Machine Learning

Méthodes de classification supervisée :

- Arbres de décision
- Forêts aléatoires

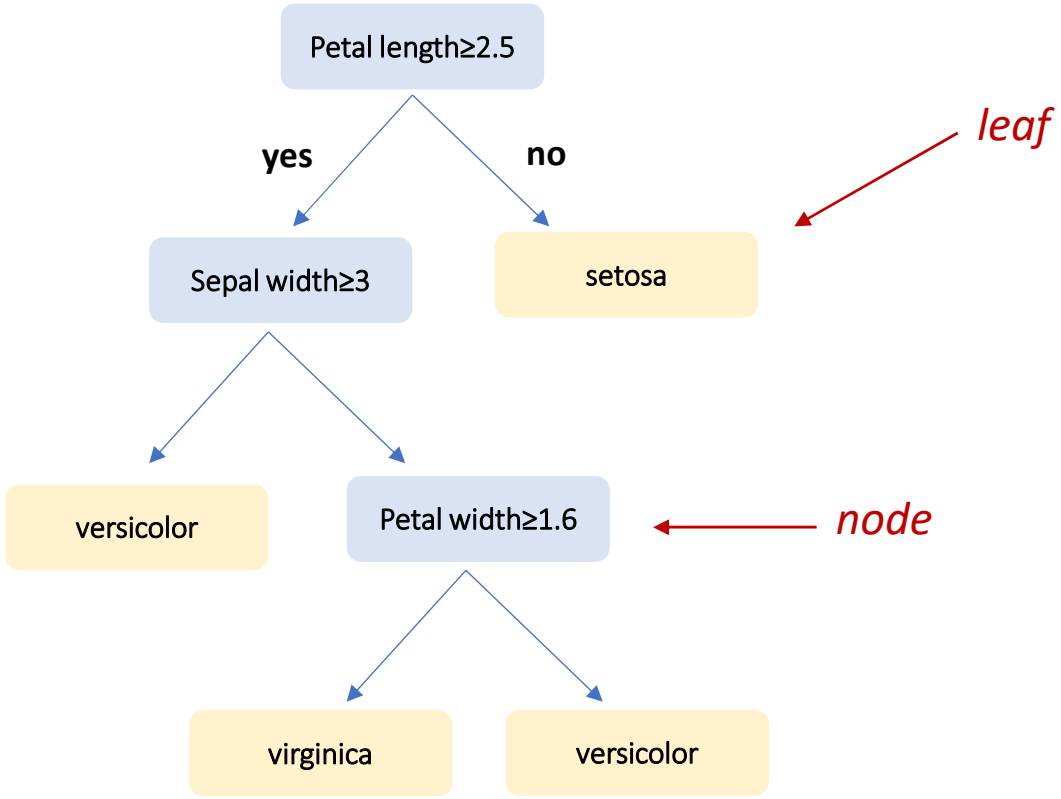
Les arbres de décision

- Principe
- Construction
- Utilisation
- Elagage

Définition

Un arbre de décision est une méthode d'apprentissage supervisée très intuitive. Son résultat est un graphe où chaque *nœud* intermédiaire représente un *test* et chaque nœud final (*feuille*) représente une décision (classe).

Sepal length	Sepal width	Petal length	Petal width	Species
6.3	2.8	6	2.5	virginica
5.8	2.7	5.1	1.9	virginica
7	3.2	4.7	1.4	versicolor
5.1	3.5	1.2	0.4	setosa
4.9	3	1.4	0.2	setosa
6.9	3.1	4.9	1.5	versicolor
5.5	2.9	4	1.3	versicolor
6.3	2.9	5.6	1.8	virginica

