Apprentissage supervisé

Partie 1

Analyse de données et Classification 2
ENSEEIHT - 3ème année Sciences du Numérique

Contact :
Sandrine.Mouysset@irit.fr
sandrine.mouysset@toulouse-inp.fr

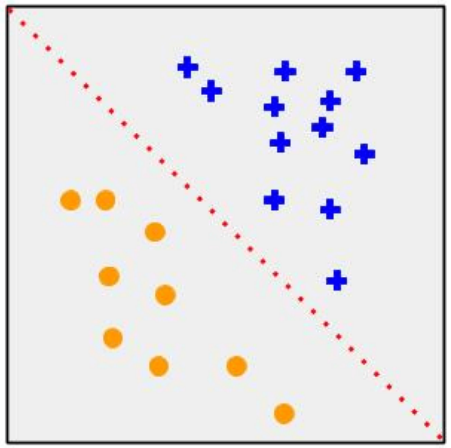
Chaîne d'analyse des données

Modèles supervisés :

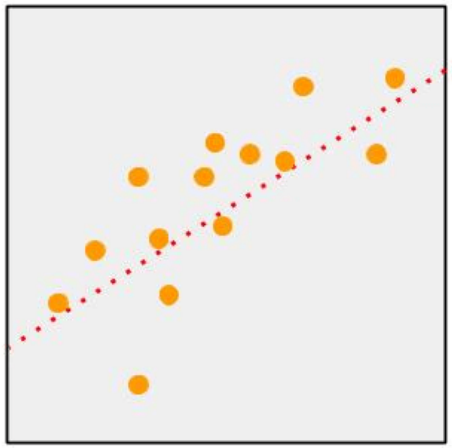
On dispose d'une **base d'apprentissage**, sous ensemble de données "étiquetées" par des experts du type

X	$X^1 \ldots X^i \ldots X^m$	Classe
1 ⋮ i ⋮ n	Caractéristiques variables	S variables nominales

2 principaux types d'apprentissage supervisé :



Classification



Regression

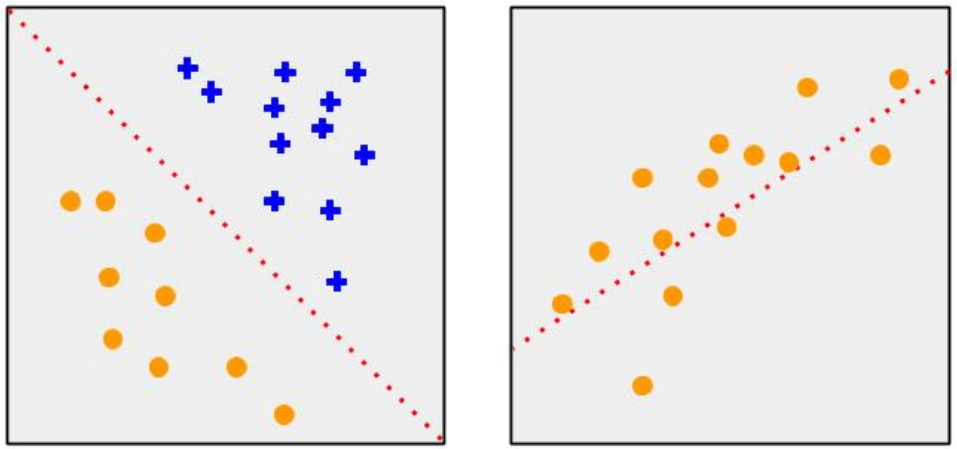
Classification : Assigner une catégorie à chaque observation :

- Les catégories sont discrètes
- La cible est un indice de classe : $y \in \{0, .., K-1\}$
- *Exemple :* reconnaissance de chiffres manuscrits :
  - x : vecteur ou matrice des intensités des pixels de l'image
  - t : identité du chiffre

Régression : Prédire une valeur réelle à chaque observation :

- les catégories sont continues
- la cible est un nombre réel $y \in \mathbb{R}$
- *Exemple :* prédire le cours d'une action
  - x : vecteur contenant l'information sur l'activité économique
  - y : valeur de l'action le lendemain

2 principaux types d'apprentissage supervisé :



Classification Regression

Approches par modèles supervisés :

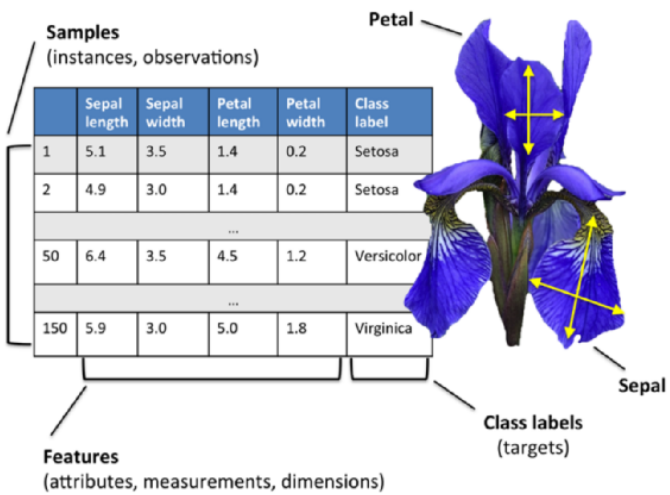
- Arbres de Décision
- Apprentissage d'ensemble : Forêts aléatoires
- Réseaux de neurones
- SVM

Méthodes d'évaluation :

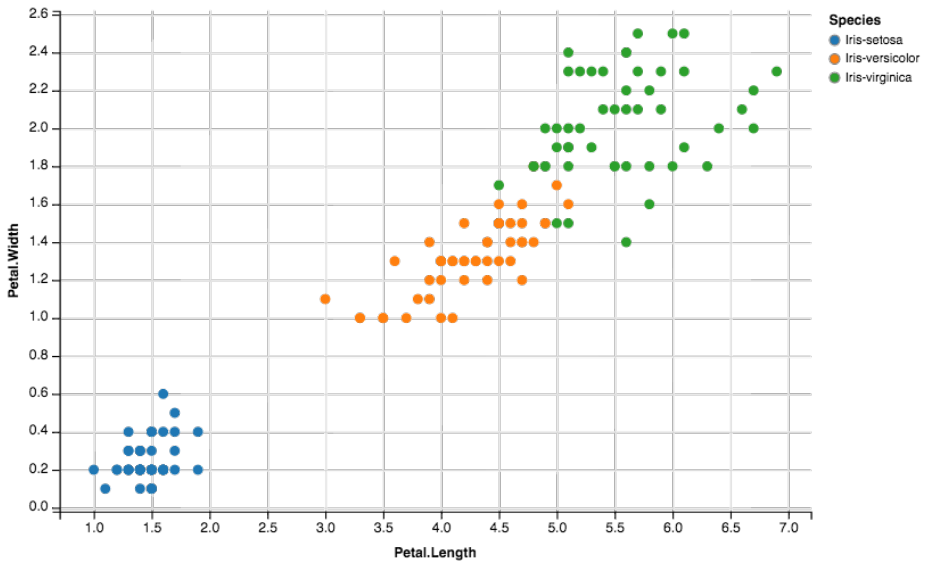
- Validation croisée
- Matrice de confusion
- Precision, Rappel, F-mesure
- Courbe ROC

