporta2006])



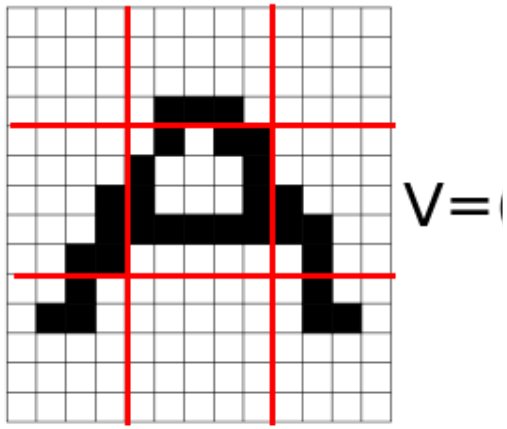



# Exercice – Choix de caractéristiques ?

**Decrire le contenu d'une image**

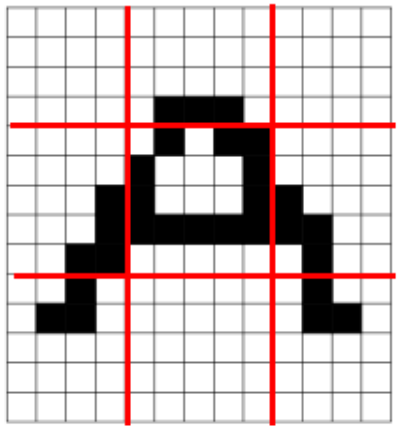
- Transformation en vecteur de descripteurs ?
- V = ($Nb_1$, $Nb_2$, … , $Nb_n$) ?



# Exercice – Choix de caractéristiques ?

Différentes possibilités

- Tous les pixels sont utilisés ➔ pb ?
- Découpage en n blocs ➔Descripteurs calculés pour chaque bloc
- Caractèristiques réflechies par l'humain (nb de trous, nb de trait horizontaux, verticaux, …)
- …

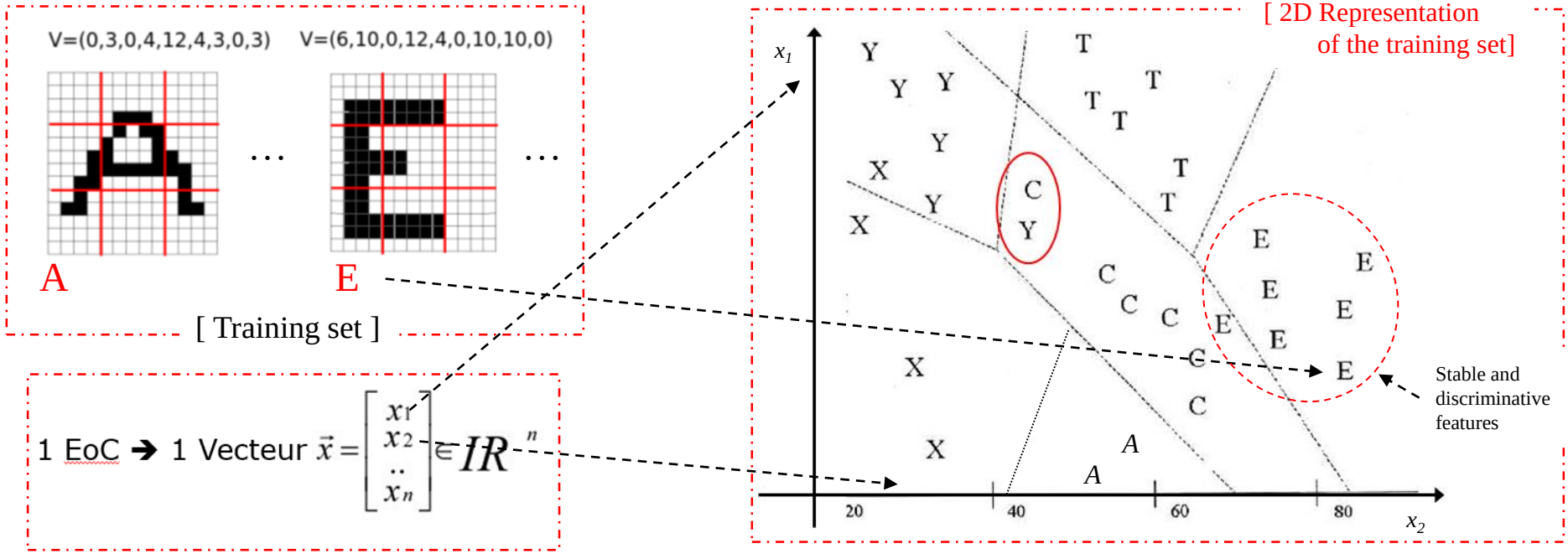


V=(0,3,0,4,12,4,3,0,3)

# Exercice – Choix de caracteristiques ?

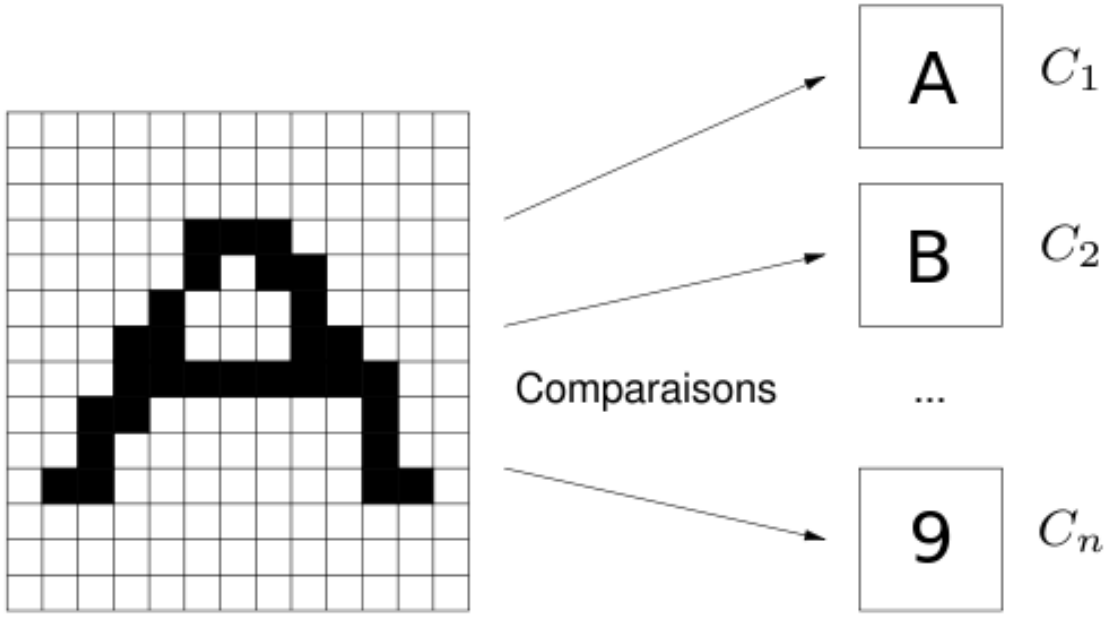
**Representation space vizualisation and analysis**

- We need a **large set of (labelled) examples** similar to the patterns to be recognized ➔ a training set
- We need a list of **stable and discriminative** features (shape, color, size,…) used to describe the patterns (labelled ones and unknown one)
- What is the feature space here? Is it a good one?



# 2e problème : la comparaison des individus ?

- Vos propositions ?



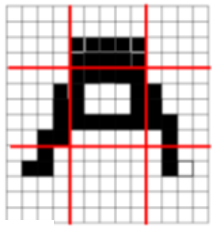

$C$ : Objet incunnu

B Base d'apprentissage (objets labelisés)

# 2e problème : la comparaison des individus ?

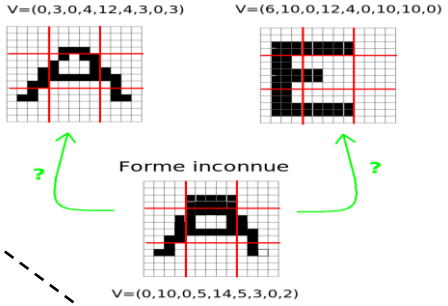
- Calcul de distances entre objets / vecteurs / points

Unknown object



V=(0,10,0,5,14,5,3,0,2)

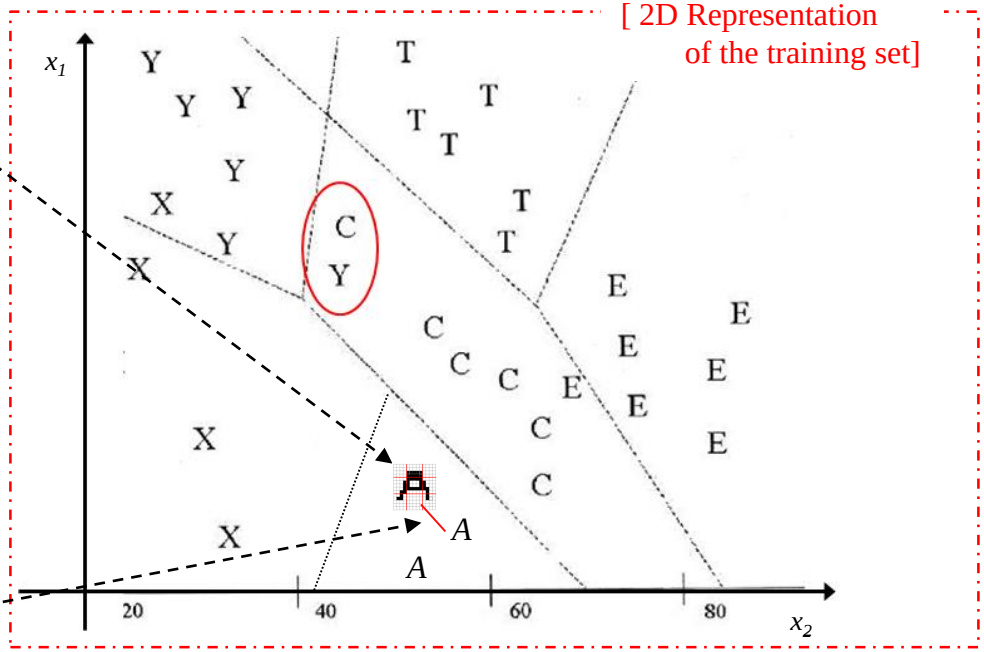
A




$$D(A,?) = \sqrt{(0-0)^2 + (3-10)^2 + (0-0)^2 + (4-5)^2 + \ldots + (3-2)^2}$$

$$D(A,?) = 7,48 \text{ et } D(B,?) = 19,05$$

[ 2D Representation of the training set]



# 2e problème : la comparaison des individus

## Distance, dissimilarité, métrique

- **Motivation**
  afin de pouvoir considérer la structure du nuage des individus, il faut définir une distance, qui induira une géométrie.

