

IFT870/BIN710

Forage de données

Thème 1 : Introduction

Davy Ouedraogo
Département d'informatique



Partie I : Théorie

Définition

- ❑ **Forage de données (Datamining)** : extraction d'informations non-triviales, cachées, et utiles de grandes bases de données.
- ❑ Ensemble de méthodes sophistiquées pour l'analyse de grandes bases de données afin de découvrir des informations (patterns/modèles) utiles cachées par la quantité de données, pour aider à la décision.

Nécessité du forage de données

□ **Croissance explosive des données**

❖ Collection et disponibilité des données

- Facilité de collection automatique, explosion des bases de données, Web, société informatisée

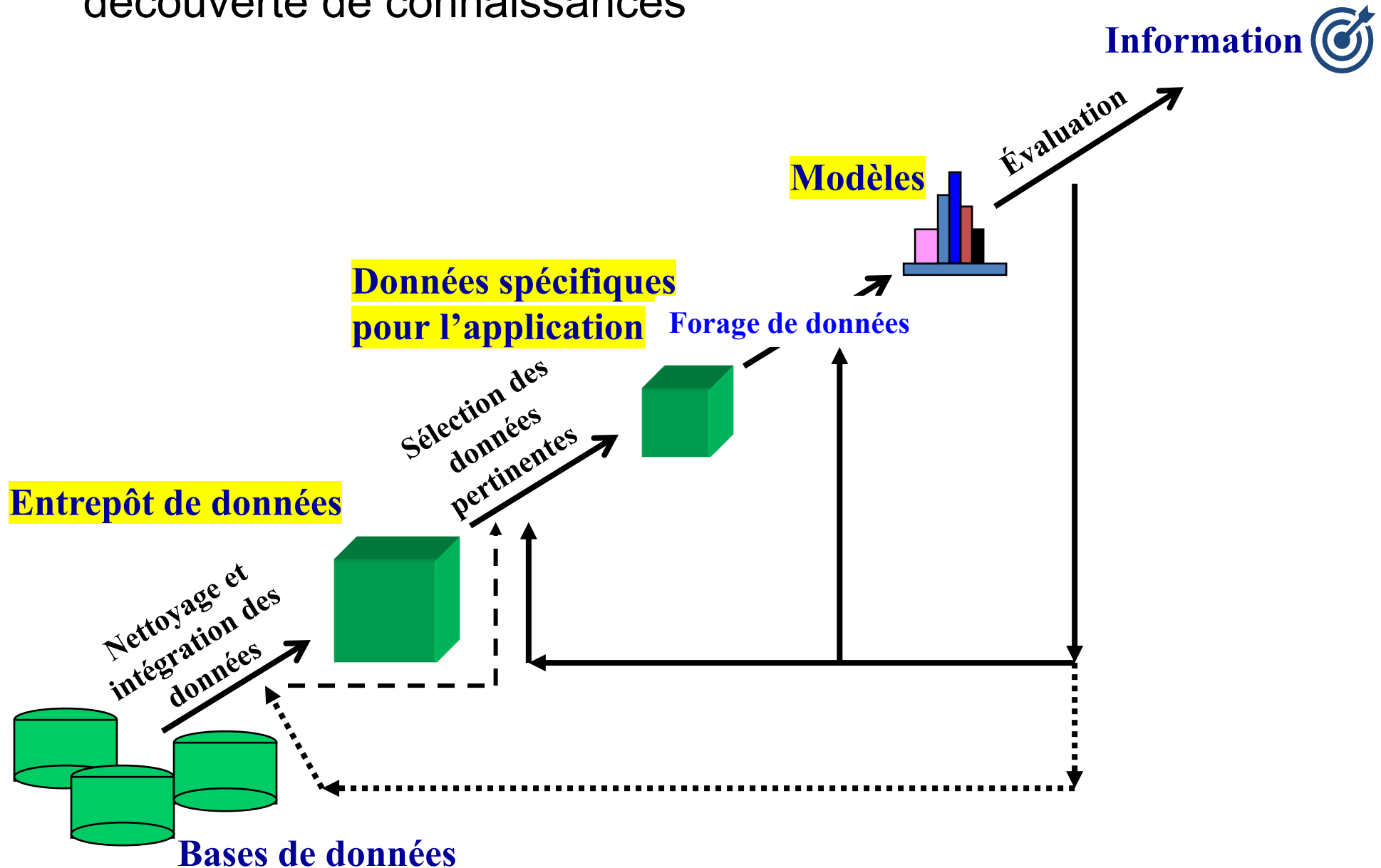
❖ Des sources de données massives

- **Industrie** : Web, e-commerce, transactions, finances, stocks, ... (productivité)
- **Science** : bio-informatique, cyber sécurité, médecine, simulation scientifique, (données → hypothèses)
- **Société** : réseaux sociaux, caméras digitales, ...

□ **Besoin de méthodes automatisées pour analyser ces données**

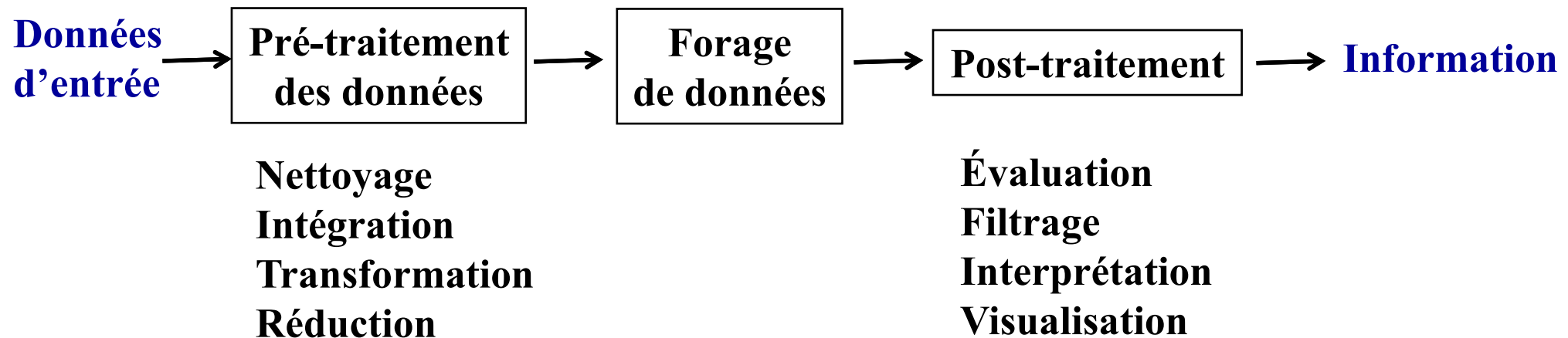
Découverte de connaissances et forage de données (Knowledge Discovery and Datamining (KDD))

- ❑ Le **forage de données** joue un rôle essentiel dans le processus de découverte de connaissances

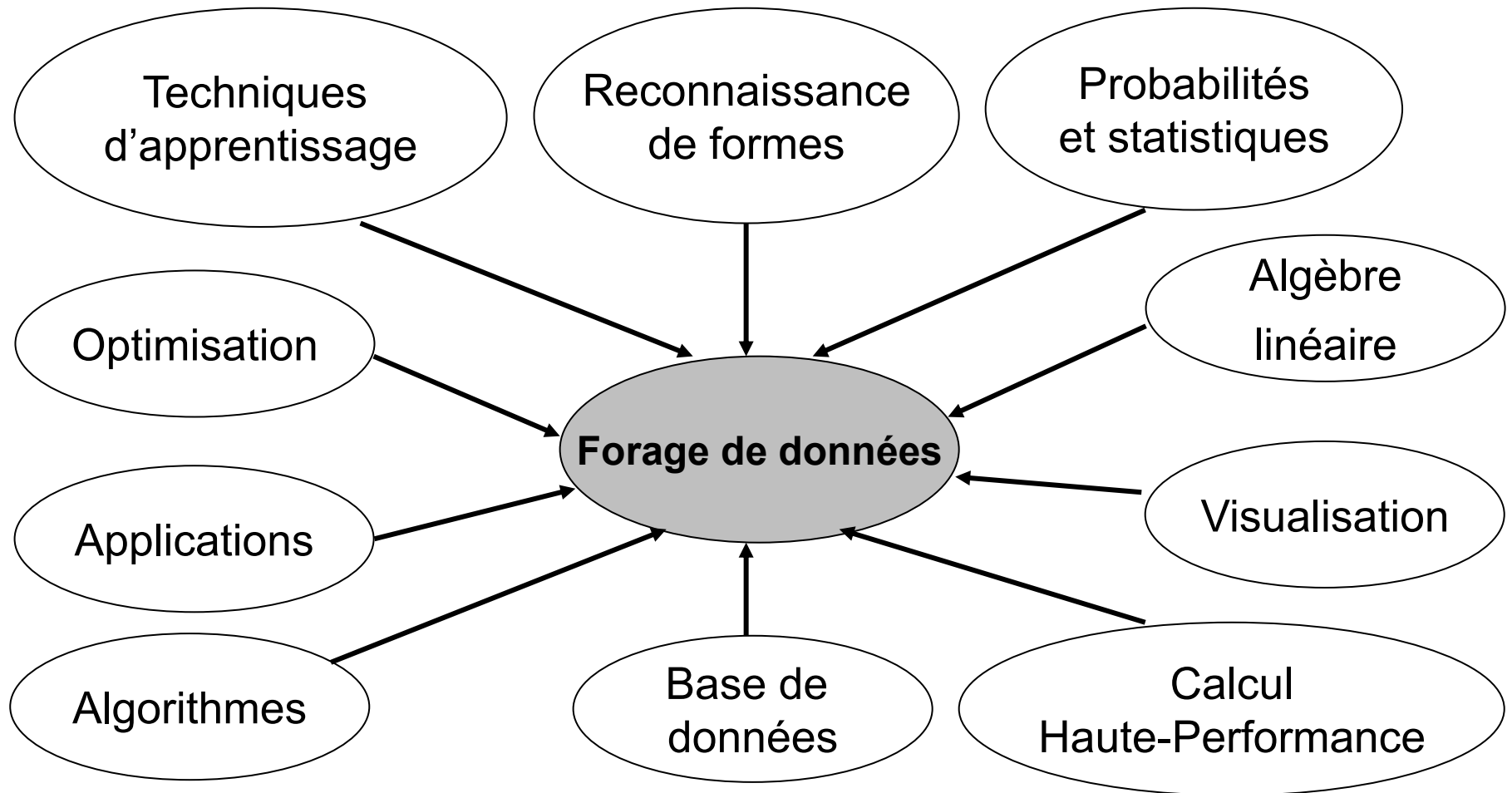


Découverte de connaissances et forage de données (Knowledge Discovery and Datamining (KDD))

- ❑ Le forage de données joue un rôle essentiel dans le processus de découverte de connaissances



Forage de données : intégration de plusieurs disciplines



Type de données

□ **Données non-structurées**

- ❖ Vecteurs de valeurs d'attributs (ex : matrices numériques)
- ❖ Données de documents (ex: matrices documents-termes)
- ❖ Données de transactions (ex. ensembles d'items)

Type de données

- ❖ Vecteurs de valeurs d'attributs (ex: matrices numériques)
- ❖ Données de documents (ex: matrices documents-termes)

	équipe	coach	pays	balle	score	jeu	gagné	perdu	saison
Document 1	3	0	5	0	2	6	0	2	0
Document 2	0	7	0	2	1	0	0	3	0
Document 3	0	1	0	0	1	2	2	0	3

- ❖ Données de transactions (ex: ensembles d'items)

<i>TID</i>	<i>Items</i>
1	Bread, Coke, Milk
2	Beer, Bread
3	Beer, Coke, Diaper, Milk
4	Beer, Bread, Diaper, Milk
5	Coke, Diaper, Milk

Type de données

❑ **Données non-structurées**

- ❖ Vecteurs de valeurs d'attributs (ex : matrices numériques)
- ❖ Données de documents (ex: matrices documents-termes)
- ❖ Données de transactions (ex. ensembles d'items)

❑ **Données structurées**

- ❖ Ordonnées : vidéo (séquence d'images), données séquentielles (séquences de données, ex: séquences biologiques), données temporelles (série de données ordonnées dans le temps)
- ❖ Graphes : réseaux sociaux, données du web

❑ **Données spatiales et de multimedia**

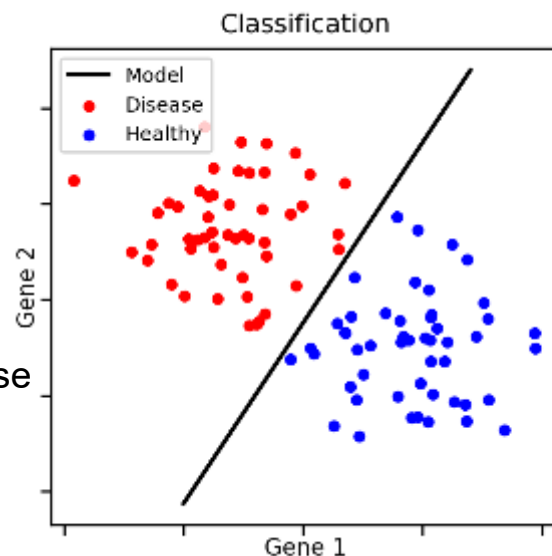
- ❖ Données spatiales (cartes)
- ❖ Images

Fonctions du forage de données

❑ Fonctions prédictives (apprentissage supervisé)

❖ Pour un ensemble de données observées sous la forme (X,y) où X est un vecteur de valeurs d'attributs et y la valeur d'un attribut cible correspondant, trouver un modèle h qui estime y avec précision étant donné un nouveau vecteur X , i.e. $y \approx h(X)$.