Arbres de décision :

- une des structures de données majeures de l'apprentissage statistique ;
- adapatable à toute nature de données ;
- fonctionnement reposant sur des heuristiques qui, tout en satisfaisant l'intuition, donnent des résultats remarquables en pratique (notamment lorsqu'ils sont utilisés en " forêts aléatoires") ;
- structure arborescente lisible contrairement à d'autres approches où le prédicteur construit est une "boîte noire".



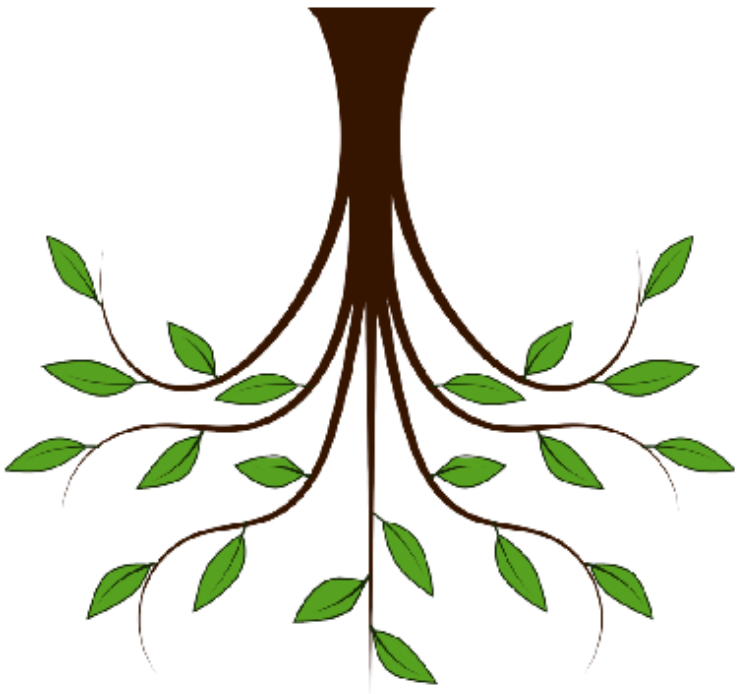
	Sepal length	Sepal width	Petal length	Petal width	Class label
1	5.1	3.5	1.4	0.2	Setosa
2	4.9	3.0	1.4	0.2	Setosa
...					
50	6.4	3.5	4.5	1.2	Versicolor
...					
150	5.9	3.0	5.0	1.8	Virginica

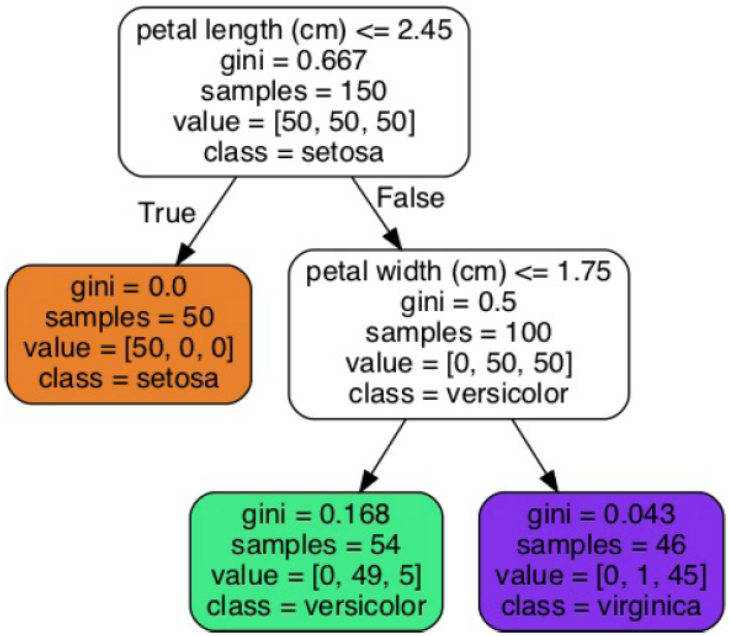


Exemple sur les données *Iris*

150 données décrites par 4 mesures représentant 3 espèces

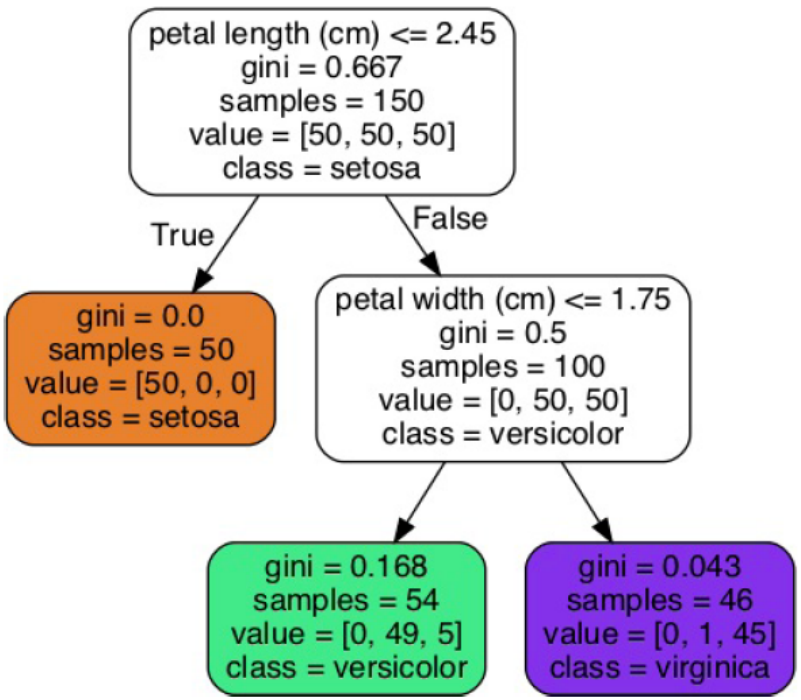
Exemple : Arbre de décision sur les données *Iris*





Résultat pour une profondeur 2

Exemple : Arbre de décision sur les données *Iris*



Résultat pour une profondeur 2

- Nom et valeur seuil de la variable discriminante ;
- *Sample d'un noeud* : nombre d'observations d'entraînement passées par ce noeud ;
- *Value d'un noeud :* répartition des observations sur les classes ;
- *Indice de Gini :* mesure d'impureté du noeud ;
- *Class :* classe majoritaire du noeud.