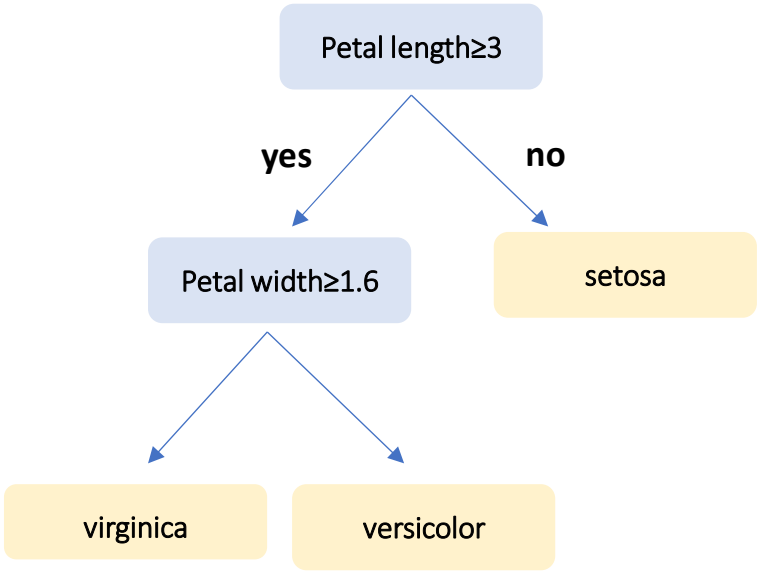


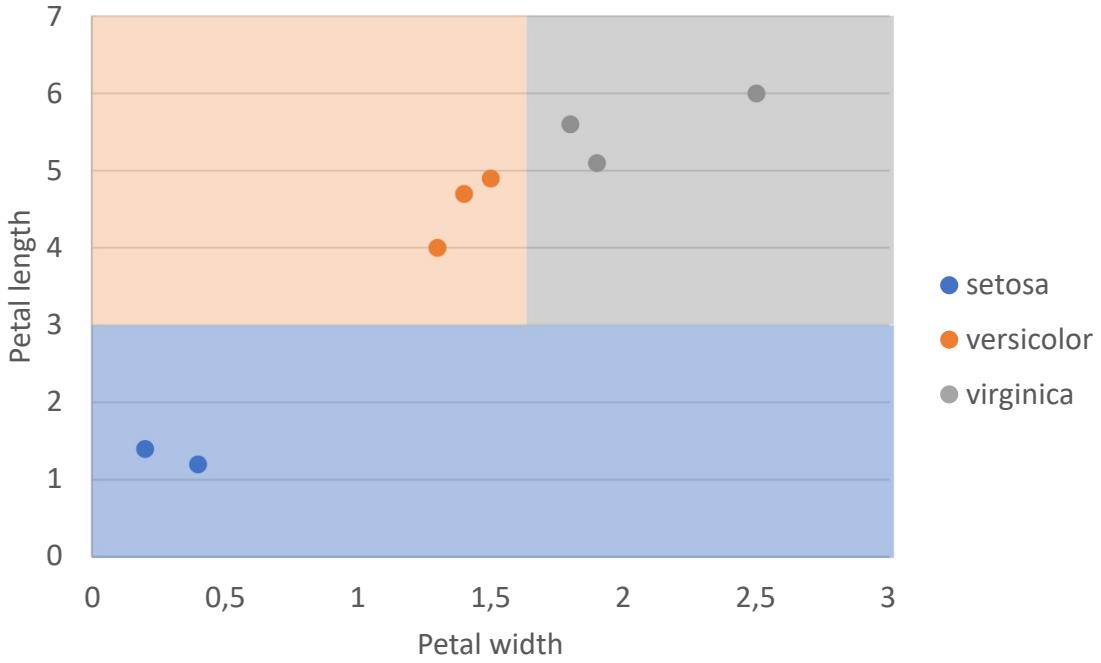
On parle d'arbre binaire quand chaque noeud est divisé en deux branches.

Un partage est défini par un test $X \geq a$ où a est un seuil quand la variable X est numérique, $X = 0$ ou $X = 1$ pour une variable binaire, ou par une combinaison de modalités quand la variable X est qualitative.

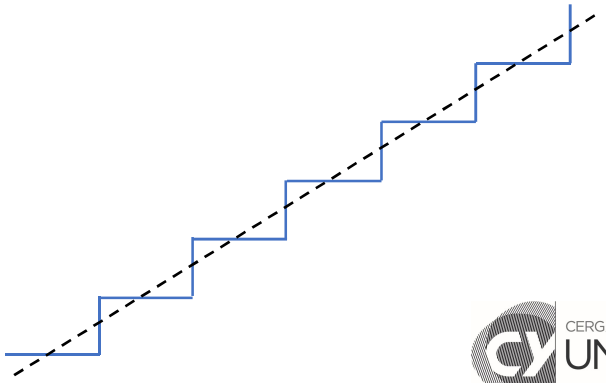
Forme des frontières

Avec un arbre de décision, les frontières sont soit verticales, soit horizontales (pour les variables continues)





Un arbre de décision permet de trouver des frontières relativement simples. Même une frontière linéaire oblique nécessitera un nombre de nœuds important.



