

# LE MACHINE-LEARNING EN PRATIQUE

Vincent Guigue  
[vincent.guigue@agroparistech.fr](mailto:vincent.guigue@agroparistech.fr)

# INTRODUCTION

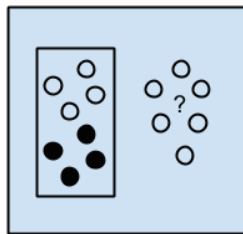
# Différents cadres de machine learning

Supervisé

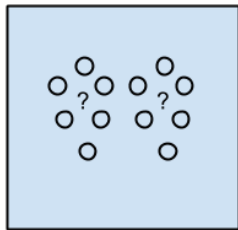
Non-supervisé

Semi-supervisé

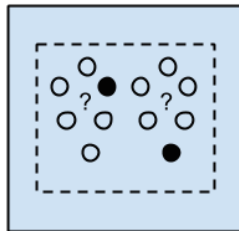
Renforcement



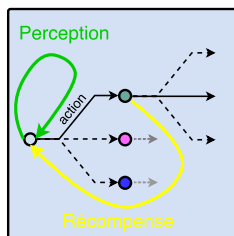
Supervised Learning Algorithms



Unsupervised Learning Algorithms



Semi-supervised Learning Algorithms



■ Différents algorithmes...

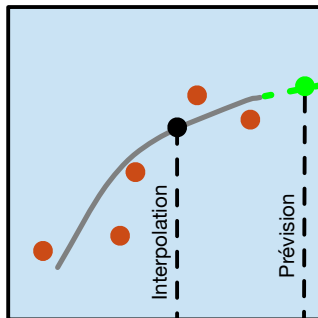
... et différentes évaluations

■ Différentes **données**, différents **coûts**...

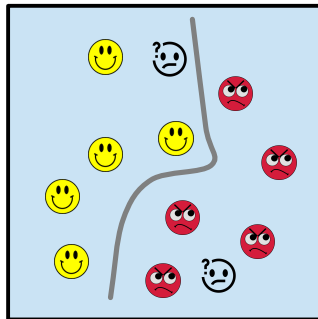
Et une nouvelle donne avec [Amazon Mechanical Turk](#)

