

FASCICULE DE COURS

Modélisation

4^{ème} année Génie Informatique

Semestre 2 ♣



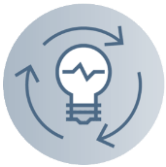
PRÉSENTÉ PAR

✓ **Dr. Chaymae MILOUDI**

chaymae.Miloudi@gmail.com

Plan du cours

- 1** Les familles de techniques
- 2** Règles d'association
- 3** Segmentation
- 4** Classification & régression
- 5** Estimation de performances & Validation Méthodes



APPRENTISSAGE NON-SUPERVISÉES

1

- Les algorithmes n'intègrent pas la notion d'entrée-sortie. Toutes les données sont équivalentes (on pourrait dire qu'il n'y a que des entrées). Dans ce cas, les algorithmes cherchent à organiser les données en groupes.
- Classes non connues à l'avance.

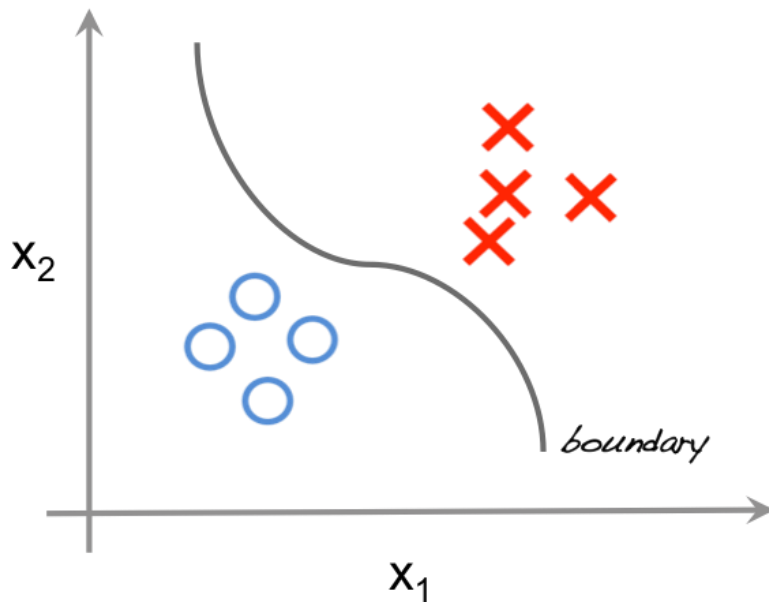
APPRENTISSAGE SUPERVISÉES

1

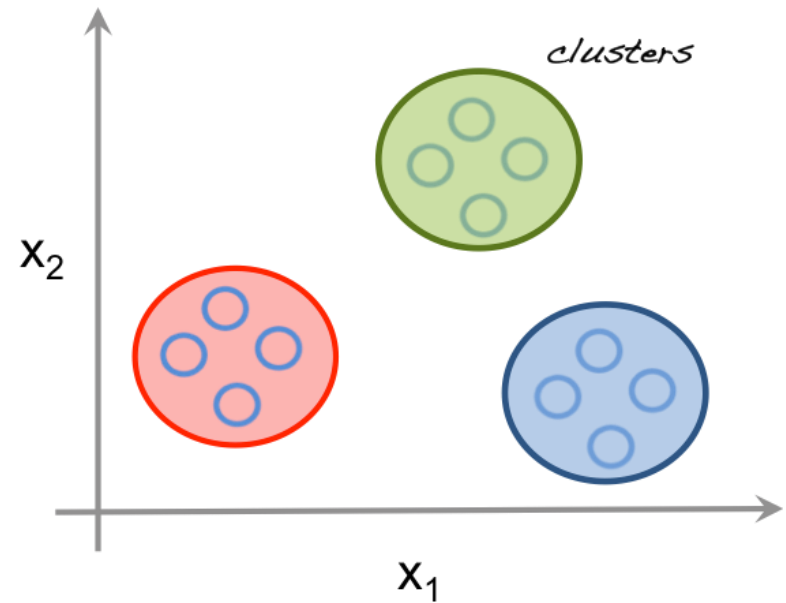
- Les algorithmes supervisés extraient de la connaissance à partir d'un ensemble de données contenant des couples entrée-sortie. Ces couples sont déjà « connus », dans le sens où les sorties sont définies a priori.
- Classes connues à l'avance.

Familles de techniques






1 Supervised learning












Unsupervised learning



Apprentissage non-supervisé: Quelques techniques

	CLUSTERING	DIMENSIONALITY REDUCTION	OUTLIER DETECTION
 Classification Ascendante Hiérarchique	✓		
 K-Moyenne (K-means)	✓		
 Principal component analysis		✓	
 Isomap		✓	
 One class SVM			✓
 Isolation Forest			✓

Apprentissage supervisé: Quelques techniques

		CLASSIFICATION	RÉGRESSION
	Régression Logistique	✓	
	Support Vector Machine (SVM)	✓	✓
	Régression Linéaire		✓
	Réseau de neurones	✓	✓
	Arbres de décision	✓	✓
	Random Forest	✓	✓
	Gradient Boosting Decision Tree (GBDT)	✓	✓
	K-Nearest Neighbors (KNN)	✓	
	Naïve Bayes	✓	

Plan du cours

- 1 Les familles de techniques
- 2 Règles d'association
- 3 Segmentation
- 4 Classification & régression
- 5 Estimation de performances & Validation Méthodes



- Repérer des règles liant les données avec un bon niveau de probabilité.
 - Découverte de relations fines entre attributs (ou variables).
 - Généralisation des dépendances fonctionnelles.
- Mettre en évidence les produits / des articles achetés ensemble.
- Transcrire la connaissance sous forme de règles d'association
- Règles du style:
< si $[P(tid, X) := \text{prémisse}]$ alors $[P(tid, Y) := \text{conséquence}]$ >

Exemple : ANALYSE DES TICKETS DE CAISSE

N° Transaction (Caddie)	Contenu du caddie			
1	Poulet	Moutarde	Œufs	Pates
2	Moutarde	Œufs		
3	Pain	Beurre	Poulet	
4	Pates			
5	Pain	Lait	Beurre	
6	Œufs	Pain		
7	Confiture			



- Une observation = un caddie.
- Ne tenir compte que de la présence des produits: peu importe leur quantité.
- Dans un caddie: le nombre de produits est variables.
- La liste des produits est immense et variable.

Exemple: **TABLEAU DES TRANSACTIONS**

- ✓ Mettre en évidence les produits / des articles achetés ensemble
- ✓ Transcrire la connaissance sous forme de règles d'association

2 si $[P(tid, X) := \text{prémisse}]$ alors $[P(tid, Y) := \text{conséquence}]$

4

N° Transaction (Caddie)	Contenu du caddie			
1	Poulet	Moutarde	Œufs	Pates

si Poulet et Moutarde alors Œufs et Pates

N° Transaction (Caddie)	Contenu du caddie	
6	Œufs	Pain

si Œufs alors Pain

Exemple: TABLEAU BINAIRE

N° Transaction (Caddie)	Contenu du caddie			
1	Poulet	Moutarde	Œufs	Pates
2	Moutarde	Œufs		
3	Pain	Beurre	Poulet	
4	Pates			
5	Pain	Lait	Beurre	
6	Œufs	Pain		
7	Confiture			



	P1	P2	P3	P4	P5	P6	P7	P8
1	1	1	1	1	0	0	0	0
2	0	1	1	0	0	0	0	0
3	1	0	0	0	1	1	0	0
4	0	0	0	1	0	0	0	0
5	0	0	0	0	1	1	1	0
6	0	0	1	0	1	0	0	0
7	0	0	0	0	0	0	0	1

désignation

P1 = Poulet

P2 = Moutarde

P3 = Œufs

P4 = Pates

P5 = Pain

P6 = Beurre

P7 = Lait

P8 = Confiture

Exemple: CODAGE DISJONCTIF COMPLET

Observation	Taille	Corpulence
1	Petit	Mince
2	Grand	Enveloppé
3	Grand	Mince



2

Observation	Taille = Petit	Taille = Grand	Corpulence = Mince	Corpulence = Enveloppé
1	1	0	1	0
2	0	1	0	1
3	0	1	1	0



Dès que l'on peut se ramener à des données 0/1 :

Il est possible de construire des règles d'association

Critères d'évaluation

1

Règle d'association :
p1 → p2

SUPPORT

- ✓ indicateur de « **fiabilité** »
- ✓ probabilité absolue :
 $P(X \cup Y)$
- ✓ $||X \cup Y|| / ||BD|| = \%$
de transactions vérifiant la règle

CONFIANCE

- ✓ Indicateur de « **précision** »
- ✓ probabilité conditionnelle :
 $P(Y/X)$
- ✓ $||X \cup Y|| / ||X|| = \%$
de transactions vérifiant l'implication

$sup(R_1) = 2$: en termes absolus
ou $sup(P_1) = 2 / 6 = 33\%$: en termes relatifs

$Conf(R_1) = sup(R_1) / sup(\text{antécédant } R_1)$
 $= sup(p_1 \rightarrow p_2) / sup(p_1) = 2 / 4 = 50\%$

Caddie	p1	p2	p3	p4
1	1	1	1	0
2	1	0	1	0
3	1	1	1	0
4	1	0	1	0
5	0	1	1	0
6	0	0	0	1



« Bonne » règle = règle avec un support et une confiance élevée

Critères d'évaluation: **ANALYSE DES TICKETS**

{ "crème" } \rightarrow { "pain" }

1



2

ID	PRODUITS
1	pain, crème, eau
2	crème
3	pain, crème, vin
4	eau
5	crème, eau

Support = Prob. (crème et pain) :

4

$$\text{Sup} = \frac{\text{nb}(\text{tran. contenant crème et pain})}{\text{nb_total}(\text{tran.})} = \frac{2}{5} = 0.4$$

Confiance = Prob(crème et pain / crème) :

$$\text{Conf} = \frac{\text{nb}(\text{tran. contenant crème et pain})}{\text{nb}(\text{tran. contenant crème})} = \frac{2}{4} = 0.5 = \frac{\text{sup}(\text{crème et pain})}{\text{sup}(\text{crème})}$$

Démarche d'extraction des règles d'association

Paramètres : Fixer un degré d'exigence sur les règles à extraire

- ✓ Support min. (exp. 2 transactions)
- ✓ Confiance min. (exp. 75%)

L'idée est surtout de contrôler (limiter) le nombre de règles produites

Démarche : Construction en deux temps

- ✓ recherche des itemsets fréquents (support \geq support min.)
- ✓ à partir des itemsets fréquents, produire les règles (conf. \geq conf. min.)

Caddie	p1	p2	p3	p4
1	1	1	1	0
2	1	0	1	0
3	1	1	1	0
4	1	0	1	0
5	0	1	1	0
6	0	0	0	1

Quelques définitions :

- item = produit
- itemset = ensemble de produits (ex. $\{p1, p3\}$)
- $\text{sup}(\text{itemset})$ = nombre de transactions d'apparition simultanée des produits (ex. $\text{sup}\{p1, p3\} = 4$)
- $\text{card}(\text{itemset})$ = nombre de produits dans l'ensemble (ex. $\text{card}\{p1, p3\} = 2$)

ALGORITHME APRIORI

ETUDE DE CAS DE RECHERCHE D'ASSOCIATIONS INTÉRESSANTES

1

- Le principe de l'algorithme est de rechercher l'ensemble L_1 de tous les items apparaissant dans au moins S_{\min} transactions.
- Puis, parmi C_2 qui est le produit cartésien de L_1 avec lui-même, on construit l'ensemble L_2 de tous les couples d'items apparaissant dans au moins S_{\min} transactions.
- L'algorithme s'arrête quand L_k est vide.

Simulation



On va prendre la valeur du Support minimal = **3**

1

Caddie	p1	p2	p3	p4
1	1	1	1	0
2	1	0	1	0
3	1	1	1	0
4	1	0	1	0
5	0	1	1	0
6	0	0	0	1

1-Itemsets	Support
{p1}	4
{p2}	3
{p3}	5
{p4}	1

$$L1 = \{ \{p1\}, \{p2\}, \{p3\} \}$$

$$C2 = \{ \{p1,p2\}, \{p1,p3\}, \{p2,p3\} \}$$

Simulation



On va prendre le Support minimal = 3

1

Caddie	p1	p2	p3	p4
1	1	1	1	0
2	1	0	1	0
3	1	1	1	0
4	1	0	1	0
5	0	1	1	0
6	0	0	0	1



2-Itemsets	Support
{p1,p2}	2
{p1,p3}	4
{p2,p3}	3

$$L_2 = \{ \{p1,p3\}, \{p2,p3\} \}$$

$$C_3 = \{ \{p1,p2,p3\} \}$$



Simulation



On va prendre le Support minimal = 3

1

Caddie	p1	p2	p3	p4
1	1	1	1	0
2	1	0	1	0
3	1	1	1	0
4	1	0	1	0
5	0	1	1	0
6	0	0	0	1



3-Itemsets	Support
{p1,p2, p3}	2

$$L_3 = \emptyset$$

$$C_4 = \emptyset$$

$$F = \{ \{p1\}, \{p2\}, \{p3\}, \{p1,p3\}, \{p2,p3\} \}$$

Simulation



On va prendre le pourcentage de la confiance minimale = **65%**

2

Caddie	p1	p2	p3	p4
1	1	1	1	0
2	1	0	1	0
3	1	1	1	0
4	1	0	1	0
5	0	1	1	0
6	0	0	0	1

$\{p1, p3\}$

$p1 \rightarrow p3$: confiance = $4/4 = 100\%$

$p3 \rightarrow p1$: confiance = $4/5 = 80\%$



$\{p2, p3\}$

$p2 \rightarrow p3$: confiance = $3/3 = 100\%$

$p3 \rightarrow p2$: confiance = $3/5 = 60\%$



Utilité des règles d'association

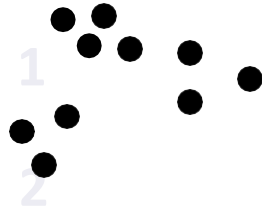
- 1 La distribution réussit à mieux cibler ses mailings.
- 2 La course à la fidélisation des clients.
- Réductions personnalisées à la caisse.
- Profil-client.
- 4 Le test des nouveaux produits.
- Le panier moyen.
- Cartes de fidélité.

Plan du cours

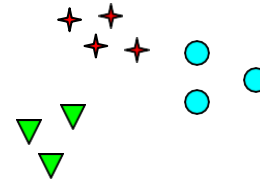
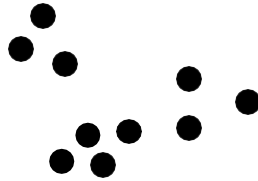
- 1 Les familles de techniques
- 2 Règles d'association
- 3 Segmentation
- 4 Classification & régression
- 5 Estimation de performances & Validation Méthodes



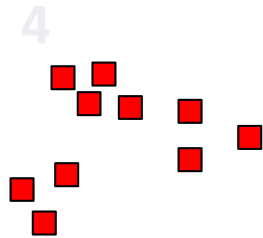
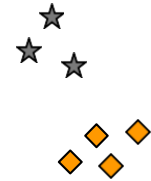
Problématique



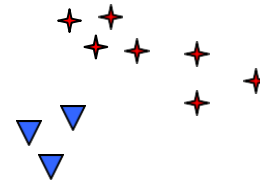
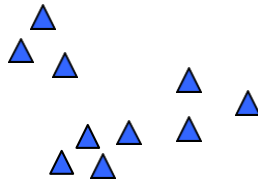
How many Clusters?