Prediction avec un arbre de décision

Prédiction d'une classe

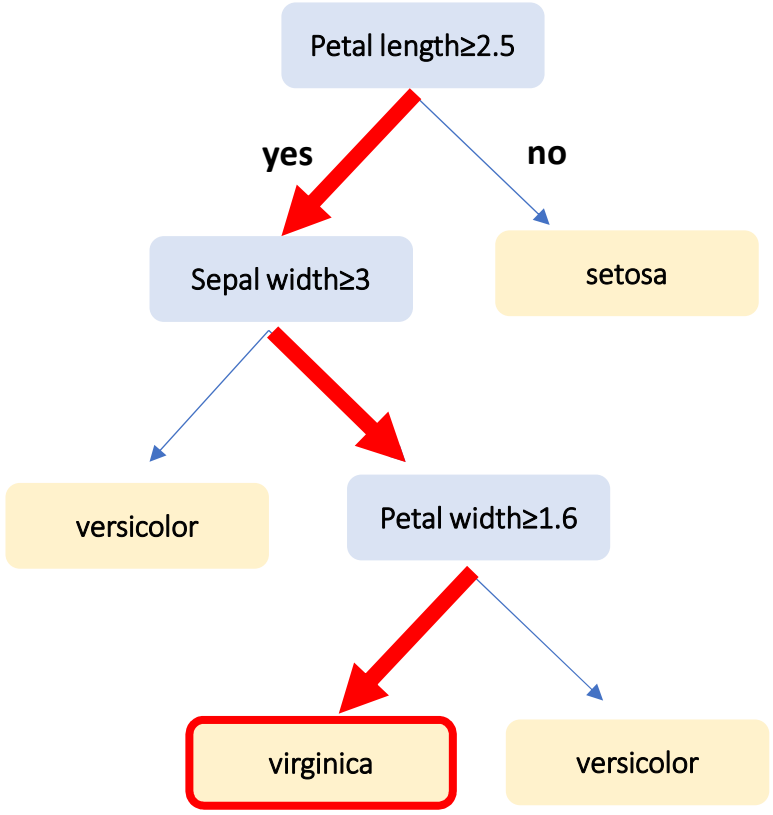
Pour faire la prédiction d'une nouvelle observation, il suffit de lui faire suivre dans l'arbre le chemin correspond aux valeurs de ses variables.

Soit un nouvel iris défini par:

- *Sepal length = 5.5*
- *Sepal width = 2.9*
- *Petal length = 2.8*
- *Petal width = 1.7*

Modèle f

virginica



Prediction avec un arbre de décision

Prédiction de la probabilité d'une classe

Il est intéressant de connaitre, non pas la classe prédite, mais la probabilité pour une nouvelle observation d'appartenir à une classe. Cela permet de différencier une observation qui serait proche de la frontière (donc avec une classe incertaine) d'une observation plus éloignée.

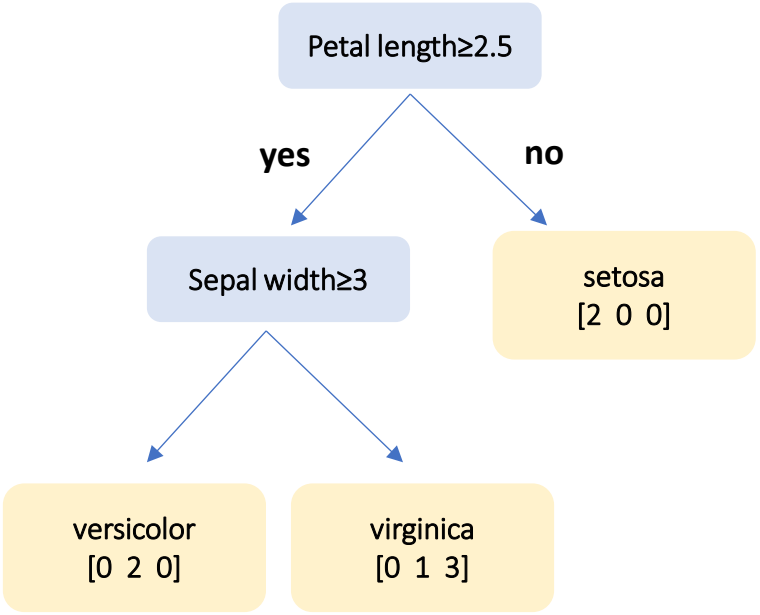
Pour chaque feuille, on comptabilise le nombre d'observations dans chaque classe.