Mesures d'impureté

- Indice de Gini d'un noeud i comportant n éléments :

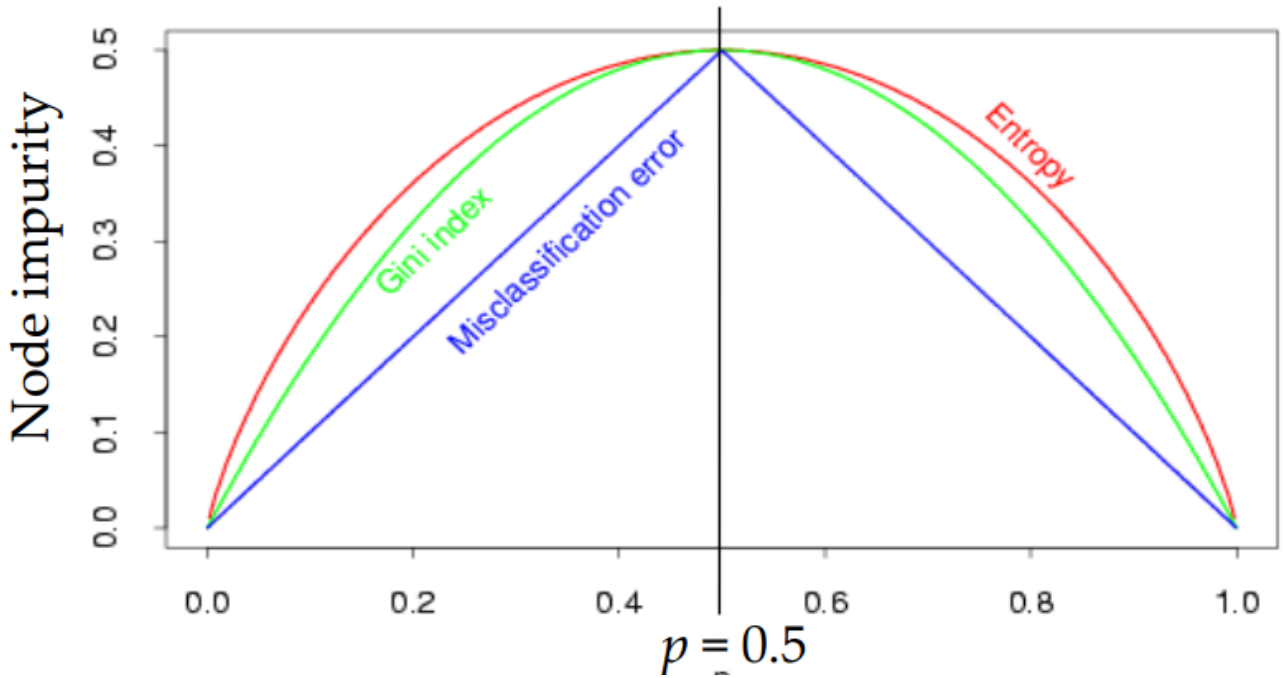
$$G_i = 1 - \sum_{k=1}^{n} p_{i,k}^2$$

- Entropie d'un noeud i :

$$H_i = - \sum_{k=1,\ p_{i,k} \neq 0}^{n} p_{i,k} log(p_{i,k})$$

où $p_{i,k}$ est le pourcentage d'observations de la classe k parmi toutes les observations d'entraînement dans le $i^{\text{ème}}$ noeud.

Mesures d'impureté

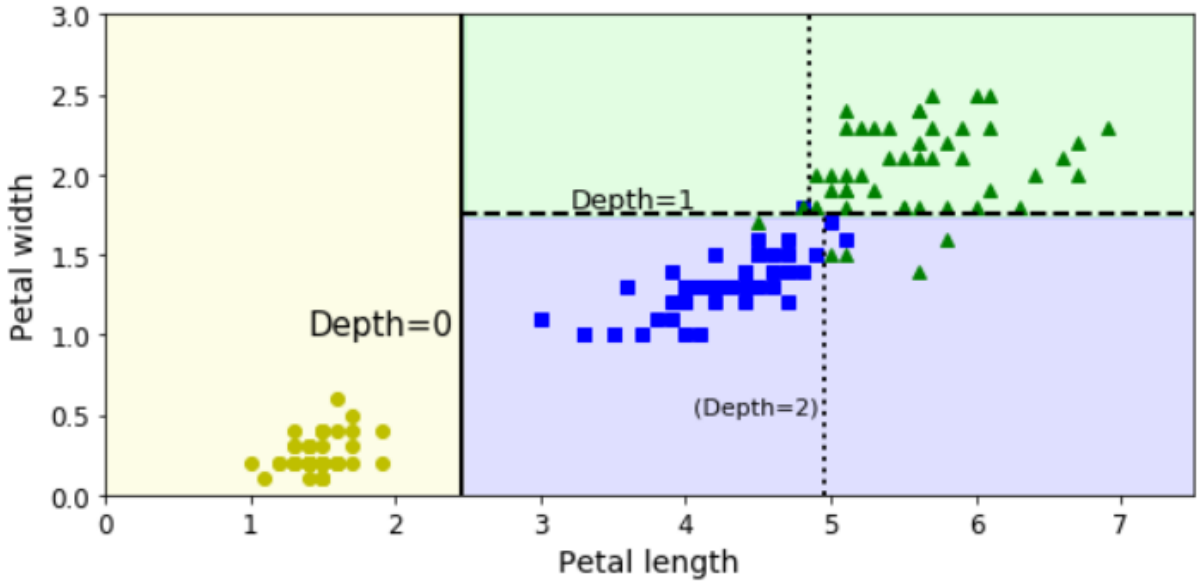


Mesures d'impureté

- Si $G_i = 0$ *ou* $H_i = 0$ *:* le noeud i est dit *pur* c-à-d si toutes les observations d'entraînement qui y aboutissent appartiennent à la même classe ;
- Indice de Gini un peu plus rapide à calculer ;
- En cas de différence entre H et G, l'impureté de Gini a tendance à isoler la classe la $+$ fréquente dans une branche de l'arbre tandis que l'entropie produit en général des arbres légèrement plus équilibrés.