# Grande familles de problématiques supervisées

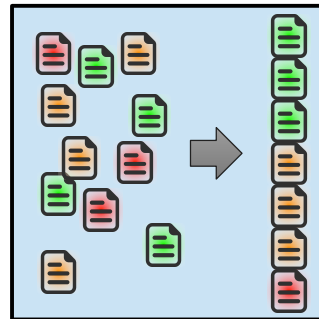
## Régression



## Classification

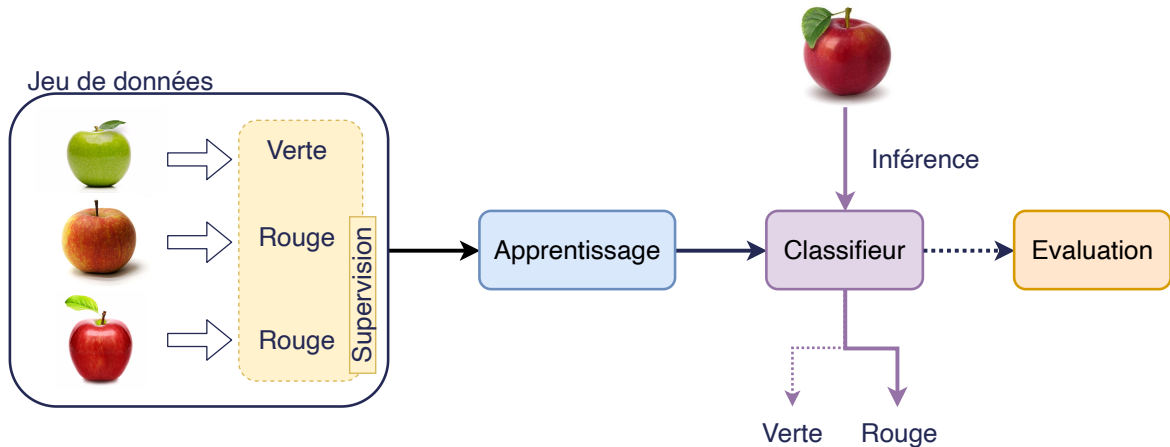


## Ordonnement



# Chaine de traitement

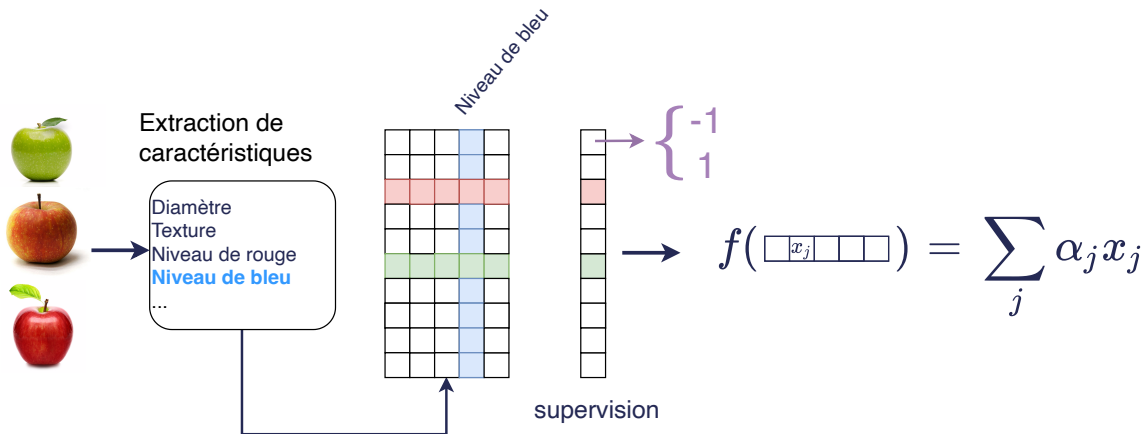
## Identifier les entrées / sorties + évaluation



... En version abstraite

# Chaine de traitement

En plus concret :



- Sélection des bonnes colonnes
- Ajout de colonnes intéressantes (calculs, sources de données externes, ...)

# Et évaluation

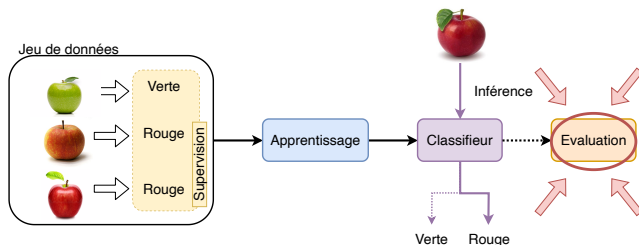
L'évaluation est **aussi importante** que le modèle lui-même !

- Modèle **sans** mesure de performances =
- Modèle + performance fausse =
- Modèle + performance basse =
- Analyse des performances  $\Rightarrow$  modèle v2

pas de sens

danger

difficile à vendre



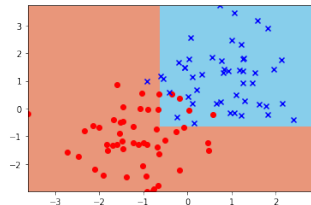
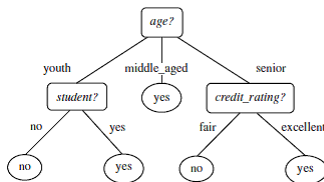
# CLASSES DE MODÈLES



# Modèles de ML : références historiques

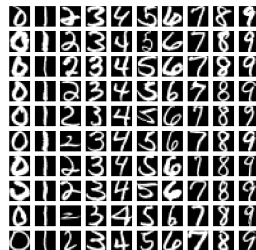
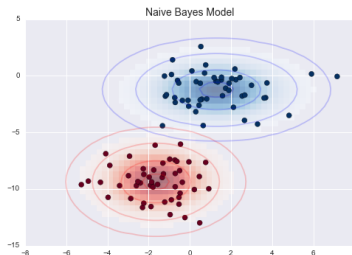
## ■ Arbre de décision : entre IA symbolique & apprentissage statistique

- Ensemble de règles
- Interprétable (selon profondeur)
- Apprenable (critère entropique)
- Adapté données mixtes cat/num



## ■ Modélisation bayésienne

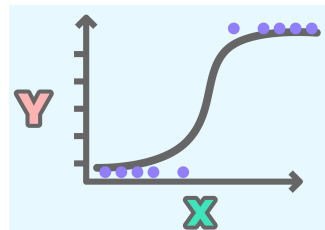
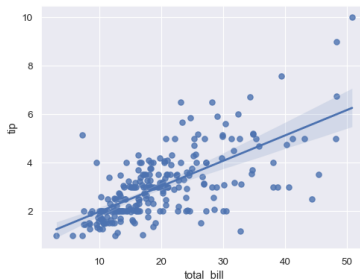
- Loïs de probabilité
- Max. de vraisemblance
- Naive Bayes
- A priori des experts