Dans la feuille virginica, il y a 0 setosa, 1 versicolor et 3 virginica

Ensuite on calcule la probabilité de chaque classe à l'intérieur de la feuille.

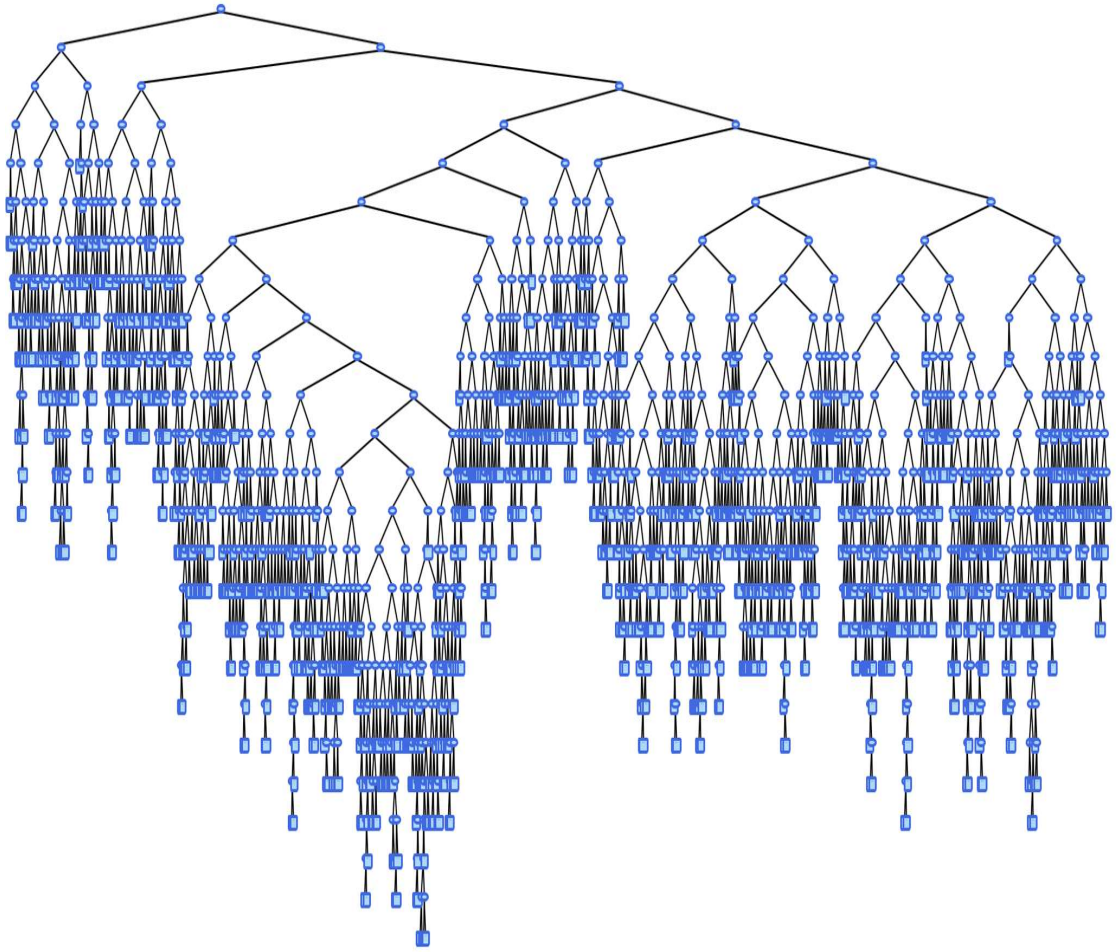
Dans la feuille virginica, la probabilité d'avoir un setosa est 0%, un versicolor est 25% et un virginica est 75%

La classe prédite est celle ayant la plus grande probabilité.