Interprétation des arbres de décision

- *Frontières de décision* facilement interprétables :



- *Estimation de la probabilité* qu'une observation appartienne à une classe donnée k :
  - l'algorithme traverse d'abord l'arbre pour trouver le noeud terminal de cette observation,
  - renvoie le pourcentage d'observations d'entraînement de la classe k dans ce noeud.

Algorithme CART produit que des arbres binaires

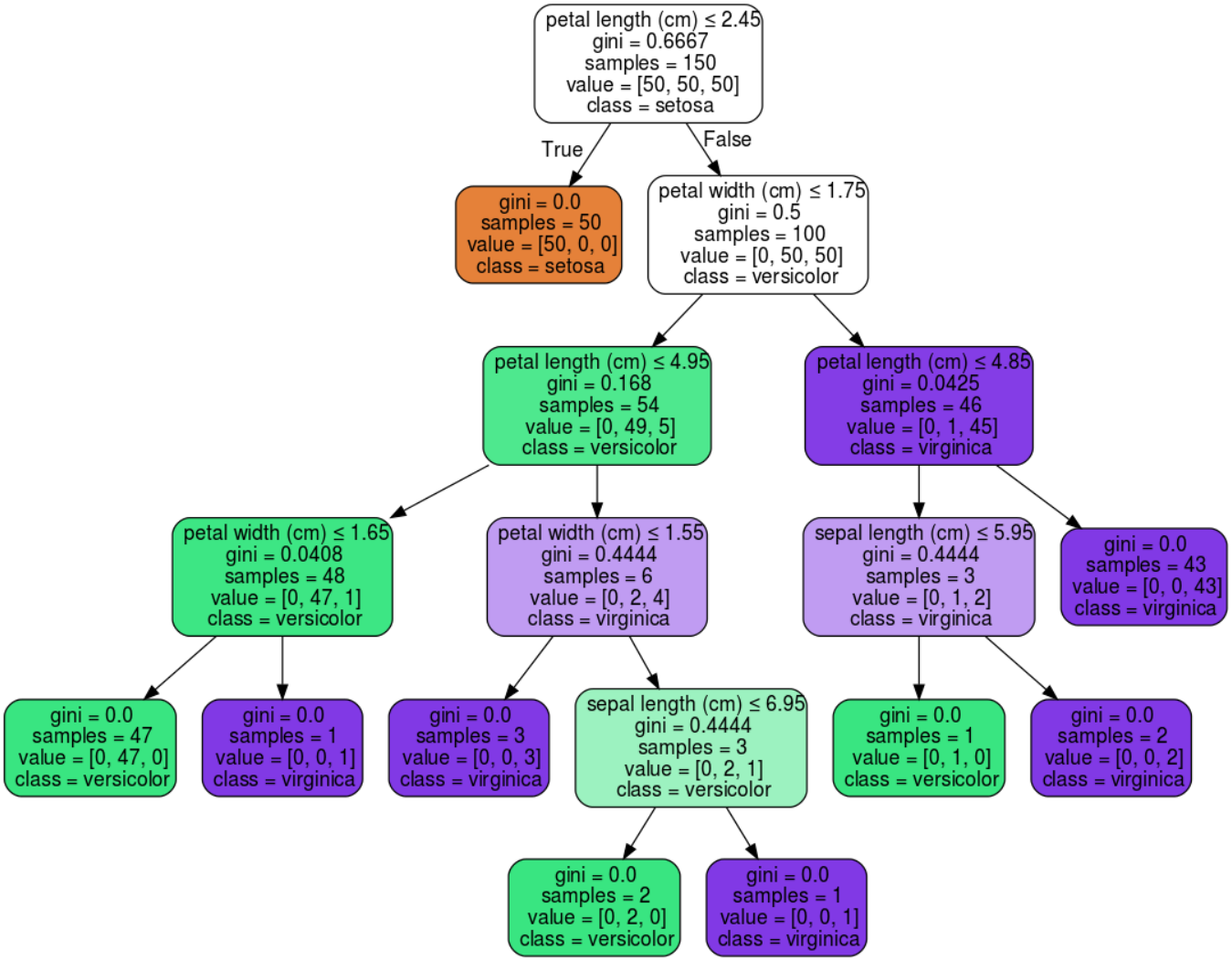
- *Séparation du jeu d'entraînement en deux sous-ensembles* en utilisant une seule caractéristique k et un seuil t_k ;
- *Choix du (k, t_k) ?* Recherche de la paire (k, t_k) qui produit les sous-ensembles les plus purs (pondérés par leur taille) ;
- *Fonction de coût J à minimiser :*

$$J(k, t_k) = \frac{m_{gauche}}{m} G_{gauche} + \frac{m_{droite}}{m} G_{droite}$$

où $G_{gauche/droite}$ mesure l'impureté du sous-ensemble de gauche/ de droite ; et $m_{gauche/droite}$ est le nombre d'observations du sous-ensemble de gauche/de droite ;

- *Récursion :* une fois le jeu de d'entraînement partagé en deux, on applique la même logique aux sous-ensembles de manière récursive ;
- *Critère d'arrêt :* la récursion s'interrompt que lorsque la profondeur maximale (hyperparamètre) est atteinte ou qu'il n'existe plus de partage réduisant l'impureté.