- **Distance euclidienne classique**
  la distance la plus simple entre deux points de Rp est définie par

$$d^2(u,v) = \sum_{j=0}^{p} (u_j - v_j)^2 = \|u - v\|$$

- **Généralisation simple ➔ métrique**
  on multiplie la variable j par $\sqrt{a_j}$

$$d^2(u,v) = \sum_{j=0}^{p} a_j (u_j - v_j)^2$$

$\alpha_j$ = *Poids associés aux variables*

# Similarités, dissimilarités, distances

## Métriques particulières

- **Métrique usuelle**
  - M = I correspond au produit scalaire usuel et à une distance classique

    Problèmes
  - la distance entre individus dépend de l'unité de mesure.
  - la distance privilégie les variables les plus dispersées.

- **Métrique réduite**
  - c'est la plus courante, on prend la matrice diagonale des inverses des variances (cf Mahalanobis)

$$M = D_{1/s^2} = \begin{bmatrix} \frac{1}{s_1^2} & & 0 \\ & \ddots & \\ 0 & & \frac{1}{s_p^2} \end{bmatrix}$$

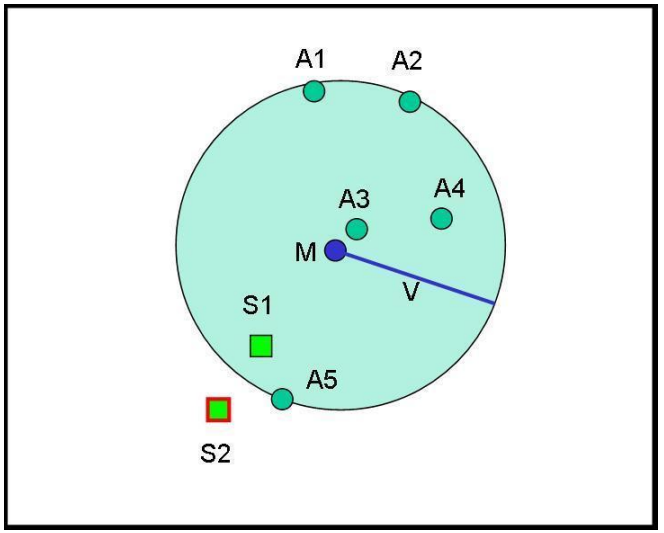
- **Métriques par classe**

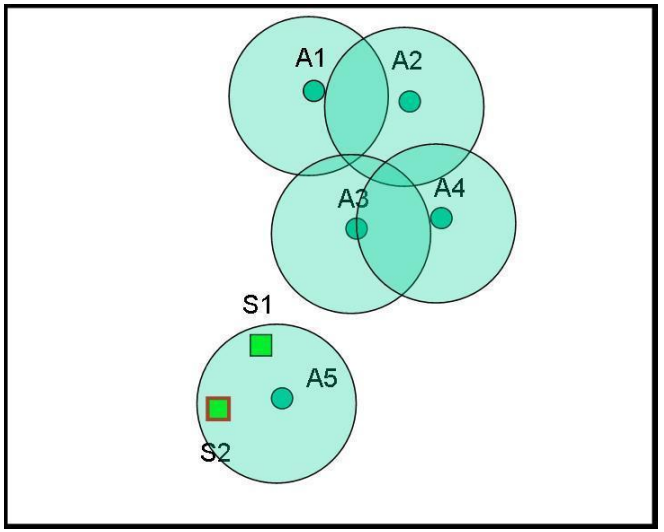
- **Metric Learning** [Bellet2015]
  - Il est aussi possible d'essayer d'adapter la métrique aux données et objectifs (de classification) d'un problème particulier

# 3e problème : Sélection des exemples d'apprentissage et choix (ou génération) des modèles

- Un modèle par classe
  - Un seul modèle par classe fonction de la moyenne et de l'écart-type de chacune des caractéristiques des exemples de l'ensemble d'apprentissage
    → Point moyen : *G*

- Plusieurs modèles par classe
  - Conservation de toute l'information, c'est-à-dire le vecteur de caractéristiques de chacun des exemples de l'ensemble d'apprentissage
  - Couverture, discrétisation de l'espace de représentation
  - Prendre en compte la variabilité
  - → KPPV



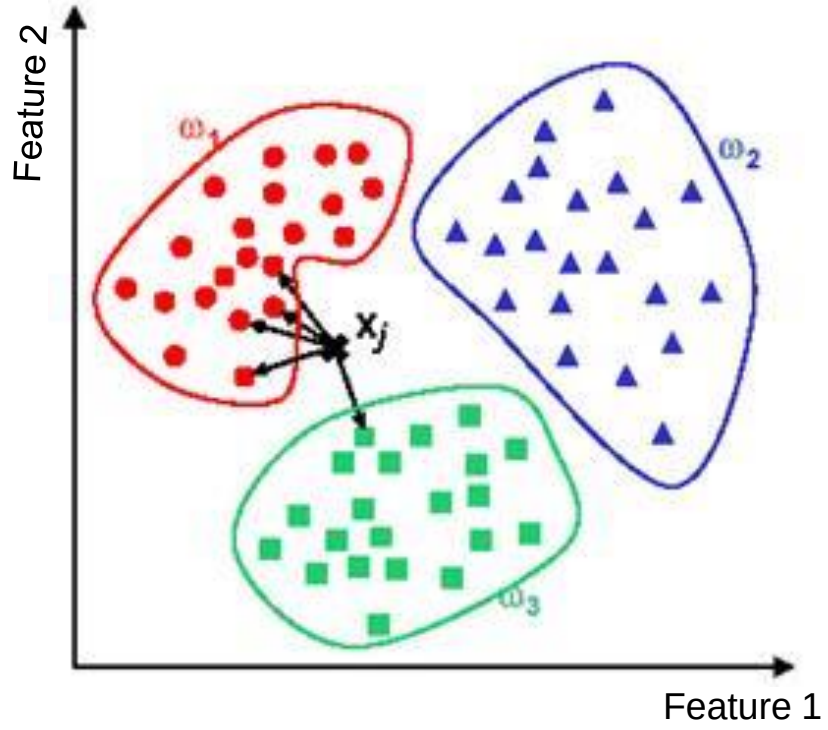




Ceci est une vision simpliste du problème pour illustrer l'importance du choix des exemples d'apprentissage et du modèle choisi ensuite

# Premiers modèles…

**Supervised classification : k-Nearest-Neighbors (kNN)**

- We have a training set with feature vectors tagged with the corresponding classes ($w_i$)
- The unknown vector $X_j$ is classified with/inside the most represented class among its k nearest neighbors



# Premiers modèles…

**Supervised classification : k-Nearest-Neighbors (kNN)**

- We have a training set with feature vectors tagged with the corresponding classes ($w_i$)
- The unknown vector $X_j$ is classified with/inside the most represented class among its k nearest neighbors

**Algorithme 1 :** Algorithme implémentant la règle des k-ppv

**Données :**
- *appbase* : base d'exemples de référence, contenant les valeurs des descripteurs
- *etiquettes* : étiquettes des exemples de la base de référence
- *individu* : exemple à classer
- $k$ : nombre de voisins à prendre en compte

**Résultat :**
- *classe* : étiquette de la classe proposée

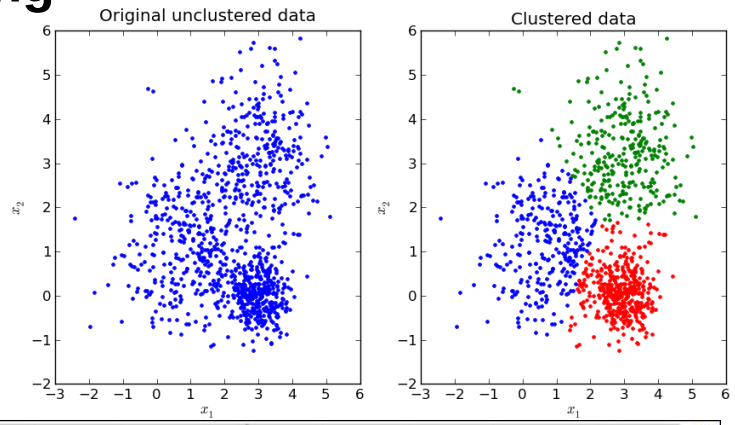
**début**
  **pour** *chaque individu* $I_r \in$ *dans appbase* **faire**
    dist[r] $\leftarrow$ distance (individu, $I_{app}$)
  **end**
  Trier conjointement (*dist*, *etiquettes*) sur la valeur de *dist* croissante
  *classe* $\leftarrow$ étiquette majoritaire dans *etiquettes*$[0 : k]$
  **retourner** *classe*
**end**