# Modèles de ML : les bonnes affaires

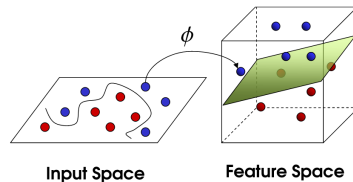
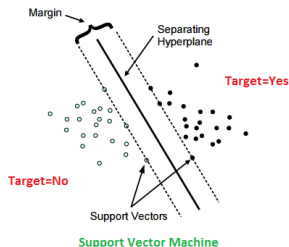
## ■ Modèles linéaires : Moindre carrés (MSE), régression logistique, ...

- Formulation simple & efficace
- Classif, régression
- Références très solides / modèle discriminant
- Descente de gradient



## ■ SVM, noyaux et méthodes discriminantes

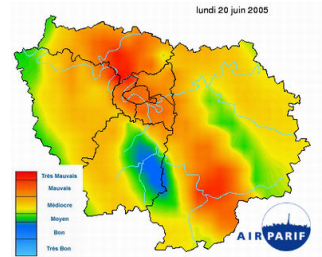
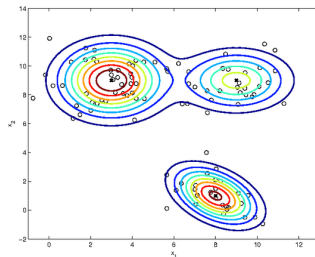
- Perceptron
- Régularisation
- SVM
- Projection non linéaire



# Modèles de ML : approches non-supervisées

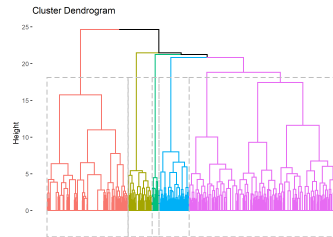
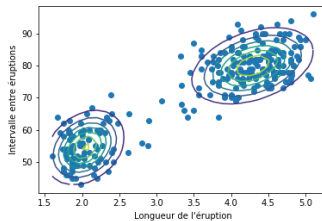
## ■ Estimation de densité

- Parzen
- Nadaraya-Watson
- Détour par les Knn
- EM



## ■ Clustering

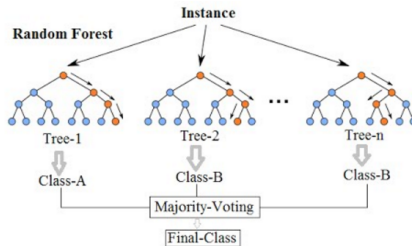
- clustering hiérarchique
- k-means / C-EM
- Clustering spectral
- A Priori



# Modèles de ML : l'état de l'art

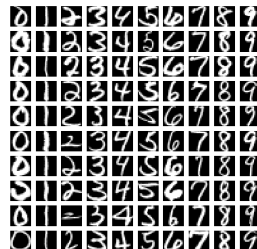
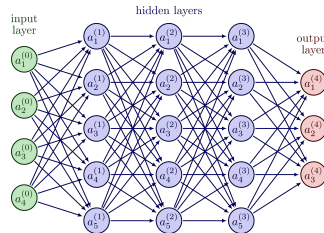
## ■ Approches ensemblistes

- Bagging
- Boosting
- Forêt, forêt aléatoire
- XGBoost



## ■ Réseaux de neurones ( $\Rightarrow$ pytorch)

- Perceptron
- Réseaux de neurones
- Rétropropagation du gradient
- Différentes architecture



# FOCUS SUR LES ARBRES DE DÉCISION

# Notations usuelles en classification

## On dispose :

- Espace de représentation  $\mathcal{X}$

Une caractéristique/variable/attribut  $X_j$

peut être continue, ordinaire ou discrète

Souvent  $\mathcal{X} = \mathbb{R}^d$

$d$  est la dimension de l'espace de représentation

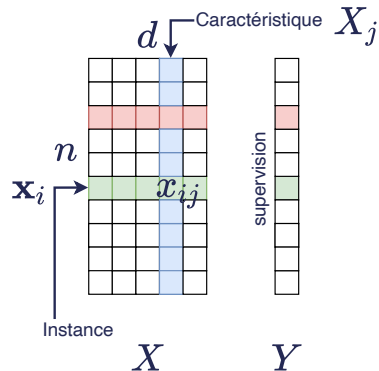
- Ensemble d'exemples/instances

$X = \{\mathbf{x}_1, \dots, \mathbf{x}_i, \dots, \mathbf{x}_n\}, \mathbf{x}_i \in \mathcal{X}$