Exemple : Arbre de décision sur les données *Iris*



Résultat pour une profondeur 5

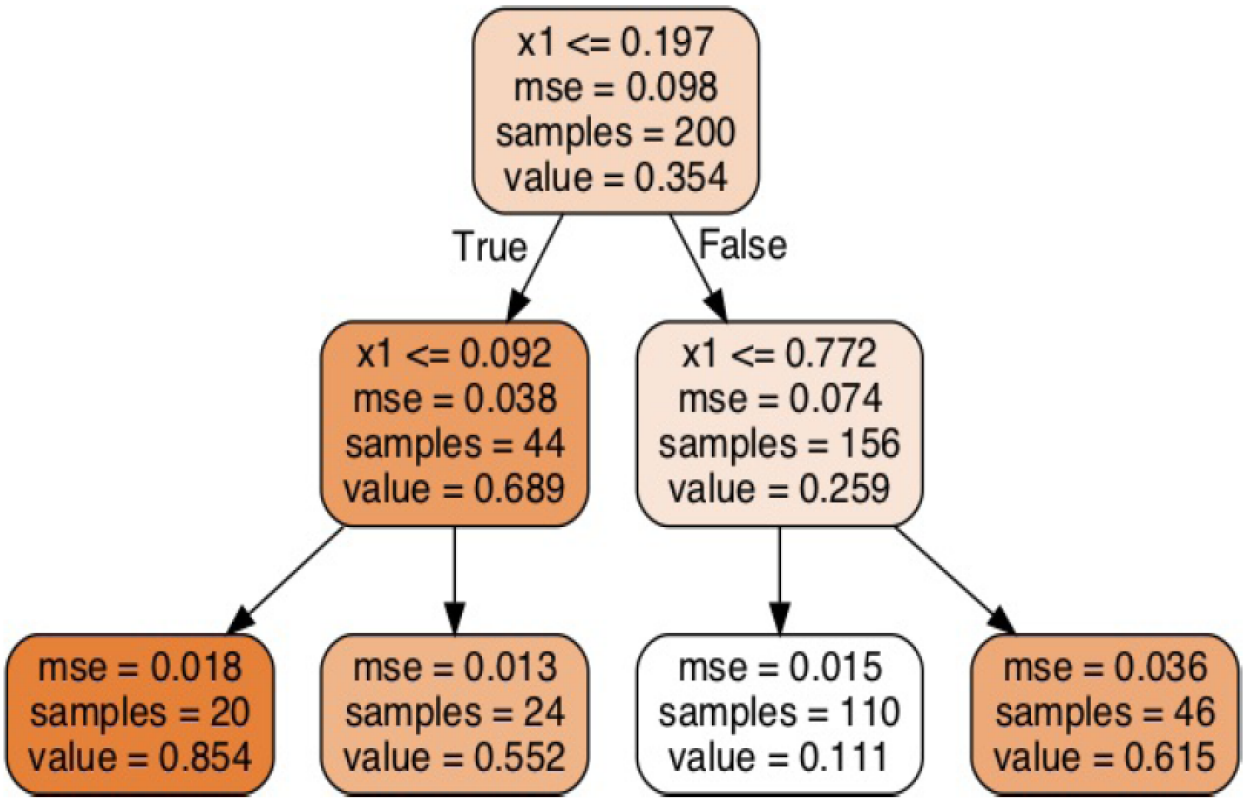
Exercice : Rugby !

On cherche à construire un arbre de décision permettant de décider si une équipe de rugby va gagner ou perdre le prochain match.
Une base d'apprentissage a été construite en considérant les données suivantes qui récapitulent les conditions qui accompagnent les succès et les échecs de cette équipe de rugby.

Match à domicile	Ciel	Match précédent gagné ?	Match gagné ?
oui	Soleil	oui	oui
oui	Pluie	non	non
oui	Soleil	non	oui
non	couvert	oui	oui
non	Pluie	oui	oui
non	Soleil	non	non

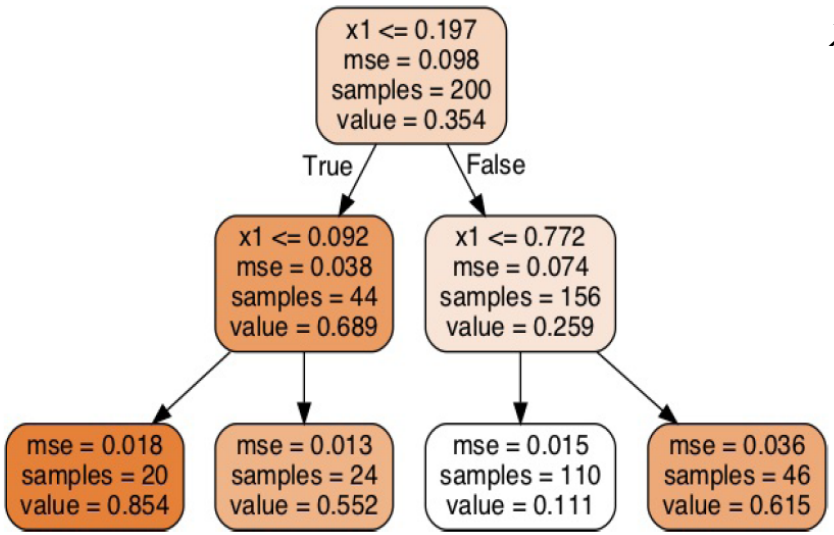
1. Déterminer l'indice de Gini associé à cette base d'apprentissage vis-à-vis des deux classes "Gagner le match" et "Perdre le match".
2. Déterminer la variation de l'indice de Gini lorsqu'on coupe les données à l'aide des variables "Match à domicile", "Ciel" et "Match précédent gagné". En déduire la variable qui sera utilisée au premier niveau de l'arbre de décision.

Régression par arbre de décision : au lieu de prédire une classe dans chaque noeud, l'arbre de décision prédit une valeur.



Résultat pour une profondeur 2

Régression par arbre de décision : au lieu de prédire une classe dans chaque noeud, l'arbre de décision prédit une valeur.



Résultat pour une profondeur 2

Exemple : Valeur prédite pour $x_1 = 0.6$?

- Parcours de l'arbre pour trouver le noeud qui contiendrait x_1 ;
- *Value* $= 0.111$: valeur égale à la moyenne des 110 observations associées ;
- *MSE* $= 0.015$: erreur quadratique moyenne sur les 110 observations.

Algorithme CART pour la régression : fonctionnement similaire à la classification mais la fonction de coût est remplacer par l'erreur moyenne quadratique (MSE) définie par :

$$J(k, t_k) = \frac{m_{gauche}}{m} MSE_{gauche} + \frac{m_{droite}}{m} MSE_{droite}$$

où

- $MSE_{noeud} = \sum_{i \in noeud} (\hat{y}_{noeud} - y^{(i)})^2$
- $\hat{y}_{noeud} = \frac{1}{m_{noeud}} \sum_{i \in noeud} y^{(i)}$

avec $y^{(i)}$ rassemble les observations associés au noeud i et m_{noeud} nombre d'observations associé au noeud.