# Premiers modèles…

## Unsupervised classification : k-means clustering

- No tagged data available ⇨ learning impossible
- We look for k classes starting from k centers ($G_i$)

**Objectif :** minimising the intra-class variance



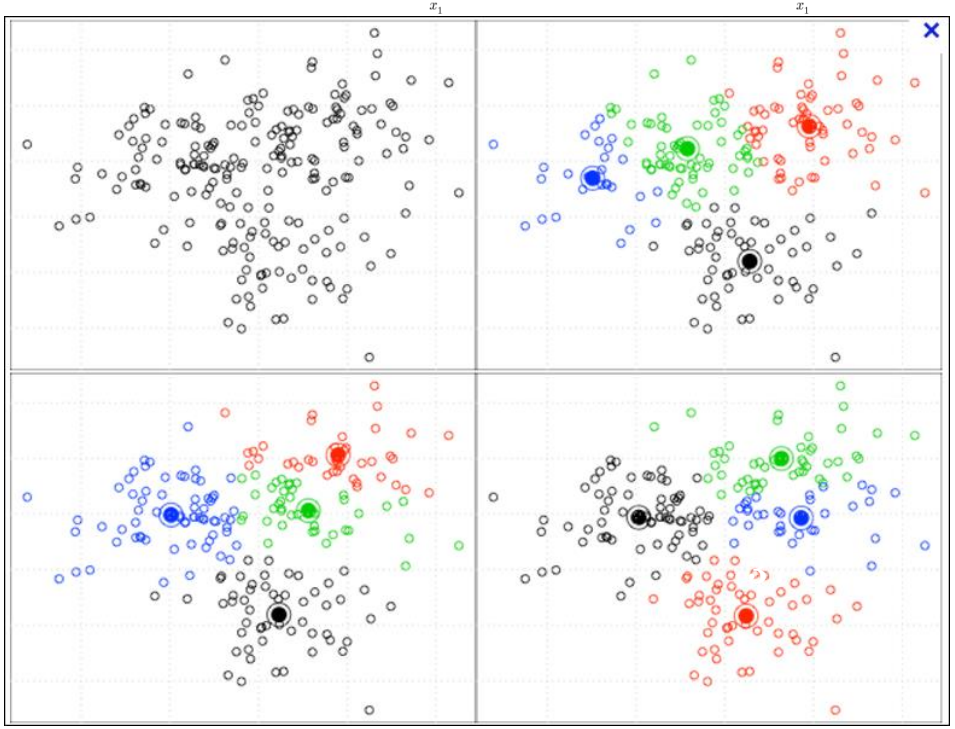

**Algorithm:**

**1 - Choose K centers randomly**

**2 – Repeat :**

**a/ Allocate each x to the closest center Gi**

**b/ Compute the new Gi until stabilization**



**Dependent from initialization →**

# Premiers modèles…

## Clustering on images

Group together pixels by color, automatic segmentation: k-means, k = 2

# Pour aller plus loin, prenons un exemple…

**Prédiction de l'infarctus du myocarde**

On sait que la probabilité de développer un infarctus du myocarde (IM) augmente avec l'âge et avec le taux de cholestérol LDL.

- Comment développer un programme qui prédise le risque d'IM à partir de ces deux variables ?

*Exemple tiré et adapté de [Denoeux2018]*

# Approche à base de règles (système expert)

**Prédiction de l'infarctus du myocarde**

On sait que la probabilité de développer un infarctus du myocarde (IM) augmente avec l'âge et avec le taux de cholestérol LDL.

- Comment développer un programme qui prédise le risque d'IM à partir de ces deux variables ?

- **Approche de type « système expert » :**
  - Modéliser la connaissance d'un médecin par des règles de la forme :
  - SI âge > 60 ET ldl > 10 ALORS risque élevé
  - SI 50 < âge  ET ldl > 8  ALORS risque moyen
  - ...

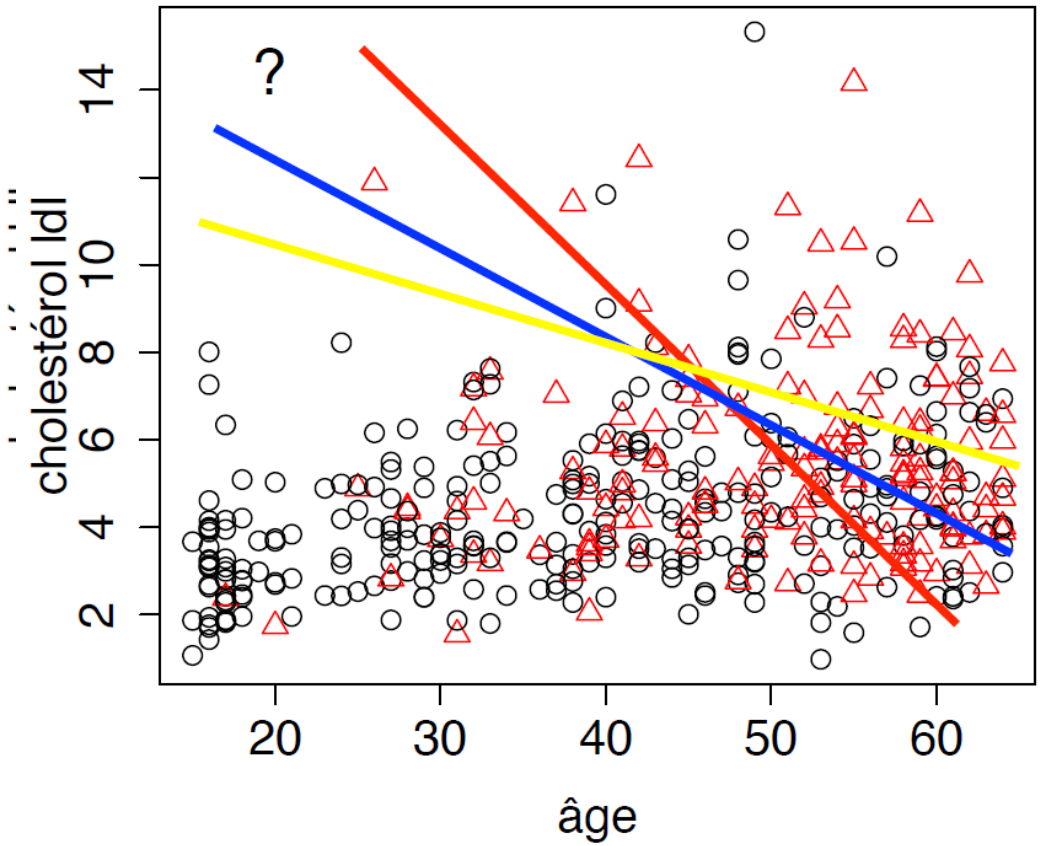
- Boite blanche → Mais approche difficile à mettre en œuvre lorsque le nombre de variables explicatives devient important.

# Approche par apprentissage

**Approche de type ML ➔ Premier modèle : Régression logistique**

**Préparation des données**

- Constituer une base d'apprentissage
- Comment distinguer / séparer au mieux les deux classes ?
- Trouver la frontière
- Modèle linéaire :
  la droite séparatrice optimale ?



# Approche par apprentissage

**Approche de type ML ➔ Premier modèle : Régression logistique**

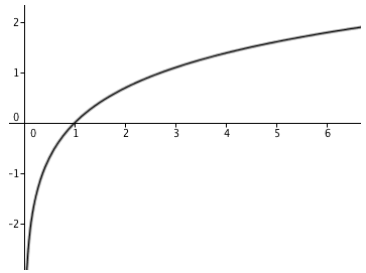
- On ne peut pas prédire à coup sûr l'occurence d'un IM à partir de l'âge et du taux de cholestérol, mais on peut chercher à estimer sa probabilité

  notée p(x) = $\mathbb{P}(\underbrace{\text{IM}}_{y=1} \mid \underbrace{\text{âge, Idl}}_{\mathbf{x}})$ sous une formulation mathématique

- Modèle linéaire, on pose : $\ln \frac{p(\mathbf{x})}{1-p(\mathbf{x})} = w_0 + w_1 \times \text{âge} + w_2 \times \text{Idl}$

- Formulation équivalente : $p(\mathbf{x}) = \frac{1}{1+\exp(-w_0 - w_1 \times \text{âge} - w_2 \times \text{Idl})}$

- Problème : comment déterminer les coefficients w0, w1 et w2 ?
  - Si y = 1 (IM avéré), on veut avoir p(x) aussi grand que possible.
    On définit l'erreur dans ce cas par ***- ln (p(x))*** ➔ erreur grande qd p(x) est proche de 0
  - Symétriquement, si y = 0 (pas d'IM), on veut avoir p(x) aussi petit que possible.
    L'erreur est alors ***- ln (1 - p(x))*** ➔ erreur grande qd p(x) est proche de 1
  - Formule générale : $\text{erreur} = -y \ln p(\mathbf{x}) - (1-y) \ln (1-p(\mathbf{x}))$
  - Erreur totale pour un ensemble d'apprentissage {(x1; y1), … ,(xn; yn) } mesurée par l'**entropie-croisée** : $C(\underbrace{w_0, w_1, w_2}_{\mathbf{w}}) = \sum_{i=1}^{n} \text{erreur}_i$