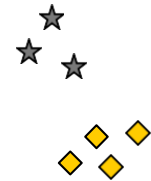
Six Clusters



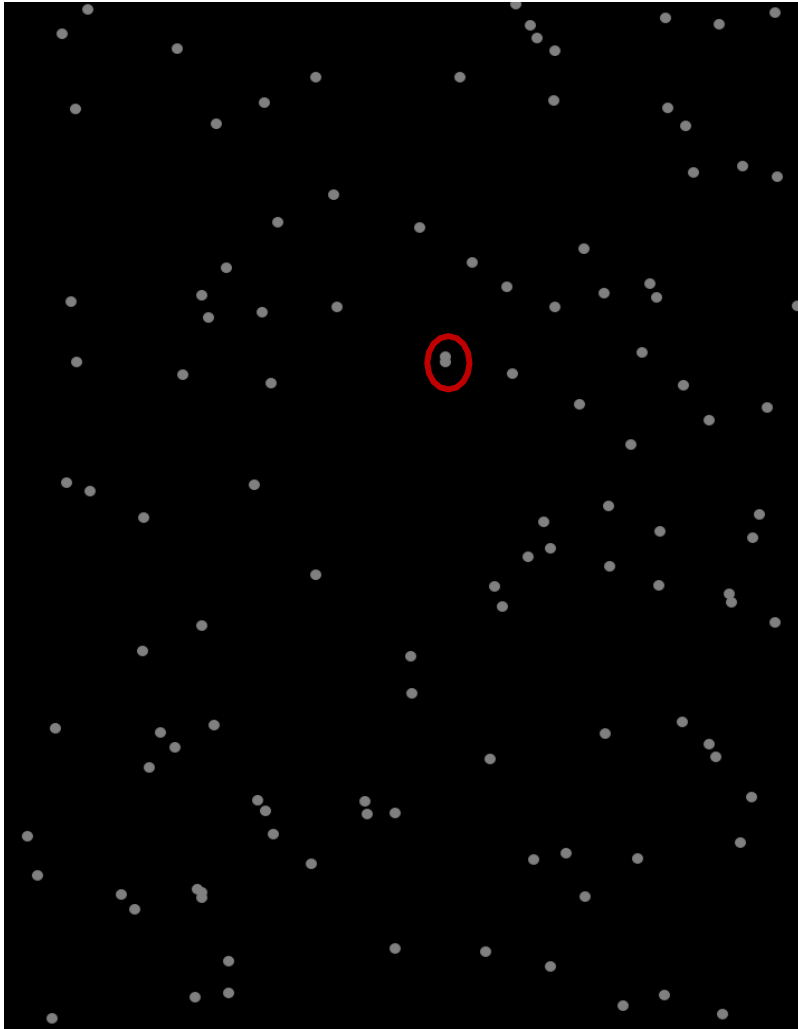
Two Clusters



Four Clusters



Principe de la segmentation



Deux individus se ressemblent le plus

SI

les points qui les représentent dans le nuage sont les plus proches

Nécessité d'une métrique de la distance

Distance Euclidienne

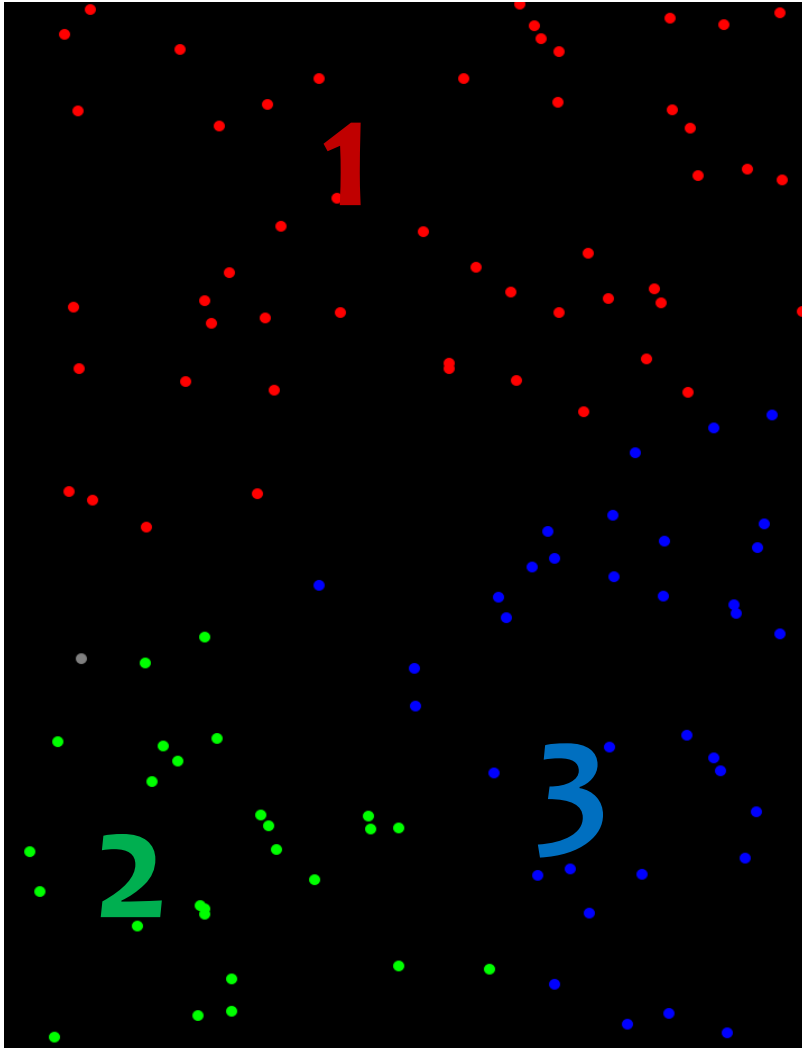
Distance de Mahalanobis

Distance de Manhattan

Distance de Ward

....

Principe de la segmentation



Deux individus se ressemblent le plus

SI

les points qui les représentent dans le nuage sont les plus proches

Nécessité d'une métrique de la distance

Distance Euclidienne

Distance de Mahalanobis

Distance de Manhattan

Distance de Ward

K-MEANS

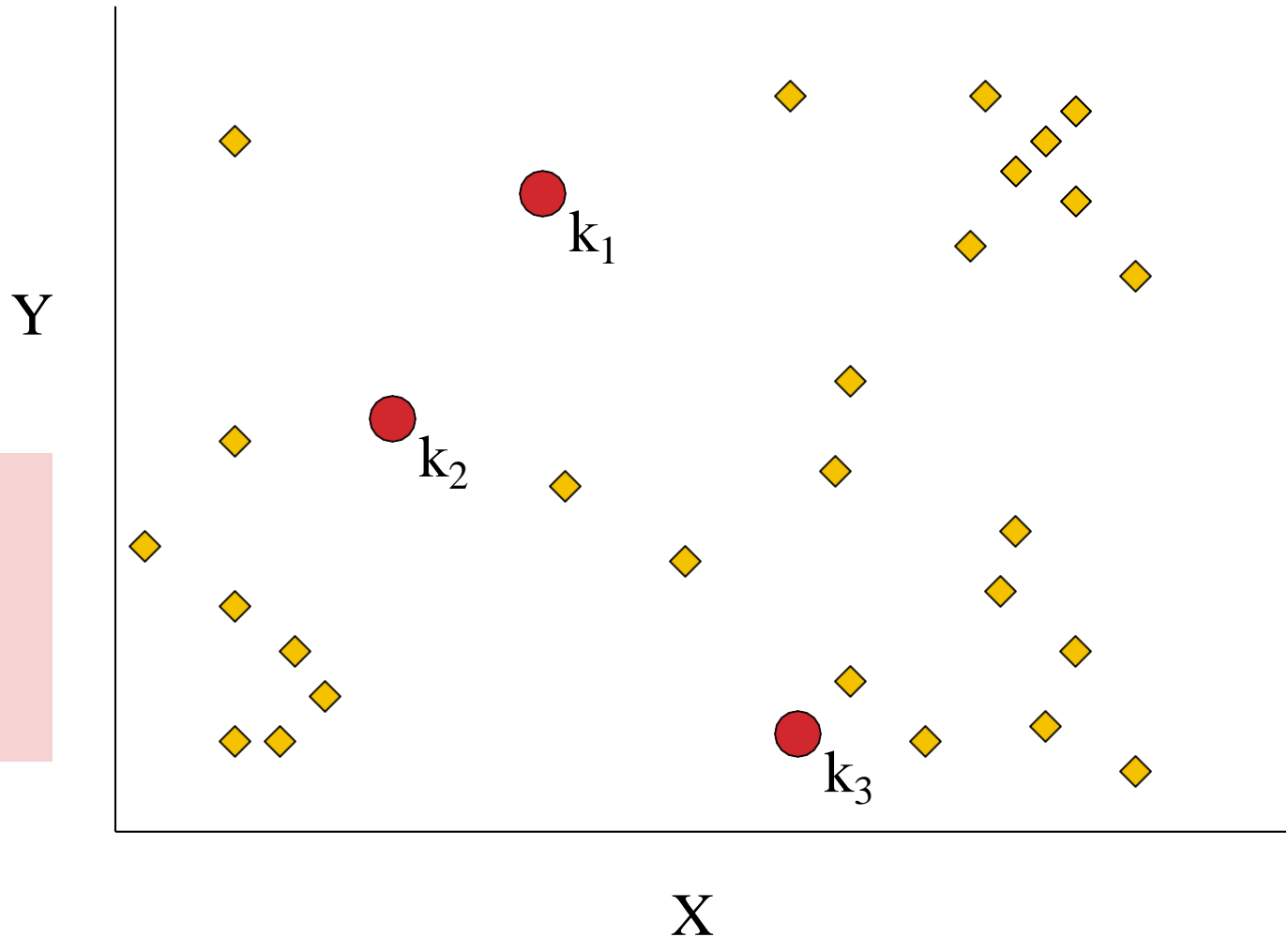
MÉTHODE DES CENTRES MOBILES

Présentation K-MEANS

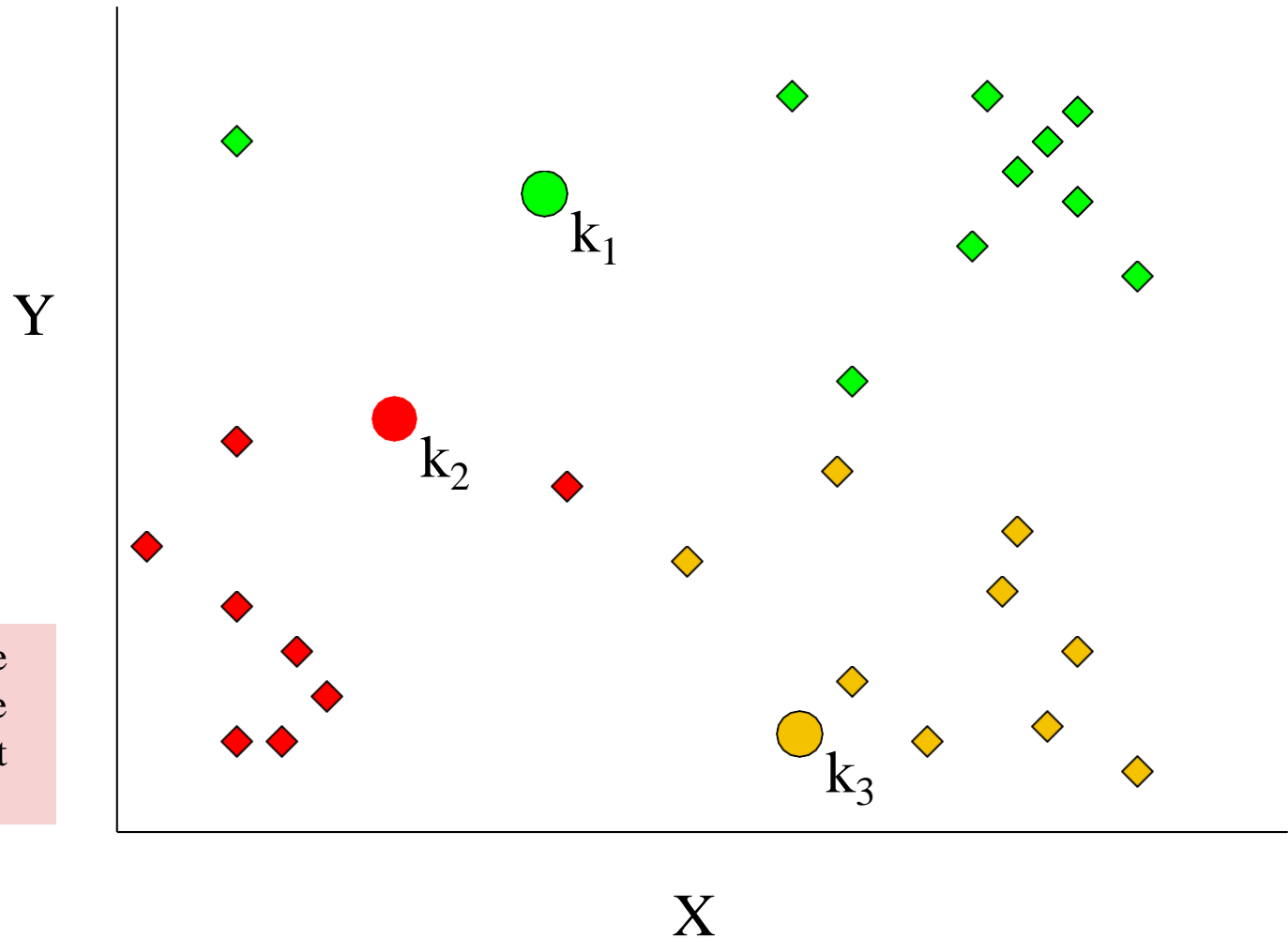
- L'algorithme des K-moyennes est un algorithme qui permet de trouver des classes dans des données.
- les classes qu'il construit n'entretiennent jamais de relations hiérarchiques: une classe n'est jamais incluse dans une autre classe
- L'algorithme fonctionne en précisant le nombre de classes attendues.
- L'algorithme calcule les distances Intra-Classe et Inter-Classe.
- Il travaille sur des **variables continues**.

Simulation du K-MEANS

Choisir **3**
Centres de
clusters
(au hasard)