$\mathbf{x}_i = (x_{i1}, \dots, x_{id})$

- Supervision = étiquettes  $Y = \{y_1, \dots, y_i, \dots, y_n\}$

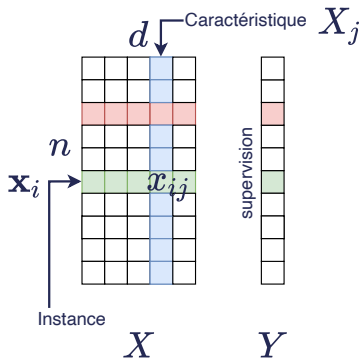
dans le cas binaire,  $y_i \in \{0, 1\}$  ou  $y_i \in \{-1, 1\}$



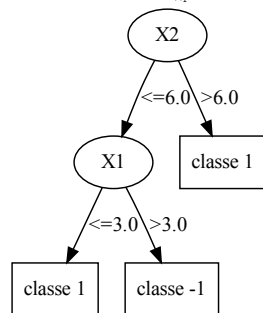
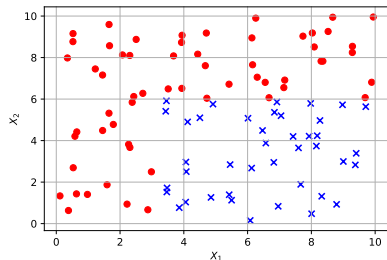
## On veut :

Trouver une fonction  $f : \mathcal{X} \rightarrow Y$  telle que la prédiction sur de futurs exemples soit la plus précise possible.

# Principe de l'arbre de décision

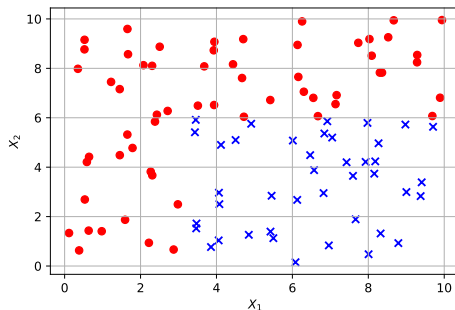


- Présenter une instance  $\mathbf{x}$  à la racine
- Nœud = test d'une variable
- Branche = résultat du test
- Feuilles = étiquette  $y$  de l'instance



# Algorithme général

classe ???



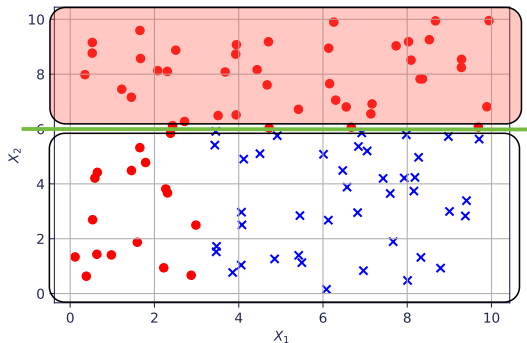
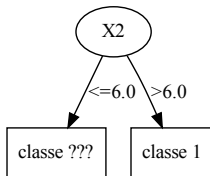
## Algorithme glouton, top-down

Initialisation à la racine avec tous les exemples

- Si le nœud n'est pas pur, alors
  - Trouver  $X_j$  la **meilleure variable** pour ce nœud et le **test associé**
  - Pour chaque test, créer un fils au nœud courant
  - Faire *tomber* les exemples du nœud courant à leur fils correspondant
- sinon transformer le nœud en feuille.