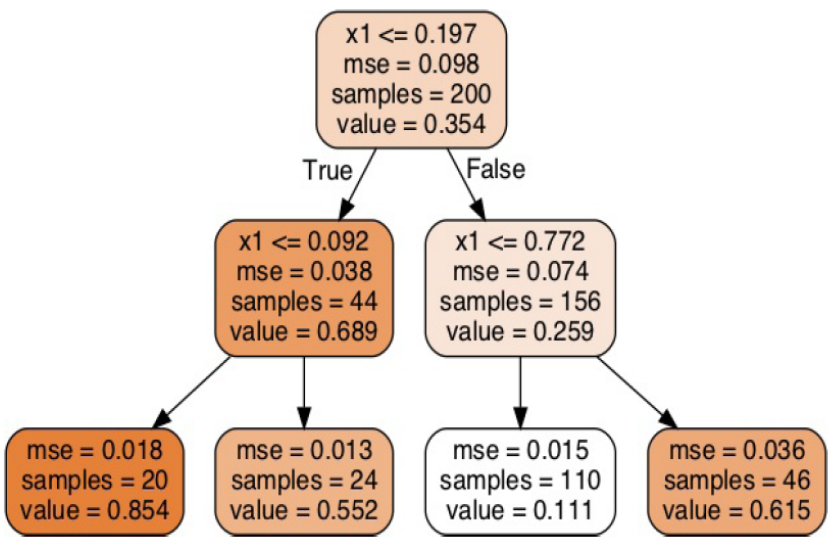
Algorithme CART pour la régression : fonctionnement similaire à la classification mais la fonction de coût est remplacer par l'erreur moyenne quadratique (MSE) définie par :

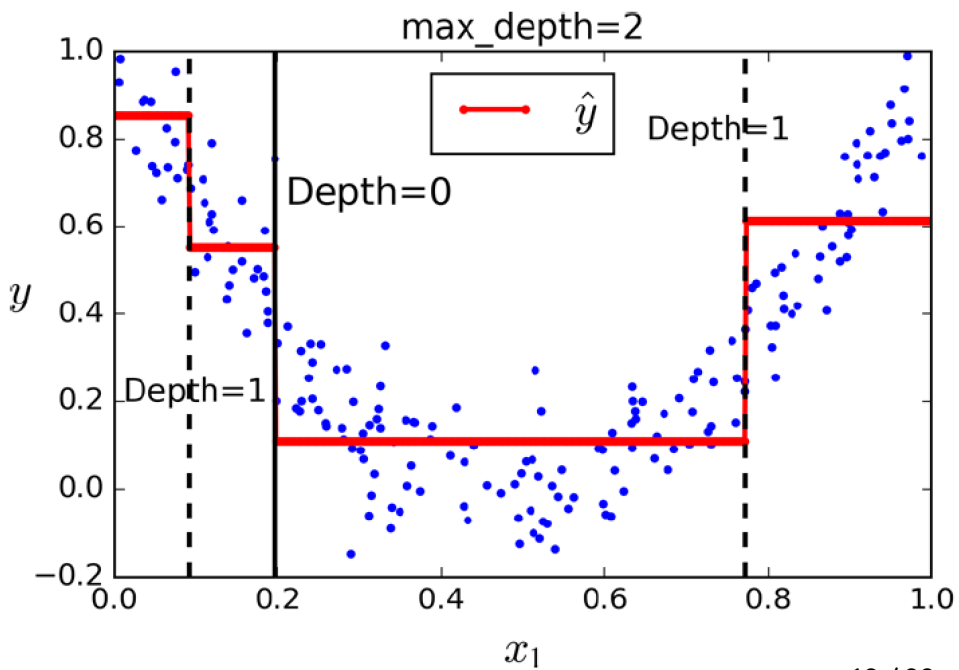
$$J(k, t_k) = \frac{m_{gauche}}{m} MSE_{gauche} + \frac{m_{droite}}{m} MSE_{droite}$$

où

- $MSE_{noeud} = \sum_{i \in noeud} (\hat{y}_{noeud} - y^{(i)})^2$
- $\hat{y}_{noeud} = \frac{1}{m_{noeud}} \sum_{i \in noeud} y^{(i)}$

avec $y^{(i)}$ rassemble les observations associés au noeud i et m_{noeud} nombre d'observations associé au noeud.





Méthodes d'ensemble :

- Entraîner un ensemble d'arbres de décision, chacun sur un sous-ensemble aléatoire différent du jeu d'entraînement ;
- Calcul des prédictions pour chacun des arbres ;
- Choisir la classe obtenant le plus de votes pour la *classification* ou choisir la moyenne des résultats pour la *régression*.

$\Rightarrow$ un tel ensemble d'arbres de décision s'appelle une *forêt aléatoire* : un des algorithmes d'apprentissage automatique les plus puissants.

3 catégories d'apprentissage par ensembles :

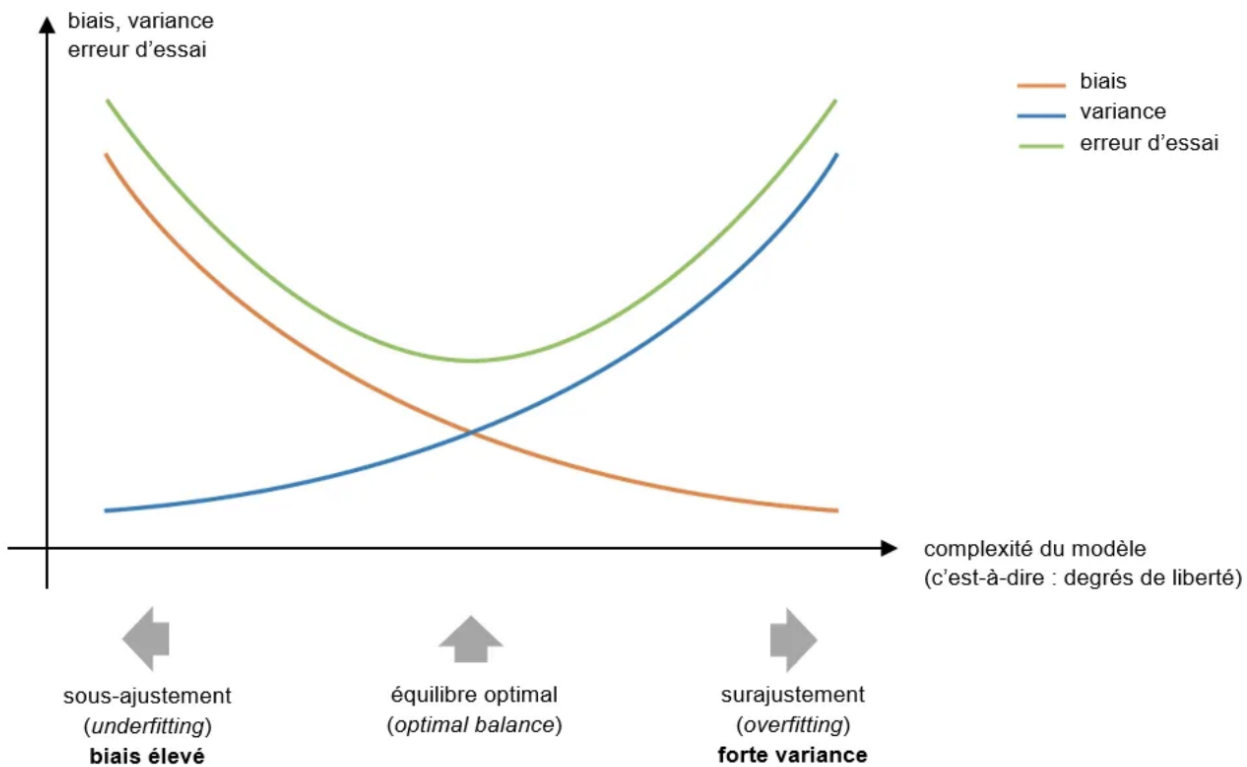
- **le bagging** apprend, en parallèle et indépendamment, des modèles de base qui le constituent et les combine en suivant un processus de moyenne déterminé au préalable.
- **le boosting** apprend séquentiellement de manière très adaptative (un modèle de base dépend des précédents) et combine les modèles de bases selon une stratégie déterminée au préalable.
- **le stacking** apprend, en parallèle et combine les modèles de base en un méta-modèle pour produire une prédiction basée sur les prédictions des différents modèles de base.