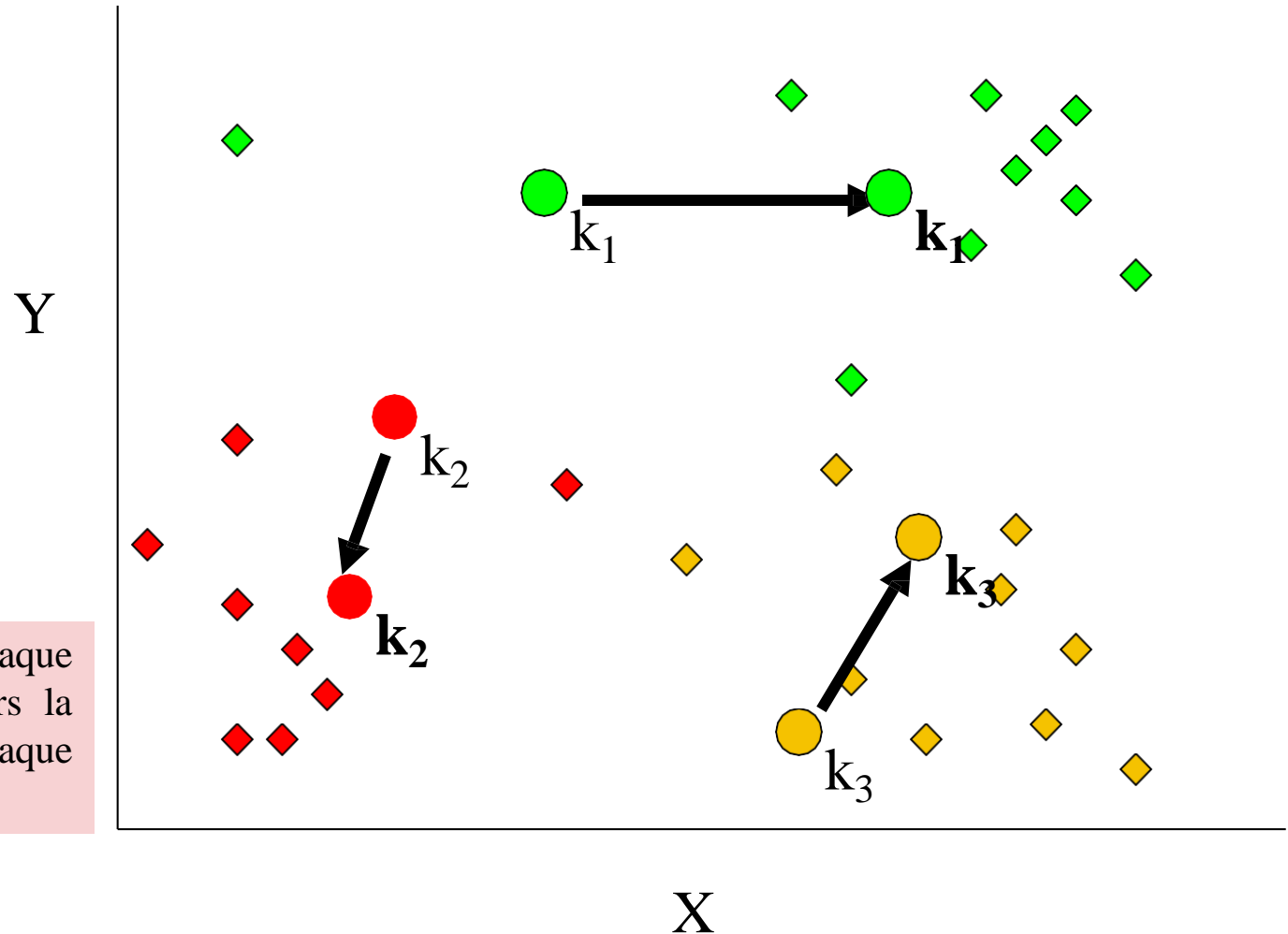
Simulation du K-MEANS



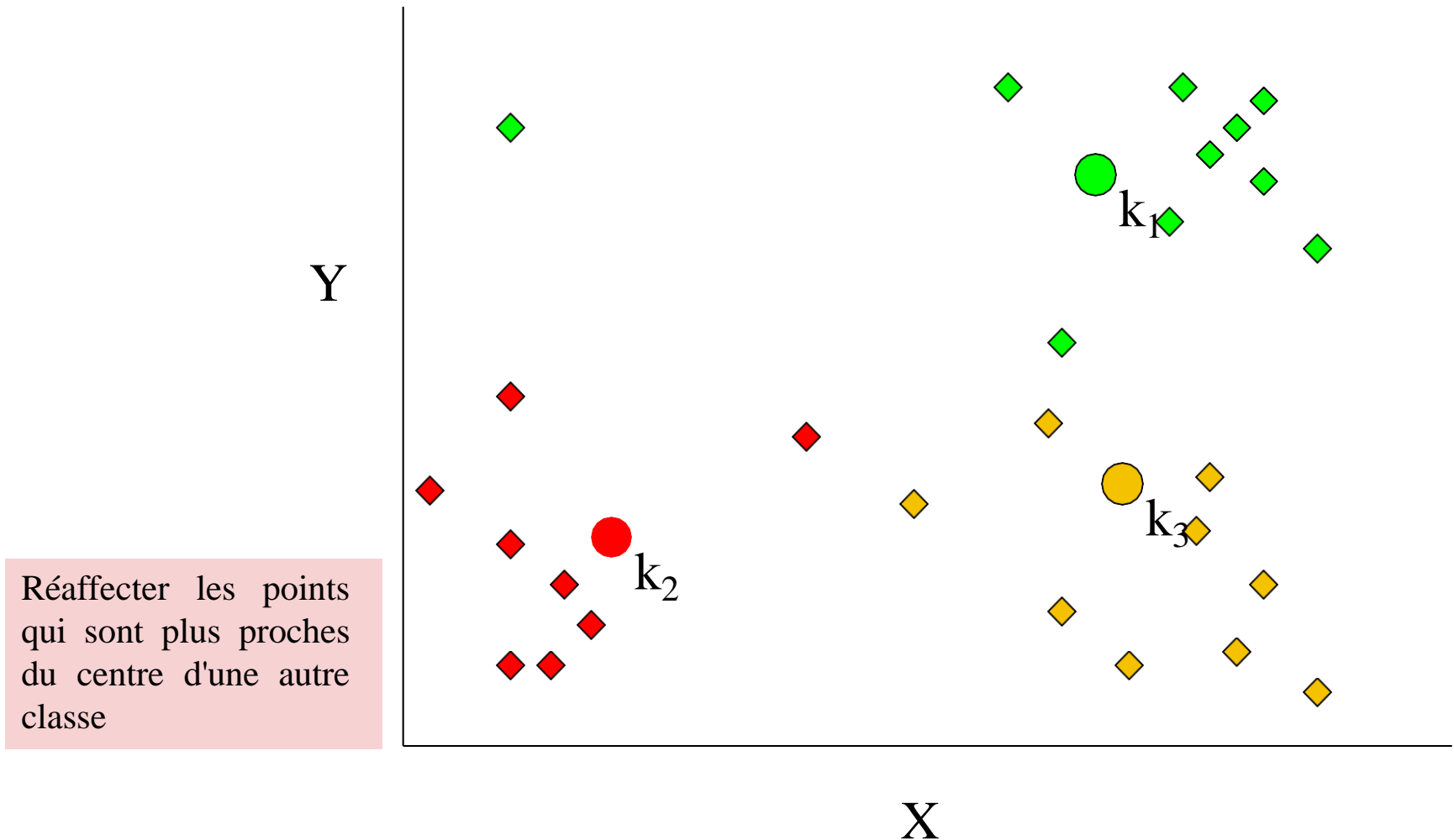
Affecter chaque point à la classe dont le centre est le plus proche

Simulation du K-MEANS

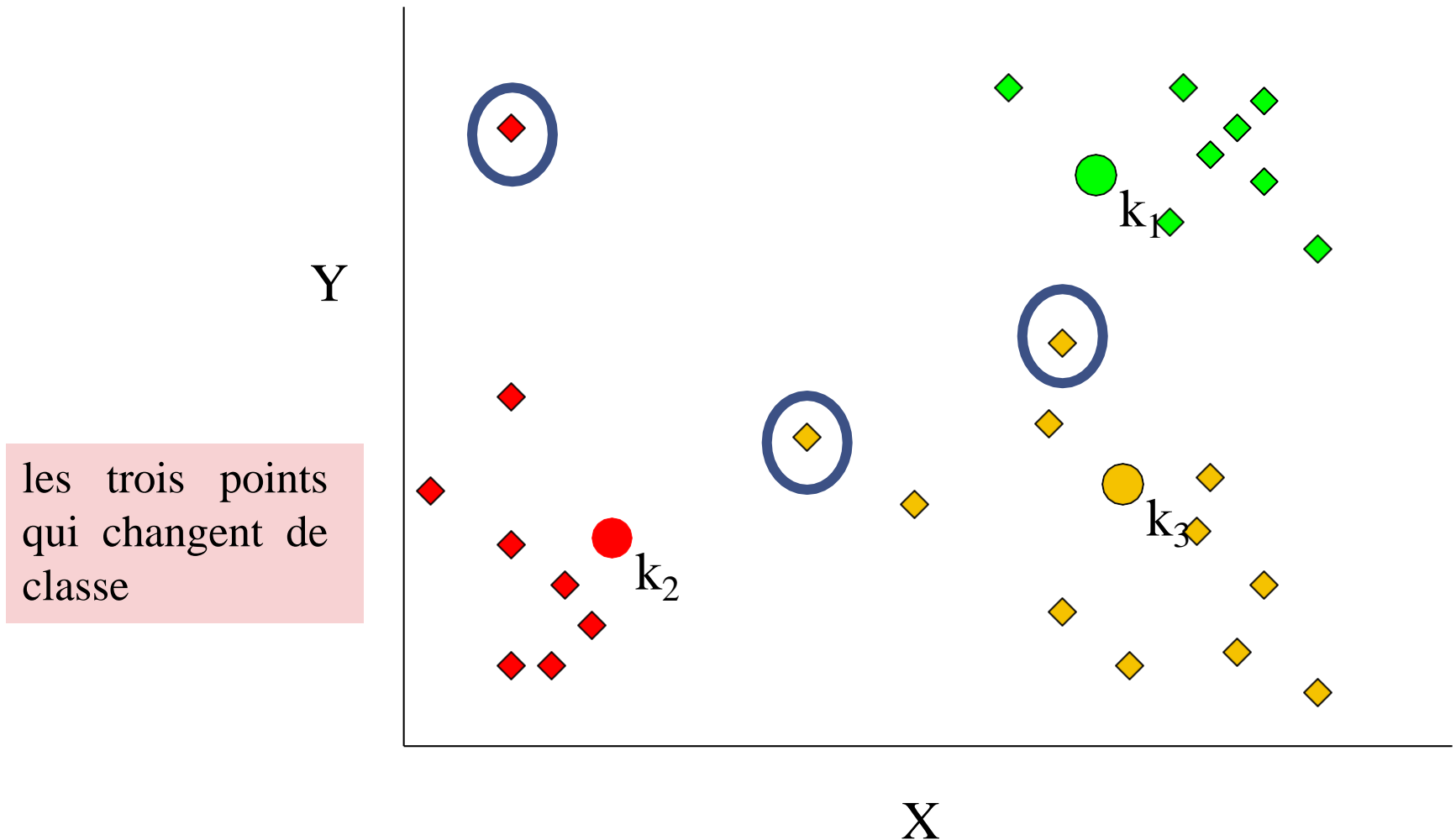
Déplacer chaque centre de classe vers la moyenne de chaque classe



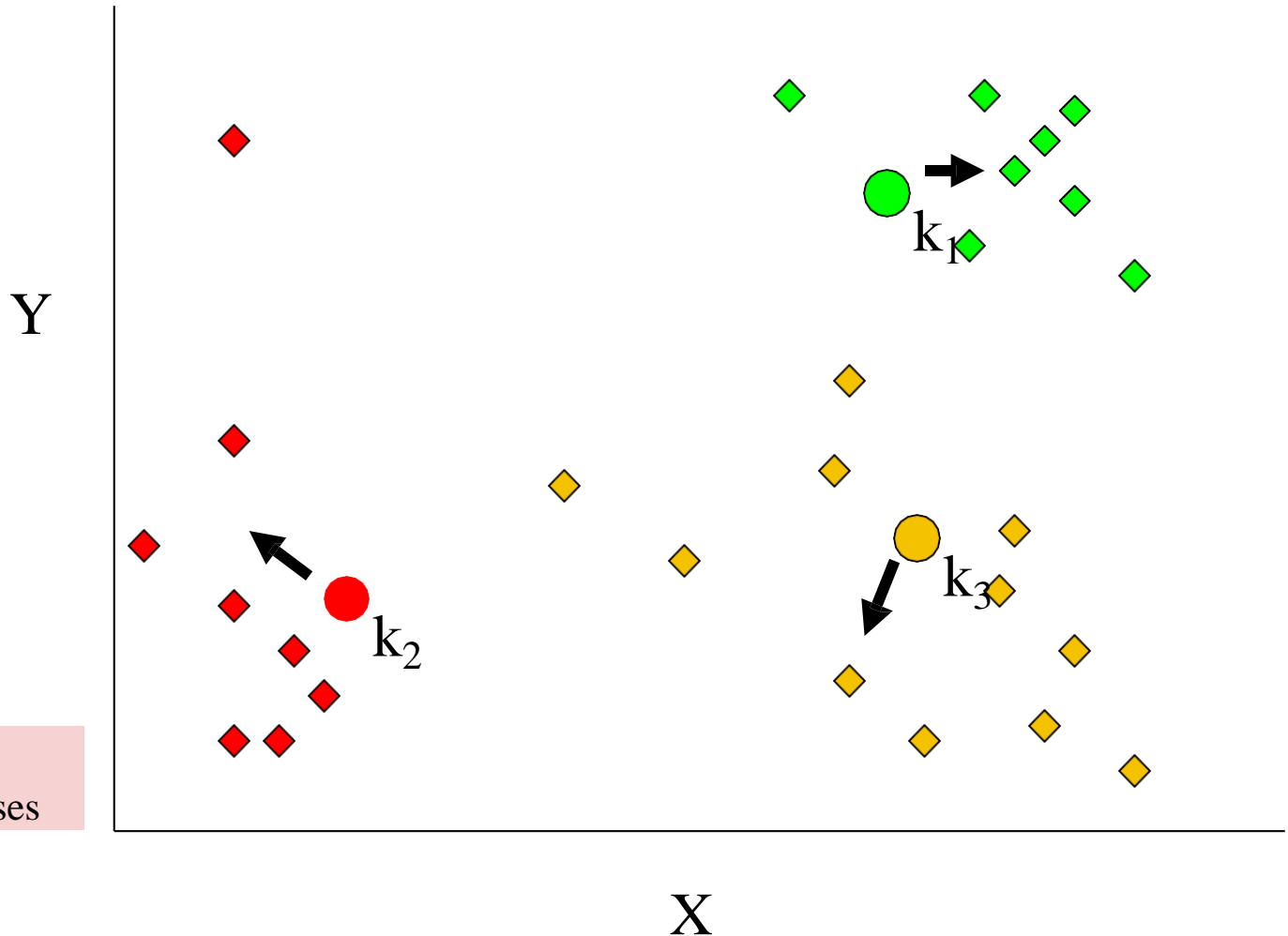
Simulation du K-MEANS



Simulation du K-MEANS

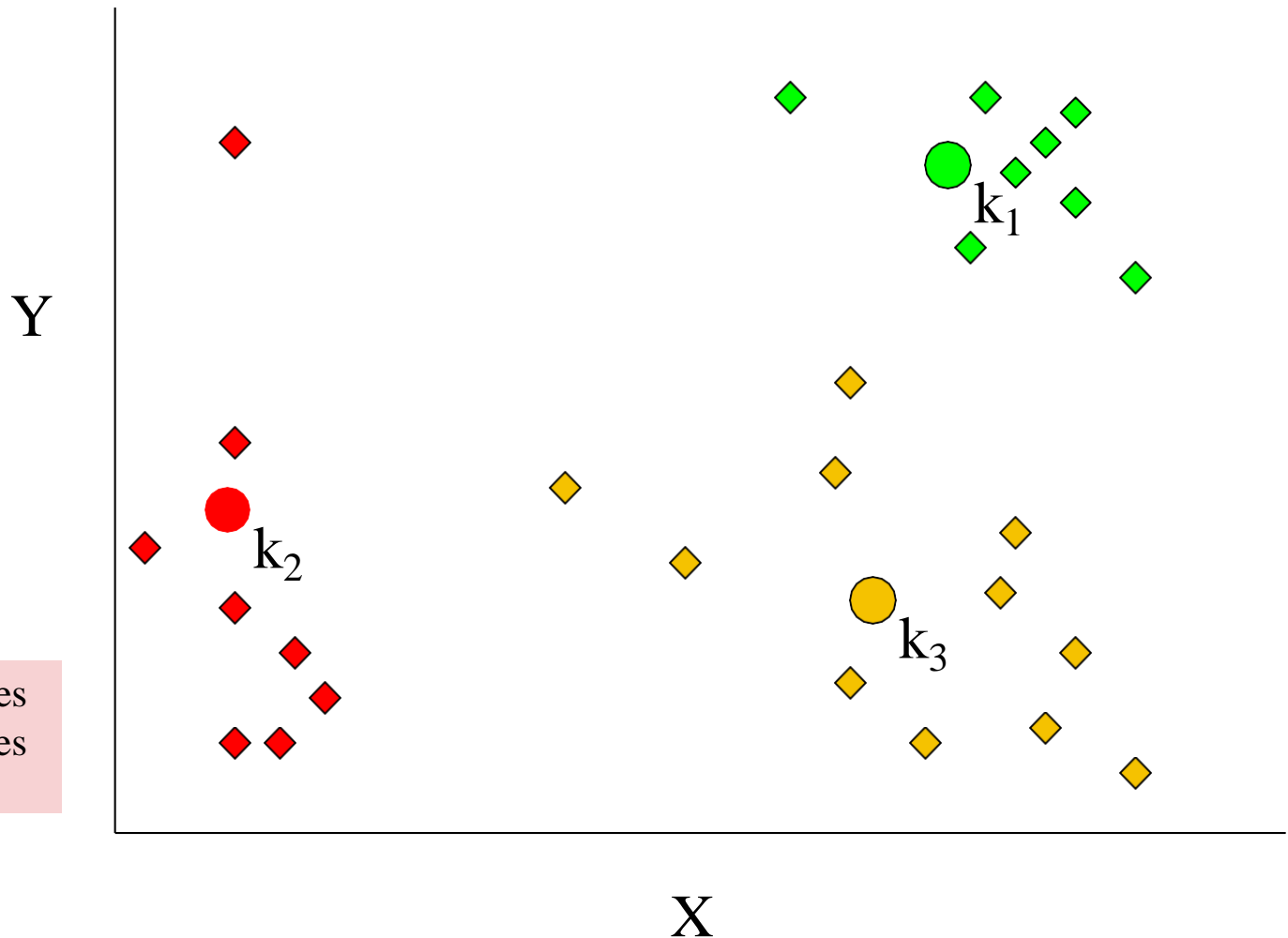


Simulation du K-MEANS



Re-calculer les
moyennes des classes

Simulation du K-MEANS



Déplacer les
centres des classes
vers les moyennes

Points faibles du K-MEANS

- Le choix du nombre de groupes est subjectif dans le cas où le nombre de classes est inconnu au sein de l'échantillon.
- L'algorithme du K-Means ne trouve pas nécessairement la configuration la plus optimale.
- Les résultats de l'algorithme du K-Means sont sensibles à l'initialisation aléatoires des centres.

CAH

CLASSIFICATION ASCENDANTE HIÉRARCHIQUE

Principe

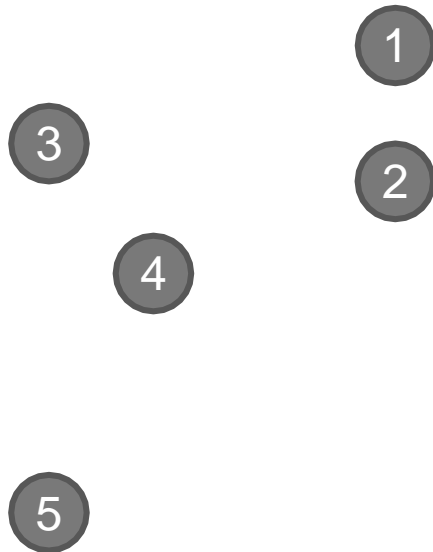
1. Créer à chaque étape une partition obtenue en agrégeant 2 à 2 les éléments les plus proches!
2. Eléments: individus ou groupe d'individus.
3. L'algorithme fournit une hiérarchie de partitions: arbre contenant l'historique de la classification et permettant de retrouver **n-1 partitions**.
4. Nécessité de se munir d'une métrique (distance euclidienne, chi2, Ward...).
5. Nécessité de fixer une règle pour agréger un individu et un groupe d'individus (ou bien 2 groupes d'individus).

Le Dendrogramme

- Durant les étapes d'un algorithmes de classification hiérarchique, on est en train de construire un dendrogramme.
- Le dendrogramme indique les objets et classes qui ont été fusionnées à chaque itération.
- Le dendrogramme indique aussi la valeur du critère choisi pour chaque partition rencontrée.
- Il donne un résumé de la classification hiérarchique
- Chaque palier correspond à une fusion de classes
- Le niveau d'un palier donne une indication sur la qualité de la fusion correspondante.
- Toute coupure horizontale correspond à une partition.