# Algorithme général

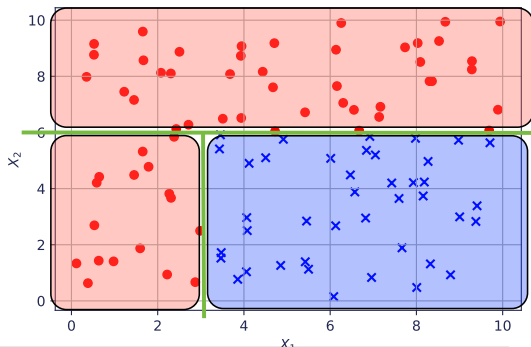
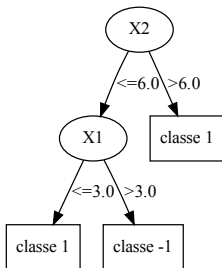


## Algorithme glouton, top-down

Initialisation à la racine avec tous les exemples

- Si le nœud n'est pas pur, alors
  - Trouver  $X_j$  la **meilleure variable** pour ce nœud et le **test associé**
  - Pour chaque test, créer un fils au nœud courant
  - Faire *tomber* les exemples du nœud courant à leur fils correspondant
- sinon transformer le nœud en feuille.

# Algorithme général

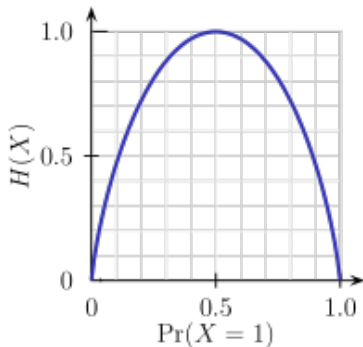


## Algorithme glouton, top-down

Initialisation à la racine avec tous les exemples

- Si le nœud n'est pas pur, alors
  - Trouver  $X_j$  la **meilleure variable** pour ce nœud et le **test associé**
  - Pour chaque test, créer un fils au nœud courant
  - Faire *tomber* les exemples du nœud courant à leur fils correspondant
- sinon transformer le nœud en feuille.

# Sélectionner la meilleure variable



## Entropie d'une variable aléatoire

Soit  $X$  une variable aléatoire pouvant prendre  $n$  valeurs  $x_i$  :

$$H(X) = - \sum_{i=1}^n P(X = x_i) \log(P(X = x_i))$$

Entropie ↗ = désordre ↗

Entropie nulle → pas d'aléa  
→ classification parfaite.

# Sélectionner la meilleure variable

## Entropie d'un échantillon : cas binaire

- $X$  un ensemble de données,  $Y$  leurs étiquettes (positif/négatif)
- $p_+$  la proportion d'exemples positifs
- $p_-$  la proportion d'exemples négatifs
- $H(Y) = -p_+ \log(p_+) - p_- \log(p_-)$

## Entropie conditionnelle

- Entropie conditionnelle :  $H(Y|X) = \sum_i P(X = x_i) H(Y|X = x_i)$
- Un test  $T$  sur une variable  $\Rightarrow$  deux partitions d'exemples de  $X$  :  $X^{(1)}$  qui vérifie le test et  $X^{(2)}$  qui ne vérifie pas le test (resp.  $Y^{(1)}$  et  $Y^{(2)}$ ).  
L'entropie conditionnelle au test  $T$  est :

$$H(Y|T) = \frac{|X^{(1)}|}{|X|} H(Y^{(1)}) + \frac{|X^{(2)}|}{|X|} H(Y^{(2)})$$

$\Rightarrow$  Gain d'info. :  $I(T, Y) = H(Y) - H(Y|T)$  à maximiser  $\Leftrightarrow$  minimiser  $H(Y|T)$

# Cas discret / Cas continu

			$X_j$		
		A		1	
		A		1	
		B		1	
		C		1	
		B		1	
		B		-1	
		A		-1	
		B		-1	
		C		-1	
		C		-1	
$X$				$Y$	

## Cas discret

$X_j \in \{A, B, C\} \Rightarrow$  Ensemble d'exemples divisé en 3  $\Rightarrow$  Calcul aisé de l'entropie :

$$H(Y|X_j) = \frac{|A|}{|X|} H(Y^{(A)}) + \frac{|B|}{|X|} H(Y^{(B)}) + \frac{|C|}{|X|} H(Y^{(C)})$$

			$X_j$		
	0.1	A		1	
	1.1	A		1	
	0.5	B		1	
	1.6	C		1	
	1.3	B		1	
	0.2	B		-1	
	0.7	A		-1	
	1.1	B		-1	
	0.8	C		-1	
	1.9	C		-1	
$X$				$Y$	

## Cas continu

$X_j \in [0, 2] \Rightarrow$  il faut tester toutes les valeurs de coupure !