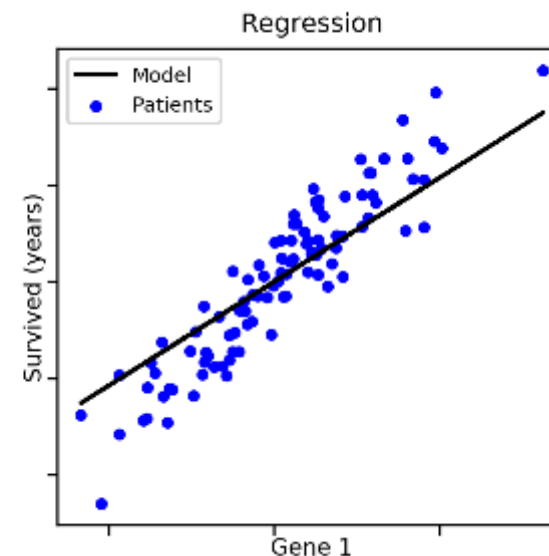
- **Classification** : attribut cible à valeurs discrètes (catégoriques)
- **Regression** : attribut cible à valeurs continues (numériques)



$y = \{ \text{Disease, Healthy} \}$

X de Gene1xGene2

$h : \text{Gene1xGene2} \rightarrow \text{Classe}$



y de AnneesSurvie $\in \mathbb{R}^+$

X de Gene1

$h : \text{Gene1} \rightarrow \text{AnneesSurvie}$

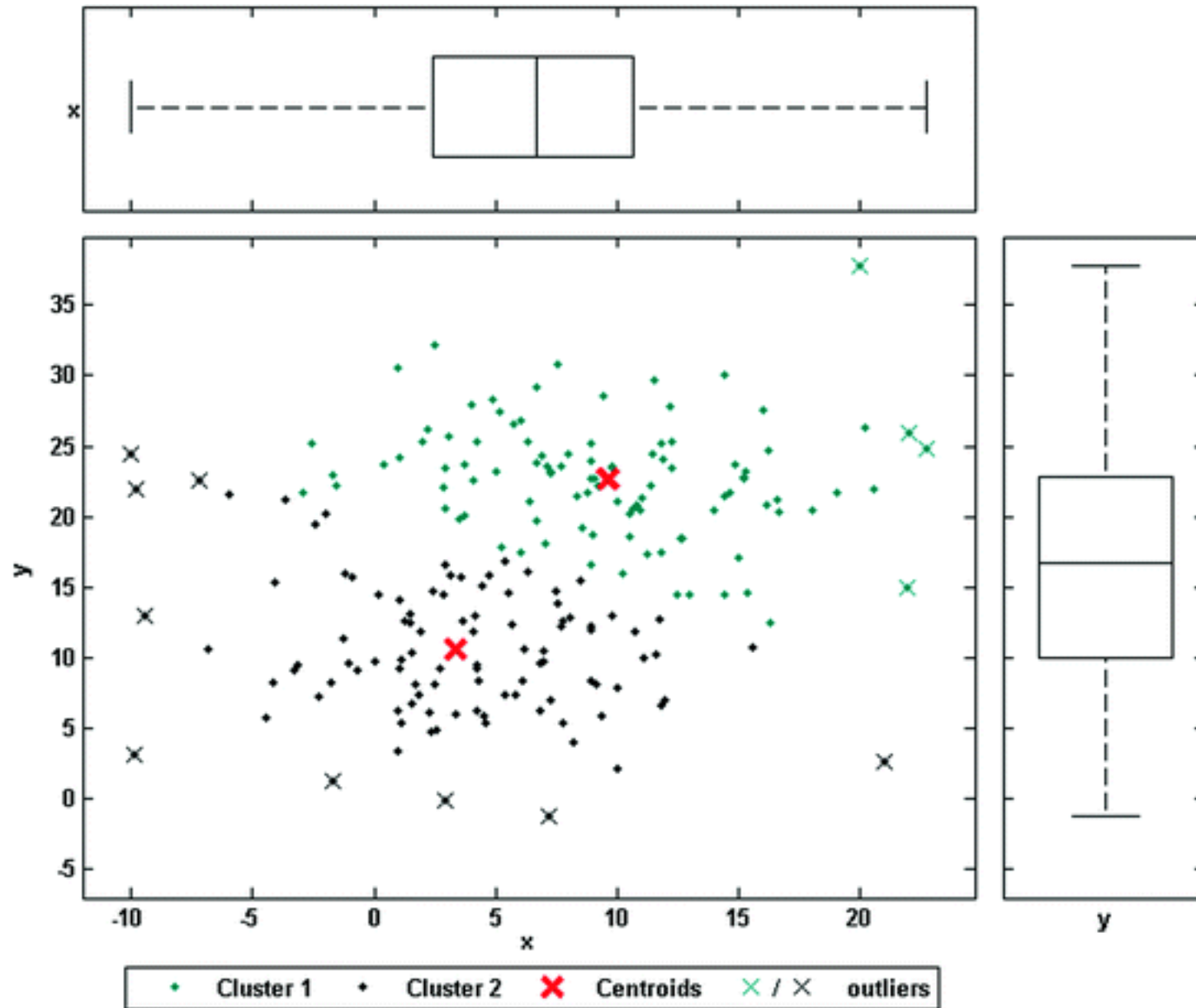
Fonctions du forage de données

❑ Fonctions descriptives (apprentissage non-supervisé)

- ❖ Pour un ensemble de données observées sous la forme de vecteurs de valeurs d'attributs X , trouver un modèle h qui **résume les caractéristiques de l'ensemble**.
 - **Clustering** : h estime la classe y de chaque vecteur X donnée en entrée, i.e. $y \approx h(X)$, de sorte à maximiser la similarité intra-classe, tout en minimisant la similarité inter-classe.
 - **Détection de données aberrantes (outliers)** : h permet d'identifier les cas exceptionnels qui s'écartent considérablement de la majorité des groupes de données.

Fonctions du forage de données

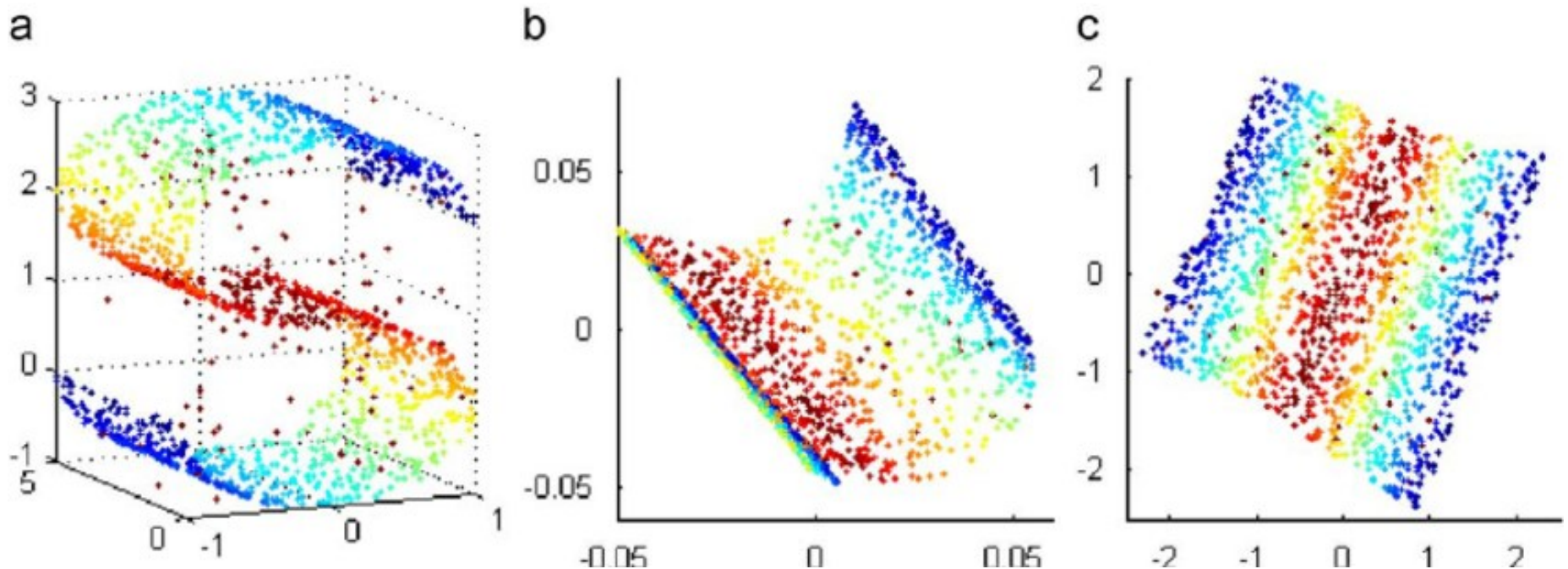
❑ Fonctions descriptives (apprentissage non-supervisé)



Fonctions du forage de données

□ Fonctions descriptives (apprentissage non-supervisé)

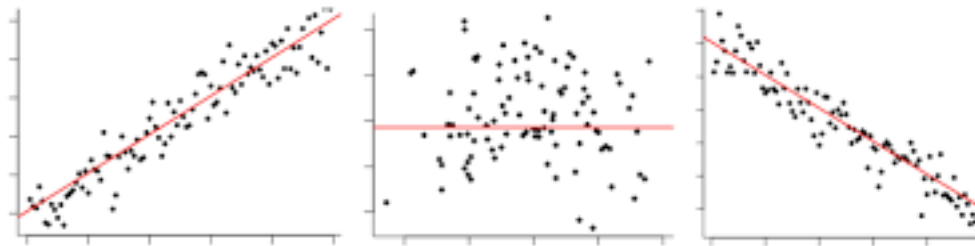
- ❖ Pour un ensemble de données observées sous la forme de vecteurs de valeurs d'attributs X , trouver un modèle h qui **résume les relations sous-jacentes entre les données**.
 - Réduction de dimension : h estime une nouvelle représentation de chaque X sur un nombre réduit d'attributs.



Fonctions du forage de données

❑ Fonctions descriptives (apprentissage non-supervisé)

- ❖ Analyse de corrélation : r permet d'estimer le degré de la relation linéaire entre deux attributs



- ❖ Analyse d'association : h permet d'identifier des relations fortes entre des attributs dans les données.

Tid	Panier
10	Bière, Noix, Couches
20	Bière, Café, Couches
30	Bière, Couches, Œufs
40	Noix, Œufs, Lait
50	Noix, Café, Couches, Œufs, Lait

Bière → Couches

Algorithmes d'apprentissage supervisé

☐ **K plus proches voisins**

☐ **Modèles linéaires**

☐ **Classificateur de Bayes Naïf**

☐ **Arbres de décision**

☐ **Ensemble d'arbres de décision**

☐ **Machines à vecteur de support et à noyau**

Hypothèses pour le choix :

- Type de données
- Existence d'une métrique
- Interprétabilité
- Explicabilité
- Séparation des classes
- Colinéarité
- Quantité des données

Algorithmes d'apprentissage supervisé

□ Méthodes des k plus proches voisins (KNN)

❖ Approche très simple :

- stocker les données (X,y) observées,
- pour toute nouvelle donnée X' , calculer y' en fonction des k données observées les plus proches

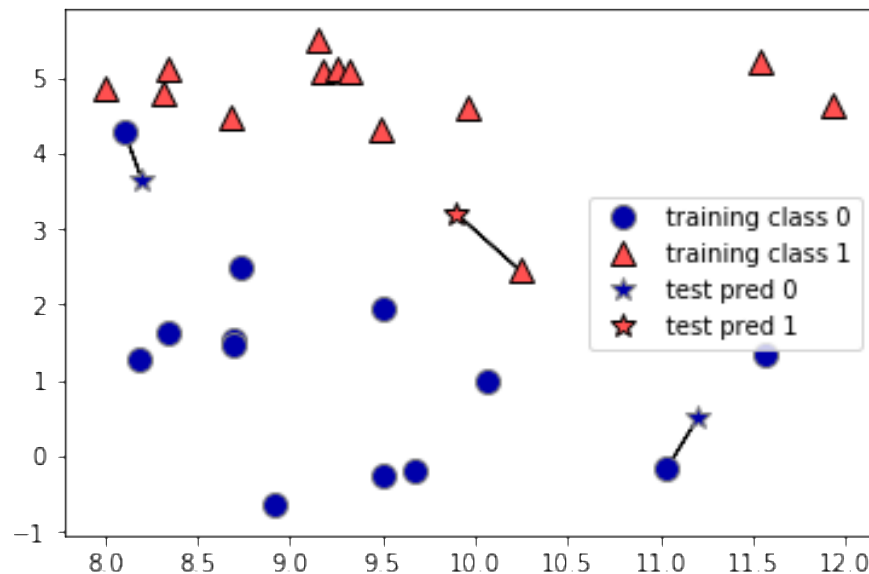
Algorithmes d'apprentissage supervisé

❑ Méthodes des k plus proches voisins (KNN)

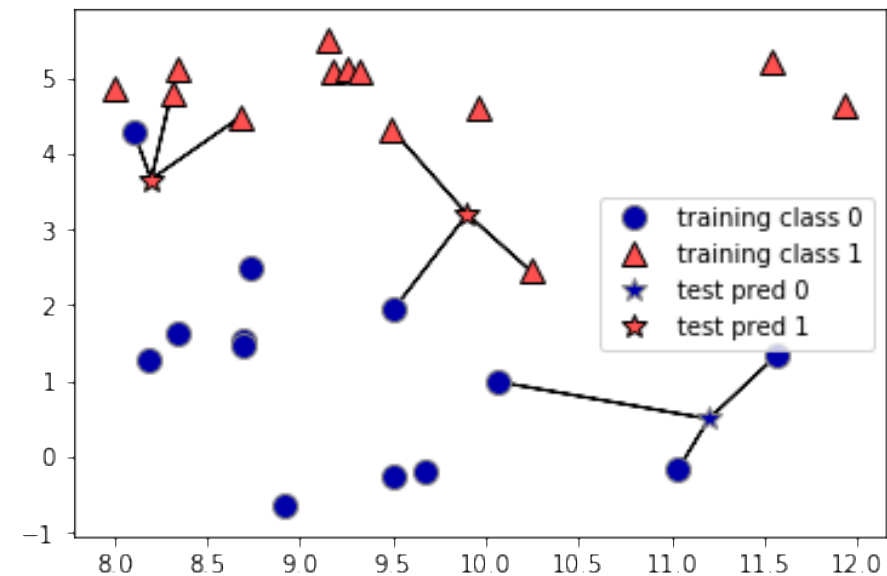
❖ Approche très simple :

- stocker les données (X,y) observées,
- pour toute nouvelle donnée X' , calculer y' en fonction des k données observées les plus proches

Classification ($k=1$)



Classification ($k=3$)



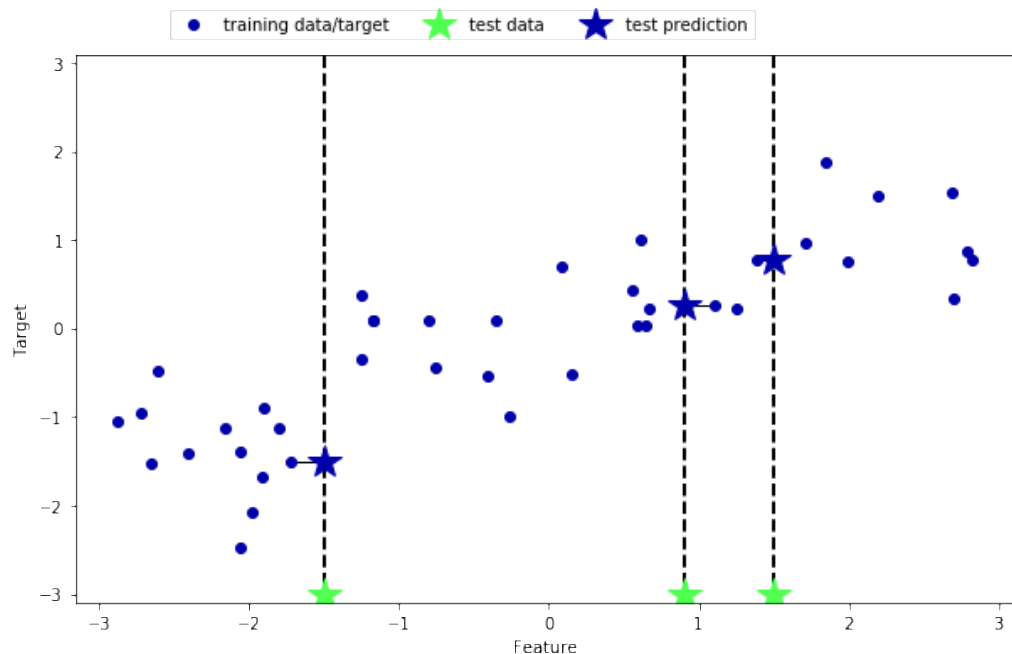
Algorithmes d'apprentissage supervisé

❑ Méthodes des k plus proches voisins (KNN)