L'algorithme

L'approche est récursive. L'algorithme divise la base d'apprentissage de façon aussi efficace que possible avec un test sur les variables explicatives, jusqu'à obtenir des sous-ensembles contenant des observations de la même classe (ou quasiment).
A chaque nœud, on choisit un test pour séparer les données du nœud en deux sous-ensembles.
On continue à diviser les nœuds de façon récursive jusqu'à obtenir un nœud terminal (feuille).



Les différents algorithmes (CART, QUINLAN,...) dépendent de

1. la façon de choisir un test,
2. du critère d'arrêt.

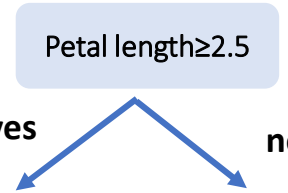
1. Comment choisir un test?

A chaque nœud, on choisit un test sur les variables explicatives pour partager les données en deux sous-ensembles (arbre binaire).

L'idée est de faire converger l'algorithme aussi rapidement que possible vers une feuille qui caractérise une classe de la variable cible.

Le meilleur test est donc celui qui permettra d'obtenir un sous-ensemble contenant des observations de la même classe quasiment.

Sepal length	Sepal width	Petal length	Petal width	Species
6.3	2.8	6	2.5	virginica
5.8	2.7	5.1	1.9	virginica
7	3.2	4.7	1.4	versicolor
5.1	3.5	1.2	0.4	setosa
4.9	3	1.4	0.2	setosa
6.9	3.1	4.9	1.5	versicolor
5.5	2.9	4	1.3	versicolor
6.3	2.9	5.6	1.8	virginica



Sepal length	Sepal width	Petal length	Petal width	Species
6.3	2.8	6	2.5	virginica
5.8	2.7	5.1	1.9	virginica
7	3.2	4.7	1.4	versicolor
6.9	3.1	4.9	1.5	versicolor
5.5	2.9	4	1.3	versicolor
6.3	2.9	5.6	1.8	virginica

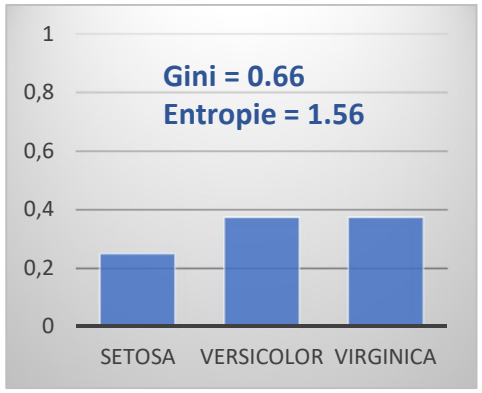
Sepal length	Sepal width	Petal length	Petal width	Species
5.1	3.5	1.2	0.4	setosa
4.9	3	1.4	0.2	setosa

Uniquement la classe setosa

1. Comment choisir un test?

Indice de Ginie et entropie

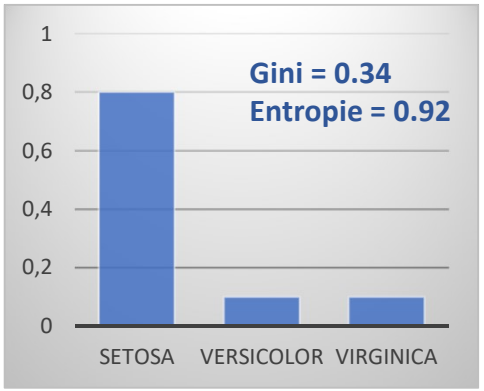
On juge de la pertinence d'un test d'après la distribution des classes de la variable cible après le partage.