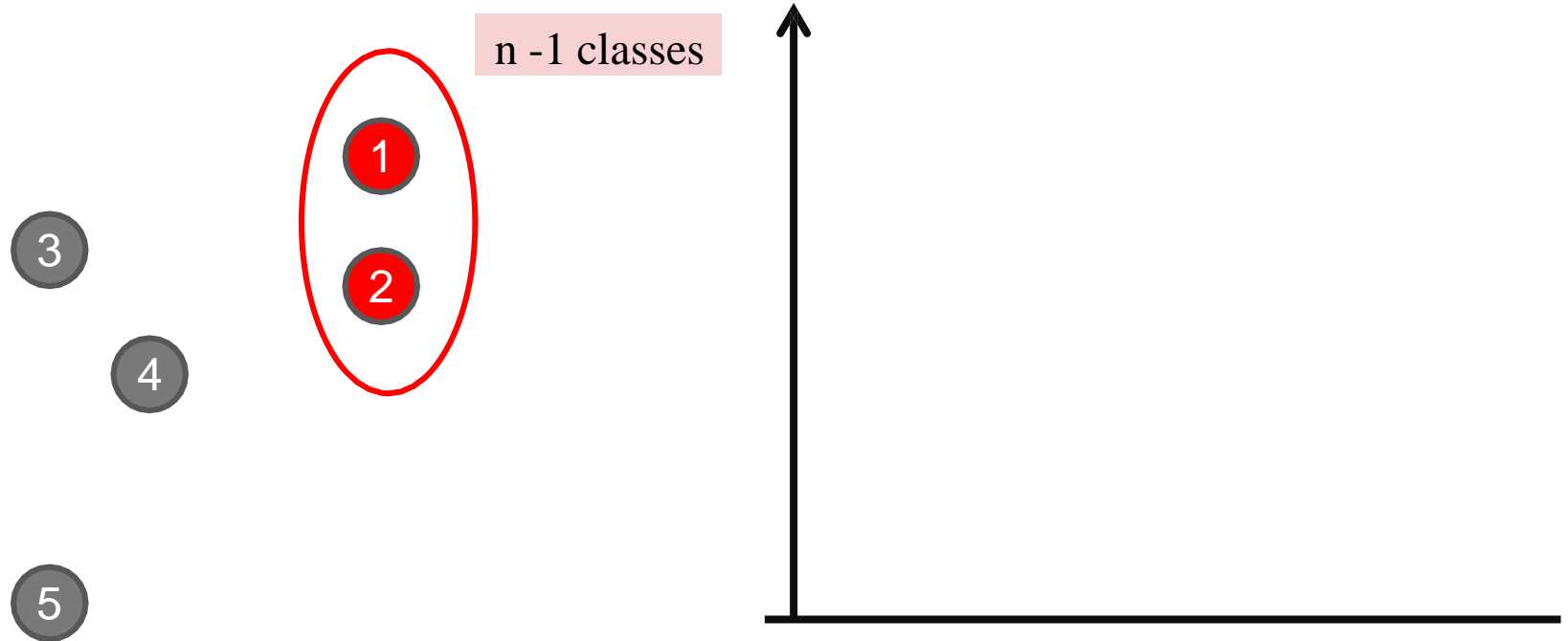
Simulation du CAH

n individus / n classes

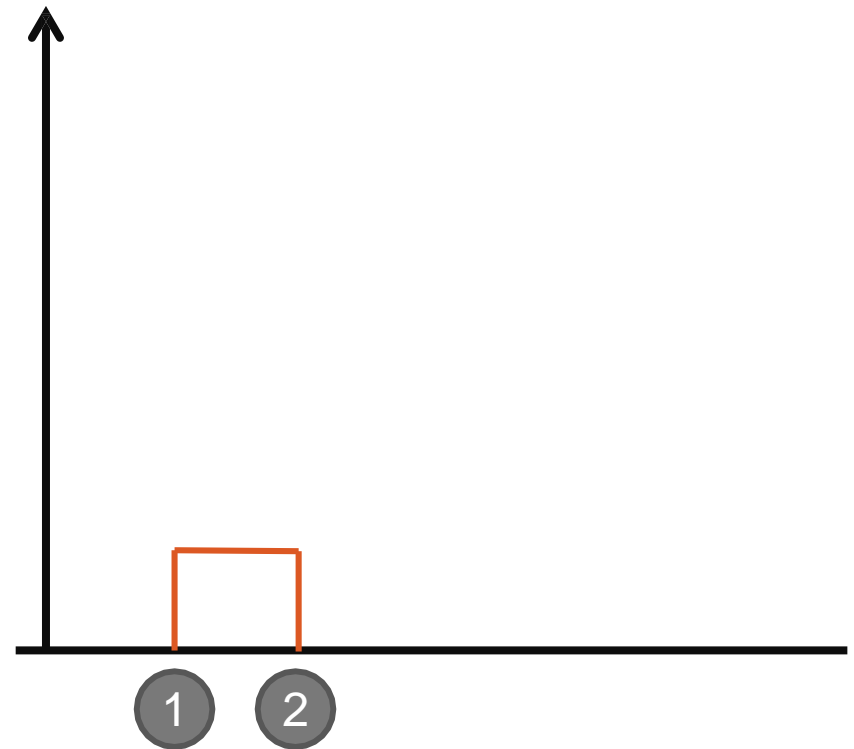
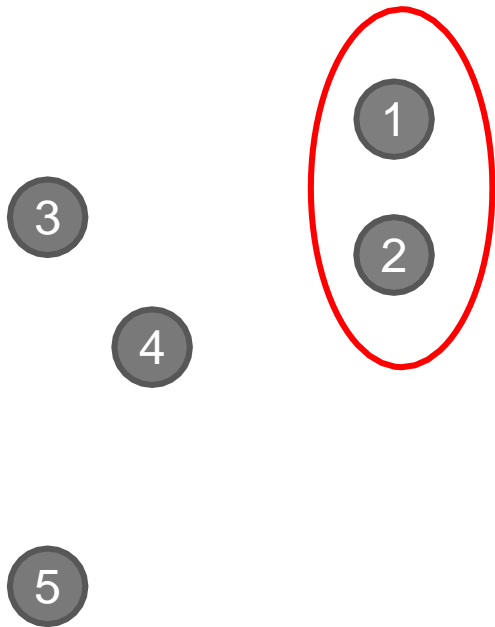


On construit la matrice de distance entre les n éléments et on regroupe les 2 éléments les plus proches

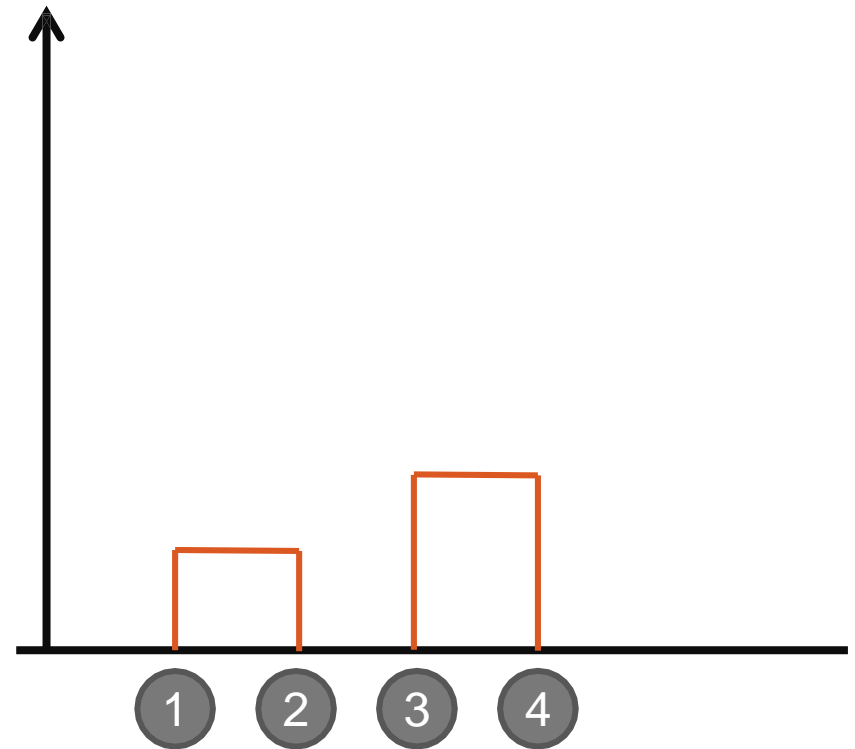
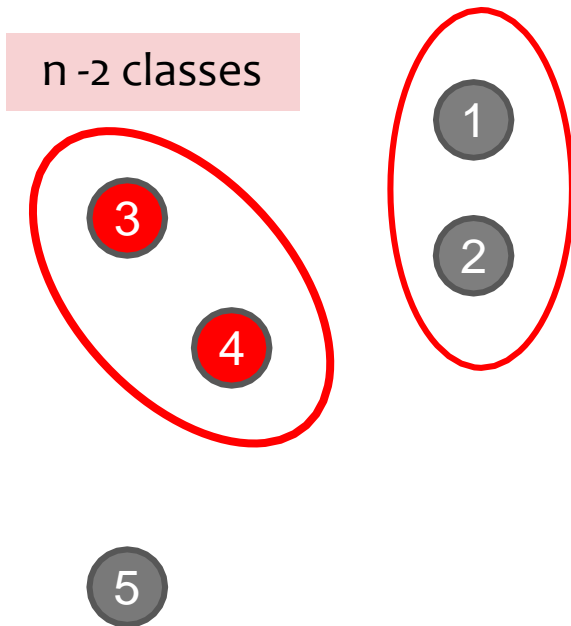
Simulation du CAH



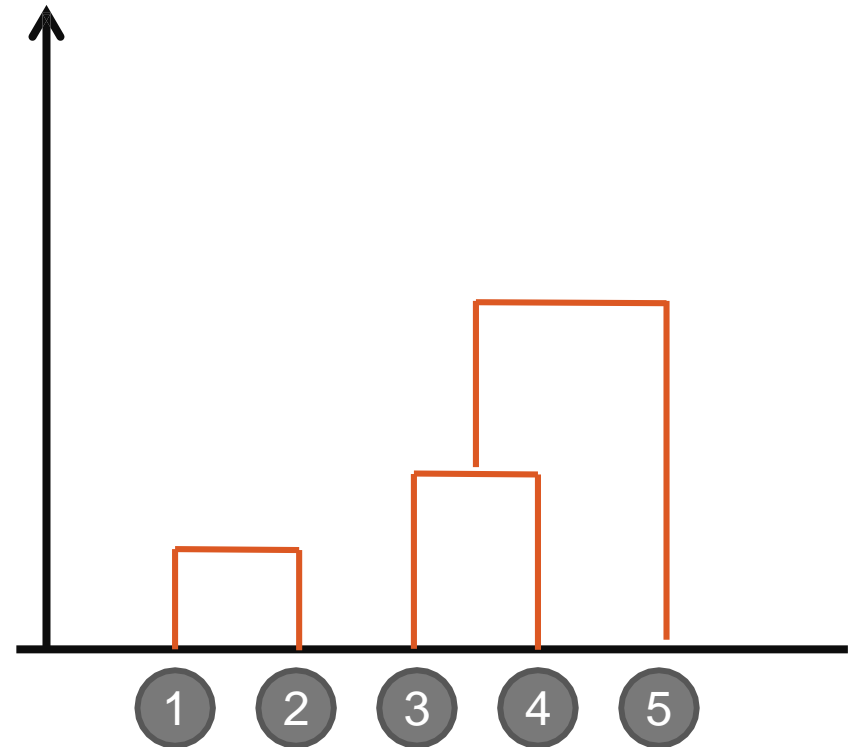
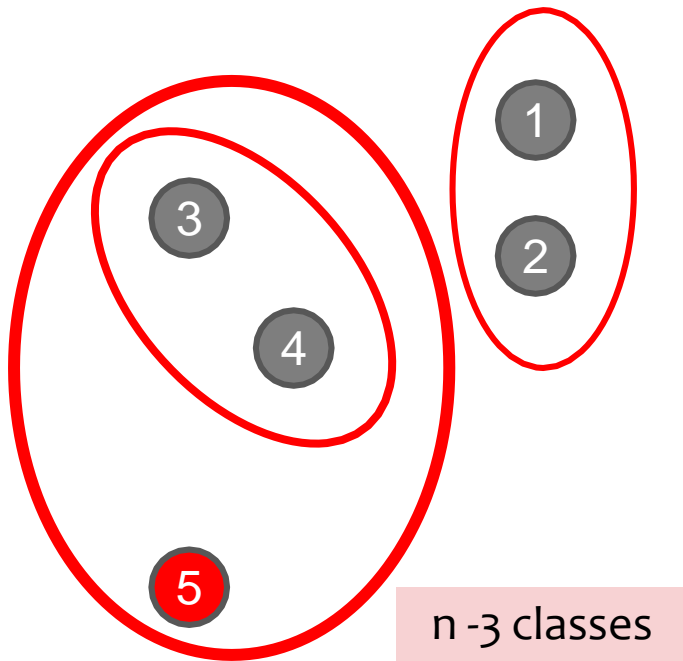
Simulation du CAH



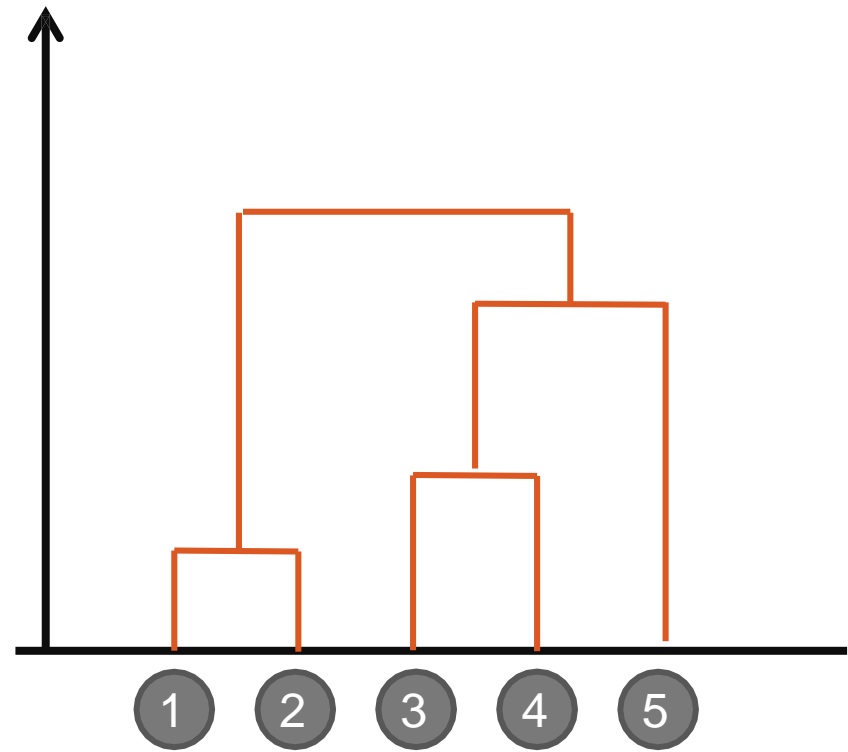
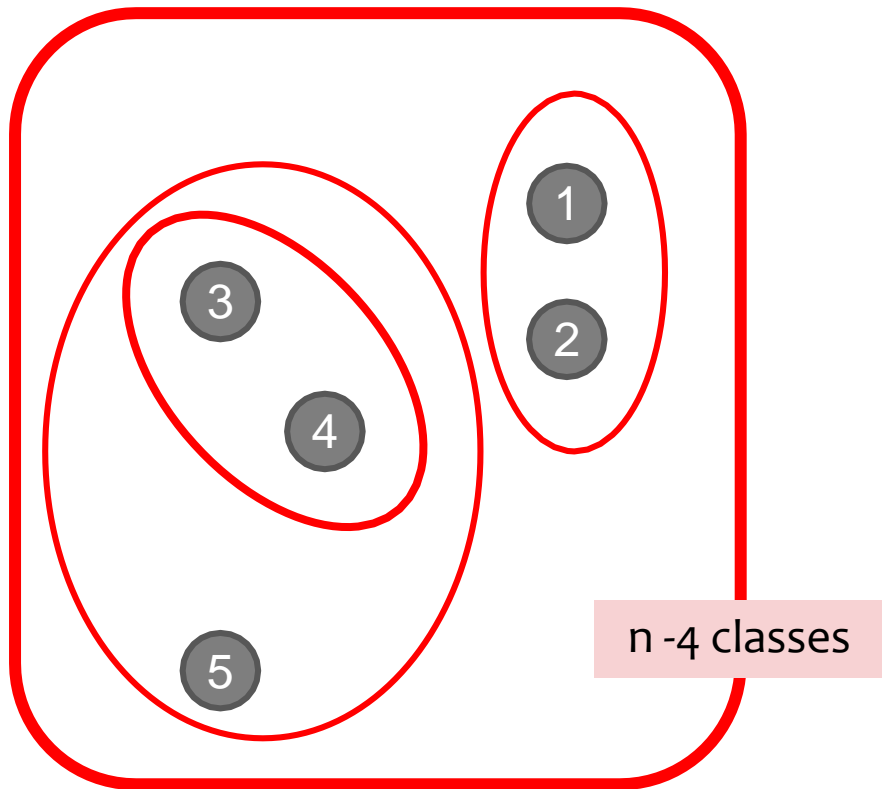
Simulation du CAH