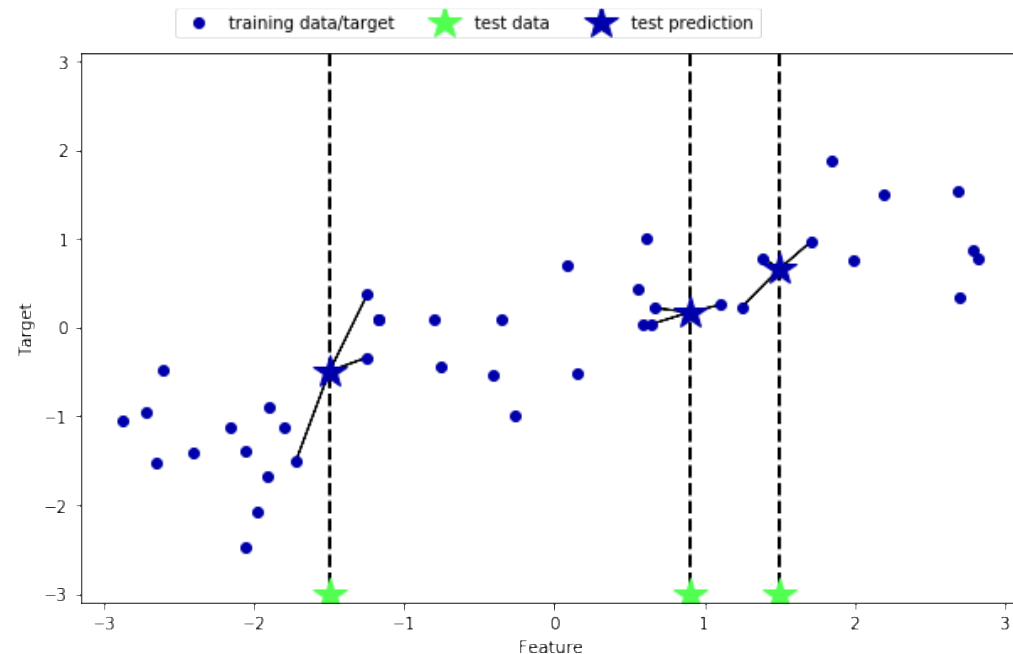
❖ Approche très simple :

- stocker les données (X,y) observées,
- pour toute nouvelle donnée X , calculer y en fonction des k données observées les plus proches

Régression ($k=1$)



Régression ($k=3$)



Algorithmes d'apprentissage supervisé

□ Modèles linéaires

❖ Très utilisés en pratique :

- pour toute nouvelle donnée X , le modèle h prédit y comme une fonction linéaire des attributs de X

$$y = h(X) = coef[0]*X[0] + coef[1]*X[1] + ... + coef[n]*X[n] + intercept$$

Algorithmes d'apprentissage supervisé

□ Modèles linéaires

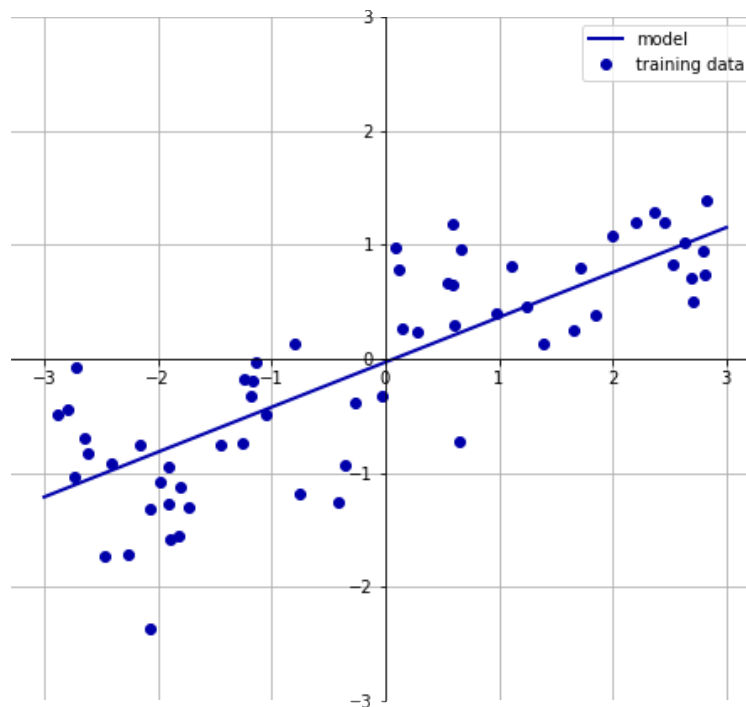
❖ Très utilisés en pratique :

- pour toute nouvelle donnée X , le modèle h prédit y comme une fonction linéaire des attributs de X

$$y = h(X) = coef[0]*X[0] + coef[1]*X[1] + ... + coef[n]*X[n] + intercept$$

Régression linéaire (Ordinary Least Square) :

Trouve les *coef* qui minimisent la somme des carrés entre les valeurs des prédictions et des vraies cibles.



Algorithmes d'apprentissage supervisé

□ Modèles linéaires

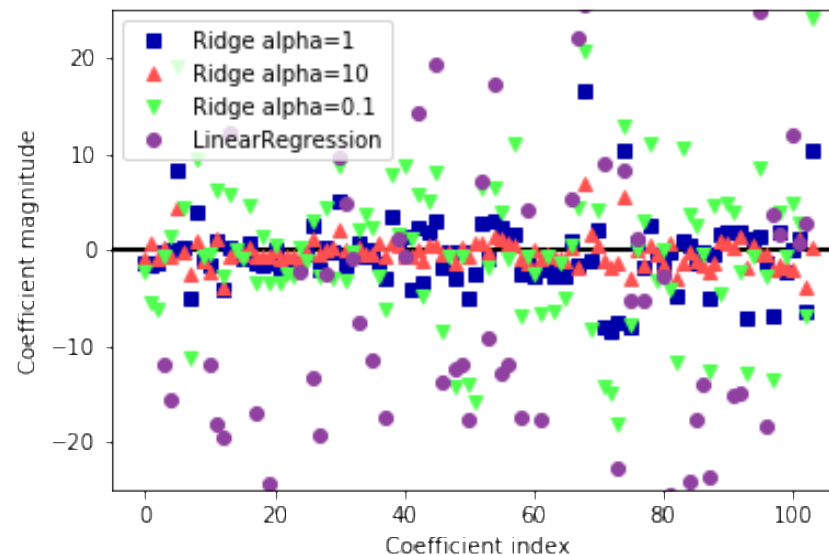
❖ Très utilisés en pratique :

- pour toute nouvelle donnée X , le modèle h prédit y comme une fonction linéaire des attributs de X

$$y = h(X) = coef[0]*X[0] + coef[1]*X[1] + ... + coef[n]*X[n] + intercept$$

Régression de Ridge:

Régression linéaire +
les valeurs des coefficients
doivent être les plus
petites possibles
(Régularisation L2)



Algorithmes d'apprentissage supervisé

❑ Modèles linéaires

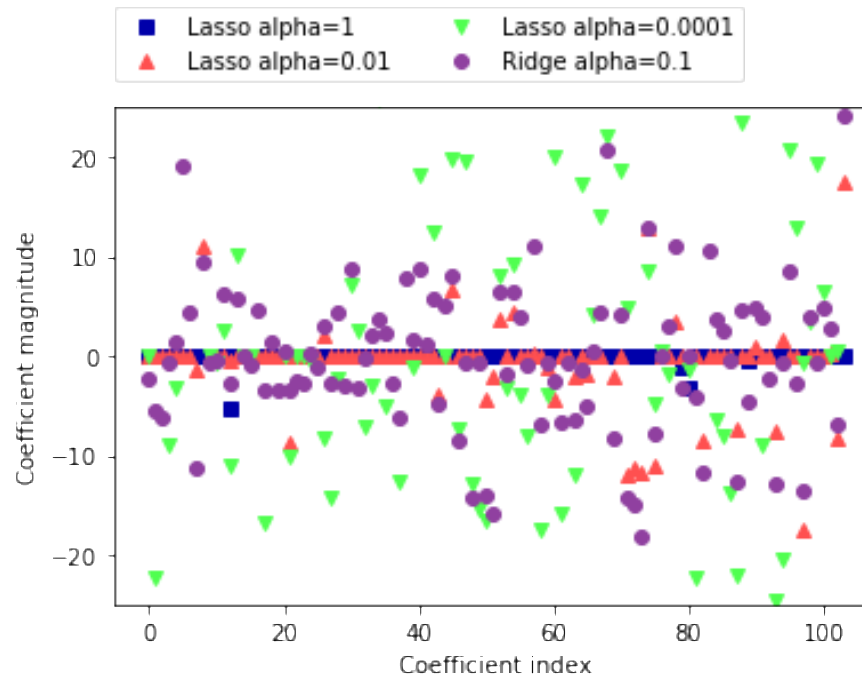
❖ Très utilisés en pratique :

- pour toute nouvelle donnée X , le modèle h prédit y comme une fonction linéaire des attributs de X

$$y = h(X) = coef[0]*X[0] + coef[1]*X[1] + ... + coef[n]*X[n] + intercept$$

Régression de Lasso:

Régression linéaire +
Régularisation L2 +
les valeurs de certains
coefficients sont à 0
(Régularisation L1)



Algorithmes d'apprentissage supervisé

❑ Modèles linéaires

❖ Très utilisés en pratique :

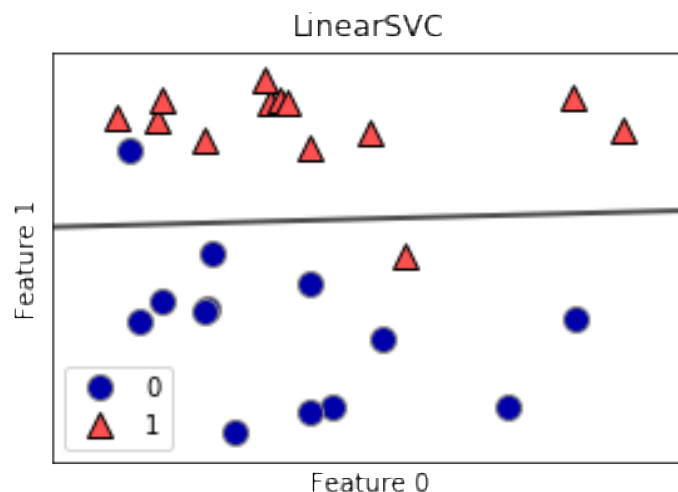
- pour toute nouvelle donnée X , le modèle h prédit y comme une fonction linéaire des attributs de X

Classification

$$y = (coef[0]*X[0] + coef[1]*X[1] + ... + coef[n]*X[n] + intercept > 0)$$

**Machine à vecteur de support linéaire
(LinearSVC) :**

Régularisation L2 par défaut



**Régression logistique
(LogisticRegression):**

Régularisation L2 par défaut



Algorithmes d'apprentissage supervisé

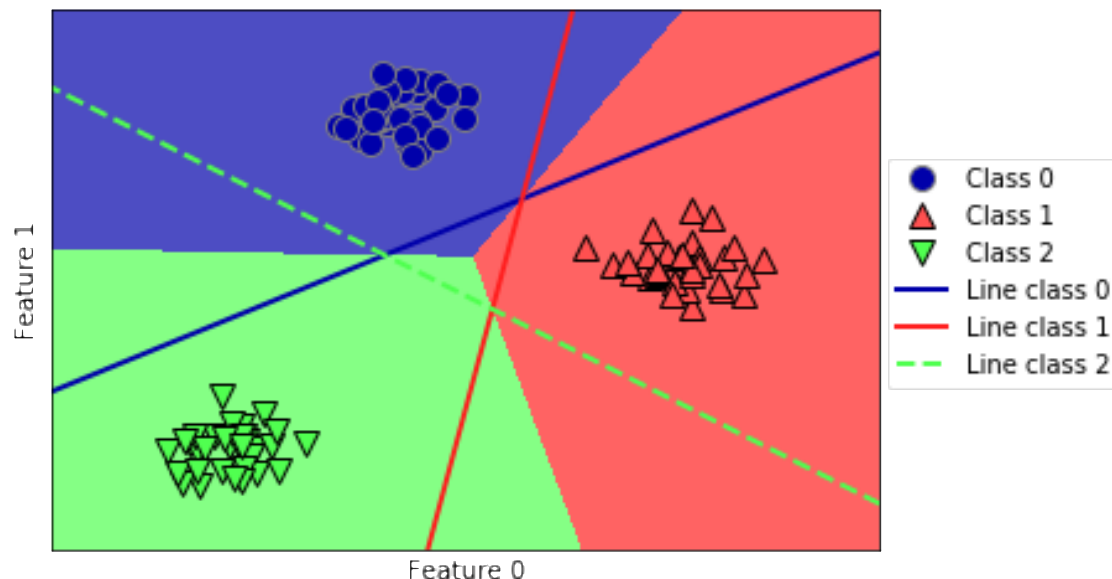
❑ Modèles linéaires

❑ Très utilisés en pratique :

❑ pour toute nouvelle donnée X , le modèle h prédit y comme une fonction linéaire des attributs de X

$$y = (coef[0]*X[0] + coef[1]*X[1] + ... + coef[n]*X[n] + intercept > 0)$$

Classification multi-classe (un modèle par classe un-contre-tous)



Algorithmes d'apprentissage supervisé

❑ Classificateur de Bayes Naïf

- ❖ Similaires aux modèles linéaires, mais plus rapide :
 - analyse individuellement chaque attribut et calcule des statistiques par classe

		Play Golf	
		Yes	No
Outlook	Sunny	3	2
	Overcast	4	0
	Rainy	2	3

		Play Golf	
		Yes	No
Outlook	Sunny	3/9	2/5
	Overcast	4/9	0/5
	Rainy	2/9	3/5

		Play Golf	
		Yes	No
Humidity	High	3	4
	Normal	6	1

		Play Golf	
		Yes	No
Humidity	High	3/9	4/5
	Normal	6/9	1/5

		Play Golf	
		Yes	No
Temp.	Hot	2	2
	Mild	4	2
	Cool	3	1

		Play Golf	
		Yes	No
Temp.	Hot	2/9	2/5
	Mild	4/9	2/5
	Cool	3/9	1/5

		Play Golf	
		Yes	No
Windy	False	6	2
	True	3	3

		Play Golf	
		Yes	No
Windy	False	6/9	2/5
	True	3/9	3/5

Algorithmes d'apprentissage supervisé

❑ Classificateur de Bayes Naïf

- ❖ Similaires aux modèles linéaires, mais plus rapide :
 - analyse individuellement chaque attribut et calcule des statistiques par classe

Frequency Table				Likelihood Table			
		Play Golf				Play Golf	
		Yes	No			Yes	No
Outlook	Sunny	3	2	Outlook	Sunny	3/9	2/5
	Overcast	4	0		Overcast	4/9	0/5
	Rainy	2	3		Rainy	2/9	3/5
		Play Golf				Play Golf	
		Yes	No			Yes	No
Humidity	High	3	4	Humidity	High	3/9	4/5
	Normal	6	1		Normal	6/9	1/5
		Play Golf				Play Golf	
		Yes	No			Yes	No
Temp.	Hot	2	2	Temp.	Hot	2/9	2/5
	Mild	4	2		Mild	4/9	2/5
	Cool	3	1		Cool	3/9	1/5
		Play Golf				Play Golf	
		Yes	No			Yes	No
Windy	False	6	2	Windy	False	6/9	2/5
	True	3	3		True	3/9	3/5

Méthode Gaussienne (GaussianNB)

Données de type continu (numérique)

Méthode de Bernoulli (BernoulliNB)

Données binaires

Méthode Multinomiale (MultinomialNB)

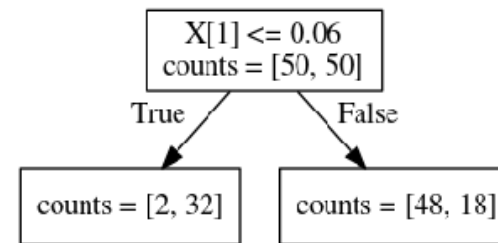
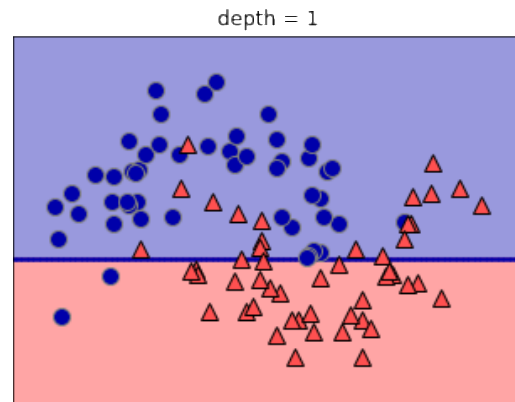
Données de type « nombres d'occurrences »

Algorithmes d'apprentissage supervisé

□ Arbres de décision

❖ Très utilisés en pratique

- pour toute nouvelle donnée X , le modèle h prédit y en fonction d'une hiérarchie de questions si/sinon, avec l'objectif d'utiliser le moins de questions possible.