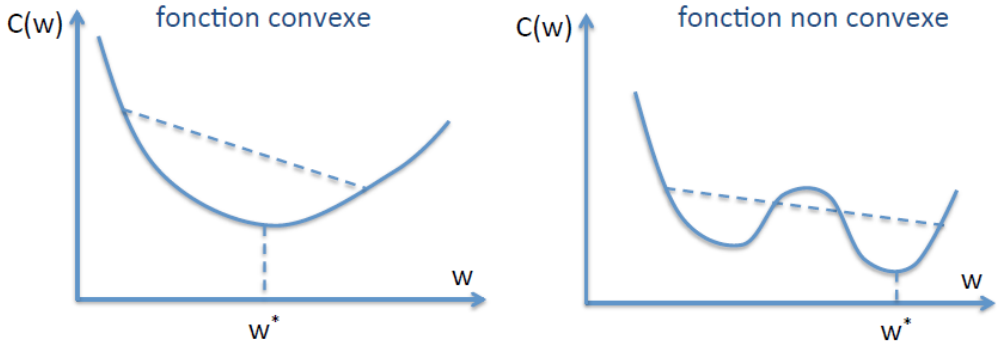


# Approche par apprentissage

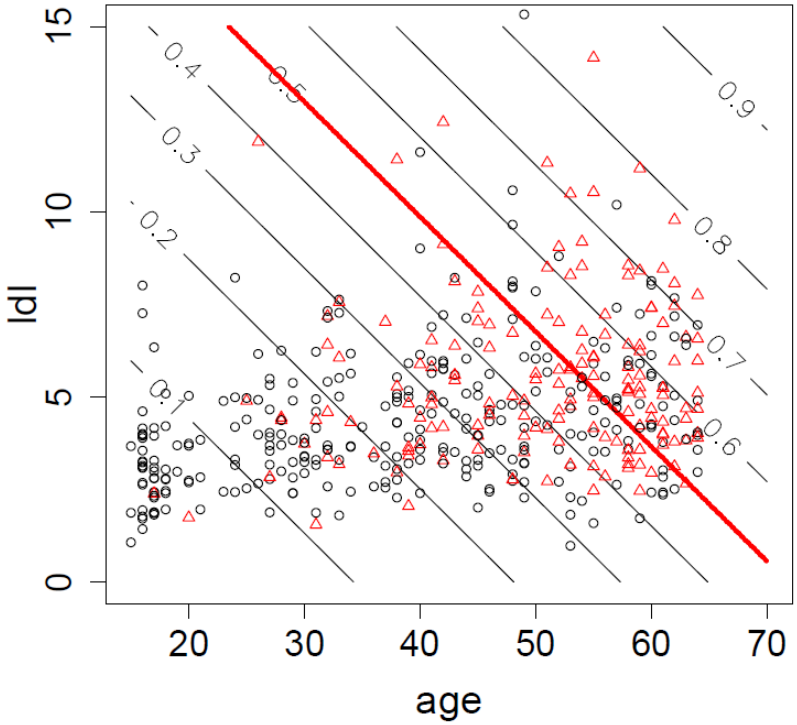
## Approche de type ML ➔ Premier modèle : Régression logistique

### Apprentissage

- Une fois définie une fonction d'erreur, le problème de l'apprentissage devient un problème d'optimisation : rechercher le vecteur de coefficient w* qui minimise l'erreur.
- Dans le cas de la régression logistique, ce vecteur est unique car la fonction d'erreur est convexe.




- Le vecteur solution w* peut être obtenu par un algorithme itératif.



# Approche par apprentissage

## Approche de type ML ➔ Premier modèle : Régression logistique

### Exploitation

- Une fois déterminé le vecteur de coefficient w optimal, on dispose d'un **programme classifieur permettant de classer de nouveaux individus**.

### Evaluation des performances

- Pour estimer la probabilité d'erreur du classifieur, il faut disposer d'un ensemble de test indépendant (**base de test**).
- On construit alors la **matrice de confusion** pour cet ensemble
- Avec 100 exemples

| | | Vraie classe | |
|---|---|---|---|
| | | Positif | Négatif |
| Prediction | Positif | 14 | 10 |
| | Négatif | 21 | 55 |

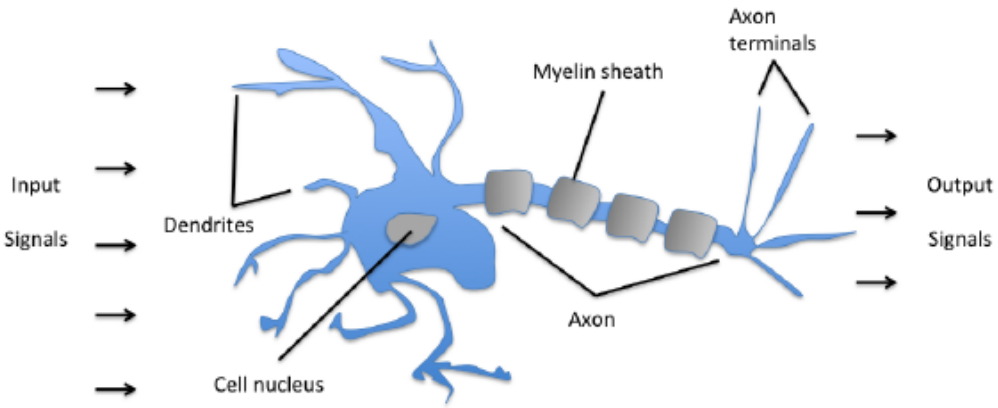
- Taux d'erreur = (10+21)/100=31%.

# Approche par apprentissage

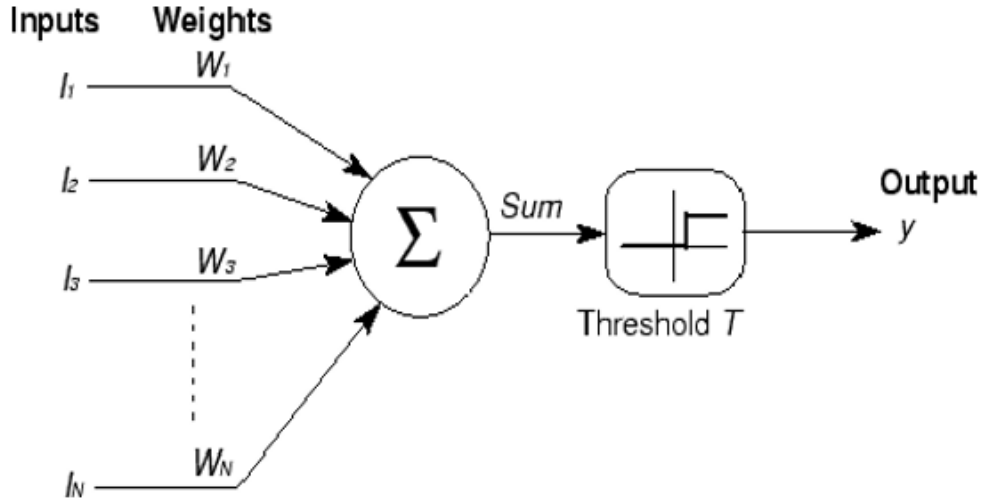
## De la régression logistique aux réseaux de neurones

**Le modèle de McCulloch et Pitts** [McCulloch1943]

- Idée : neurones biologiques vus comme des portes logiques effectuant des opérations de la logique booléenne




Schematic of a biological neuron.

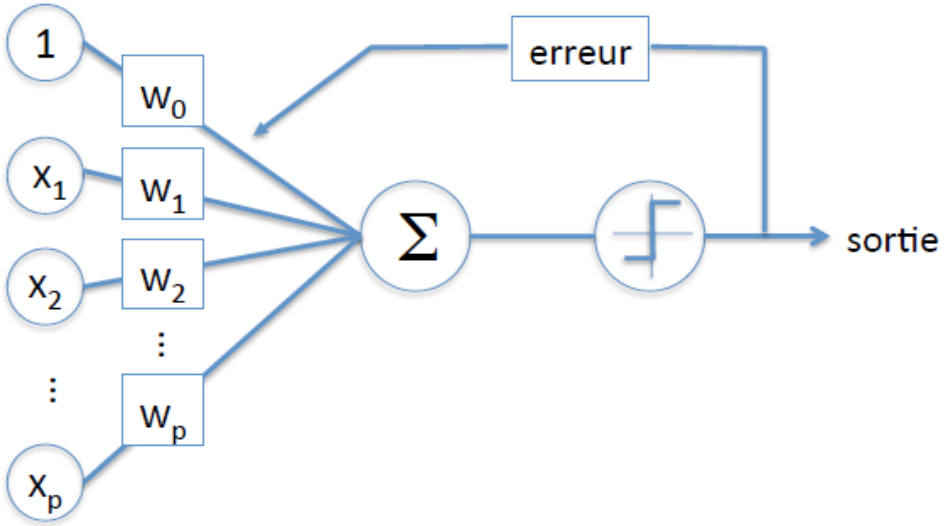


# Approche par apprentissage

## De la régression logistique aux réseaux de neurones

**Le Perceptron** [Rosenfeld1957]

- Idée : une algorithme qui apprend les poids pour résoudre des problèmes de classification binaire.



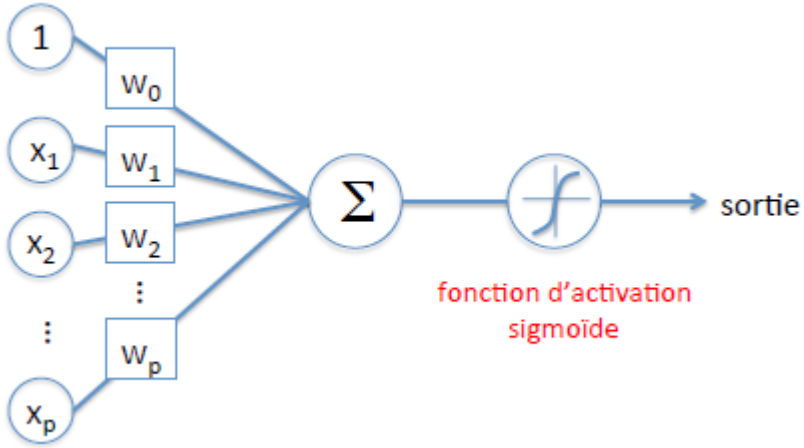

Limites :

- L'algorithme ne converge que si les deux classes sont bien séparées
- Difficilement généralisable à plus de deux classes

# Approche par apprentissage

## De la régression logistique aux réseaux de neurones

Version moderne du perceptron




Sortie :

$$g(\mathbf{x}) = \frac{1}{1 + \exp\left[-(w_0 + w_1 x_1 + \ldots w_p x_p)\right]}$$

Apprentissage des poids par minimisation de l'entropie croisée

C'est exactement le modèle de la régression logistique !