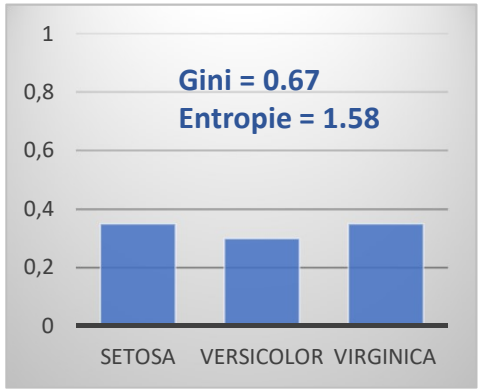


Distribution de la variable cible avant partage

Good split

Bad split





Distribution de la variable cible après partage

Pour mesurer l'uniformité des la distribution, on utilise l'indice de Gini ou l'entropie.

Soient E un ensemble et $E_1,\dots,E_k$ une partition de E. La fréquence d'une classe E_i est données par

$$p_i = \frac{card(E_i)}{card(E)}$$

Indice de Gini : $$Gini(E) = \sum_{i=1}^{k} p_i \times (1 - p_i)$$

Entropie : $$Ent(E) = -\sum_{i=1}^{k} p_i \times \log_2(p_i)$$

Par convention $0*\log_2(0)=0$

Ces fonctions sont positives et maximales s'il y a uniformité.

L'indice de Gini ou l'entropie se calculent uniquement sur la fonction cible

1. Comment choisir un test?

Le gain d'information

La distribution de la variable cible après le partage dépend de sa distribution avant partage.

Le gain d'information compare l'indice de Gini ou l'entropie avant et après partage.

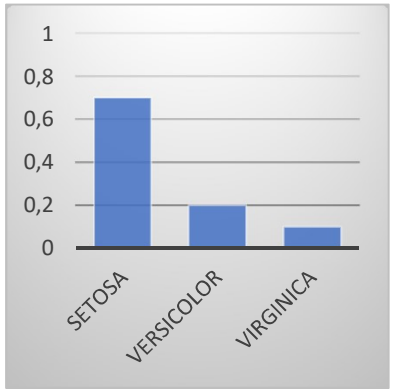
Dans le cas d'un arbre binaire, on note E les données avant partage et E_1 et E_2 les deux sous-ensembles obtenus après partage, alors le gain est défini par,

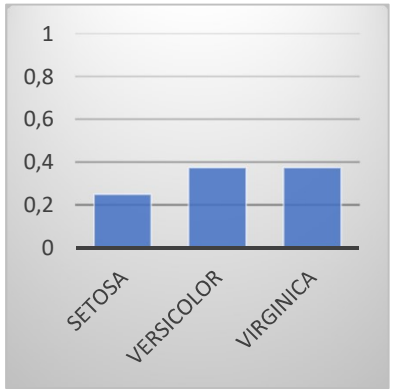
$$Gain = Gini(E) - p_1 Gini(E_1) - p_2 Gini(E_2)$$

$$Gain = Ent(E) - p_1 Ent(E_1) - p_2 Ent(E_2)$$

où $p_i = \frac{card(E_i)}{card(E)}$.

A chaque noeud, le gain est calculer pour tous les tests possibles (toutes les variables explicatives et tous les niveaux de partage sur ces variables). Le test retenu est celui qui a le plus grand gain.

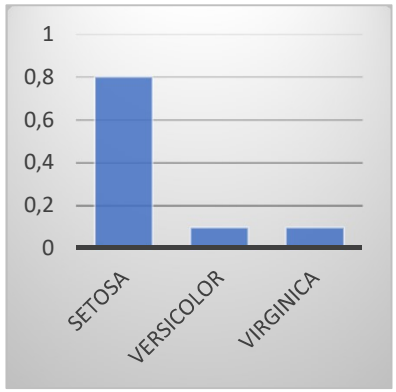




Distribution de la variable cible avant partage

Partage sans intérêt

Partage intéressant



Distribution de la variable cible après partage

2. Critères d'arrêt

Il y a des conditions automatiques d'arrêt, par exemple quand:

- les observations d'un nœud sont (presque) tous dans la même classe,
- tous les test ont été essayés.

Et il y a des critères d'arrêt que l'on peut fixer:

- le minimum d'observation dans un nœud,
- le nombre de nœuds,
- ...