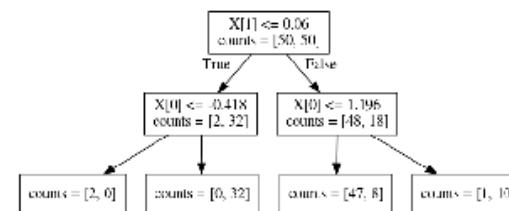
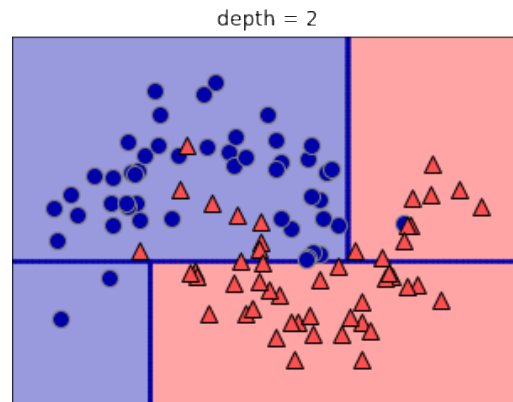


Algorithmes d'apprentissage supervisé

□ Arbres de décision

❖ Très utilisés en pratique

- pour toute nouvelle donnée X , le modèle h prédit y en fonction d'une hiérarchie de questions si/sinon, avec l'objectif d'utiliser le moins de questions possible.

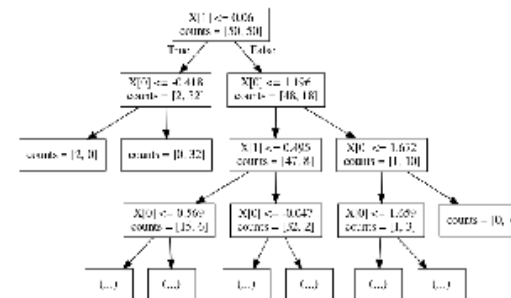
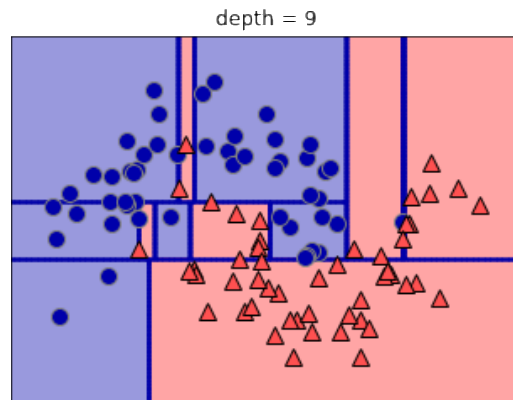


Algorithmes d'apprentissage supervisé

□ Arbres de décision

❖ Très utilisés en pratique

- pour toute nouvelle donnée X , le modèle h prédit y en fonction d'une hiérarchie de questions si/sinon, avec l'objectif d'utiliser le moins de questions possible.

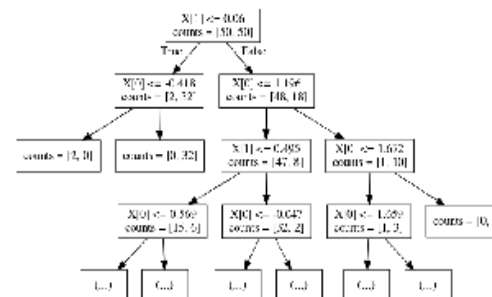
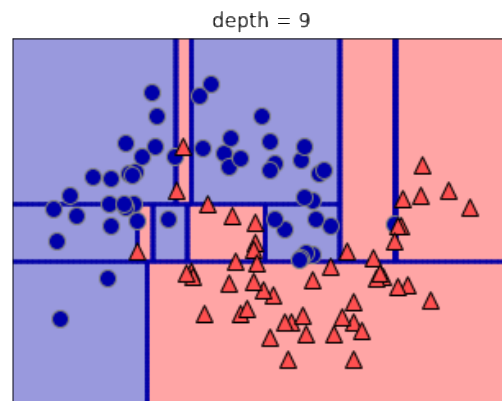


Algorithmes d'apprentissage supervisé

□ Arbres de décision

❖ Très utilisés en pratique

- pour toute nouvelle donnée X , le modèle h prédit y en fonction d'une hiérarchie de questions si/sinon, avec l'objectif d'utiliser le moins de questions possible.



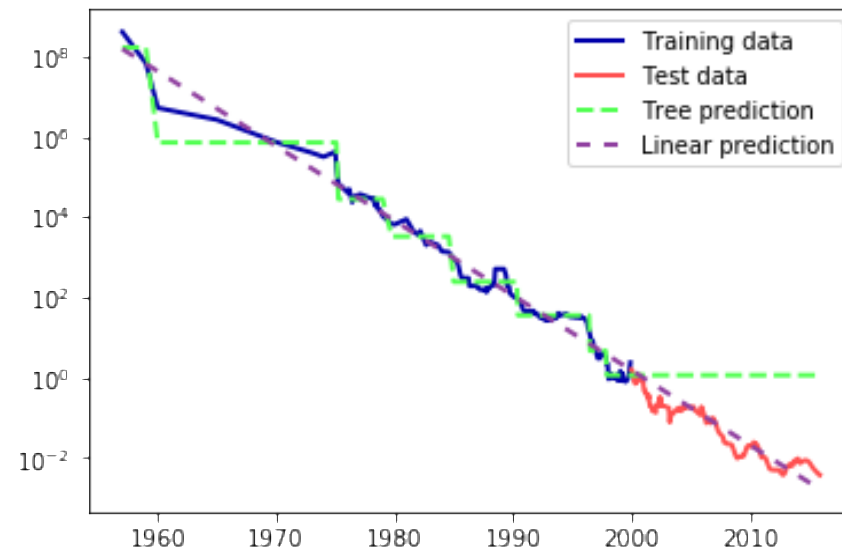
Classification par arbre de décision (DecisionTreeClassifier)

Algorithmes d'apprentissage supervisé

□ Arbres de décision

❖ Très utilisés en pratique

- pour toute nouvelle donnée X , le modèle h prédit y en fonction d'une hiérarchie de questions si/sinon, avec l'objectif d'utiliser le moins de questions possible.



Régression par arbre de décision (DecisionTreeRegressor)

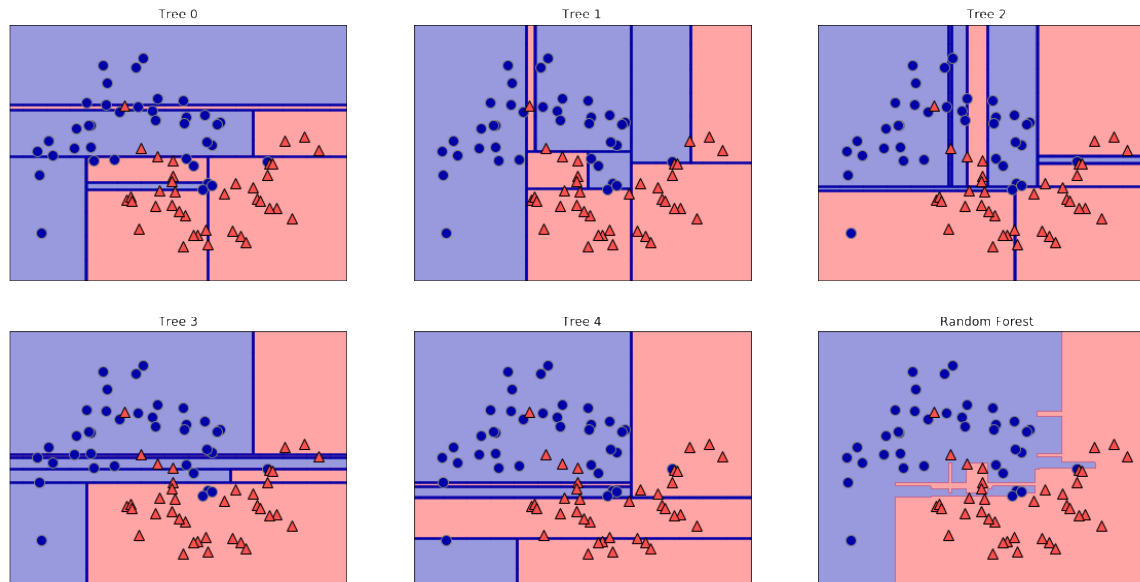
Ne peut prédire des valeurs en dehors de l'intervalle des données observées

Algorithmes d'apprentissage supervisé

❑ Ensemble d'arbres de décision

❖ Parmi les plus utilisés:

- Combinaison de modèles simples pour créer des modèles plus précis : utilisation de plusieurs arbres de décision

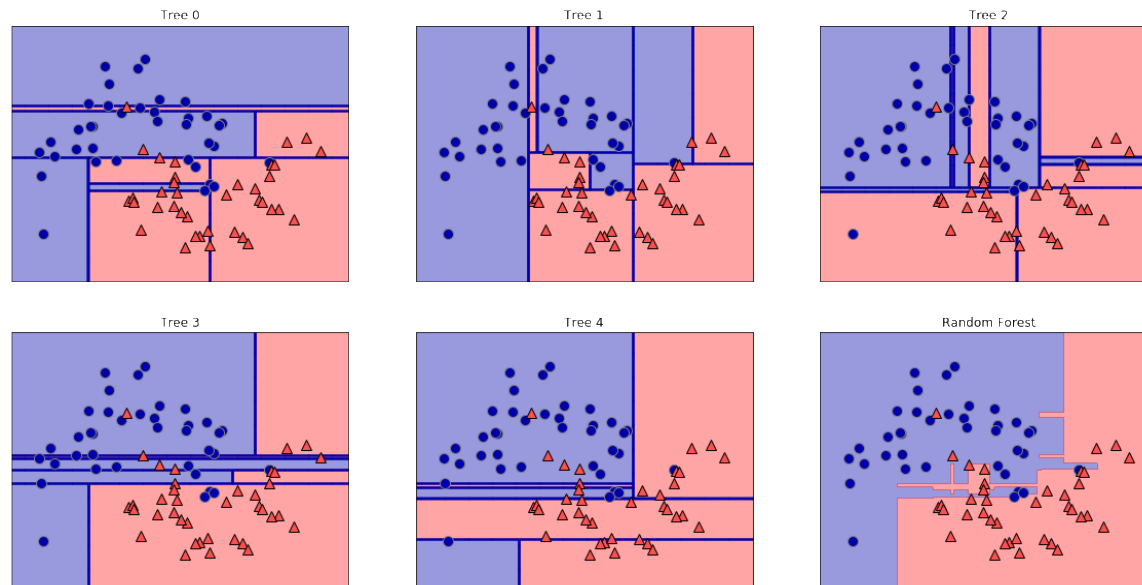


Algorithmes d'apprentissage supervisé

❑ Ensemble d'arbres de décision

❖ Parmi les plus utilisés:

- Combinaison de modèles simples pour créer des modèles plus précis : utilisation de plusieurs arbres de décision



Forêt aléatoire (RandomForest):

RandomForestClassifier

RandomForestRegressor

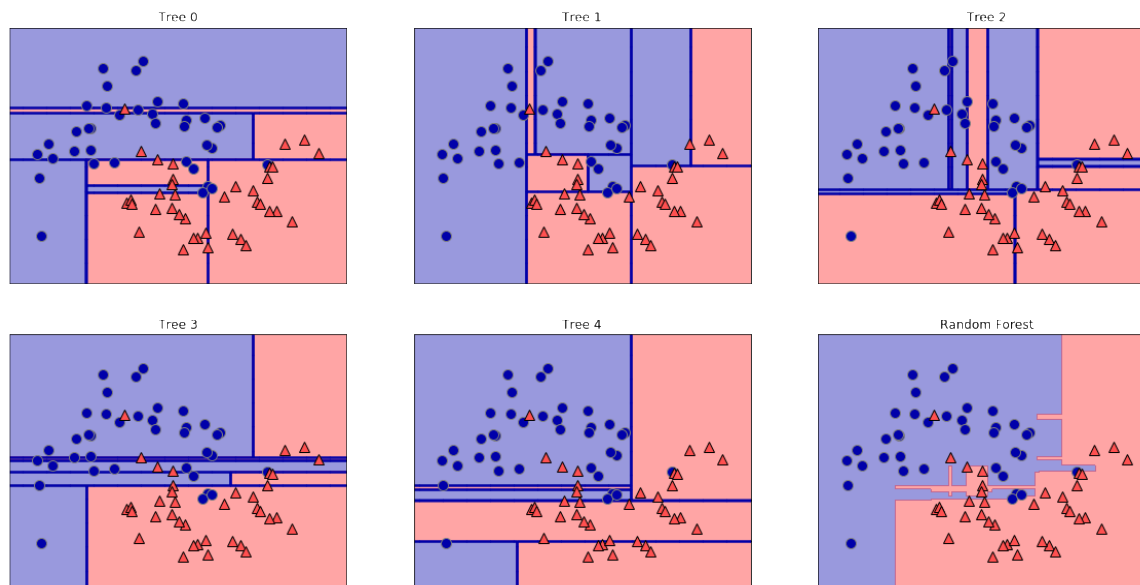
Pour la construction de chaque arbre, sélection aléatoire
d'un échantillon des données ou un échantillon des attributs

Algorithmes d'apprentissage supervisé

❑ Ensemble d'arbres de décision

❖ Parmi les plus utilisés:

- Combinaison de modèles simples pour créer des modèles plus précis : utilisation de plusieurs arbres de décision



Machines à booster les gradients (GradientBoosting):

GradientBoostingClassifier

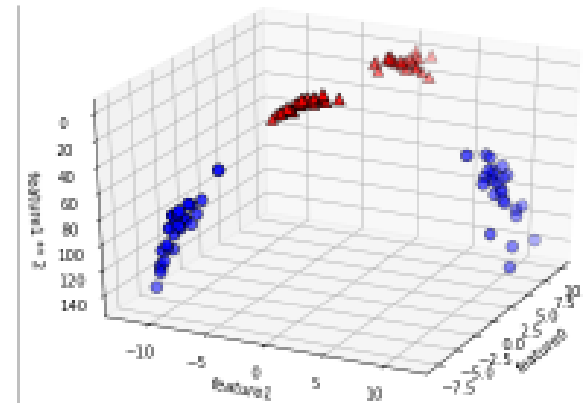
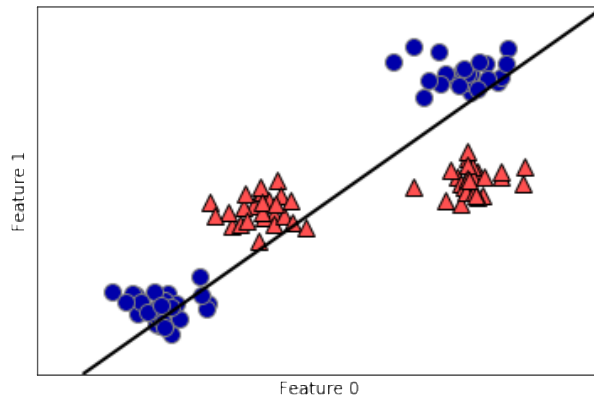
GradientBoostingRegressor

Arbres construits en série de manière itérative, de telle sorte que chaque arbre essaie de corriger les erreurs de l'arbre précédent.