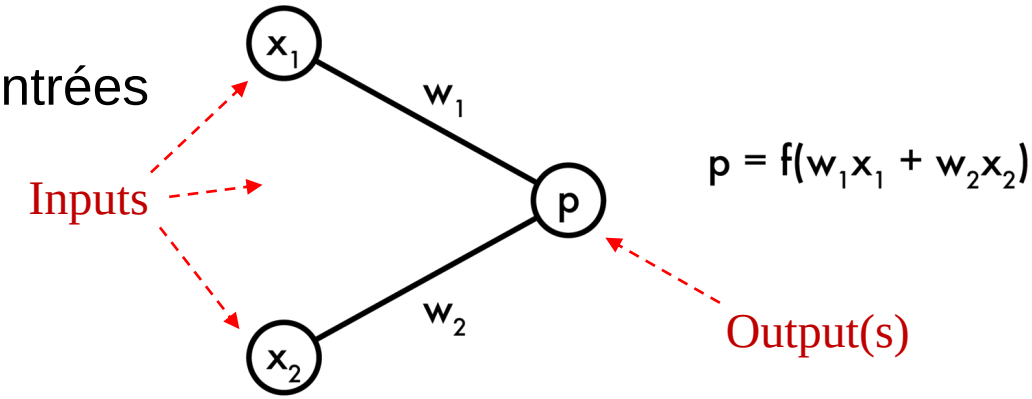
# Outrepasser le problème de "feature engineering”

## Le choix des caracteriistiques → de l’espace de représentation

- Il s’agit ici de la limitation principale du machine learning (surtout en pratique)
- ML fonctionne bien si il existe une relation claire entre les entrées du système (espace de représentation: $x_i$) et les sorties désirées
- La relation *p* est la fonction, le modéle que l’on cherche à apprendre

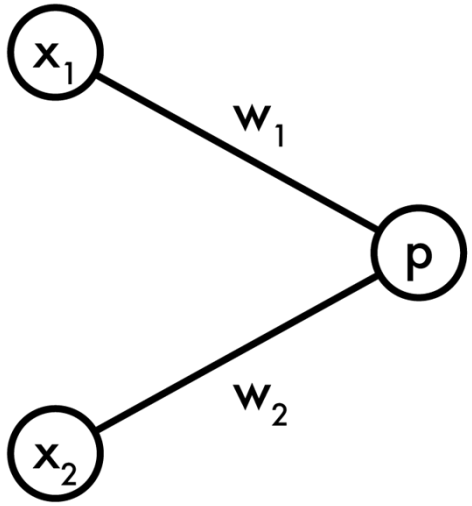
## Avec les modèles linéaires

- Calcul d’une probabilité d’appartenance pour chaque classe à partir des valeurs prises par les feature choisies
- Sortie = combinaison linéaire des entrées
- Apprentissage des poids



# Outrepasser le problème de "feature engineering”

Et si on ajoutait des transformations?

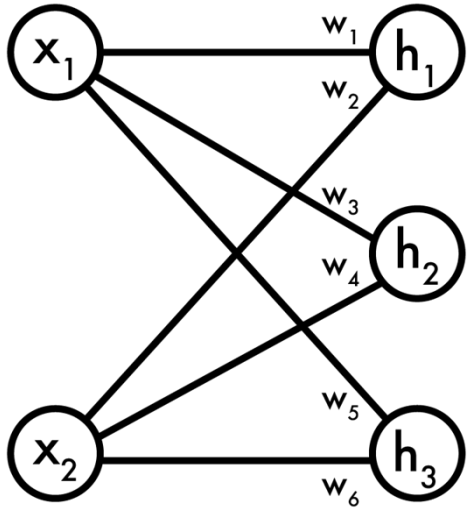



$$p = f(w_1x_1 + w_2x_2)$$

# Outrepasser le problème de "feature engineering”

Et si on ajoutait des transformations?

- Créer de “nouvelles” caractéristiques à partir des précédentes
- Ajout d’une couche supplémentaire (cachée) nommée **H**

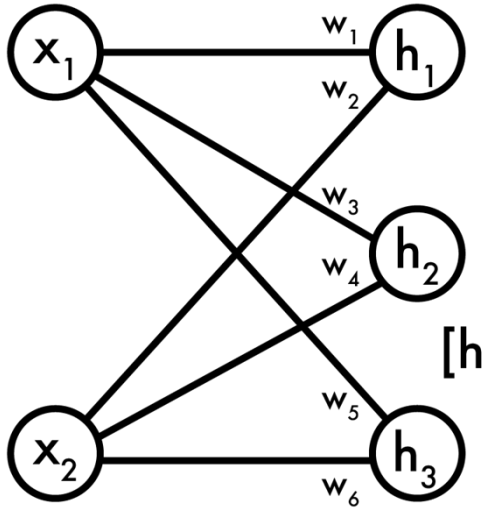



$$h_1 = \varphi(w_1 x_1 + w_2 x_2)$$
$$h_2 = \varphi(w_3 x_1 + w_4 x_2)$$
$$h_3 = \varphi(w_5 x_1 + w_6 x_2)$$

# Outrepasser le problème de "feature engineering”

Et si on ajoutait des transformations?

- Comme le modele lineaire, **H** peut etre exprimé sous forme matricielle




$$h_1 = \varphi(w_1 x_1 + w_2 x_2)$$
$$h_2 = \varphi(w_3 x_1 + w_4 x_2)$$
$$h_3 = \varphi(w_5 x_1 + w_6 x_2)$$

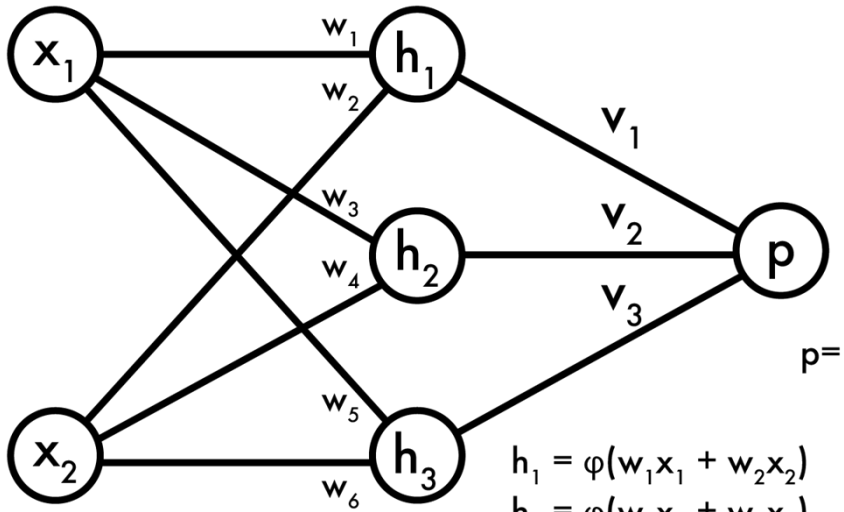
$$[h_1\ h_2\ h_3] = \varphi([x_1\ x_2]\begin{bmatrix} w_1 & w_3 & w_6 \\ w_2 & w_4 & w_5 \end{bmatrix})$$

$$\mathbf{H} = \varphi(\mathbf{Xw})$$

# Outrepasser le problème de "feature engineering”

Et si on ajoutait des transformations?