Très grossièrement, le *bagging* se concentrera principalement sur l'obtention d'un modèle d'ensemble avec moins de variance que ses composants alors que le *boosting* et le *stacking* essaieront principalement de produire des modèles forts moins biaisés que leurs composants (même si la variance peut également être réduite).

$\Rightarrow$ Biais et Variance

Compromis entre Biais et variance :

- Biais : erreur de généralisation (supposer que le modèle est linéaire alors qu'il est quadratique par exemple).
  ⇒ Modèle à haut biais sous-ajustera les données d'entraînement.
- Variance : sensibilité excessive du modèle à de petites variations dans le jeu d'entraînement.
  ⇒ Modèle ayant beaucoup de degrés de liberté surajustera les données d'entraînement.



On décompose l'erreur de prédiction en x par :

$$Err(x) = Biais^2 + Variance + Erreur\ Irreductible$$

où l'Erreur Irreductible est due au bruit présent dans les données.

Le but est de *minimiser cette erreur* en prenant en compte les comportements opposés du biais et de la variance dépendants de la complexité du modèle

En probabilité, le biais et la variance d'un modèle $\hat{f}$ sur une variable x sont définis par :

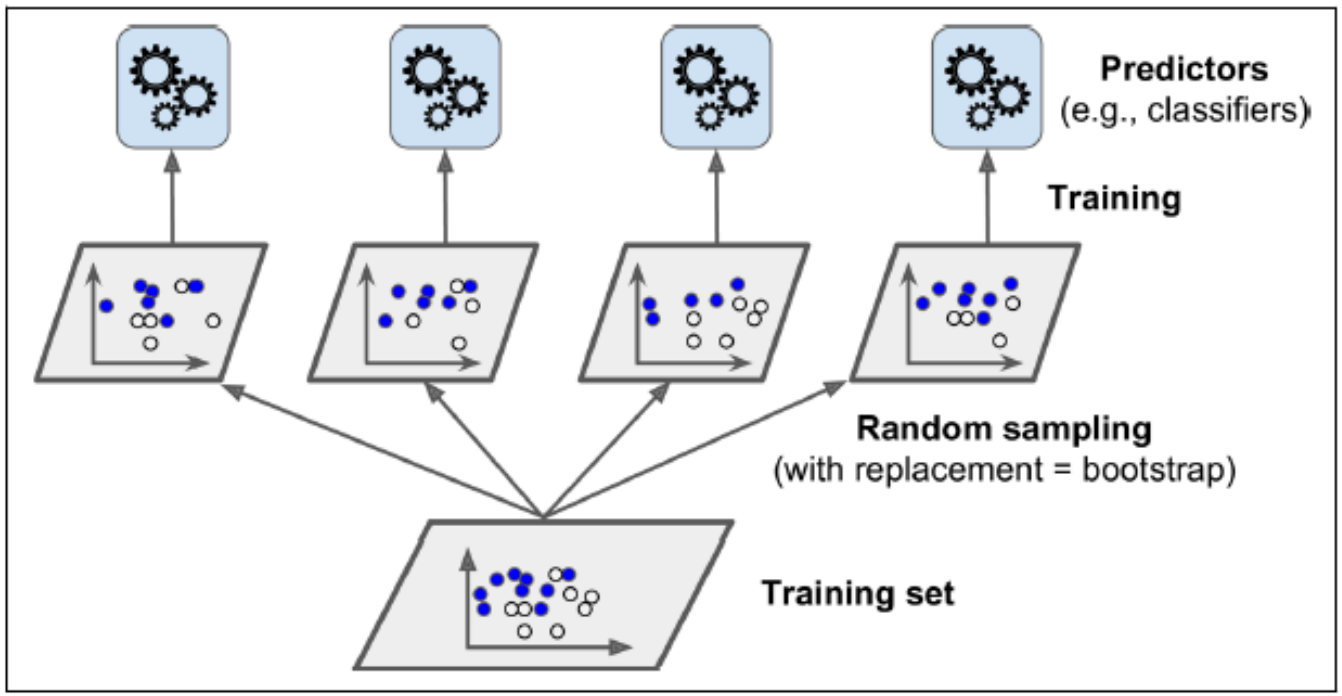
$$Biais(\hat{f}(x)) = E(\hat{f}(x) - f(x))$$