Algorithmes d'apprentissage supervisé

❑ Machines à vecteur de support et à noyau

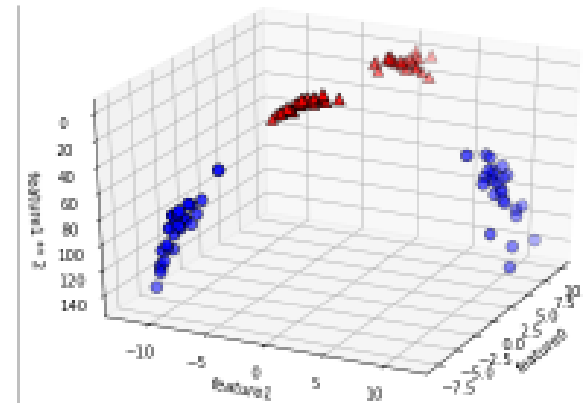
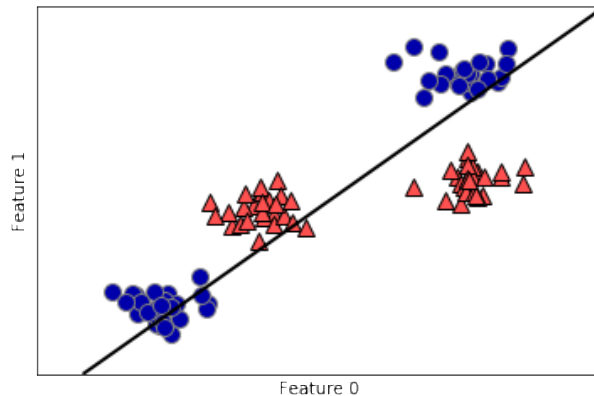
- ❖ Extension des modèles linéaires permettant de trouver des modèles plus complexes non linéaires:
 - ajout d'attributs non-linéaires correspondant à des interactions ou des polynômes des attributs initiaux



Algorithmes d'apprentissage supervisé

❑ Machines à vecteur de support et à noyau

- ❖ Extension des modèles linéaires permettant de trouver des modèles plus complexes non linéaires:
 - ajout d'attributs non-linéaires correspondant à des interactions ou des polynômes des attributs initiaux

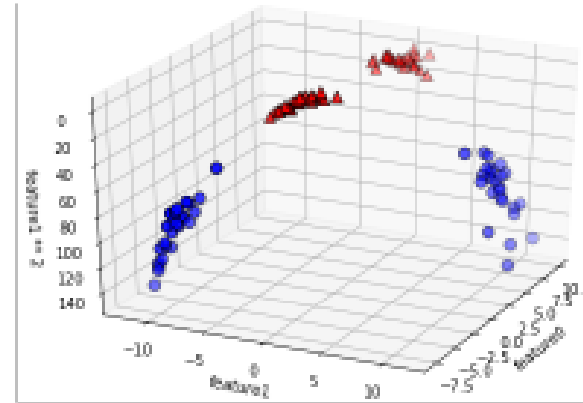
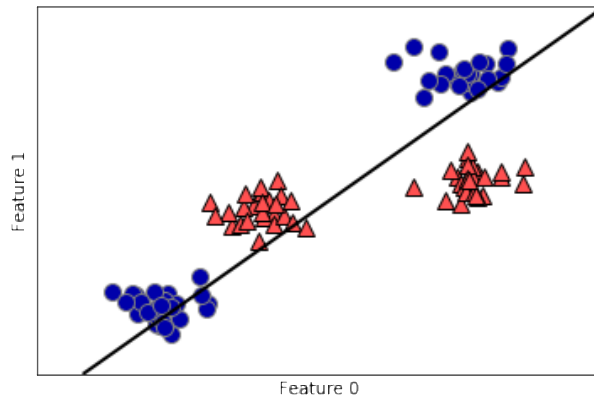


SupportVectorClassifier (SVC)
SupportVectorRegressor (SVR)

Algorithmes d'apprentissage supervisé

❑ Machines à vecteur de support et à noyau

- ❖ Extension des modèles linéaires permettant de trouver des modèles plus complexes non linéaires:
 - ajout d'attributs non-linéaires correspondant à des interactions ou des polynômes des attributs initiaux

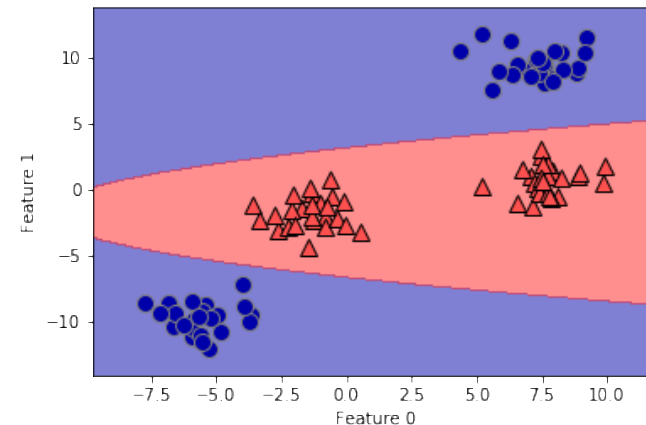
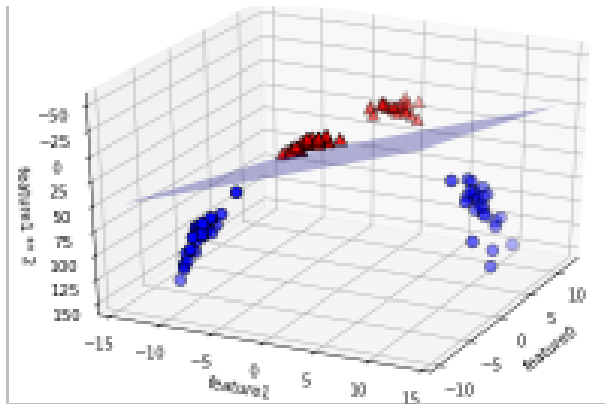


Astuce du noyau (Kernel trick) : calculer les distances entre les données (produits scalaires) sans calculer toutes les représentations possibles dans les nouveaux espaces.

Algorithmes d'apprentissage supervisé

❑ Machines à vecteur de support et à noyau

- ❖ Extension des modèles linéaires permettant de trouver des modèles plus complexes non linéaires:
 - ajout d'attributs non-linéaires correspondant à des interactions ou des polynômes des attributs initiaux



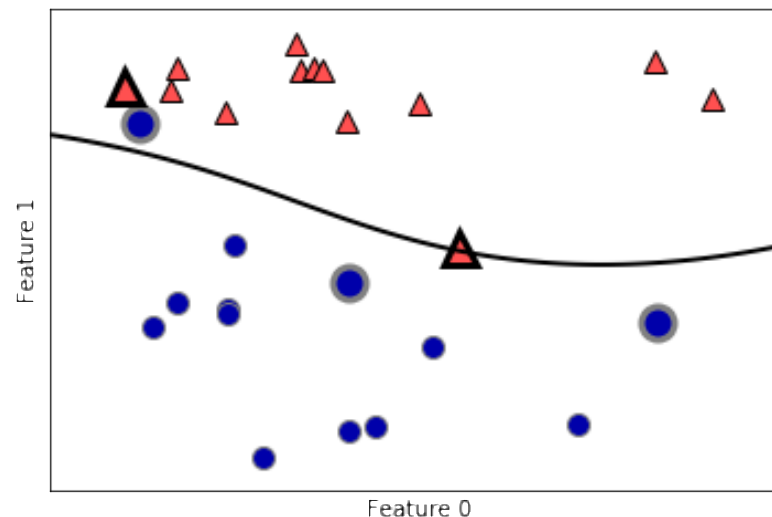
Noyau polynomial (polynomial kernel) : calcule tous les polynômes possibles
Jusqu'à un certain degré

Noyau gaussien (gaussian kernel or radial basis function, RBF) : considère tous les polynômes possibles, mais l'importance des attributs décroît lorsque leur degré croît.

Algorithmes d'apprentissage supervisé

❑ Machines à vecteur de support et à noyau

- ❖ Extension des modèles linéaires permettant de trouver des modèles plus complexes non linéaires:
 - ajout d'attributs non-linéaires correspondant à des interactions ou des polynômes des attributs initiaux



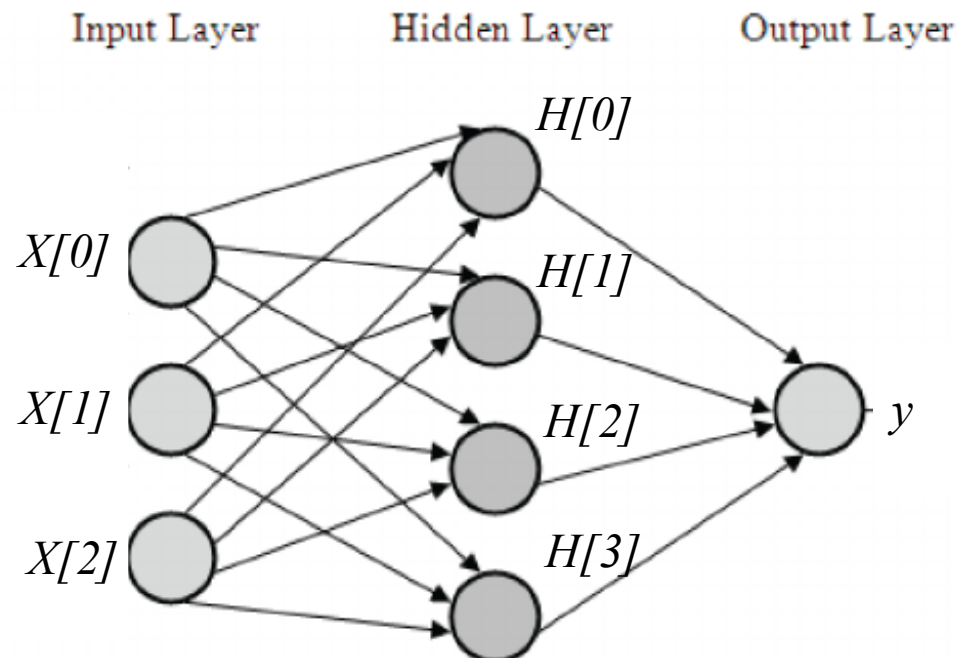
Vecteurs de support : le modèle estime et n'utilise qu'un sous-ensemble des données observées pour définir les limites de décisions : les données se situant à la séparation des classes.

Algorithmes d'apprentissage supervisé

❑ **Réseaux neuronaux** : perceptron multicouche (multilayer perceptron MLP)

❖ Extension des modèles linéaires avec plusieurs étapes (couches) de traitement.

❑ combinaisons des entrées pour obtenir une couche d'unités cachés, qui sont ensuite combinés pour obtenir les unités de la couche cachée suivante



Algorithmes d'apprentissage supervisé

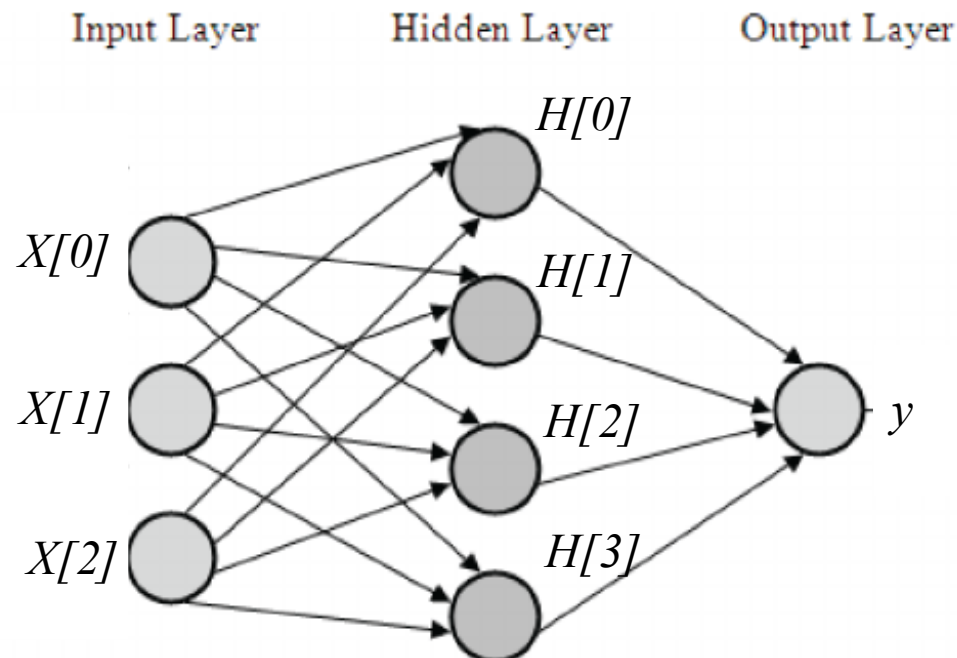
❑ Réseaux neuronaux : perceptron multicouche (multilayer perceptron MLP)

$$H(0) = coef_1[0,0]*X[0] + coef_1[1,0]*X[1] + coef_1[2,0]*X[2] + intercept_1[0]$$

$$H(1) = coef_1[0,1]*X[0] + coef_1[1,1]*X[1] + coef_1[2,1]*X[2] + intercept_1[1]$$

$$H(2) = coef_1[0,2]*X[0] + coef_1[1,2]*X[1] + coef_1[2,2]*X[2] + intercept_1[2]$$

$$y = coef_2[0]*H[0] + coef_2[1]*H[1] + coef_2[2]*H[2] + coef_2[3]*H[3] + intercept_2$$



Algorithmes d'apprentissage supervisé

❑ Réseaux neuronaux : perceptron multicouche (multilayer perceptron MLP)

$$H(0) = \text{relu}(\text{coef_1}[0,0]*X[0] + \text{coef_1}[1,0]*X[1] + \text{coef_1}[2,0]*X[2] + \text{intercept_1}[0]))$$

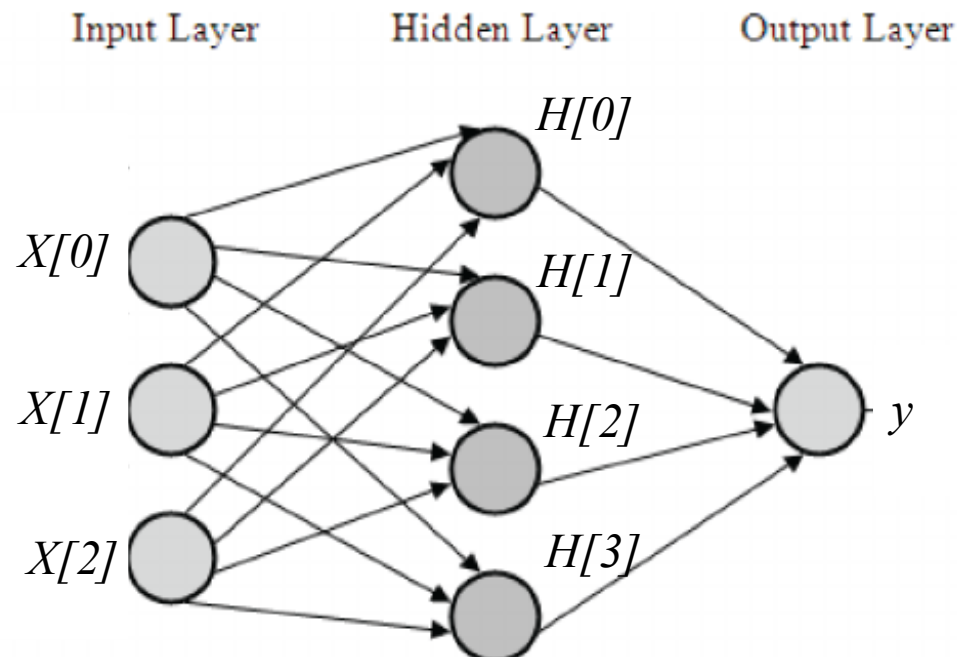
$$H(1) = \text{relu}(\text{coef_1}[0,1]*X[0] + \text{coef_1}[1,1]*X[1] + \text{coef_1}[2,1]*X[2] + \text{intercept_1}[1]))$$

$$H(2) = \text{relu}(\text{coef_1}[0,2]*X[0] + \text{coef_1}[1,2]*X[1] + \text{coef_1}[2,2]*X[2] + \text{intercept_1}[2]))$$

$$y = \text{coef_2}[0]*H[0] + \text{coef_2}[1]*H[1] + \text{coef_2}[2]*H[2] + \text{coef_2}[3]*H[3] + \text{intercept_2}$$