- 1 Ordonnancement des valeurs de  $X_j$
- 2 Calcul de la valeur de  $H(Y|X_j)$  pour tous les tests
- 3 Conservation de la meilleure valeur





# Conclusions sur les arbres

- Approche ancienne
  - Systèmes experts
  - C4.5
  - ...
- Mais toujours d'actualité !
  - Random Forest
  - XGBoost
  - cat Boost

Comment aborder des problèmes de régression avec des arbres ?

# FOCUS SUR LES SUPPORT VECTOR MACHINE



# Focus sur les SVM

$$X = \{\mathbf{x}_1, \dots, \mathbf{x}_i, \dots, \mathbf{x}_n\}, \mathbf{x}_i \in \mathbb{R}^d, Y = \{y_1, \dots, y_i, \dots, y_n\}, y_i \in \{-1, 1\}$$

## ■ Coût (discriminant) :

$$\mathcal{L} = \sum_{i=1}^n (1 - y_i f(\mathbf{x}_i))_+, \quad (a)_+ = \max(a, 0) = \text{Partie positive}$$

### ■ Quelle différence avec les moindres carrés ?

## ■ Forme de la décision (duale/par rapport aux points d'apprentissage) :

$$f(\mathbf{x}_i) = \sum_{j=1}^{n_{app}} w_j \cdot k(\mathbf{x}_i, \mathbf{x}_j)$$

### ■ $k$ : kernel/noyau [linéaire = produit scalaire, polynomial, gaussien, ...]

## ■ Régularisation :

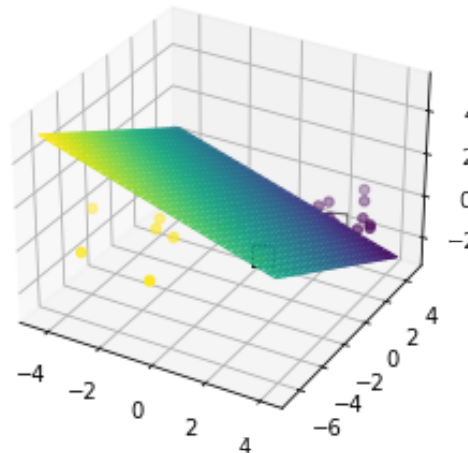
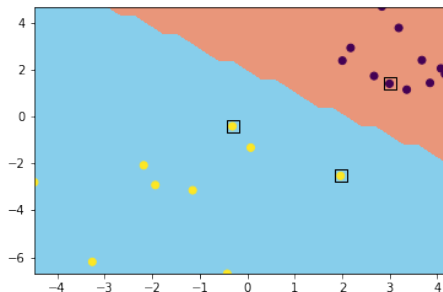
$$\mathcal{L} = \sum_{i=1}^n (1 - y_i f(\mathbf{x}_i))_+ + C \|\mathbf{w}\|^2$$

### ■ Hyper-paramètre $C$

### ■ Réflexion sur les cas extrêmes

# Notion de vecteur support de la décision

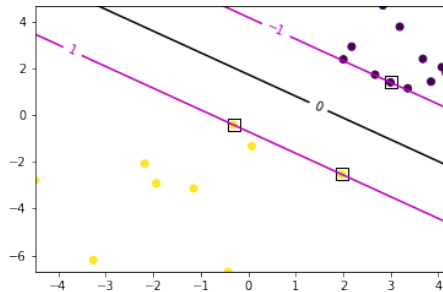
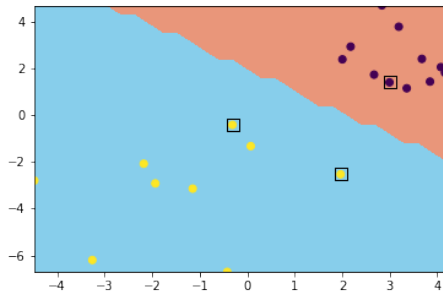
Cas linéaire :



La décision pour l'ensemble de l'espace ne repose que sur 2 vecteurs supports  
⇒ 2 points d'apprentissage ont été sélectionnés

# Notion de marge

Cas linéaire :



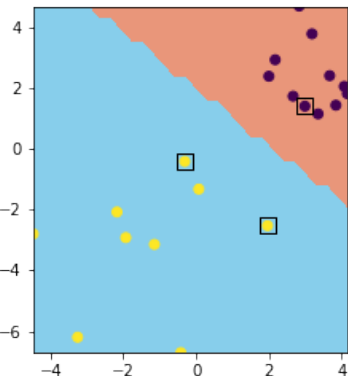
$$\mathcal{L} = \sum_{i=1}^n (1 - y_i f(\mathbf{x}_i))_+ + C \|\mathbf{w}\|^2$$