- Maintenant, les prédictions p sont fonction de la couche cachée



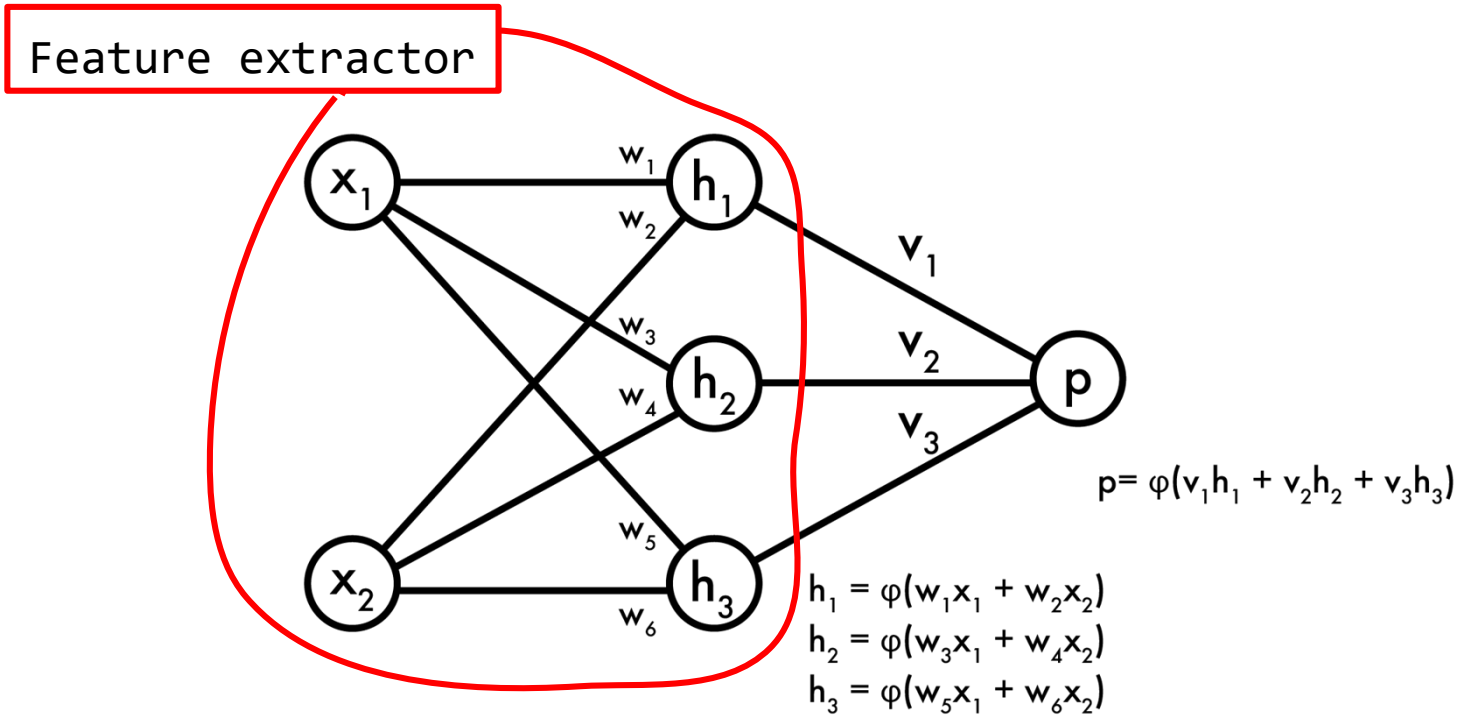

$p = \varphi(v_1 h_1 + v_2 h_2 + v_3 h_3)$

$h_1 = \varphi(w_1 x_1 + w_2 x_2)$
$h_2 = \varphi(w_3 x_1 + w_4 x_2)$
$h_3 = \varphi(w_5 x_1 + w_6 x_2)$

# Outrepasser le problème de "feature engineering”

Et si on ajoutait des transformations?

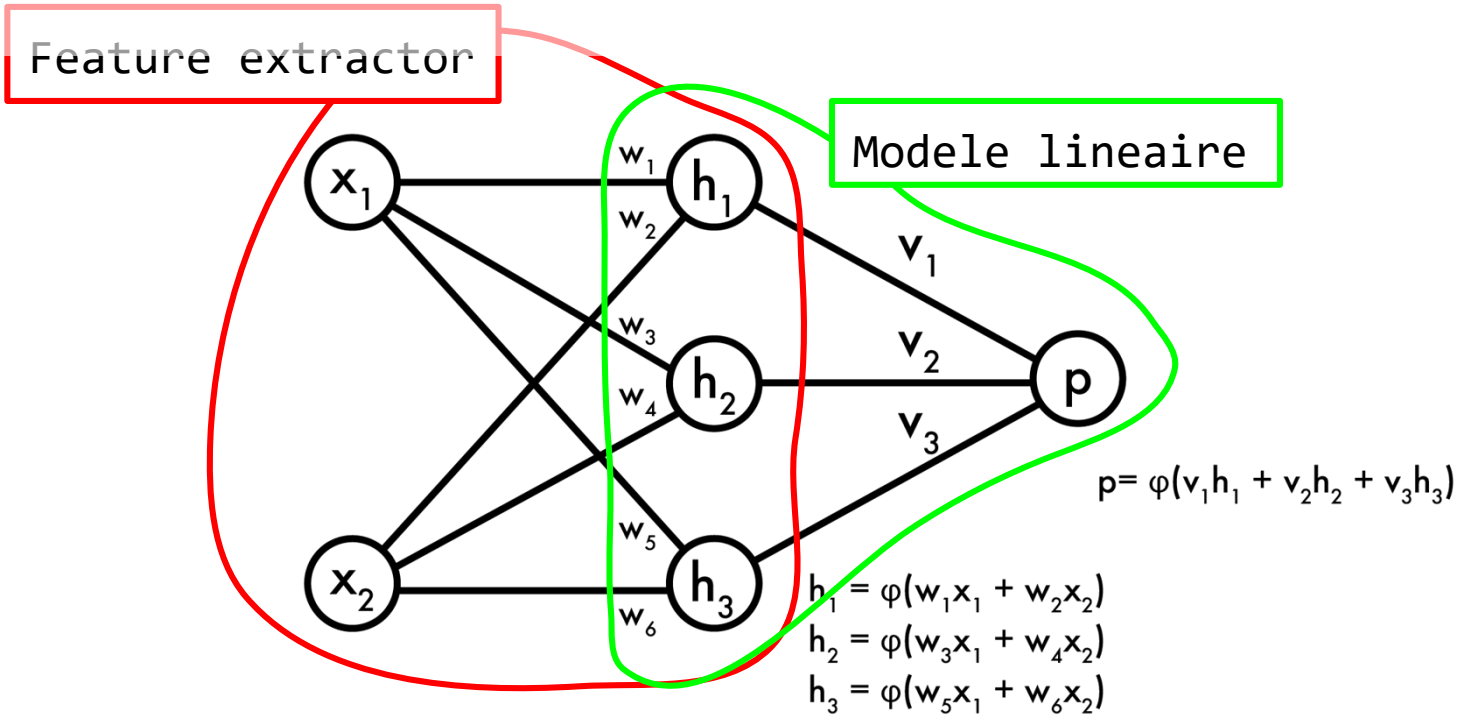
- Maintenant, les prédictions p sont fonction de la couche cachée



# Outrepasser le problème de "feature engineering”

Et si on ajoutait des transformations?

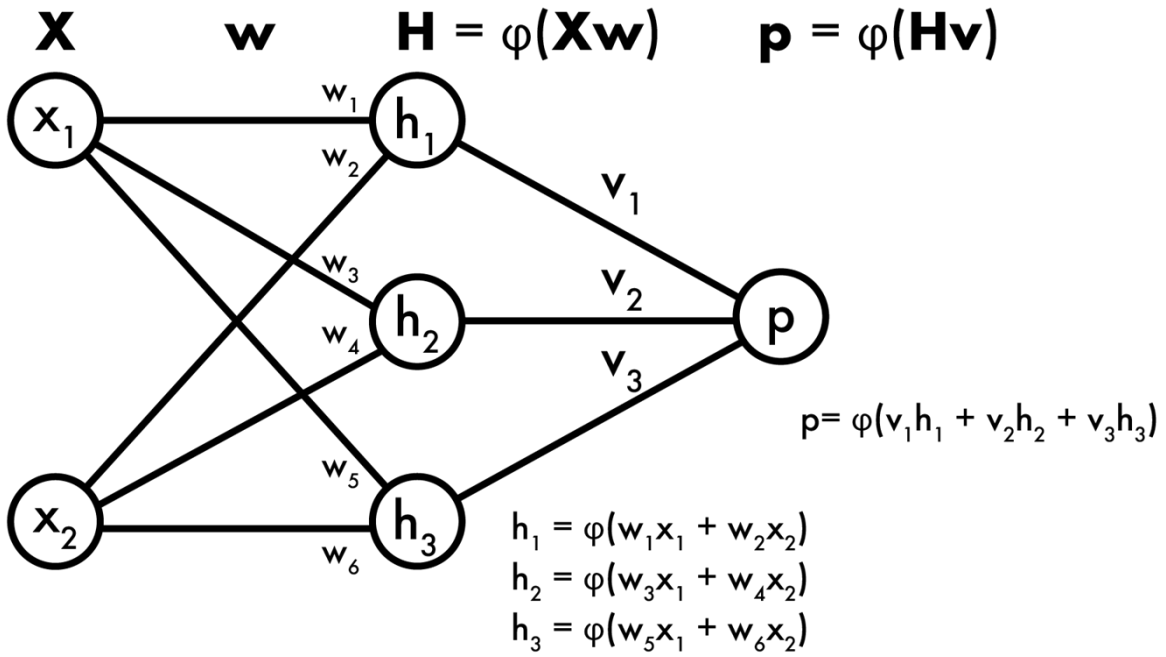
- Maintenant, les prédictions p sont fonction de la couche cachée



# Outrepasser le problème de "feature engineering”

Et si on ajoutait des transformations?

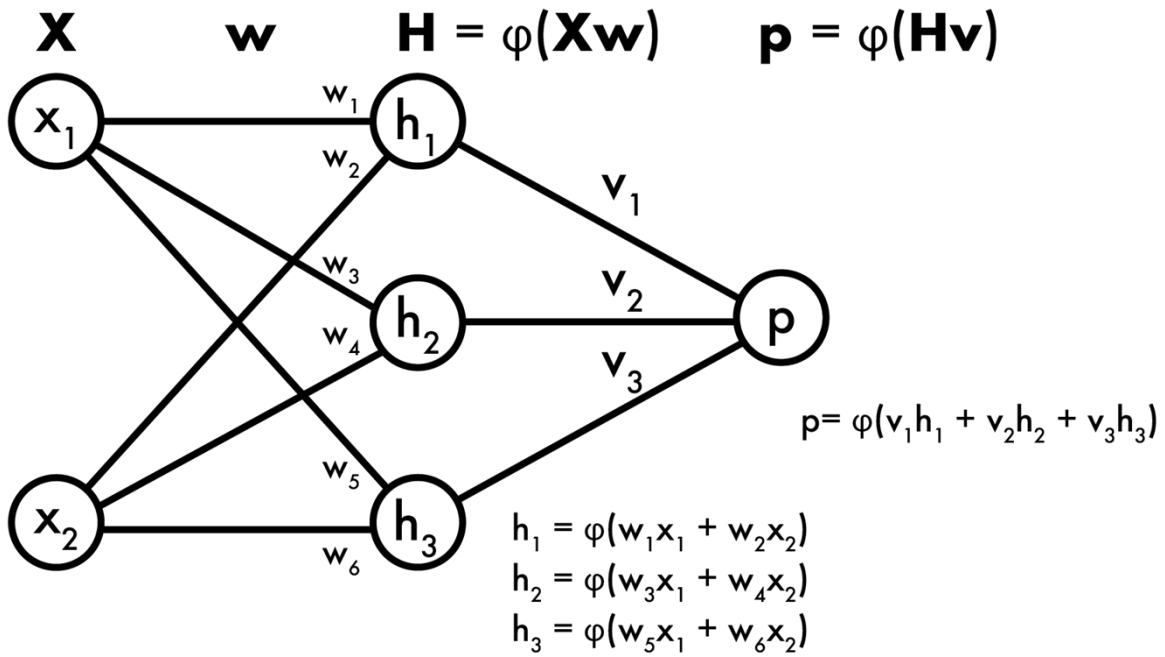
- L’ensemble du processus peut s’exprimer sous forme matricielle
- Très important pour des questions de temps de calcul