Application d'une fonction non-linéaire aux unités cachées:

- rectified linear unit (relu) (par défaut)
- tangens hyperbolicus (tanh)



Algorithmes d'apprentissage supervisé

❑ Réseaux neuronaux : perceptron multicouche (multilayer perceptron MLP)

$$H(0) = \text{relu}(\text{coef_1}[0,0]*X[0] + \text{coef_1}[1,0]*X[1] + \text{coef_1}[2,0]*X[2] + \text{intercept_1}[0]))$$

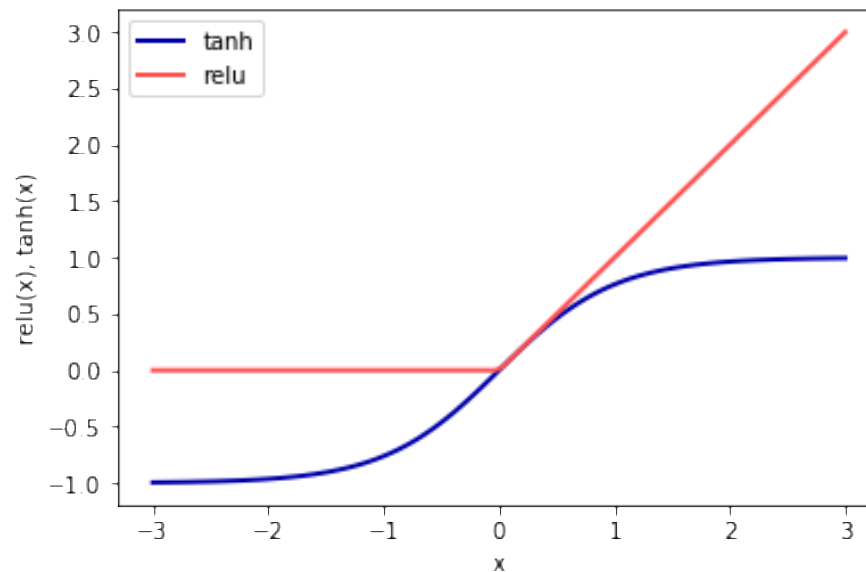
$$H(1) = \text{relu}(\text{coef_1}[0,1]*X[0] + \text{coef_1}[1,1]*X[1] + \text{coef_1}[2,1]*X[2] + \text{intercept_1}[1]))$$

$$H(2) = \text{relu}(\text{coef_1}[0,2]*X[0] + \text{coef_1}[1,2]*X[1] + \text{coef_1}[2,2]*X[2] + \text{intercept_1}[2]))$$

$$y = \text{coef_2}[0]*H[0] + \text{coef_2}[1]*H[1] + \text{coef_2}[2]*H[2] + \text{coef_2}[3]*H[3] + \text{intercept_2}$$

Application d'une fonction non-linéaire aux unités cachées:

- rectified linear unit (relu) (par défaut)
- tangens hyperbolicus (tanh)



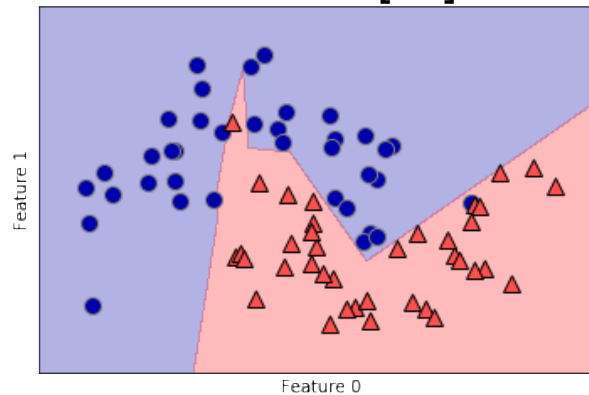
Algorithmes d'apprentissage supervisé

- ❑ **Réseaux neuronaux** : perceptron multicouche (multilayer perceptron MLP)

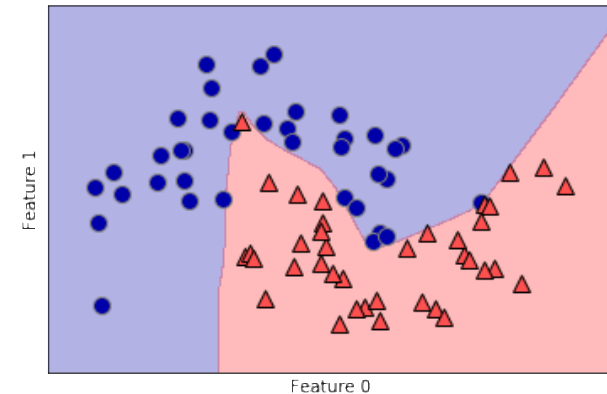
MLPClassifier
MLPRegressor

Possible d'appliquer
une régularisation L2

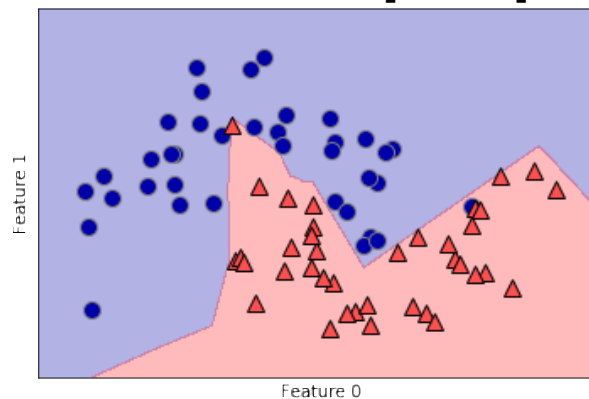
Couches = [10]



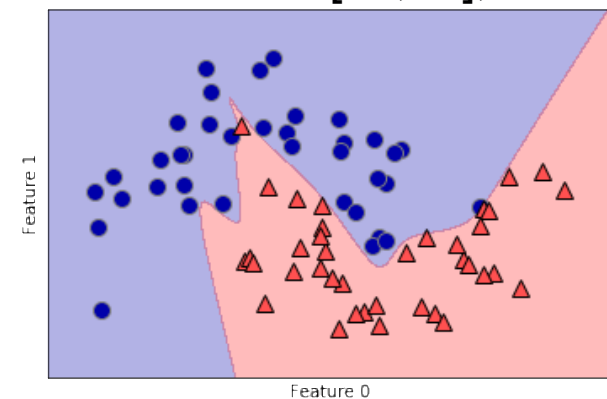
Couches = [100]



Couches = [10,10]



Couches = [10,10], tanh



Algorithmes d'apprentissage non-supervisé

❑ Réduction de dimension

❖ Méthodes de décomposition

- Analyse en composantes principales (PCA)
- Factorisation par matrices non-négatives (NMF)

❖ Réduction dans un espace préservant les distances (Manifold)

❑ Méthodes de clustering

❖ Méthodes par partition

❖ Clustering hiérarchique

❖ Méthodes basées sur la densité

Algorithmes d'apprentissage non-supervisé

❑ Réduction de dimension : Méthodes de décomposition

- ❖ Transformation des attributs initiaux en un ensemble de nouveaux attributs expliquant (séparant) au mieux les données
- ❖ Sélection des composantes les plus importantes (informatives) et représentation dans le nouvel espace

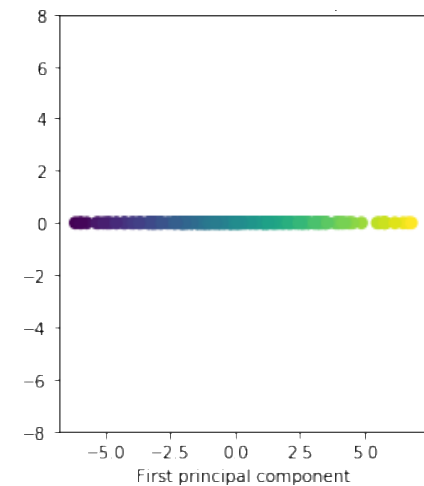
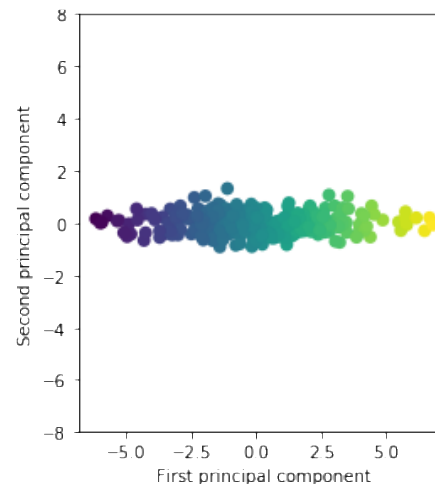
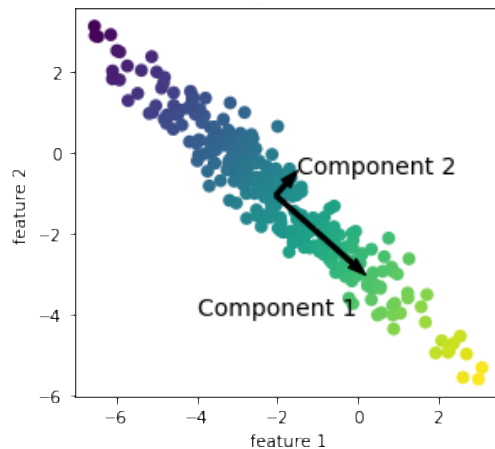
Algorithmes d'apprentissage non-supervisé

❑ Méthodes de décomposition

❖ Analyse en composantes principales (PCA)

- Transformation des attributs initiaux en un ensemble de nouveaux attributs orthogonaux (non-corrélés) (composantes principales)

PCA : explique au maximum la variance des données



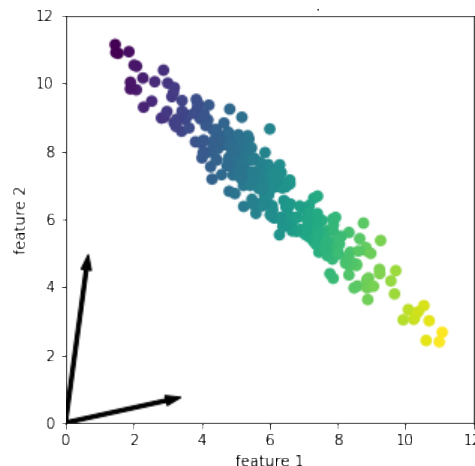
Algorithmes d'apprentissage non-supervisé

❑ Méthodes de décomposition

❖ Factorisation par matrices non-négatives (NMF)

Transformation des attributs initiaux en un ensemble de nouveaux attributs tels que les coordonnées d'un objet dans le nouvel espace sont toutes non-négatives

NMF : représente les données comme des sommes d'attributs non-négatifs. Ex: données obtenues par addition de plusieurs sources indépendantes.



Algorithmes d'apprentissage non-supervisé

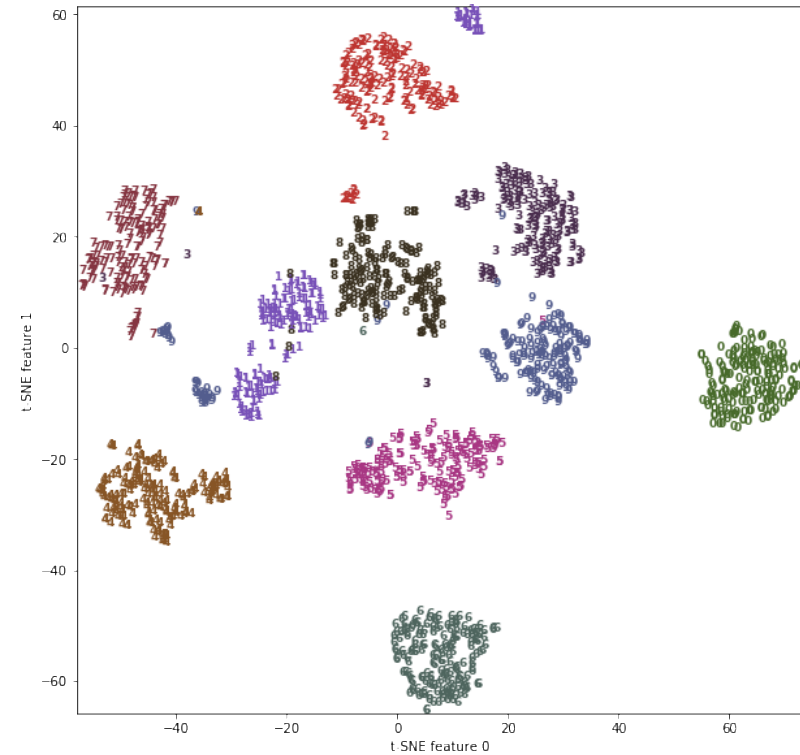
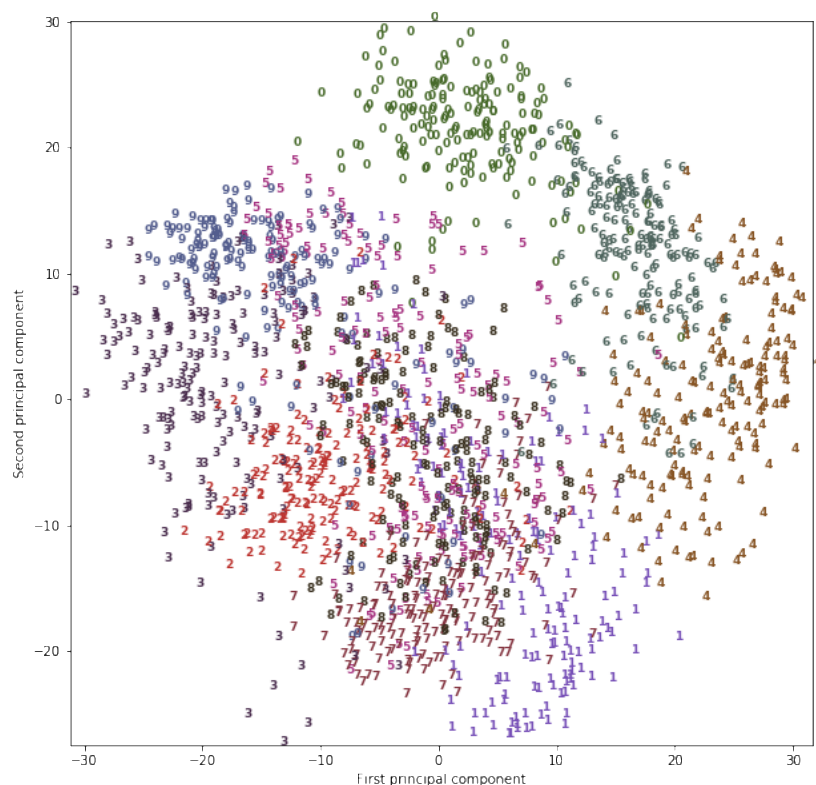
❑ Réduction de dimension : Réduction dans un espace préservant les distances (Manifold)

- ❖ Transformation des attributs initiaux en un ensemble de nouveaux attributs (généralement 2 attributs) tels que la représentation dans le nouvel espace préserve les distances relatives

Algorithmes d'apprentissage non-supervisé

❑ Réduction de dimension : Réduction dans un espace préservant les distances (Manifold)

❖ **t-distributed stochastic neighbor embedding (t-SNE) :**
Transformation des attributs initiaux en 2 nouveaux attributs tels que les objets qui étaient proches dans l'espace initial reste proches dans le nouvel espace