- Les vecteurs supports sont sur et dans la **marge**
  - = dans la zone de coût non nul
- Distinguer les cas **séparable** et **non-séparable**

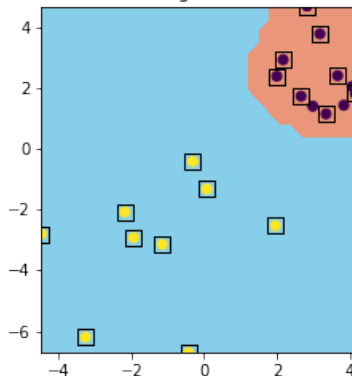
# Lien avec les mixtures de gaussiennes

En prenant :  $f(\mathbf{x}_i) = \sum_{j=1}^{n_{app}} w_j \cdot k(\mathbf{x}_i, \mathbf{x}_j)$ ,  $k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}\right)$

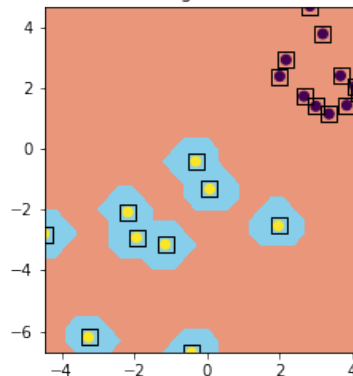
SVC linéaire



SVC gaussien

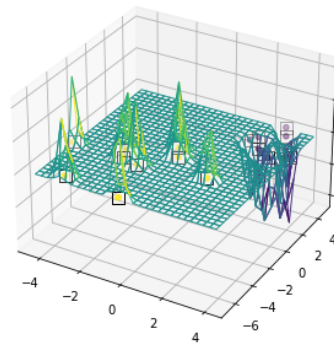
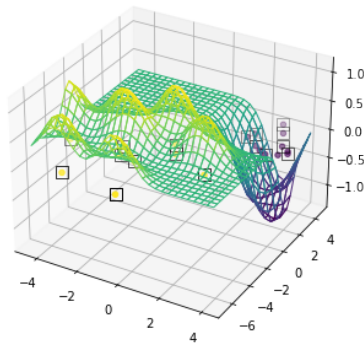
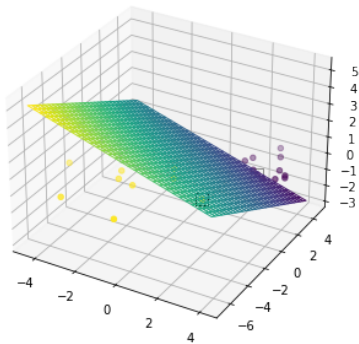


SVC gaussien



# Lien avec les mixtures de gaussiennes

En prenant :  $f(\mathbf{x}_i) = \sum_{j=1}^{n_{app}} w_j \cdot k(\mathbf{x}_i, \mathbf{x}_j)$ ,  $k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}\right)$





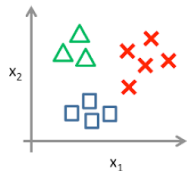
# Du cas binaire au multi-classes

Nombreux classifieurs binaires (dont les SVM)

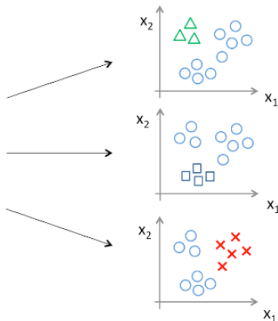
Nombreux problèmes multi-classes

Comment faire ?

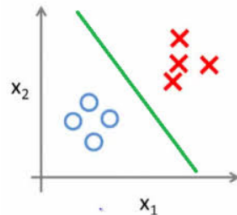
**One-vs-all (one-vs-rest):**



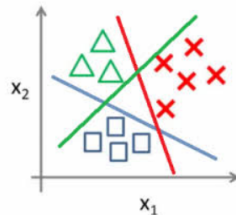
Class 1: **Green**  
Class 2: **Blue**  
Class 3: **Red**



**Binary classification:**



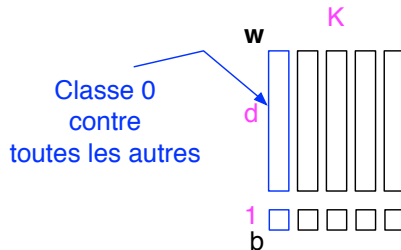
**Multi-class classification:**



Un-contre-tous (one-against-all)

# Multi-classe (2)

un contre tous (*one against all*) :  $K$  classes  
 $\Rightarrow K$  classifieurs **appris séparément** sur  
**toutes les données**



- $f(\mathbf{x}) \Rightarrow f_k(\mathbf{x})$  et critère de décision :

$$k^* = \arg \max_k f_k(\mathbf{x})$$

Quelle classe veut **le plus** l'échantillon  $\mathbf{x}$  ?