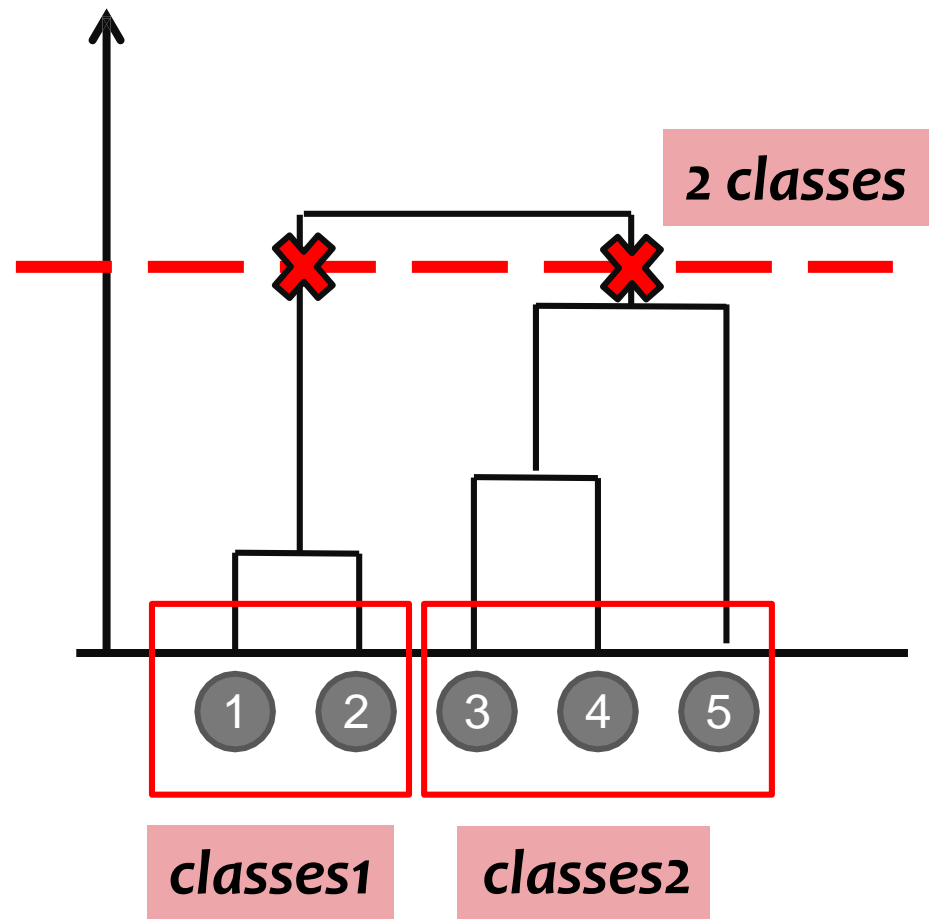
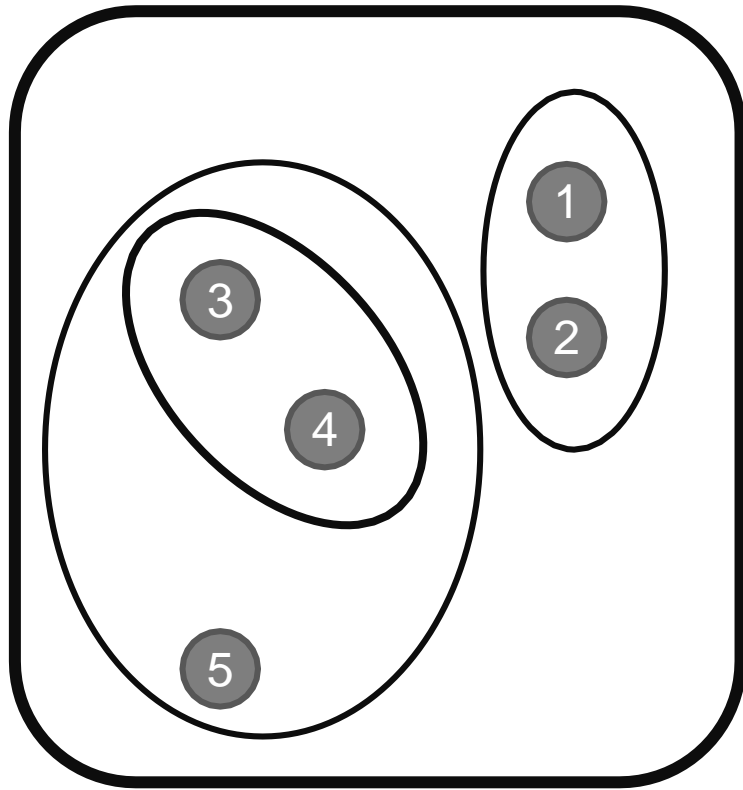
Simulation du CAH



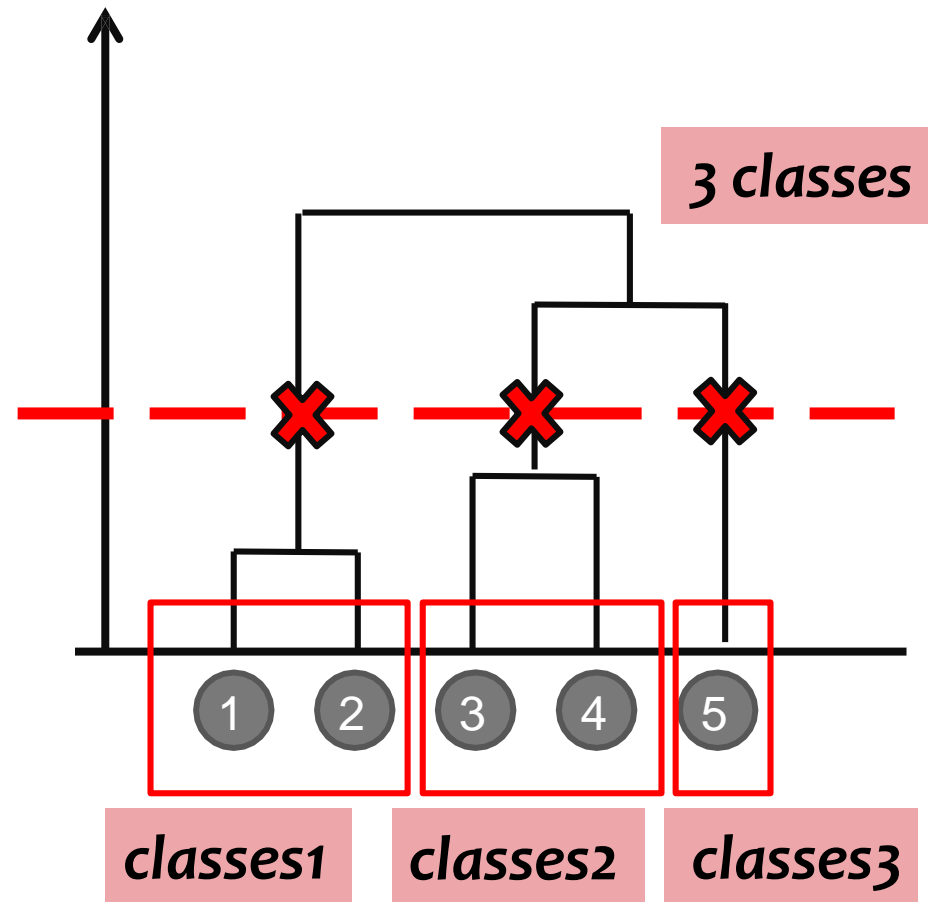
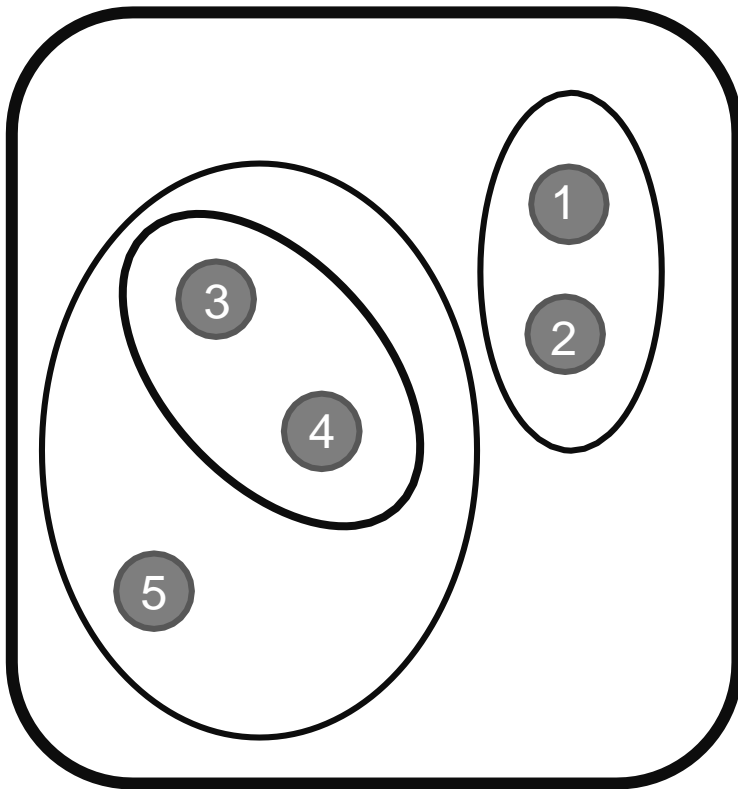
Simulation du CAH



Simulation du CAH



Simulation du CAH



Avantages du CAH

- Permet de classer: des individus, des variables, des moyennes de classes obtenues en sortie d'un algorithme des centres mobiles.
- Si on classe des moyennes, on améliore les résultats si on connaît non seulement les moyennes des classes, mais aussi les inerties intra classe et les effectifs des classes.
- S'adapte aux diverses formes de classes, par le choix de la distance.
- Permet de choisir le nombre de classes de façon optimale, grâce à des indicateurs de qualité de la classification en fonction du nombre de classes.

Points faibles du CAH

- Complexité algorithmique non linéaire (en n^2 ou n^3 , parfois $n^2 \log(n)$)
- Deux observations placées dans des classes différentes ne sont jamais plus comparées

Utilité de la segmentation

- 1 Segmentation de la clientèle.
- 2 Segmentation des images.
- Détection d'une anomalie.
- Organisation des documents.
- 1 Analyse du panier de marché.
- Analyse des réseaux sociaux.

Utilité de la segmentation

- Planification urbaine: les urbanistes utilisent le clustering pour analyser les données géographiques, comprendre la répartition de la population et gérer efficacement les ressources.
- Systèmes de recommandation: les algorithmes de clustering regroupent les utilisateurs ou les produits en fonction de leur similarité, permettant de meilleures recommandations pour les films, la musique et d'autres produits.
- Génomique: en bio-informatique, les algorithmes de clustering regroupent les gènes ou les protéines en fonction de modèles d'expression similaires.

Plan du cours

- 1 Les familles de techniques
- 2 Règles d'association
- 3 Segmentation
- 4 Classification & régression
- 5 Estimation de performances & Validation Méthodes



Classification & Régression

- Apprentissage supervisé classification/ Régression
 - Extrapoler des nouvelles informations à partir de données existantes.
 - Prédire la classe de nouvelles données observées).

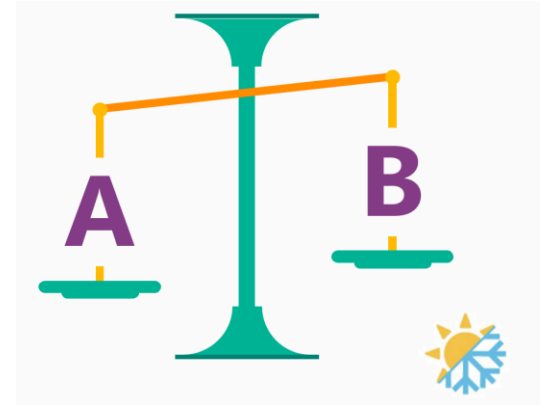
Client	Salaire	S. Familiale	Ville	Rembourse son crédit
1	Moyen	Divorcé	Tunis	? oui
2	Elevé	Célibataire	Tunis	? non
3	Faible	Célibataire	Sfax	? non
⋮	⋮	⋮	⋮	? ⋮

← Classe

Classification & Régression

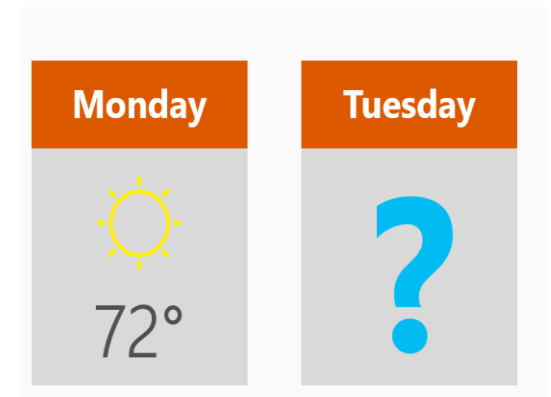
💡 Classification : Prédire des valeurs discrets

- Sera-t-il froid ou chaud demain? **Froid (A)/ chaud (B)**



💡 Régression : Prédire des valeurs continues

- Quelle est la température demain?



- Processus à deux phases :
 - **Apprentissage (off-line):** construire un modèle K (ou classifieur) qui décrit un ensemble prédéterminé de classes de données.
NB. La construction du modèle doit être basé sur des données annotés (étiquetés).
 - **Test (on-line):** utiliser le modèle obtenu K pour affecter une classe à un nouvelle observation.

Classification & Régression

Jeu de données
d'apprentissage

Classe désirée

	outlook	temperature	humidity	windy	play
0	sunny	hot	high	False	no
1	sunny	hot	high	True	no
2	overcast	hot	high	False	yes
3	rainy	mild	high	False	yes
4	rainy	cool	normal	False	yes
5	rainy	cool	normal	True	no
6	overcast	cool	normal	True	yes
7	sunny	mild	high	False	no
8	sunny	cool	normal	False	yes
9	rainy	mild	normal	False	yes

Apprentissage
supervisée

Est-ce-que le
modèle obtenu
est performant?

Modèle K

Nouvelle observation

Classe à prédire

outlook	temperature	humidity	windy	play
sunny	mild	normal	True	?