Algorithmes d'apprentissage non-supervisé

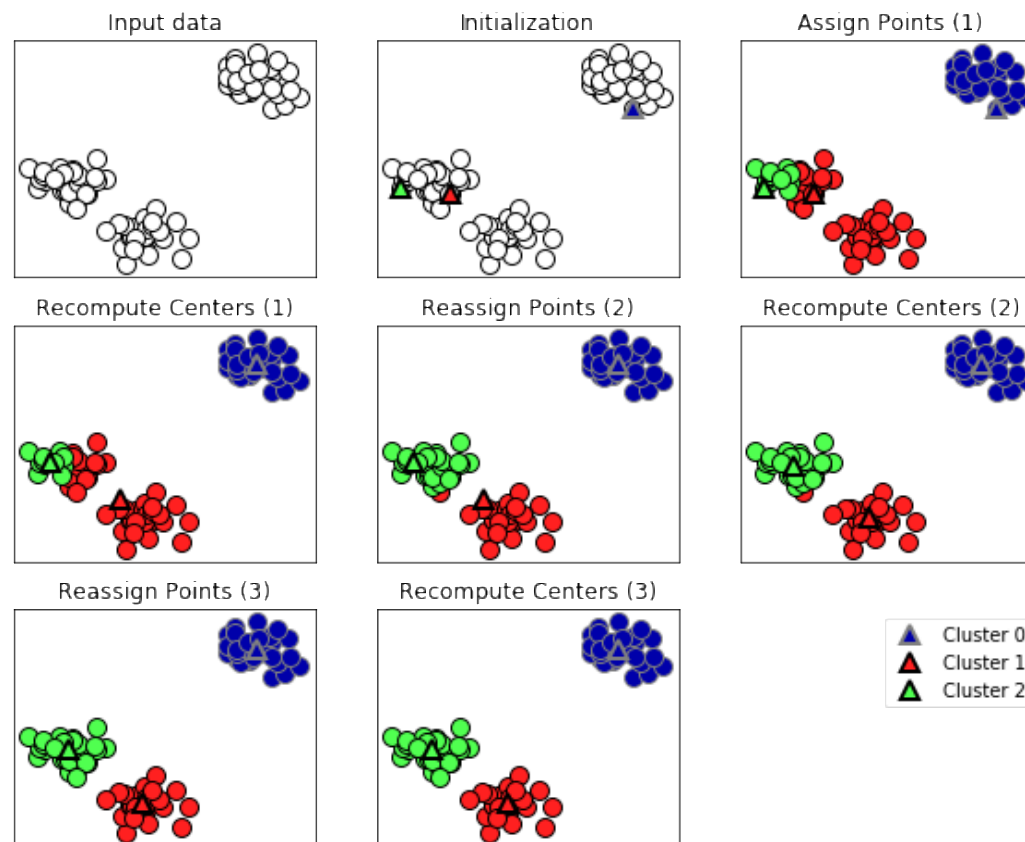
❑ Méthodes de clustering : méthodes par partition

- ❖ Trouver une partition des données qui optimise une certaine fonction objective. Ex: maximise la moyenne des distances inter-cluster et minimise la moyenne des distances intra-cluster

Algorithmes d'apprentissage non-supervisé

❑ Méthodes de clustering : méthodes par partition

- ❖ **K-Means** : cherche des centres de clusters représentant les différentes parties des données :
recalcule de façon itérative les centres des clusters et les membres des clusters jusqu'à atteindre un clustering stable.

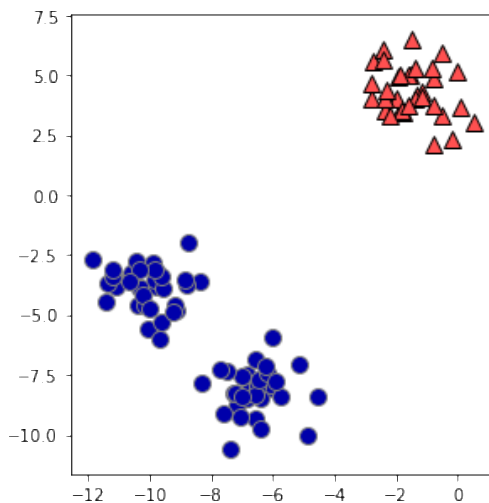


Algorithmes d'apprentissage non-supervisé

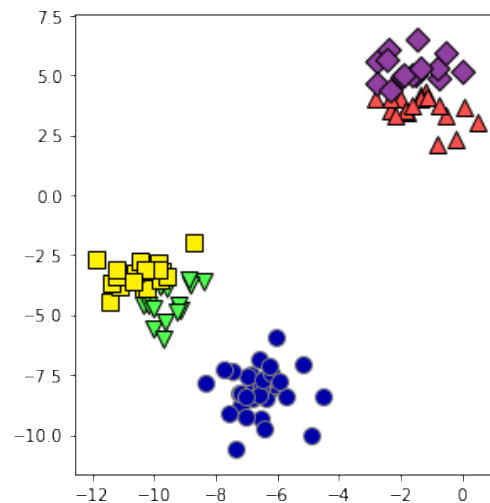
❑ Méthodes de clustering : méthodes par partition

❖ **K-Means** : cherche des centres de clusters représentant les différentes parties des données :
recalcule de façon itérative les centres des clusters et les membres des clusters jusqu'à atteindre un clustering stable.

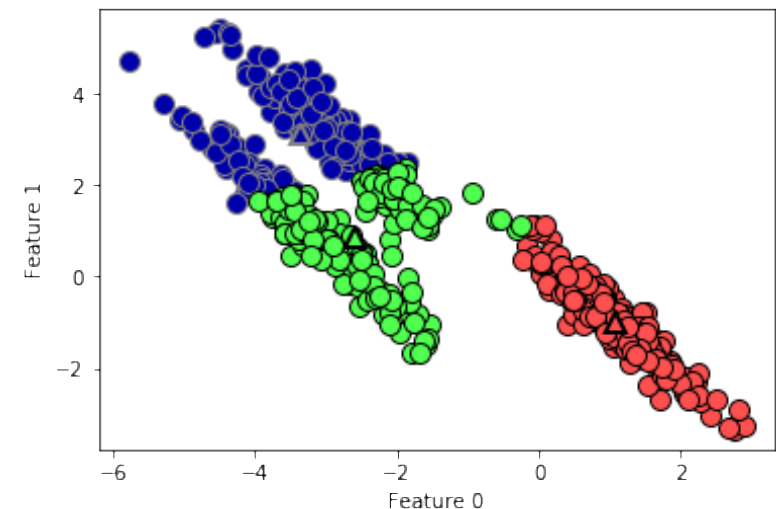
K = 2



K = 5



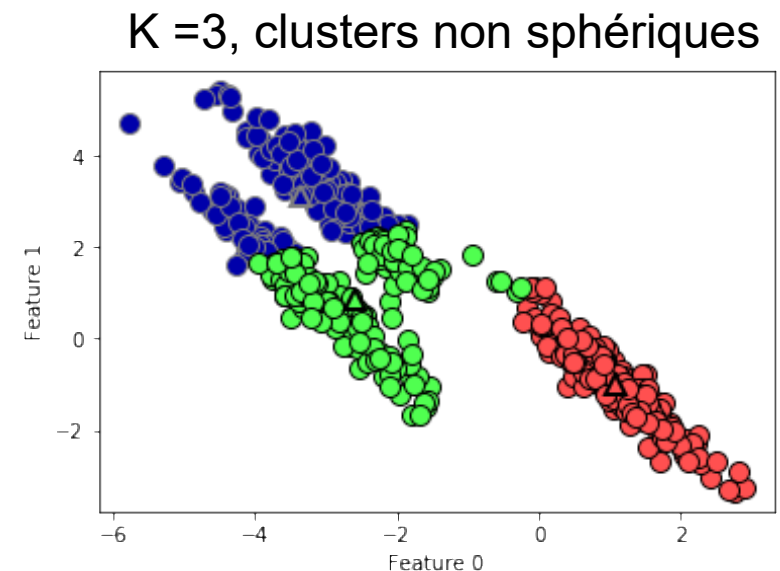
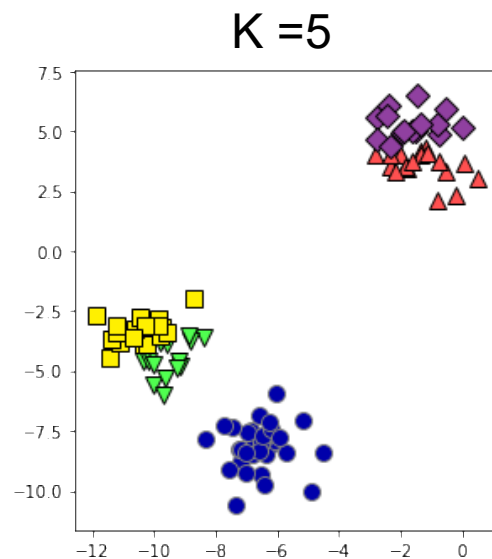
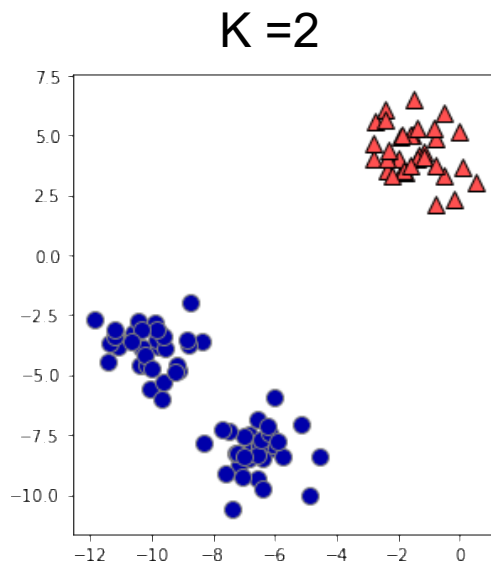
K = 3, clusters non sphériques



Algorithmes d'apprentissage non-supervisé

❑ Méthodes de clustering : méthodes par partition

- ❖ **K-Means** : cherche des centres de clusters représentant les différentes parties des données : recalcule de façon itérative les centres des clusters et les membres des clusters jusqu'à atteindre un clustering stable.

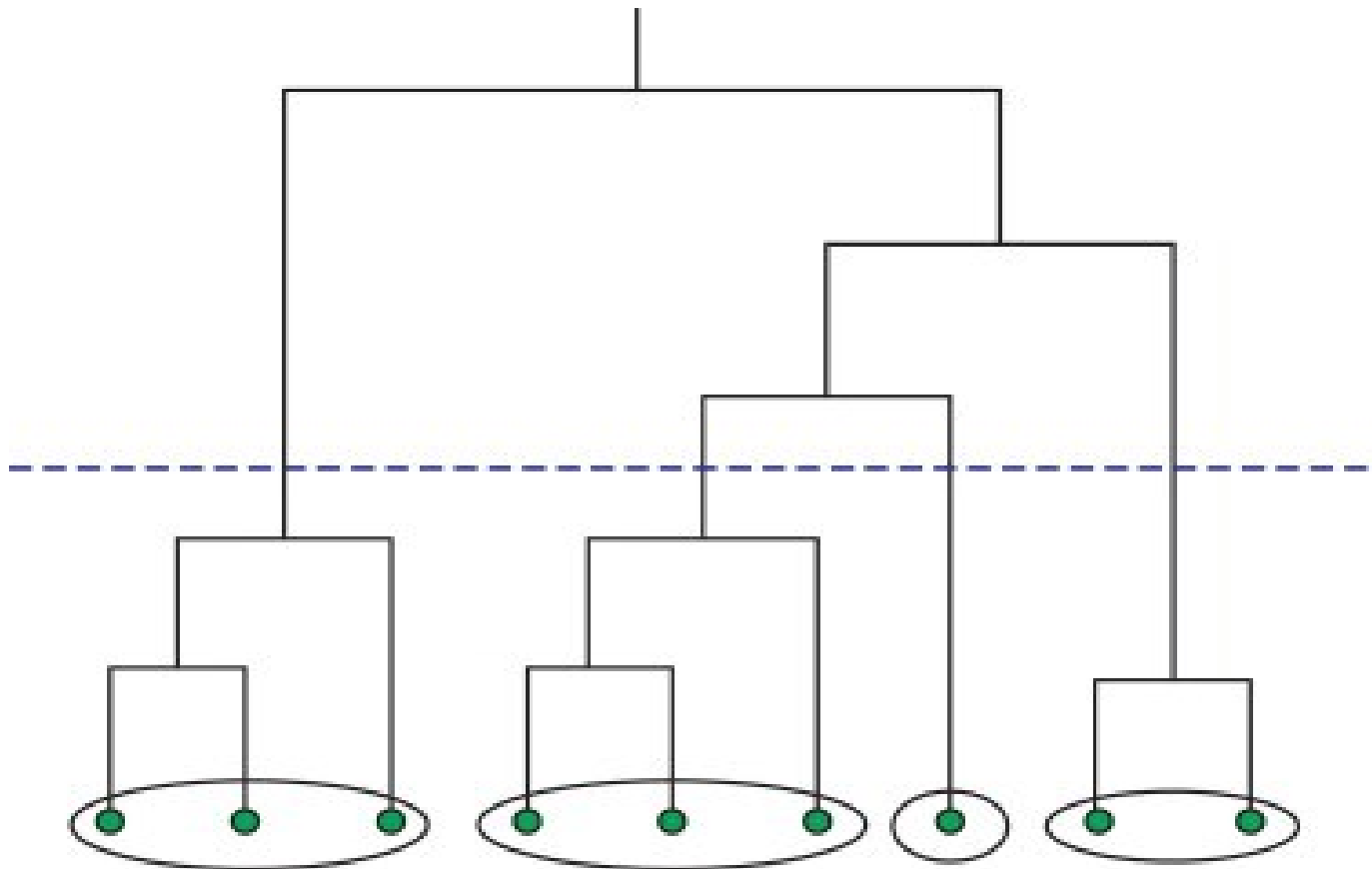


- ❖ **Fuzzy C-means** : même principe que K-Means mais produit un clustering « flou » : chaque donnée a une probabilité d'appartenance à chaque cluster.

Algorithmes d'apprentissage non-supervisé

❑ Méthodes de clustering : clustering hiérarchique

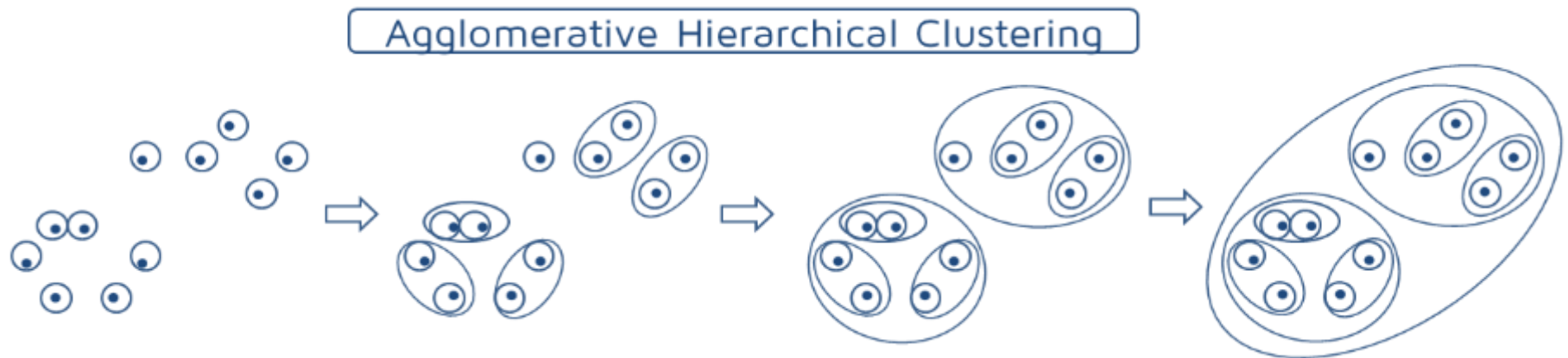
❖ Cherche une hiérarchie de clusters emboîtés



Algorithmes d'apprentissage non-supervisé

❑ Méthodes de clustering : clustering hiérarchique

- ❖ **Méthodes agglomératives (AgglomerativeClustering) :**
Initialement chaque donnée est un cluster, puis de façon itérative les deux clusters les plus proches sont fusionnés jusqu'à descendre au nombre de cluster requis.