Le choix de ces critères va influencer la profondeur de l'arbre et ses capacités à reproduire les observations de la base d'apprentissage. Un arbre profond aura tendance à avoir une erreur d'ajustement très petite, mais il est long à calculer et, surtout, il conduit très certainement à un sur-ajustement des données.

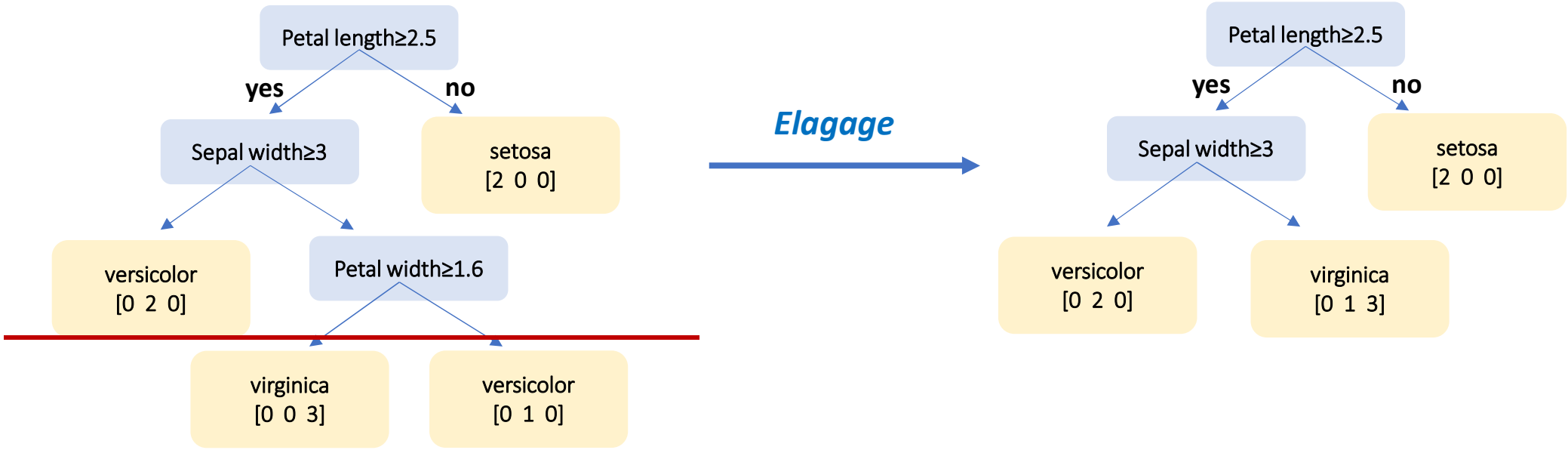
On peut considérer l'exemple extrême d'un arbre ayant autant de feuilles que d'observations dans la base d'apprentissage (une feuille pour chaque observation). Un tel arbre n'aura aucune capacité de généralisation.

Afin d'éviter le sur-apprentissage, on empêche les arbres d'être trop profonds. Pour cela, on utilise une procédure d'élagage de l'arbre.

2. Critères d'arrêt

Elagage d'un arbre

Elaguer un arbre consiste à limiter sa profondeur. L'objectif est d'éviter la formation de feuilles spécifiques à des cas particuliers.



Automatiquement élaguer va augmenter l'erreur d'ajustement (car l'arbre n'apprend plus les cas particulier). Mais cela permet une meilleure généralisation et fait diminuer l'erreur de prévision.

Pour trouver le bon compromise (le bon niveau d'élagage), on trace l'erreur de prévision en fonction de la complexité de l'arbre. Quand l'erreur de prévision augmente, il est temps d'élaguer.