$$Var(\hat{f}(x)) = E(\hat{f}(x)^2) - E(\hat{f}(x))^2$$

Bagging : Bootstrap aggregating

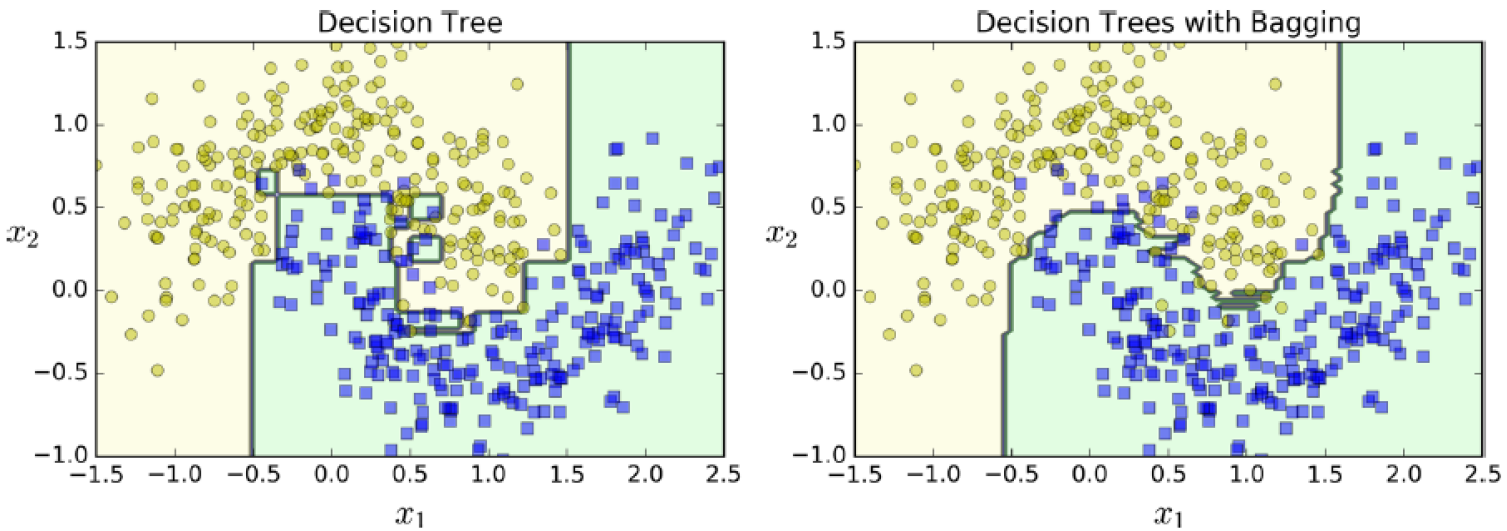
Utiliser le même algorithme d'entraînement pour chaque prédicteur mais l'entraîner sur des sous-ensembles différents extraits aléatoirement du jeu d'entraînement ; le tirage est effectué avec remise ;

Pasting : lorsque le tirage est effectué sans remise.



Méthodes d'ensemble par bagging/pasting :

- Chacun des prédicteurs a un biais plus élevé que s'il avait été entraîné sur le jeu d'entraînement originel
- mais l'agrégation réduit à la fois le biais et la variance.
- en général, l'ensemble a un biais similaire mais une variance inférieure à celle d'un unique prédicteur entraîné sur le jeu d'entraînement d'origine.



Algorithme de forêt aléatoire :

- Introduction d'une part de hasard supplémentaire lors de la construction des arbres : au lieu de rechercher la meilleure variable pour partager un noeud, il recherche la meilleure variable au sein d'un sous-ensemble de variables choisies au hasard.

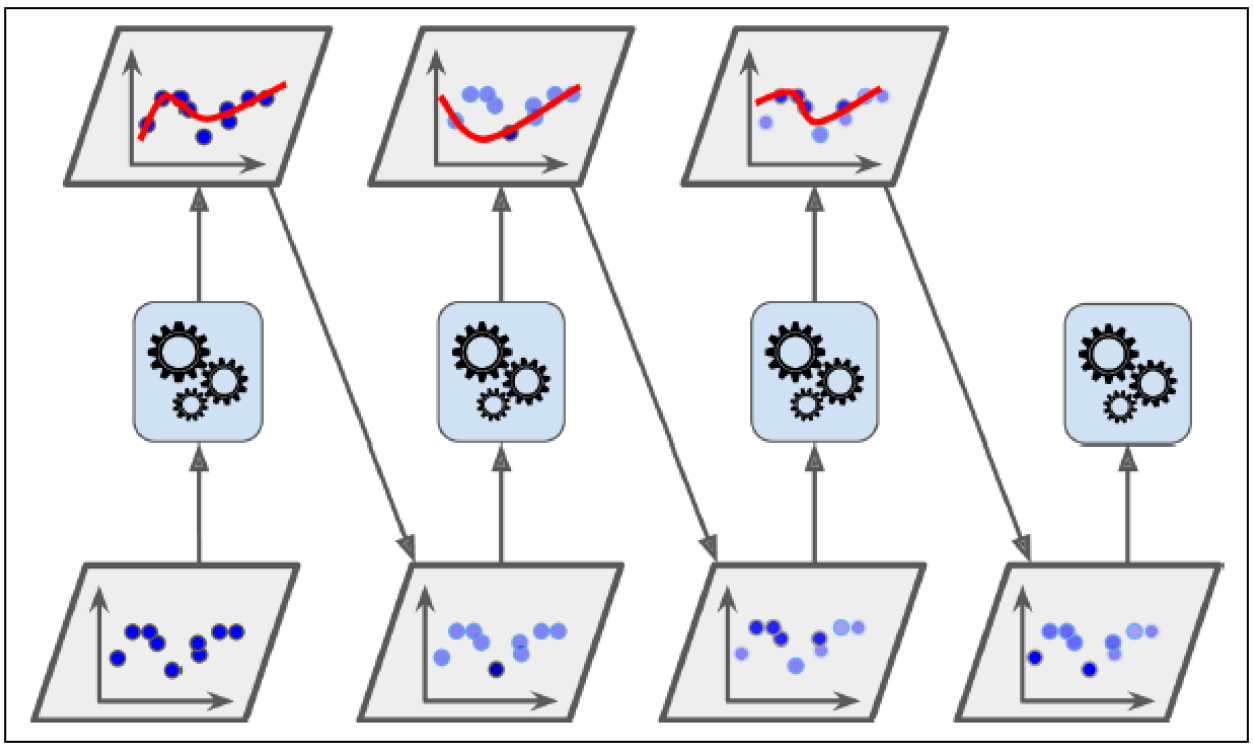
- Plus grande diversité des arbres avec le compromis d'un biais plus élevé en échange d'une variance plus faible.

- Mesure l'importance relative des variables en calculant la réduction moyenne d'impureté (*Mean Decrease Impurity*) sur l'ensemble des noeuds où cette variable intervient au travers de l'ensemble des arbres de la forêt aléatoire

- Arbres extrêmement aléatoires : variante des forêts aléatoires utilisant un seuil aléatoire pour chaque variable plutôt que de rechercher les seuils les meilleurs possibles.

Boosting :
l'idée générale est d'entraîner des prédicteurs l'un après l'autre, chacun s'efforçant de corriger son prédécesseur.

- AdaBoost : corriger son prédecesseur en prêtant plus d'attention aux observations d'entraînement sous ajustées par modification des poids des observations à chaque itération.



Boosting :
l'idée générale est d'entraîner des prédicteurs l'un après l'autre, chacun s'efforçant de corriger son prédécesseur.

- AdaBoost : corriger son prédecesseur en prêtant plus d'attention aux observations d'entraînement sous ajustées par modification des poids des observations à chaque itération.

- Gradient Boosting : ajuster un nouveau prédicteur aux erreurs résiduelles du prédicteur précédent.
  $\Rightarrow$ *Gradient Tree Boosting* ou *Gradient Boosted Regression Trees* (GBRT).

Extension des arbres de décision : Gradient Boosted Regression Trees