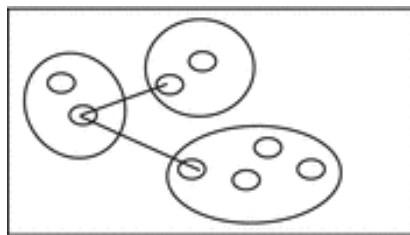


Algorithmes d'apprentissage non-supervisé

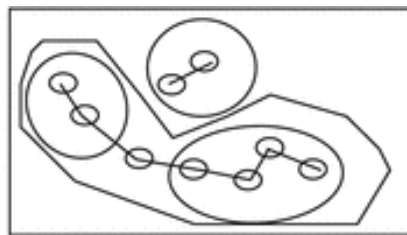
❑ Méthodes de clustering : clustering hiérarchique

❖ Méthodes agglomératives (AgglomerativeClustering) :

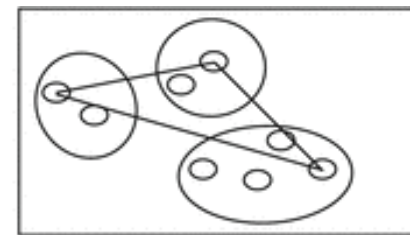
Initialement chaque donnée est un cluster, puis de façon itérative les deux clusters les plus proches sont fusionnés jusqu'à descendre au nombre de cluster requis.



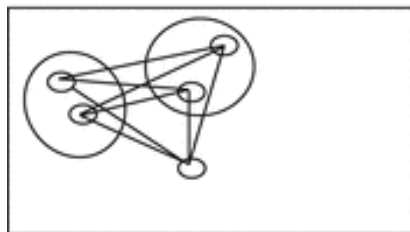
Single linkage (1)



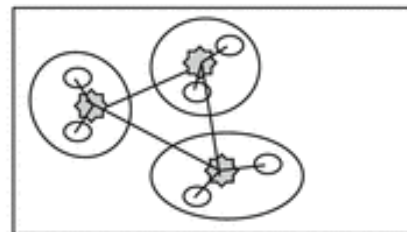
Single linkage (2): *chaining*



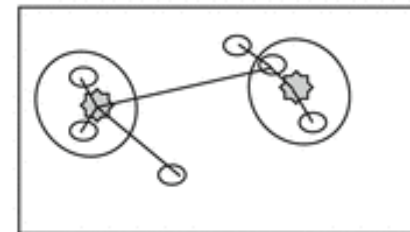
Complete linkage



Average linkage



Centroid linkage



Ward linkage

Ward : variance minimum
entre les deux clusters
fusionnés

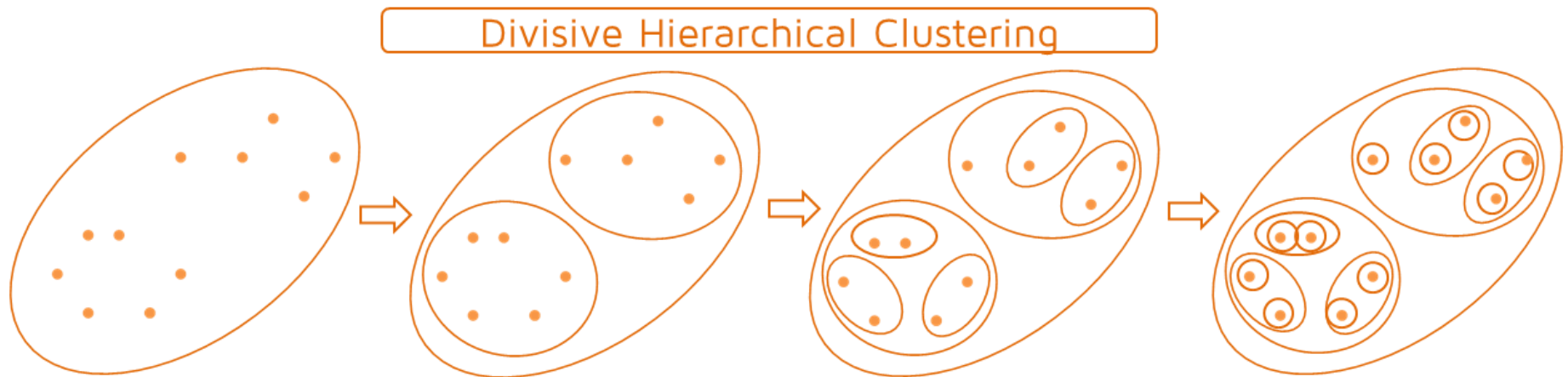
Average : distance moyenne
minimum entre les deux clusters
fusionnés

Complete : distance maximale
minimum entre les deux clusters
fusionnés

Algorithmes d'apprentissage non-supervisé

❑ Méthodes de clustering : clustering hiérarchique

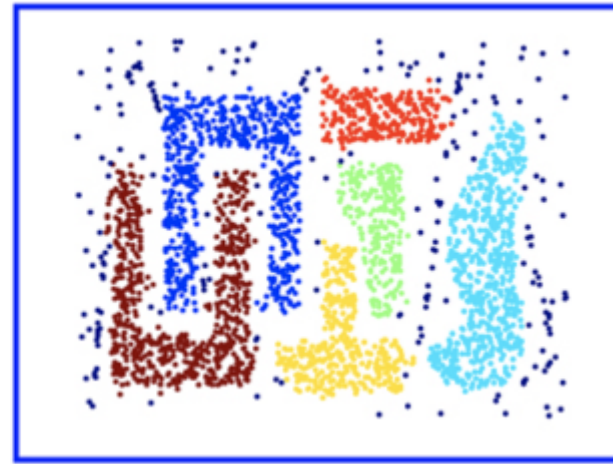
❖ **Méthodes divisives** : Initialement l'ensemble des données est un cluster, puis de façon itérative on choisit un cluster et on le divise jusqu'à atteindre le nombre de cluster requis



Algorithmes d'apprentissage non-supervisé

❑ Méthodes de clustering : méthodes basées sur la densité

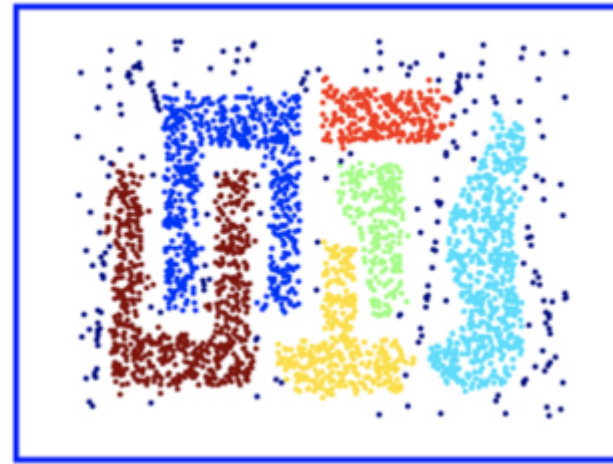
❖ Identifier des régions de grandes densité (clusters)



Algorithmes d'apprentissage non-supervisé

❑ Méthodes de clustering : méthodes basées sur la densité

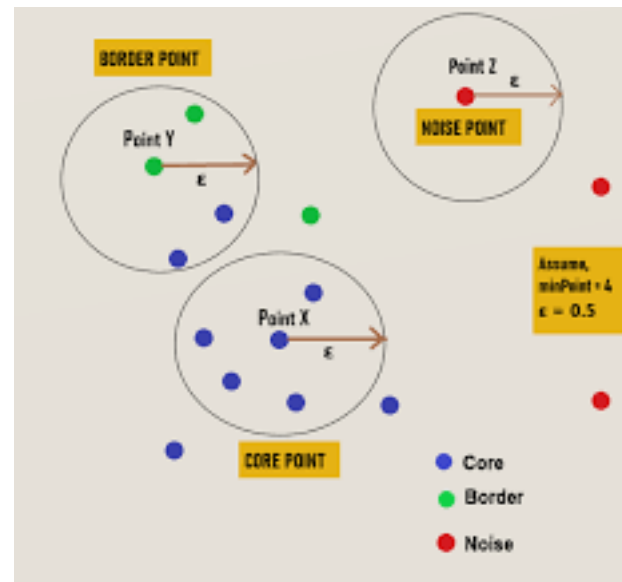
- ❖ Identifier des régions de grandes densité (clusters)
- ❖ Permet de trouver des clusters de forme non-sphérique



Algorithmes d'apprentissage non-supervisé

❑ Méthodes de clustering : méthodes basées sur la densité

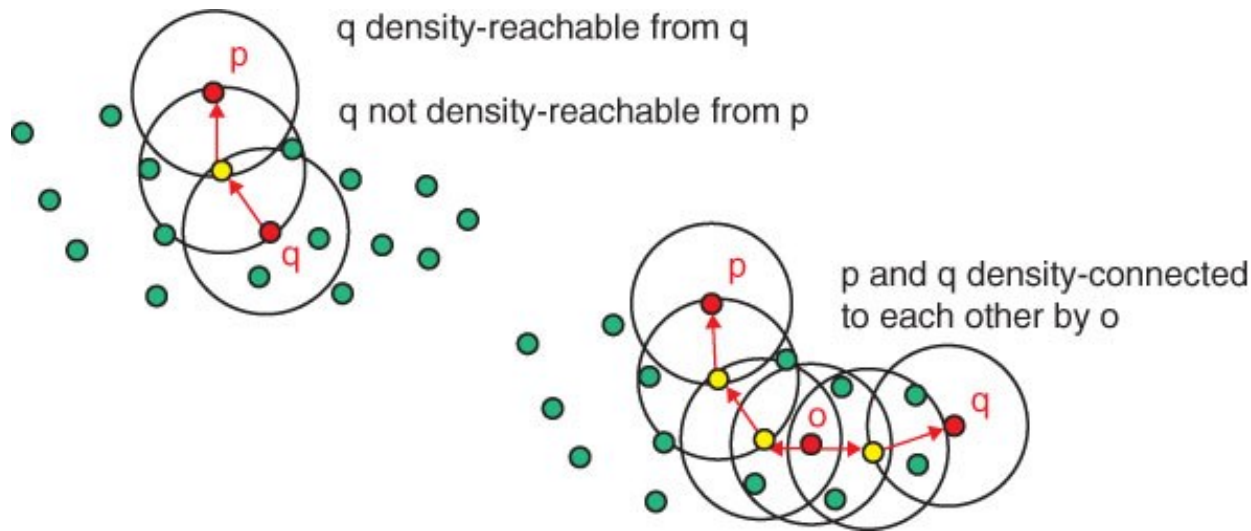
- ❖ **Density-based spatial clustering of applications with noise (DBSCAN)** : Identifie les points comme « core » (ancree), « border » (frontière), ou « noise » (bruit) suivant la densité de leur voisinage, et suivant qu'ils ont des points denses ou non dans leur voisinage



Algorithmes d'apprentissage non-supervisé

❑ Méthodes de clustering : méthodes basées sur la densité

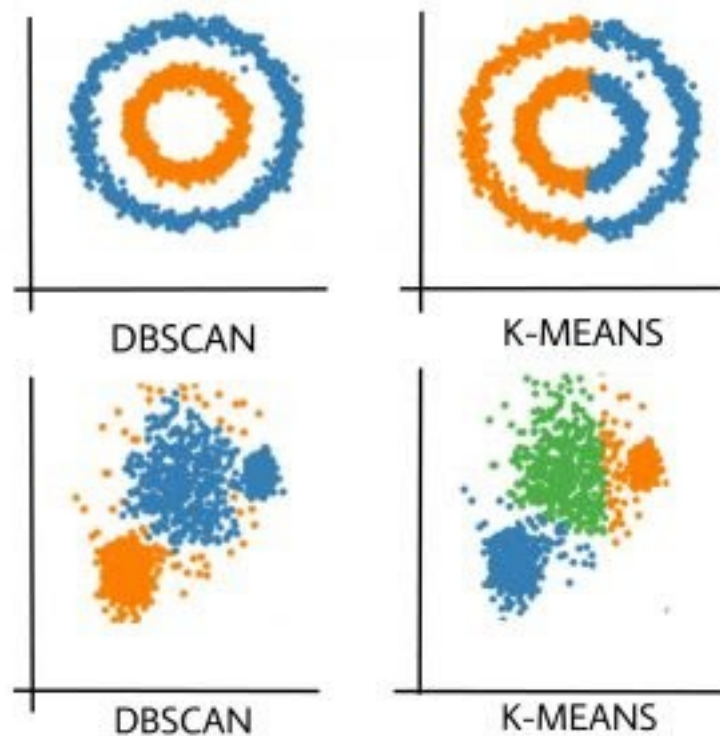
- ❖ **Density-based spatial clustering of applications with noise (DBSCAN)** : Identifie les points comme « core » (ancree), « border » (frontière), ou « noise » (bruit) suivant la densité de leur voisinage, et suivant qu'ils ont des points denses ou non dans leur voisinage



Algorithmes d'apprentissage non-supervisé

❑ Méthodes de clustering : méthodes basées sur la densité

- ❖ **Density-based spatial clustering of applications with noise (DBSCAN)** : Identifie les points comme « core » (ancree), « border » (frontière), ou « noise » (bruit) suivant la densité de leur voisinage, et suivant qu'ils ont des points denses ou non dans leur voisinage



Partie II : Pratique

Python et bibliothèques

- ❑ **Python 3**: intègre des fonctionnalités générales et de nombreuses bibliothèques spécifiques pour l'analyse de données
Installer Python (<https://www.python.org/>) et pip (<https://pip.pypa.io/en/stable/>)
- ❑ **scikit-learn**: implémentation de nombreux algorithmes de forage de données supervisé et non-supervisé, avec une documentation bien fournies (<http://scikit-learn.org/stable>):
- ❑ **Jupyter Notebook** : permet l'interaction directe avec du code via un navigateur (<https://jupyter.org/>)

Python et bibliothèques

- ❑ **numpy et scipy**: pour des calculs scientifiques en Python: calcul matricielle, opérations d'algèbre linéaire, génération de nombres aléatoires, fonctions d'optimisation, distributions statistiques,...
- ❑ **pandas**: : bibliothèque pour la manipulation de données
- ❑ **pillow**: bibliothèque pour la manipulation de données d'imagerie
- ❑ **matplotlib**: bibliothèque pour la visualisation de données
- ❑ **seaborn**: bibliothèque pour la visualisation de données statistiques

Introduction à scikit-learn

❑ **Fichier notebook:** `Introduction_scikit_learn.ipynb`

Travail dirigé 1

❑ **Fichier notebook:** Exercice_1.ipynb

Références

- [1] PEDREGOSA et al. : *Scikit-learn : Machine Learning in Python*. JMLR 12, pp. 2825-2830. (User guide and API : [https ://scikit-learn.org/stable/](https://scikit-learn.org/stable/)), 2011.

- [2] Jiawei HAN, Micheline KAMBER, Jian PEI. *DataMining: Concepts and Techniques (Third edition)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2011.

- [3] Adreas C. MÜLLER et Sarah GUIDO : *Introduction to Machine Learning with Python*. O'Reilly Media, Inc., Sebastopol, CA, 2017.

- [4] Pang-Ning TAN, Michael STEINBACH et Vipin KUMAR : *Introduction to Data Mining, (Second Edition)*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2018.