- Critères de rejet :
  - pas de  $f_k(\mathbf{x}) > 0.5$
  - plusieurs  $f_k(\mathbf{x}) > 0.5$



# SVM, quelques conclusions

- La méthode de référence des années 90+2000
- Plein de noyaux/kernel pour différents types de données
  - Graphes, images, ...
- Initialement une méthode de classification (SVM, SVC). Des extensions pour la régression (SVR).
- Problème majeur de passage à l'échelle
  - Complexité en  $\mathcal{O}(n^2)$  au mieux... Souvent  $\mathcal{O}(n^3)$  ou  $\mathcal{O}(n^4)$





# Approches ensemblistes

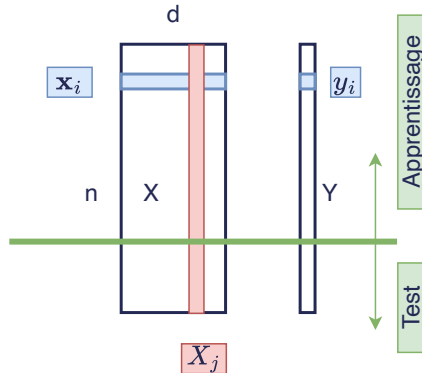
- Bagging
- Boosting
- Random Forest
- XGBoost

# EVALUATION

# Evaluation du modèle / Sélection de modèle

!! L'évaluation est aussi importante que l'apprentissage !!

- Evaluer sur les données d'apprentissage (=qui ont servi à régler les paramètres)  
⇒ **Tricherie, surestimation des performances**
- Evaluer sur des données vierges = OK



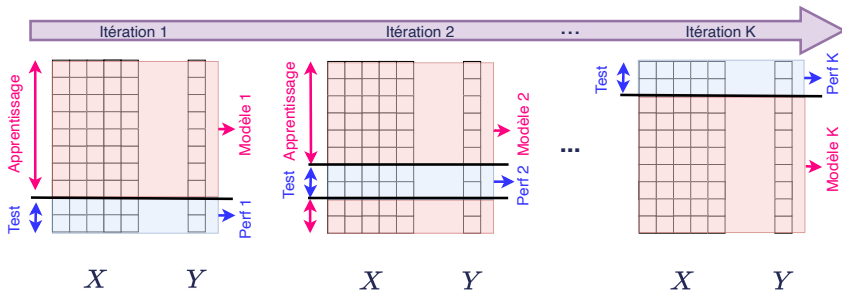
Problème de la répartition entre apprentissage et test

- La validation croisée

# Evaluation du modèle / Sélection de modèle

!! L'évaluation est aussi importante que l'apprentissage !!

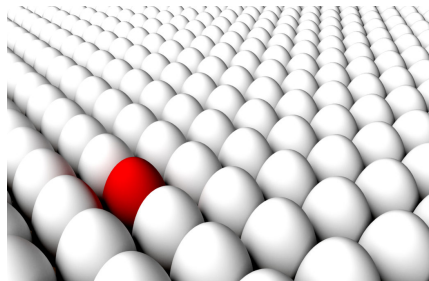
- Evaluer sur les données d'apprentissage (=qui ont servi à régler les paramètres)  
⇒ **Tricherie, surestimation des performances**
- Evaluer sur des données vierges = OK
- La validation croisée



# Attention aux cas déséquilibrés

- Anomalies,
- Fraudes,
- Entités dans les textes
- ...

⇒ Nouvelles métriques, nouvelles procédures...  
[prochain cours]



# CONCLUSION



# Conclusion

scikit-learn est un outil très puissant qui

- Propose des modèles sur l'étagère (supervisé, non-supervisé, estimation de densité, ...)
- Propose des métriques, des pré-traitements, des chaînes...
- Parallélise les calculs (cf `njob`)
- Vous permet d'insérer vos outils dans ce cadre

Allons voir ça de plus près !