Estimer la performance du modèle : Validation



Spécifier des données de validation : Echantillonnage

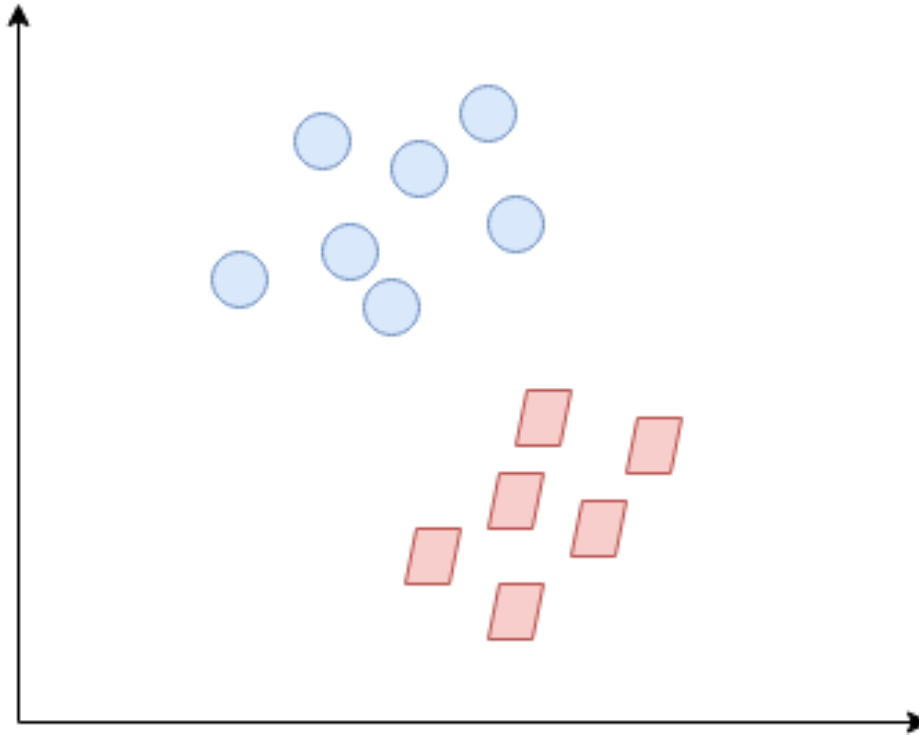
SVM

SUPPORT VECTOR MACHINE

Principe

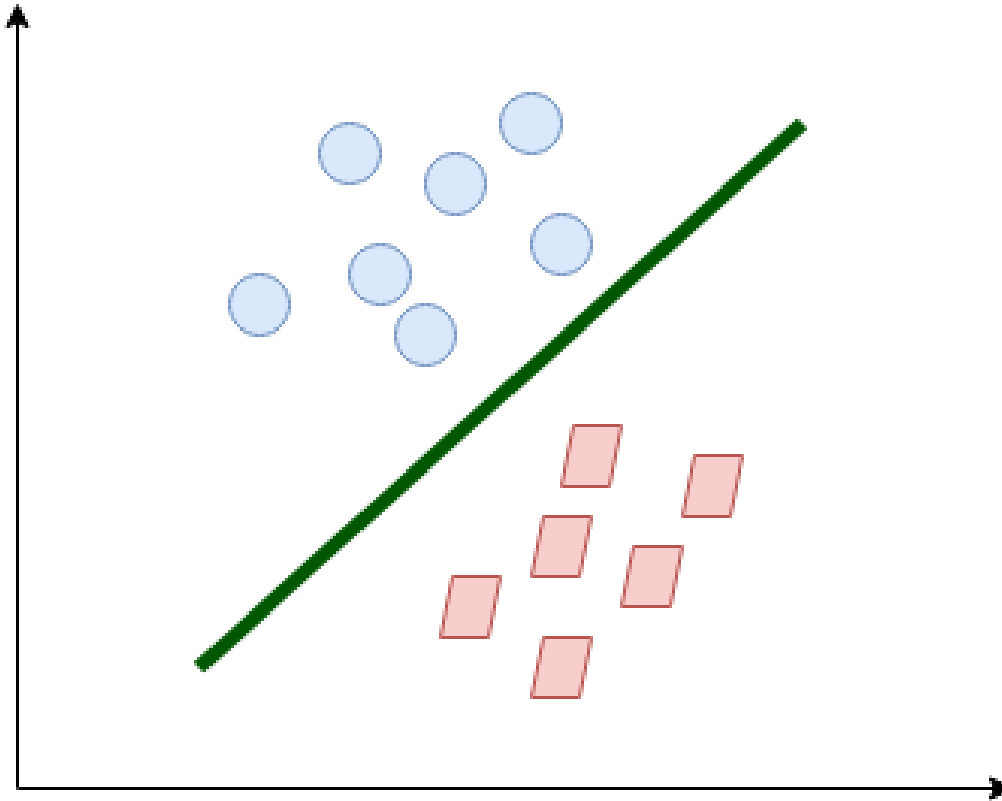
1. Support Vector Machine (SVM) est un algorithme d'apprentissage supervisé utilisé pour les tâches de classification et de régression.
2. SVM fonctionne en trouvant un **hyperplan** dans un espace de haute dimension qui sépare au mieux les données en différentes classes.
3. Elle vise à maximiser la marge (la distance entre l'hyperplan et les points de données les plus proches de chaque classe) tout en minimisant les erreurs de classification.
4. SVM peut gérer à la fois les problèmes de classification linéaire et non linéaire en utilisant diverses **Kernel fonctions**.
5. Elle est largement utilisée dans des tâches telles que la classification d'images, la catégorisation de texte, et plus encore.

Simulation du SVM



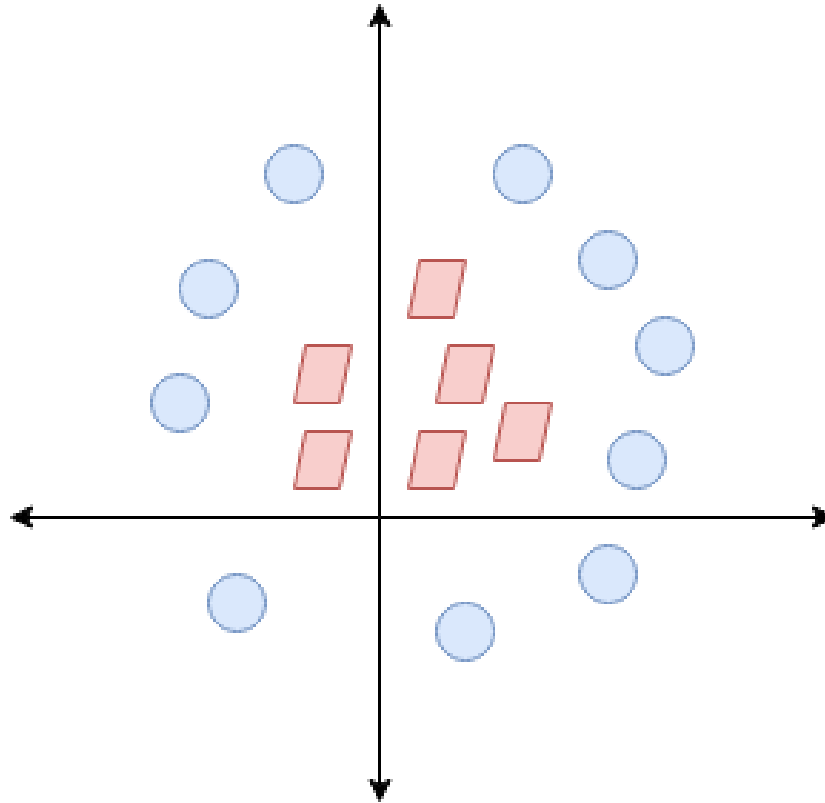
Disons que nous avons un graphique avec deux classes d'étiquettes comme indiqué dans la figure

Simulation du SVM



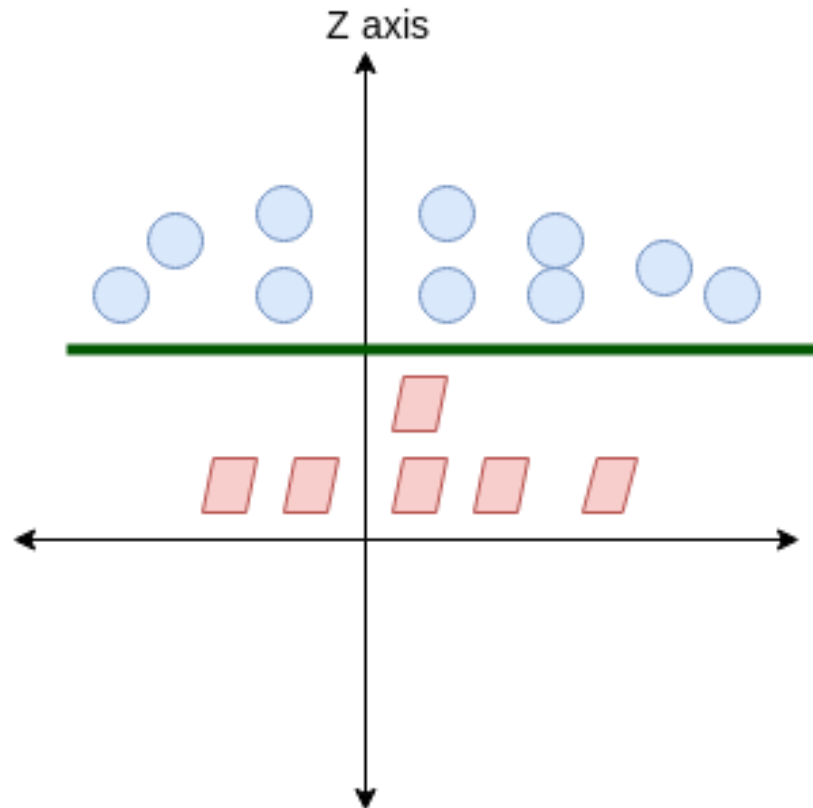
La ligne sépare équitablement les classes. C'est essentiellement ce que fait la SVM – une simple séparation des classes.

Simulation du SVM



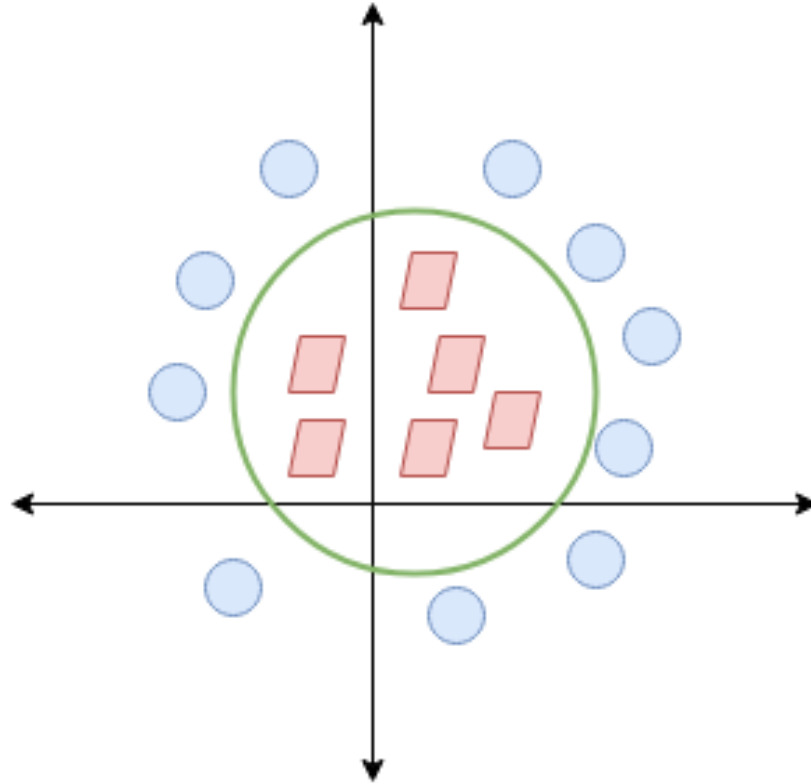
Ici, nous n'avons pas une simple ligne séparant ces deux classes.

Simulation du SVM



Nous allons donc étendre notre dimension et introduire une nouvelle dimension le long de l'axe z. Nous pouvons maintenant séparer ces deux classes

Simulation du SVM



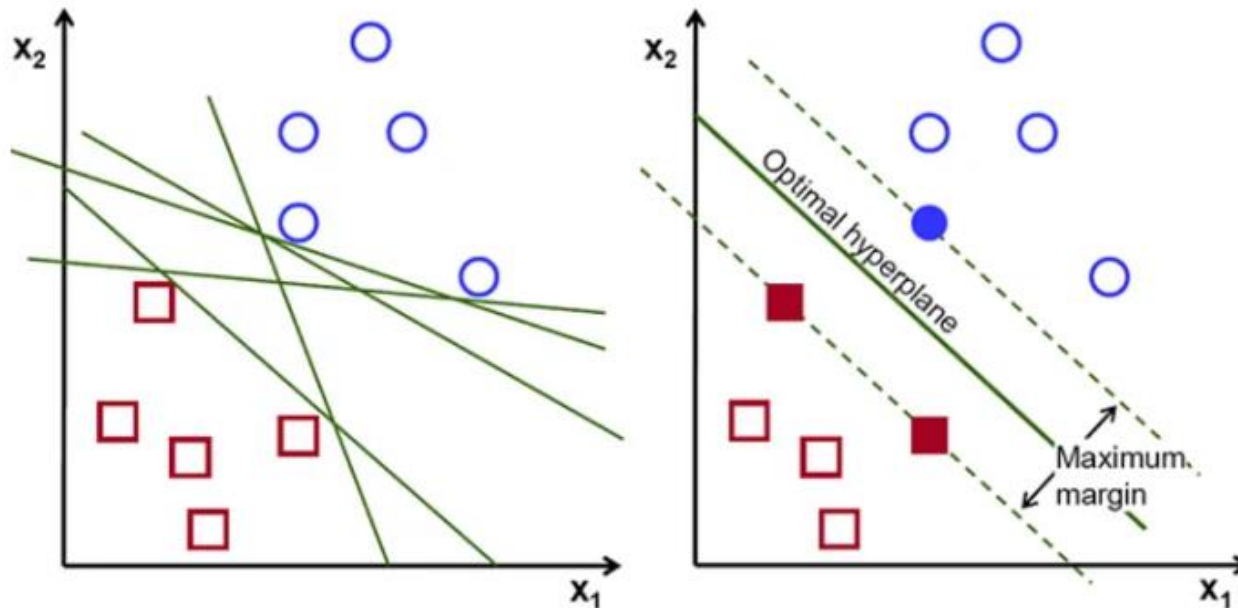
Lorsque nous transformons cette ligne dans le plan d'origine, elle se mappe à la frontière circulaire

Simulation du SVM

- C'est exactement ce que fait la SVM !

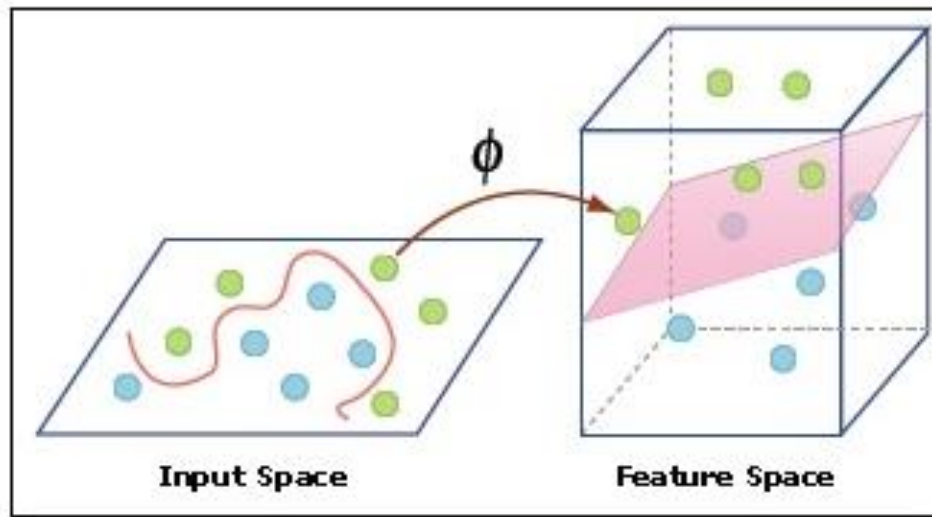
Elle essaie de trouver une ligne/un hyperplan (dans un espace multidimensionnel) qui sépare ces deux classes.

Ensuite, elle classe le nouveau point en fonction de son emplacement par rapport à l'hyperplan, c'est-à-dire s'il se trouve du côté positif ou négatif de l'hyperplan en fonction des classes à prédire.



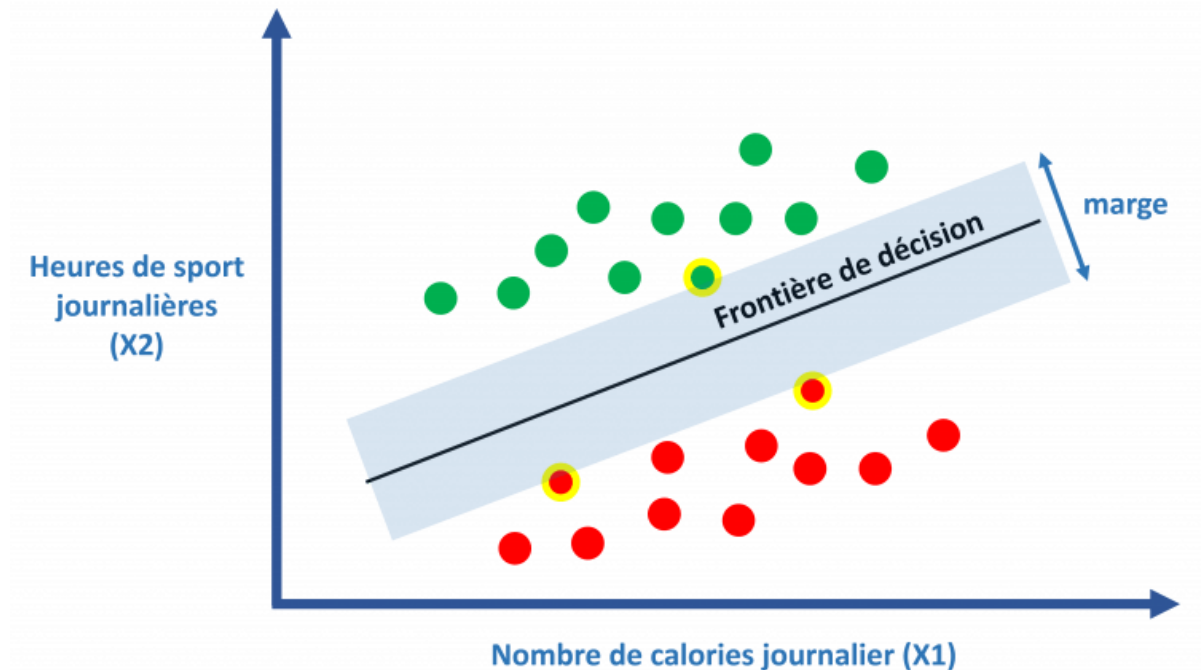
Hyperparameters

- **Noyau (Kernel):** La fonction utilisée pour mapper des données de dimension inférieure vers des données de dimension supérieure.
- Le coût de calcul augmente lorsque la dimension des données augmente. Cette augmentation de dimension est nécessaire lorsque nous ne pouvons pas trouver un hyperplan séparateur dans une dimension donnée et devons passer à une dimension supérieure.