# Outrepasser le problème de "feature engineering”

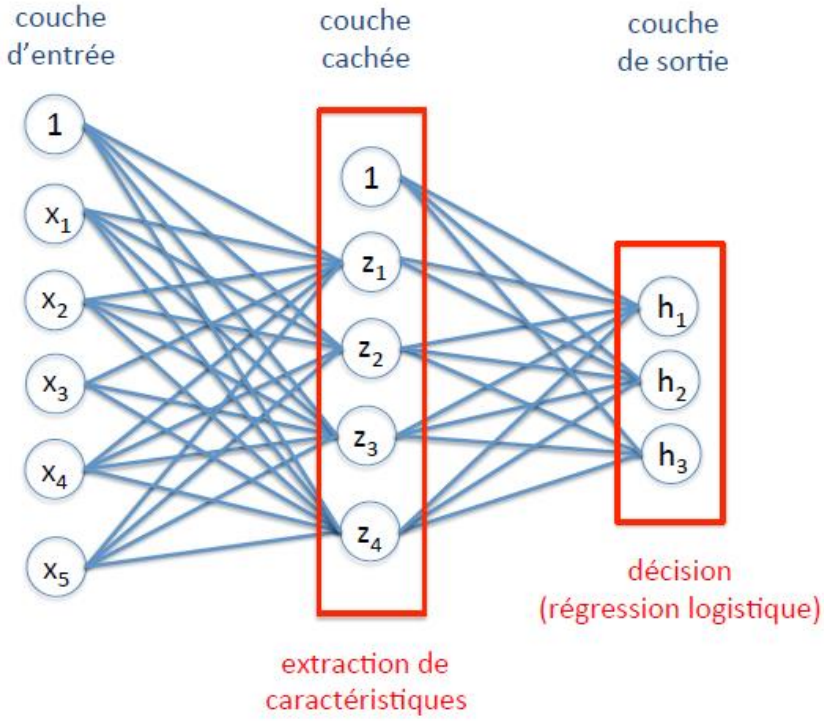
## Ici commence à apparaitre la notion / question du Deep Learning

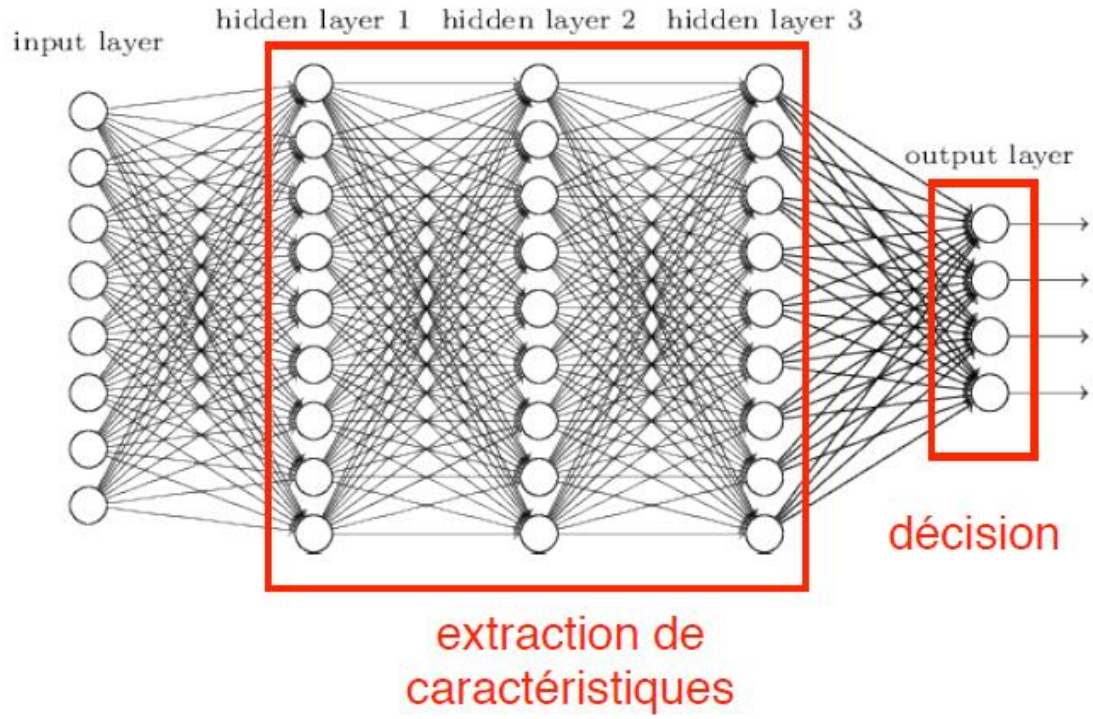
- Il est possible de multiplier le nombre de couches cachées
- Chaque couche correspondant à des fonctions φ appliquées aux combinaisons linéaires de la couche précédente



# Outrepasser le problème de "feature engineering”

## Vers les réseaux profonds




Comme pour la régression logistique, l’apprentissage des poids se fait en minimisant une fonction d’erreur telle que l’entropie-croisée.

**Problèmes:** Très grand nombre de variables (poids du réseau) - Fonction d’erreur non-convexe - beaucoup de minima locaux

➔ Nécessité de grosses capacités de calcul

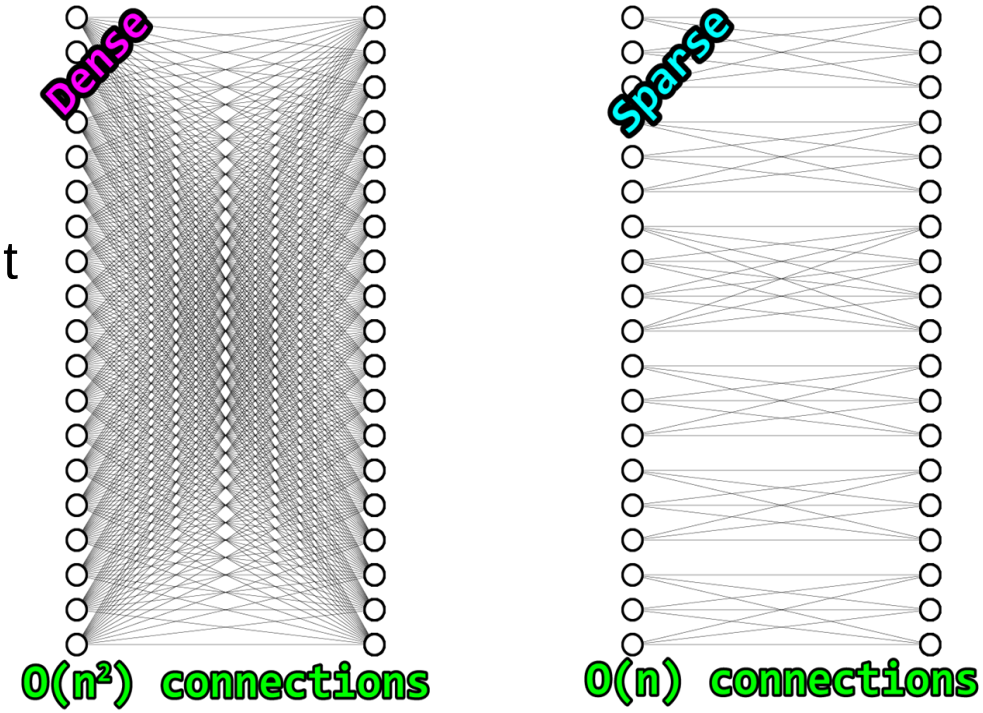
# Outrepasser le problème de "feature engineering”

Le problème a été déplacé…

- **“Architecture engineering” remplace “feature engineering”**

- Trop de poids ➔ poids partagés
  - Notion de voisinage → fenêtre d’attention
  - Convolutions : Juste des sommes pondérées de petites zones voisines dans les données (images)

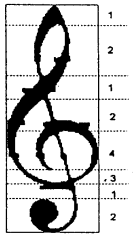
- Trop de poids ➔ boite noire produisant des résultats non interprétables par l’humain



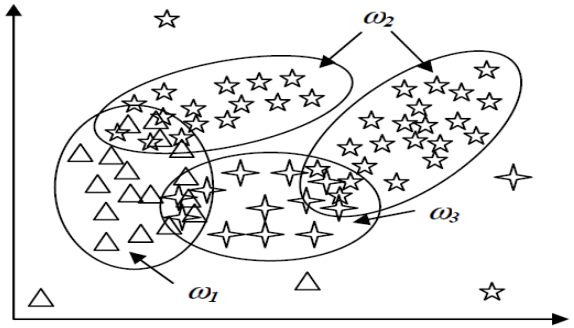
# Et si le futur était ailleurs ???