Hyperparameters

- **Frontière de décision (Hyperplan):** Une frontière de décision peut être considérée comme une ligne de démarcation (pour simplifier) d'un côté de laquelle se trouvent les exemples positifs et de l'autre côté les exemples négatifs. Sur cette même ligne, les exemples peuvent être classés comme positifs ou négatifs. Ce même concept de la SVM sera appliqué également dans la régression à vecteurs de support.



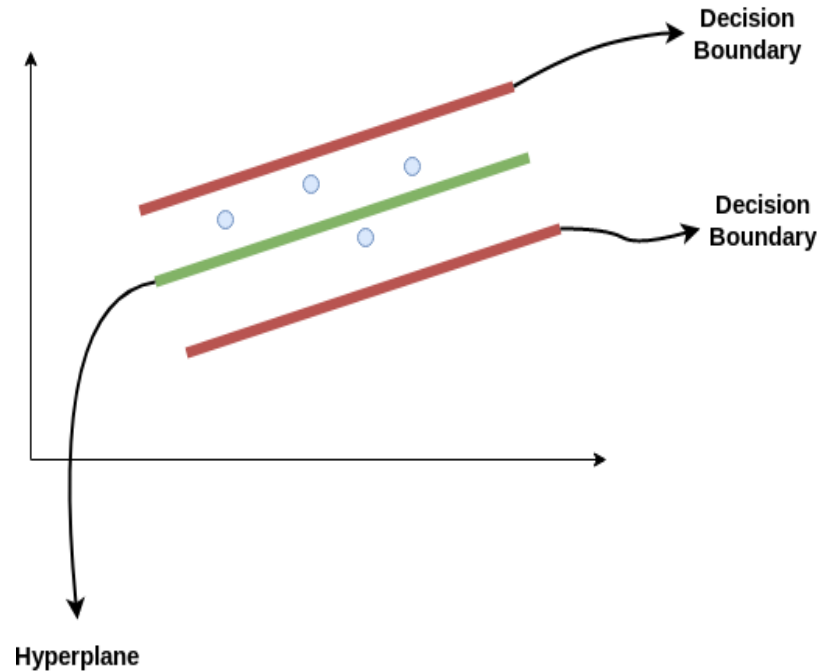
Utilisation conseillé

- Pour les données qui ont exactement deux classes.
- Pour les données à grande dimension et non linéairement séparables.
- Lorsque vous avez besoin d'un classificateur simple, facile à interpréter et précis.

Support Vector Regression

- Support Vector Regression (SVR) est un type d'algorithme d'apprentissage automatique utilisé pour l'analyse de régression.
- Contrairement à SVM utilisées pour les tâches de classification, la SVR cherche à trouver un hyperplan qui correspond le mieux aux points de données dans un espace continu. Cela est réalisé en mappant les variables d'entrée dans un espace de caractéristiques de haute dimension et en trouvant l'hyperplan qui maximise la marge (distance) entre l'hyperplan et les points de données les plus proches, tout en minimisant l'erreur de prédiction.

Simulation du SVR



Ce que nous essayons de faire ici, c'est essentiellement de déterminer une frontière de décision à une distance ' ϵ ' de l'hyperplan original de manière à ce que les points de données les plus proches de l'hyperplan, ou les vecteurs de support, se trouvent à l'intérieur de cette ligne de frontière.

- Pour les données de grande dimension (où il y aura un grand nombre de variables de prédiction)

Classification & Régression

Logistic Regression

Fonctionnement

La régression logistique est couramment utilisée comme point de départ pour les problèmes de classification binaire.

Utilisation conseillé

- Lorsque les données peuvent être clairement séparées par une seule frontière linéaire
- Line de base pour évaluer des méthodes de classification plus complexes

Résultat



k Nearest Neighbor (kNN)

Fonctionnement

KNN classe les objets en fonction des classes de leurs voisins les plus proches dans l'ensemble de données. KNN prédit que les objets proches les uns des autres sont similaires.

Utilisation conseillé

- Lorsque vous avez besoin d'un algorithme simple pour établir des règles d'apprentissage de référence
- Lorsque l'utilisation de la mémoire du modèle formé est une préoccupation moindre
- Lorsque la vitesse de prédiction du modèle formé est une préoccupation moindre

Résultat



Classification & Régression

Decision Tree

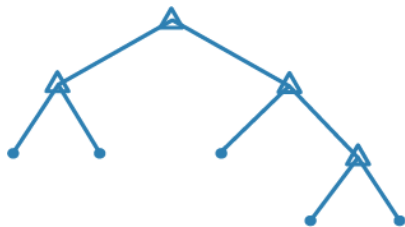
Fonctionnement

Une arborescence de décision vous permet de prédire les réponses aux données en suivant les décisions dans l'arborescence, depuis la racine (début) jusqu'à la feuille.

Utilisation conseillé

- Lorsque vous avez besoin d'un algorithme facile à interpréter et à ajuster rapidement
- Pour minimiser l'utilisation de la mémoire
- Lorsque la précision prédictive élevée n'est pas une exigence

Résultat



Bagged and Boosted Decision Trees

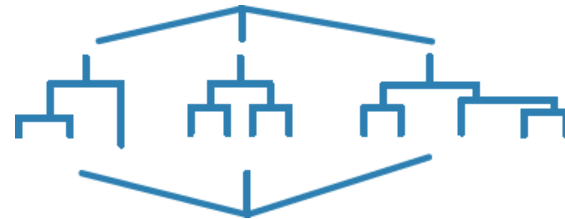
Fonctionnement

Dans ces méthodes d'ensemble, plusieurs arbres de décision «plus faibles» sont combinés dans un ensemble «plus fort».

Utilisation conseillé

- Lorsque les données d'entrées sont catégoriques (discrets) ou se comportent de façon non linéaire
- Combiner simple classificateurs en un autre plus complexe.
- Lorsqu'on veut une erreur très faible (no overfith)

Résultat



Classification & Régression

Neural Network

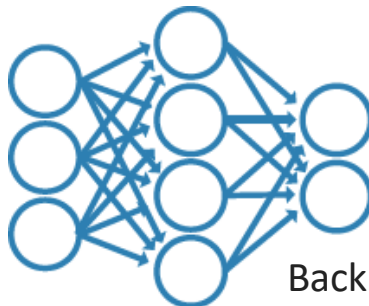
Fonctionnement

Inspiré par le cerveau humain, un réseau de neurones se compose de réseaux hautement connectés de neurones qui relient les entrées aux sorties désirées.

Utilisation conseillé

- Pour la modélisation de systèmes non linéaires
- Lorsque les données sont disponibles de façon incrémentielle et que vous souhaitez constamment mettre à jour le modèle
- Lorsqu'il peut y avoir des changements inattendus dans vos données d'entrée

Résultat



Naïve Bayes

Fonctionnement

Il classe les nouvelles données sur la base de la probabilité la plus élevée de son appartenance à une classe particulière.

Utilisation conseillé

- Pour un petit ensemble de données contenant de nombreux paramètres
- Lorsque le modèle rencontrera des scénarios qui ne figurent pas dans les données de formation, comme c'est le cas pour de nombreuses applications financières et médicales

Résultat



Classification & Régression

Linear Regression

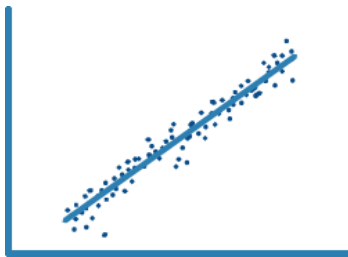
Fonctionnement

La régression linéaire est une technique de modélisation statistique utilisée pour décrire une variable de réponse continue comme une fonction linéaire d'une ou plusieurs variables.

Utilisation conseillé

- Lorsque vous avez besoin d'un algorithme facile à interpréter et à ajuster rapidement
- Comme base de référence pour l'évaluation d'autres modèles de régression plus complexes

Résultat



Nonlinear Regression

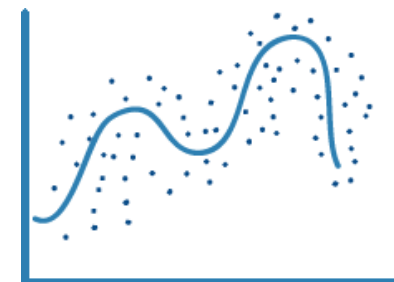
Fonctionnement

Aide à décrire les relations non linéaires dans les données expérimentales. Les modèles de régression non linéaire sont généralement considérés comme paramétriques, où le modèle est décrit comme une équation non linéaire.

Utilisation conseillé

- Lorsque les données ont des tendances non linéaires fortes et ne peuvent pas être facilement transformées en un espace linéaire
- Pour l'ajustement de modèles personnalisés aux données

Résultat



Classification & Régression

Generalized Linear Model

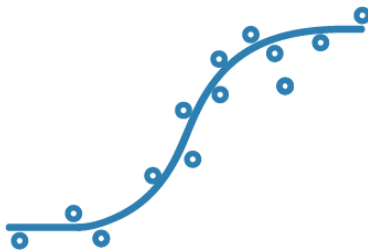
Fonctionnement

Un modèle linéaire généralisé est un cas particulier de modèles non linéaires utilisant des méthodes linéaires. Il consiste à ajuster une combinaison linéaire des entrées à une fonction non linéaire (la fonction de liaison) des sorties.

Utilisation conseillé

- Lorsque les variables de réponse ont des distributions non normales, comme une variable de réponse qui est toujours censée être positive

Résultat



Regression Tree

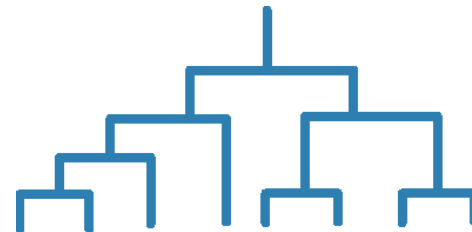
Fonctionnement

Les arbres de décision pour la régression sont semblables aux arbres de décision pour la classification, mais ils sont modifiés pour pouvoir prédire des réponses continues.

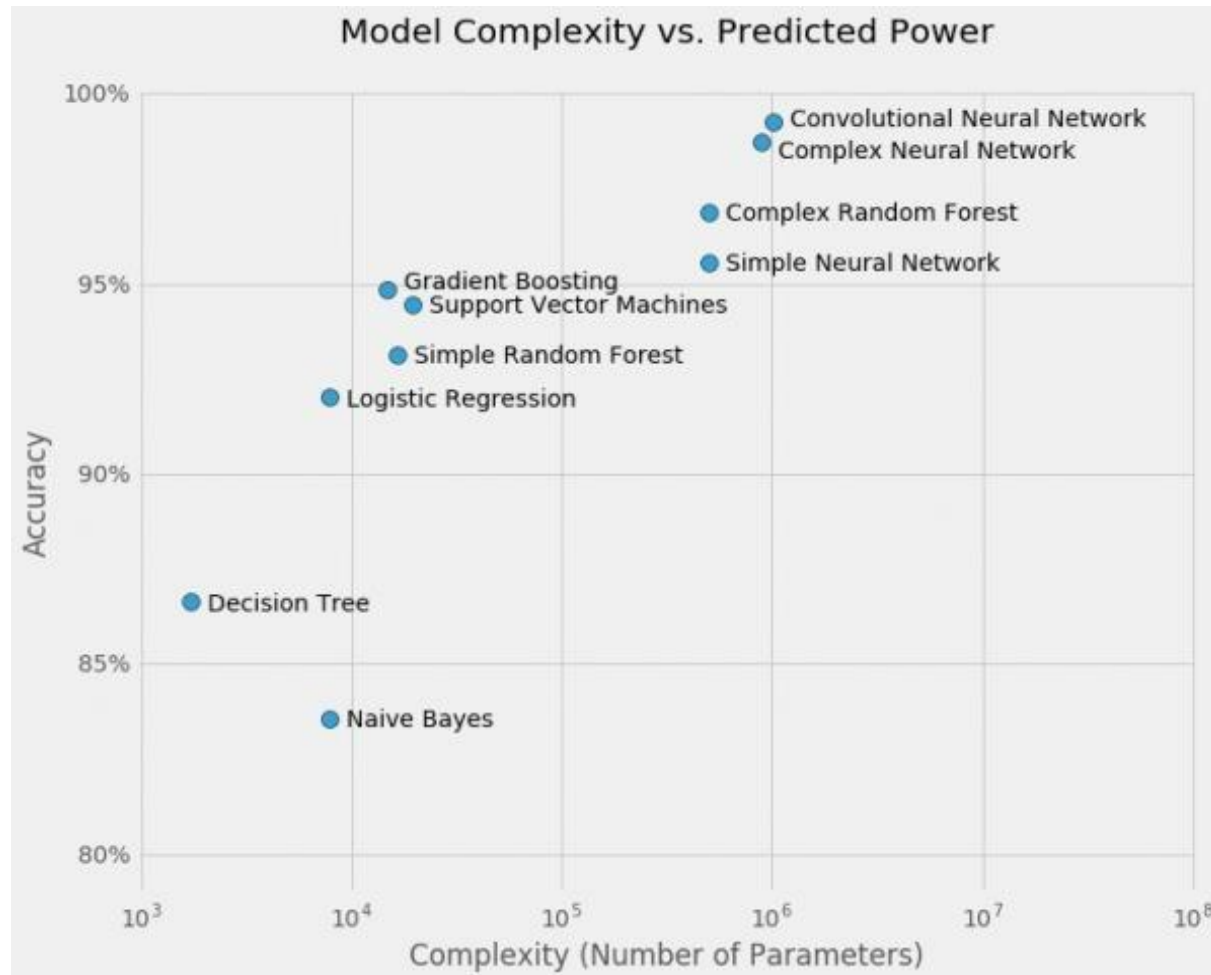
Utilisation conseillé

- Lorsque les prédicteurs sont catégoriques (discrets) ou se comportent de façon non linéaire

Résultat



MODEL DE COMPLEXITÉ VS POUVOIR DE PREDICTION



Plan du cours

- 1 Les familles de techniques
- 2 Règles d'association
- 3 Segmentation
- 4 Classification & régression
- 5 Estimation de performances & Validation Méthodes



Validation

Les modèles extraits **ne peuvent être utilisés directement en toute fiabilité**

- ✓ Il faut les évaluer, les soumettre à l'épreuve de la réalité et apprécier leur justesse
- ✓ Estimer le taux d'erreur du modèle

- 💡 Améliorer le modèle selon les résultats obtenus
- 💡 Demander l'avis d'un expert pour valider le modèle
- 💡 Déploiement & Intégration du modèle



Performance metrics

- Il est extrêmement important d'utiliser des métriques quantitatives pour évaluer un modèle d'apprentissage automatique
- Jusqu'à présent, nous nous appuyions sur la valeur de la fonction de coût pour la régression et la classification.
- D'autres métriques peuvent être utilisées pour mieux évaluer et comprendre le modèle.
- **Pour la classification:** Accuracy/Precision/Recall/F1-score, ROC curves,...
- **Pour la régression:** Normalized RMSE, Normalized Mean Absolute Error (NMAE),...

Performance metrics

What is Confusion Matrix and why you need it?

- Il s'agit d'une mesure des performances pour un problème de classification d'apprentissage automatique où le résultat peut être deux classes ou plus. Il s'agit d'un tableau avec 4 combinaisons différentes de valeurs prédites et réelles.
- Il est extrêmement utile pour mesurer Recall, Precision, Specificity, Accuracy, et AUC-ROC curves.

Confusion matrix

		Actual class	
		1 (p)	0 (n)
Predicted class	1 (Y)	True positive (TP)	False positive (FP)
	0 (N)	False negative (FN)	True negative (TN)

Performance metrics

What is Confusion Matrix and why you need it?





- Il s'agit d'une mesure des performances pour un problème de classification d'apprentissage automatique où le résultat peut être deux classes ou plus. Il s'agit d'un tableau avec 4 combinaisons différentes de valeurs prédites et réelles.
- Il est extrêmement utile pour mesurer Recall, Precision, Specificity, Accuracy, et AUC-ROC curves.

Confusion matrix

		Actual class	
		1 (p)	0 (n)
Predicted class	1 (Y)	True positive (TP)	False positive (FP)
	0 (N)	False negative (FN)	True negative (TN)

Performance metrics

What is Confusion Matrix and why you need it?

		Actual	
		CAT	NOT CAT
Predicted	CAT		
	NOT CAT		

What is Confusion Matrix and why you need it?

True Positive

- Interpretation: You predicted positive and it's true.

True Negative

- Interpretation: You predicted negative and it's true.

False Positive: (Type 1 Error)

- Interpretation: You predicted positive and it's false.

False Negative: (Type 2 Error)

- Interpretation: You predicted negative and it's false.

Performance metrics

Matrice de confusion (avec N classes)

Tableau bidimensionnel **N lignes** × **N colonnes**

CM(i,i): Nombre d'observations désirées comme C_i et prédites comme C_i

***n** le nombre total des observations

CM		Classe prédite			
		C_1	...	C_i	C_N
Classe désiré	C_1	CM(1,1)		CM(1,i)	
	\vdots		\vdots		
	C_i			CM(i,j)	
	C_N				CM(N,N)

❑ Taux de classification Accuracy **τ**

$$\tau = \frac{\sum_{i=1}^N CM(i,i)}{n}$$

❑ Taux d'erreur Error rate **e**

$$e = \frac{\sum_{i=1}^N \sum_{j=1}^N CM(i,j)}{n} = 1 - \tau$$

	A1	A2	...	C.Désirée	C.Prédite
1				Oui	Oui
2				Non	Oui
3				Non	Non
4				Non	Oui
5				Oui	Non
6				Oui	Oui
7				Non	Non
8				Oui	Oui

Performance metrics

Exemple de classification des maladies: un modèle de régression logistique est entraîné, avec $y = 1$ si maladie, $y = 0$ sinon.

Accuracy

Parmi tous les patients dont nous avons, combien d'entre eux avons-nous prédit correctement.

$$\frac{\text{True Positive} + \text{True Negative}}{\text{\# Observations}}$$

Confusion matrix

		Actual class	
		1 (p)	0 (n)
Predicted class	1 (Y)	True positive (TP)	False positive (FP)
	0 (N)	False negative (FN)	True negative (TN)

Performance metrics pour les classes asymétriques

Exemple de classification des maladies

- Un modèle de régression logistique est entraîné, avec $y = 1$ si maladie, $y = 0$ sinon.
- 1 % d'erreur est obtenu sur l'ensemble de test (99 % de diagnostics corrects).
- Seulement 0,50 % des patients sont réellement atteints de la maladie.
- La classe $y = 1$ a très peu d'exemples avec par rapport à la classe $y = 0$.



Si j'utilise un prédicteur qui prédit toujours la classe 0, j'obtiens 99,5% de précision !! Pour les classes asymétriques, la mesure d **Accuracy peut être trompeuse**

Precision and recall

Supposons que $y = 1$ (illness) en présence d'une classe rare que nous voulons détecter

Precision

Parmi tous les patients dont nous avons prédit qu'ils seraient atteints de la maladie, quelle fraction en est réellement atteinte ?

$$\frac{\text{True Positive}}{\# \text{ Predicted Positive}} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}$$

Recall

Parmi tous les patients qui souffrent réellement d'une maladie, quelle fraction avons-nous correctement détectée comme étant atteintes de la maladie ?

$$\frac{\text{True Positive}}{\# \text{ Actual Positive}} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}$$

Confusion matrix

Actual class

Predicted class	Actual class	
	1 (p)	0 (n)
1 (Y)	True positive (TP)	False positive (FP)
0 (N)	False negative (FN)	True negative (TN)

Critères de performance

Exercice: Étant donné le modèle de prédiction Γ qui permet de déterminer si l'objet dans une image est un chat

- 200 images sont utilisées pour tester la performance de Γ : 170 contiennent des chats, et 30 ne contiennent pas de chats.
- Le résultat de prédiction du modèle F est 160 contiennent des chats, et 40 ne contiennent pas de chats (140 uniquement désirées comme **Chat** et prédites comme **Chat**).
- Déterminer la matrice de confusion associée à Γ . Puis déduire l'accuracy, la précision de la classe **Chat**, et le Recall de la classe **non chat**.

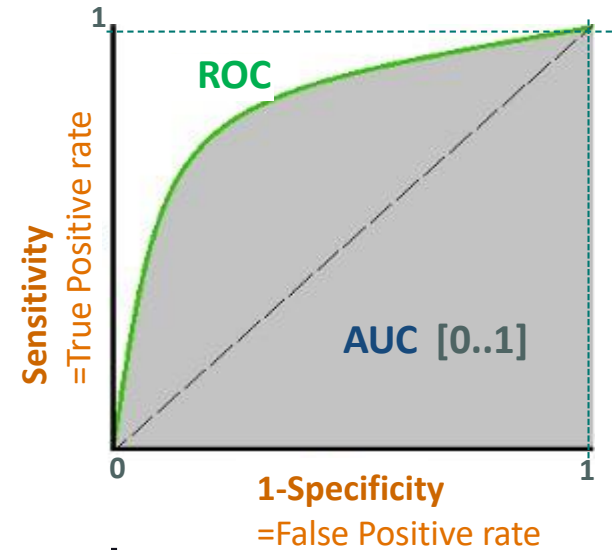
$$\text{accuracy} = \frac{140 + 10}{200} = 75\%$$
$$\text{recall}_{\overline{\text{Chat}}} = \text{sepcificity} = \frac{10}{20 + 10} = 33.3\%$$
$$\text{precision}_{\text{Chat}} = \frac{140}{140 + 20} = 87.5\%$$

		Classe désirée	
		Chat	$\overline{\text{Chat}}$
Classe prédite	Chat	140	20
	$\overline{\text{Chat}}$	30	10

Classification à différents seuils (Cas de 2 classes)

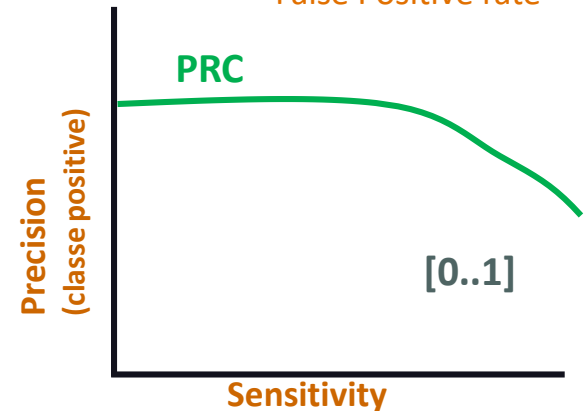
Courbe Receiver Operating Characteristics (ROC)

- L'Area Under Curve (AUC) de la courbe ROC est une métrique qui mesure la capacité d'un modèle à distinguer entre les classes. Plus la valeur de l'AUC est élevée, meilleure est la performance du modèle pour prédire correctement les classes. L'AUC varie de 0 à 1.



Precision-Recall Curve (PRC)

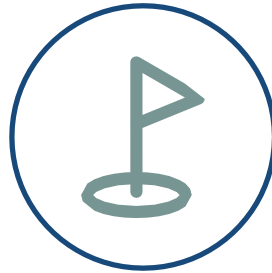
- Montre le compromis entre la précision et le rappel pour différents seuils
- Des scores élevés de rappel et précision montrent que le classifieur renvoie des résultats précis



Echantillonnage



Base de données
(Dataset)



Ensemble d'apprentissage
(Training set)



Ensemble de Test
(Test set)

L'idée d'un ensemble de validation est d'estimer les performances du modèle à partir d'un échantillon. la procédure courante est:

- Supprimer un sous-ensemble des données d'entraînement → **ce sous-ensemble n'est pas utilisé pour l'entraînement.**
- Entraîner le modèle sur les données d'entraînement restantes → **le modèle sera entraîné sur moins de données.**
- Évaluer les performances du modèle sur l'ensemble retenu (held-out set).
- Réentraîner le modèle sur toutes les données.

Méthodes de validation



Split validation

Processus d'apprentissage et de test n'est effectué **qu'une seule fois**



Cross-validation

Faire **plusieurs tests** sur différents ensembles d'apprentissage et de test

Split validation

- Processus de validation qui s'exécute qu'une seule fois
- Diviser le jeu de données en **deux groupes** :
 - **Ensemble d'apprentissage** : utilisé pour **former le classifieur**
 - **Ensemble de test** : utilisé pour **estimer la performance** du classifieur formé

Jeu de données

Ensemble d'apprentissage (70%)

Ensemble de test (30%)

Cross validation : K-Folds

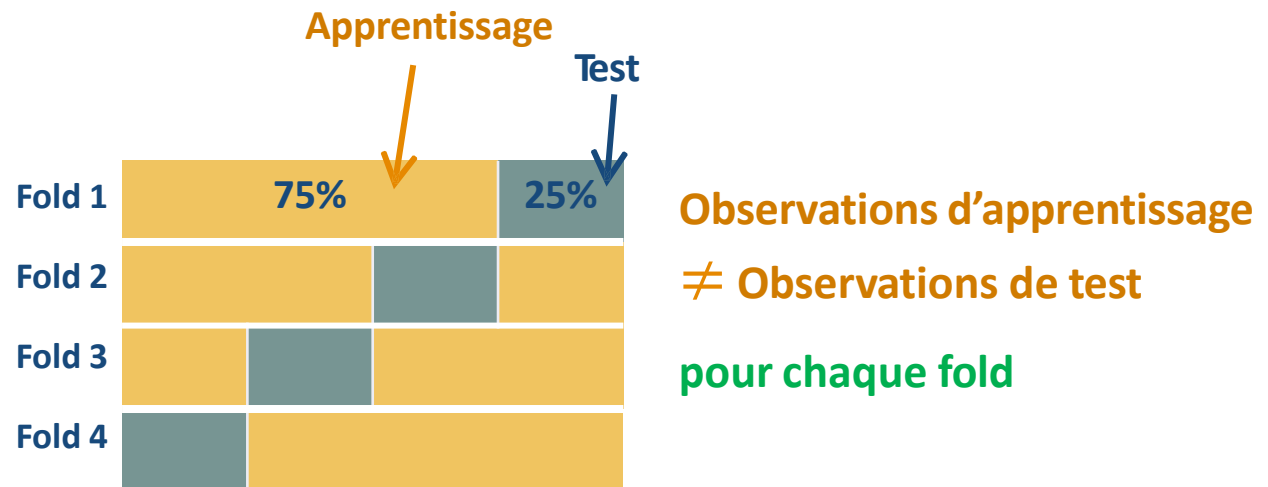
☰ Diviser **k** fois le jeu de données de **N** observations

Pour chaque fold : **1.. k**

1. Sélectionner $N \cdot \frac{k-1}{k}$ observations comme ensemble d'apprentissage
2. Les restes des observations comme ensemble de test

NB. Répéter l'opération de sorte que toutes les observations aient été utilisé exactement une fois pour le test

Exemple **k= 4**



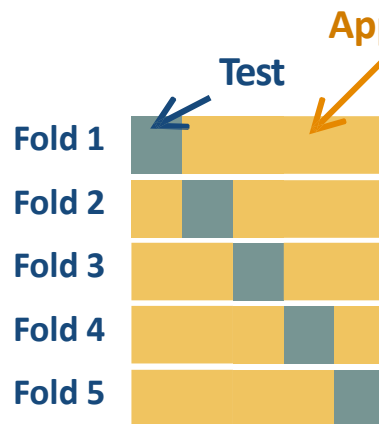
Cross validation : Leave-one-out

☰ Diviser N fois le jeu de données de N observations

Pour chaque fold : $1..N$

1. Sélectionner une observation pour le test
2. Les restes $N-1$ observations comme ensemble d'apprentissage

Exemple $k=5$

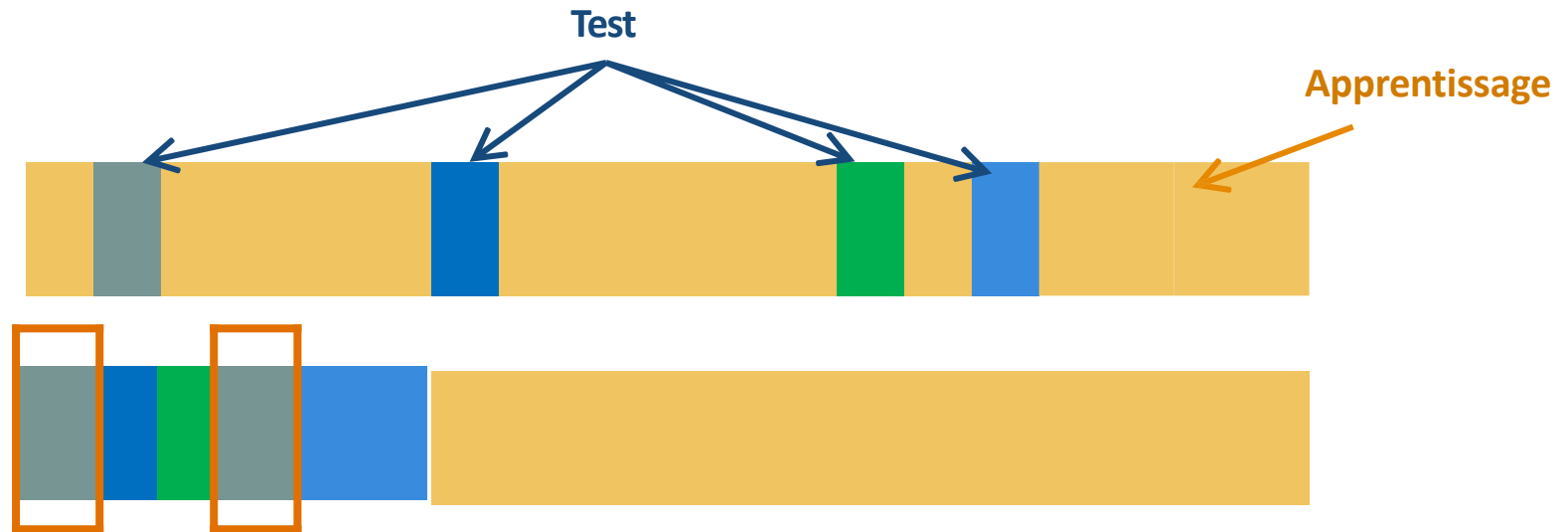


Leave-one-out= k-fold avec $k= N$

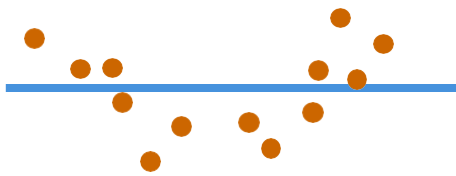
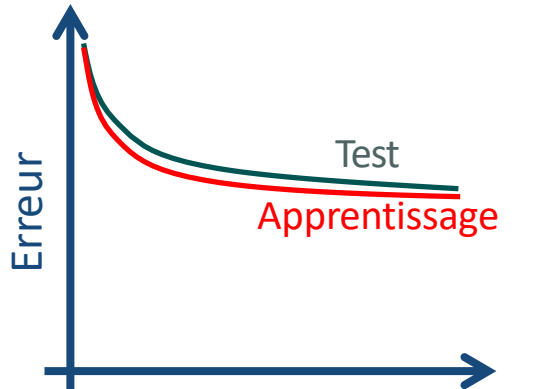
Cross validation : Bootstrapping

- Tirer **au hasard** une observations **avec remplacement** parmi N
→ Réitérer cette opération **K** fois pour obtenir une estimation moyenne stable

K=?

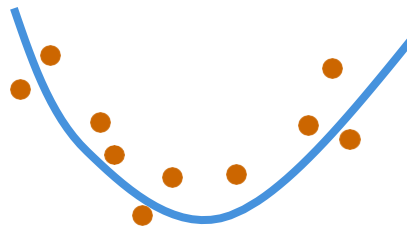
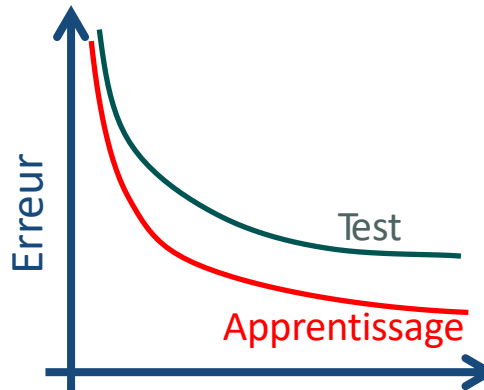


Sous-apprentissage vs. Sur-apprentissage

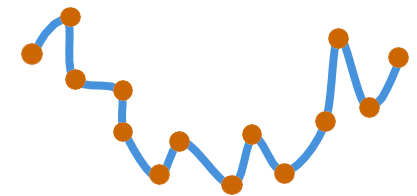
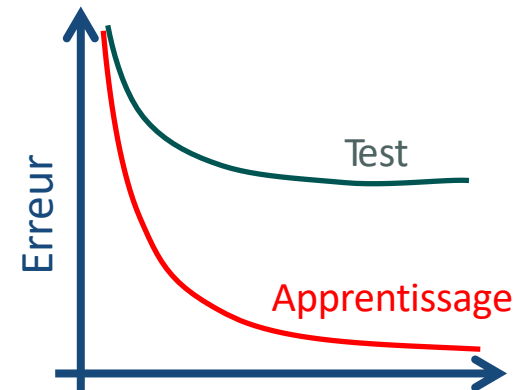


Sous-apprentissage
Underfitting

Incapacité à apprendre
les caractéristiques



Parfait
perfect



Sur-apprentissage
Overfitting

Learning noise

Quiz

1. Étant donnée un jeu de données de **20** observations; le taux de classification obtenu en appliquant la validation croisée **20-Folds** est égal à **91.6%** ; En appliquant **leave-one-out**, le taux d'erreur est égal à (une seule réponse)

- ☒ 8.4%
- ☐ 91.6%
- ☐ 50%
- ☐ Aucune bonne réponse

2. Soit la matrice de confusion suivante : (réponse multiple)

	C1	C2	C3
C1	10	1	3
C2	0	8	0
C3	2	2	11

- ☒ Le nombre total d'observations est égal à 37
- ☒ Le nombre d'observations mal classées est égal à 8
- ☒ $\text{recall}_{C2} = 0.73$
- ☒ $\text{precision}_{C2} = 1$