**Ne peut on pas aller plus loin que de simples descriptions statistiques ?**

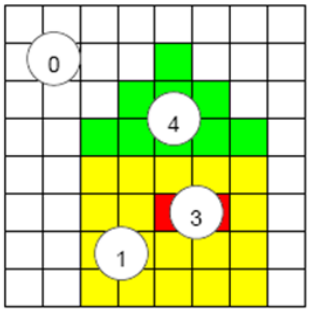
- Pendant des décennies et encore très majoritairement aujourd'hui, ML égal
- 1 Objet ➔ 1 Vecteur

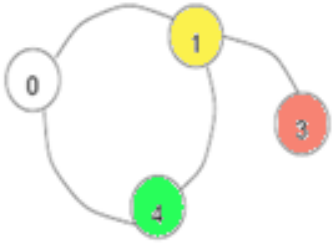


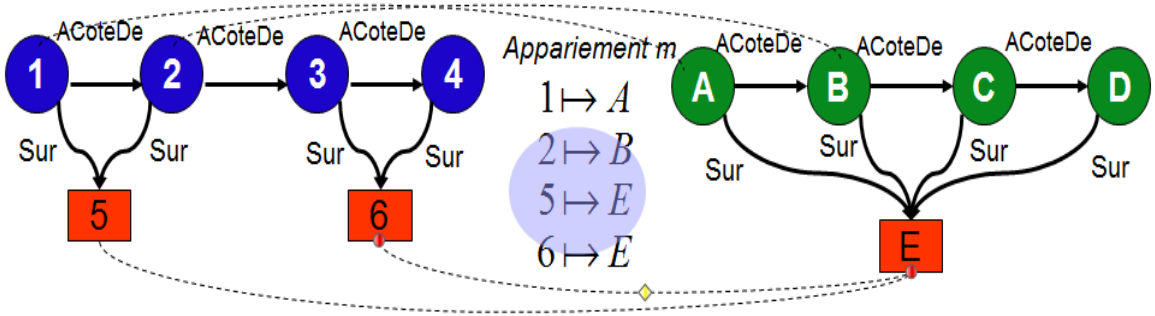
$$\vec{x} = \begin{bmatrix} x_1 \\ x_2 \\ .. \\ x_n \end{bmatrix} \in IR^{\,n}$$




- Mais une autre approche existe (et commence enfin à susciter de l'intérêt) …
- Descriptions structurelles et graph matching [Luqmann2013]
- 1 Objet ➔ 1 Graphe



# Méthodes statistiques vs structurelles ?

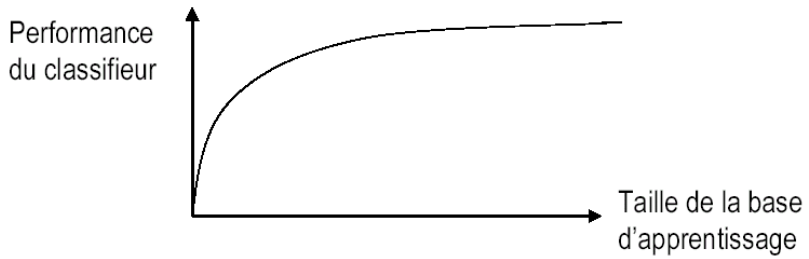
## Méthodes statistiques

- Multiplicité des descripteurs bas biveau,
- Boites noires qui qui analysent la séparabilité
  des classes et permet de conclure sur la pertinence des descripteurs choisis
- Rapide, performant sur des données connues
  Multitude de classifieurs
- Apprentissage (base, supervisé ou non)

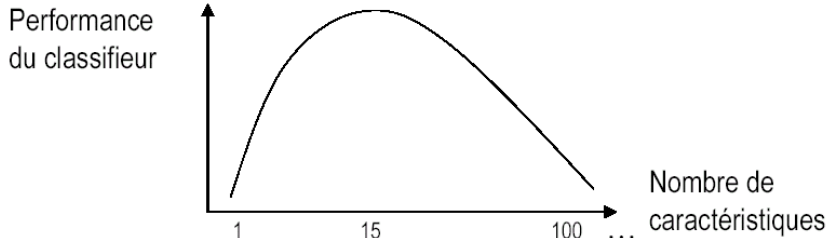
## Méthodes structurelles

- Transparence, interopérabilité
- Dimensionalité adaptative
- Prise en compte du contexte
- Reconnaissance partielle, locale et incrémentale
- Choix des modèles, des caractéristiques
- Temps de calcul, complexité ?
- Apprentissage ?

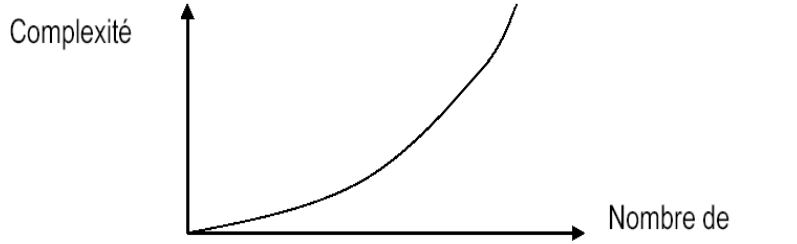
**Statistique suffisante**



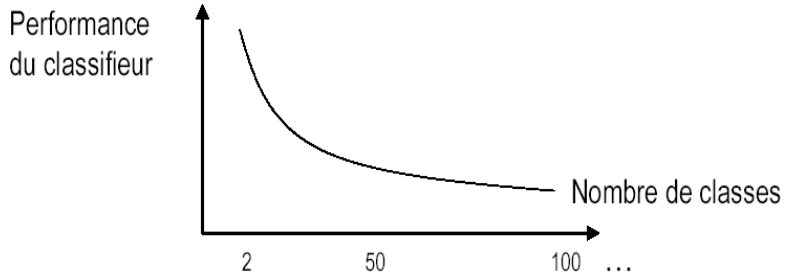

**Malédiction de la dimensionalité**




**Complexité vs nombre de caractéristiques**




**Nombre de classes**



# Dans tous les cas…

- Quelques évidences bonnes  à rappeler
  - Évidence 1 : sans de bonnes caractéristiques, aucun classifieur ne donnera de bonnes performances
  - Évidence 2 : Avec des caractéristiques raisonnablement pertinentes, tous les classifieurs ne donnent pas les mêmes résultats
  - Évidence 3 : Les données observées (base d'apprentissage) conditionnent complètement les performances
  - Évidence 4 : En cas d'échec, il faut remettre en question les descripteurs, les données observées et enfin le type de classifieur (ou son implémentation)

# Références


- Meduim 2018. Data Preprocessing With its implementation in Python, Afroz Chakure. https://towardsdatascience.com/data-preprocessing-3cd01eefd438
- George Plastiras ; Maria Terzi ; Christos Kyrkou ; Theocharis Theocharidcs Edge Intelligence: Challenges and Opportunities of Near-Sensor Machine Learning Applications. 2018 IEEE 29th International Conference on Application-specific Systems, Architectures and Processors (ASAP)
- Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville et Yoshua Bengio, « Generative Adversarial Networks », Advances in Neural Information Processing Systems 27, 2014
- Harold Hotelling. « Analysis of a Complex of Statistical Variables with Principal Components », 1933, Journal of Educational Psychology.
- Gilbert Saporta, Probabilités, Analyse des données et Statistiques, Paris, Éditions Technip, 2006,
- Kohavi, R., John, G. H. (1997), Wrappers for Feature Subset Selection, Artificial Intelligence,Volume 97, Issue 1-2, Special issue on relevance, p273 – 324.
- Aurelien Bellet, Amaury Habrard, Marc Sebban, Metric Learning (livre). Morgan & Claypool Publishers. 2015
- Muzzamil Luqman, Jean-Yves Ramel, Josep Lladós, Thierry Brouard: Fuzzy multilevel graph embedding. Pattern Recognition 46(2): 551-565 (2013)

# Références (i)


- Thierry Denoeux. Introduction à l'apprentissage automatique. Cours de l'Université de Technologie de Compiègne. Dép. Génie Informatique. Heudiasyc (UMR CNRS 7253). 2018.
- W. S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. The Bulletin of Mathematical Biophysics, 5(4) :115-133, 1943.
- F. Rosenblatt. The perceptron, a perceiving and recognizing automaton (Project PARA). Cornell Aeronautical Laboratory, 1957.
- Burr Settles, Active Learning Literature Survey. : A Computer Sciences Technical Report, University of Wisconsin–Madison, 2009
- JY Ramel, N Vincent. Semantic and interaction: when Document Image Analysis meets Computer Vision and Machine Learning ICDAR 2019. Sydney, Australia.
- Ramel JY, Vincent N., Emptoz H., Interprétation de documents techniques par "cycles perceptifs" à partir d'une perception globale du document. Revue Traitement du Signal. Vol. 15 n°2 - 1998. p1-20.

# Références (ii)


- R. Polikar, L. Udpa, S. Udpa, V. Honavar. Learn++: An incremental learning algorithm for supervised neural networks. IEEE Transactions on Systems, Man, and Cybernetics. Rowan University USA, 2001.
- Kun Deng, Yaling Zheng, Chris Bourke. New algorithms for budgeted learning May 2013 Machine Learning 90(1)
- K Trapeznikov, V Saligrama. Supervised sequential classification under budget constraints - Artificial Intelligence and Statistics, 2013
- Zhixiang Xu. Supervised Machine Learning Under Test-Time Resource Constraints: A Trade-off Between Accuracy and Cost. Washington University in St. ETDs Thesis. Louis 2014
- Xiaodan Liang Hongfei Zhou Eric Xing. Dynamic-structured Semantic Propagation Network arXiv:1803.06067v1 [cs.CV] 16 Mar 2018
- N. Ragot. Contributions à la reconnaissance de formes et applications à l'analyse de l'écrit et des documents. HDR Université de Tours. 2017.

# Licence

- Cette présentation est distribuée sous licence Creative Commons
- Attribution-ShareAlike 4.0 International