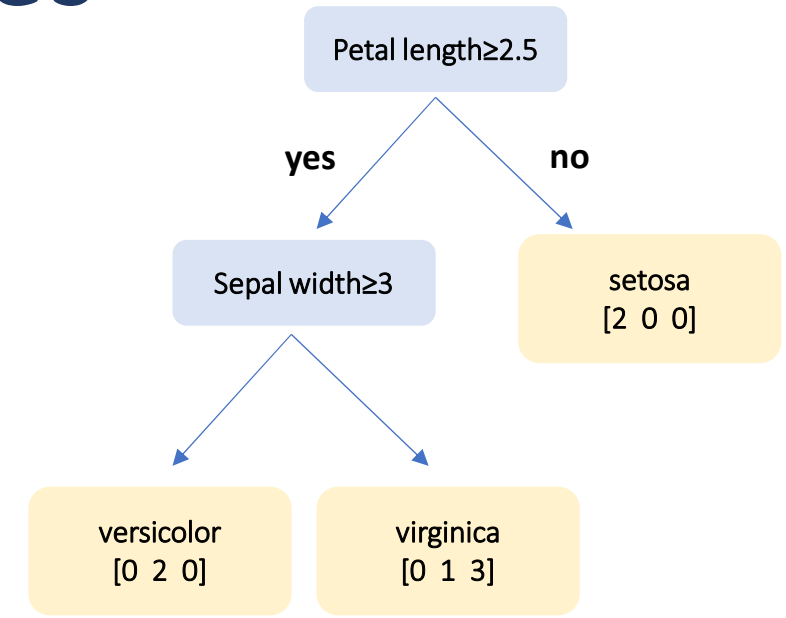
Gestion des données manquantes

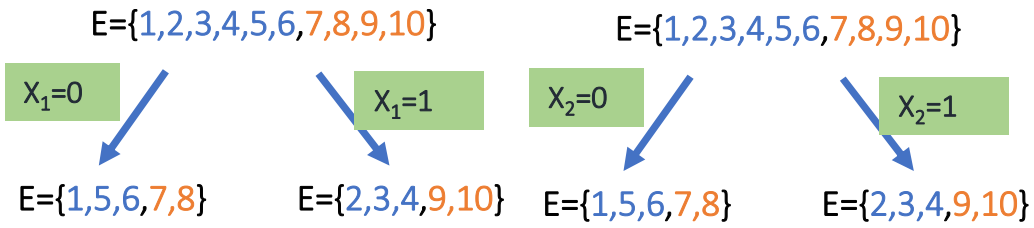
Supposons que pour un nouvel individu une variable explicative n'est pas renseignée.
Si cette variable intervient dans l'arbre alors il n'est pas possible de prédire la classe de ce nouvel individu.

Soit le nnouvel iris défini par :

- Petal length = 3,5
- Petal width = 2,5
- Setal length = 1,7
- Sepal width : Non renseigné

Impossible de lui appliquer l'arbre ci-contre au niveau du 2ème split.





X_1 et X_2 sont des variables de substitution

Une solution consiste à utiliser des variables de substitution (*surrogate-split*), c'est-à-dire une variable qui aboutit au même sous-ensembles lors du split. La variable de substitution vient remplacer la variable explicative non renseignée pour la prévision.

Bilan sur les arbres de décision

Les arbres de décision font partie des méthodes prédictives les plus utilisées car ils donnent des résultats remarquables en pratique et ils présentent beaucoup d'avantages :

- Pas d'hypothèse sur la distribution des variables explicatives
- Pas affecté par les problème d'échelle de mesure des variables ou données atypiques
- Une structure arborescente facilement compréhensible contrairement à d'autres méthodes du type « boite noire » (réseaux de neurones, svm,…).
- Gestion des données manquantes

Ils présentent beaucoup d'avantages mais aussi des limitations :

- Le problème d'optimisation est NP-complet d'où l'utilisation d'heuristiques avec un résultat sous-optimal
- Ils sont très sensibles au bruit, instables et ont tendances à sur-apprendre les données

Les solutions à ce problème sont

- L'élagage comme nous l'avons vu
- Les random forests (arbres adaptés au bagging)

L'arbre le plus utilisé est l'arbre CART. Il utilise l'indice de Gini, les variables de substitution,…

Les Forêts aléatoires

- Principe et algorithme
- Out of bag
- Importance des variables

Forêts aléatoires (Breiman 2001)

Comme le nom l'indique, une forêt aléatoire est une agrégation d'arbres de décision. L'objectif est de rendre la méthode moins sensible au bruit et aux points aberrants de la base d'apprentissage.

L'idée est simple. Il s'agit de construire plusieurs arbres sur des échantillons bootstrap de la base d'apprentissage.

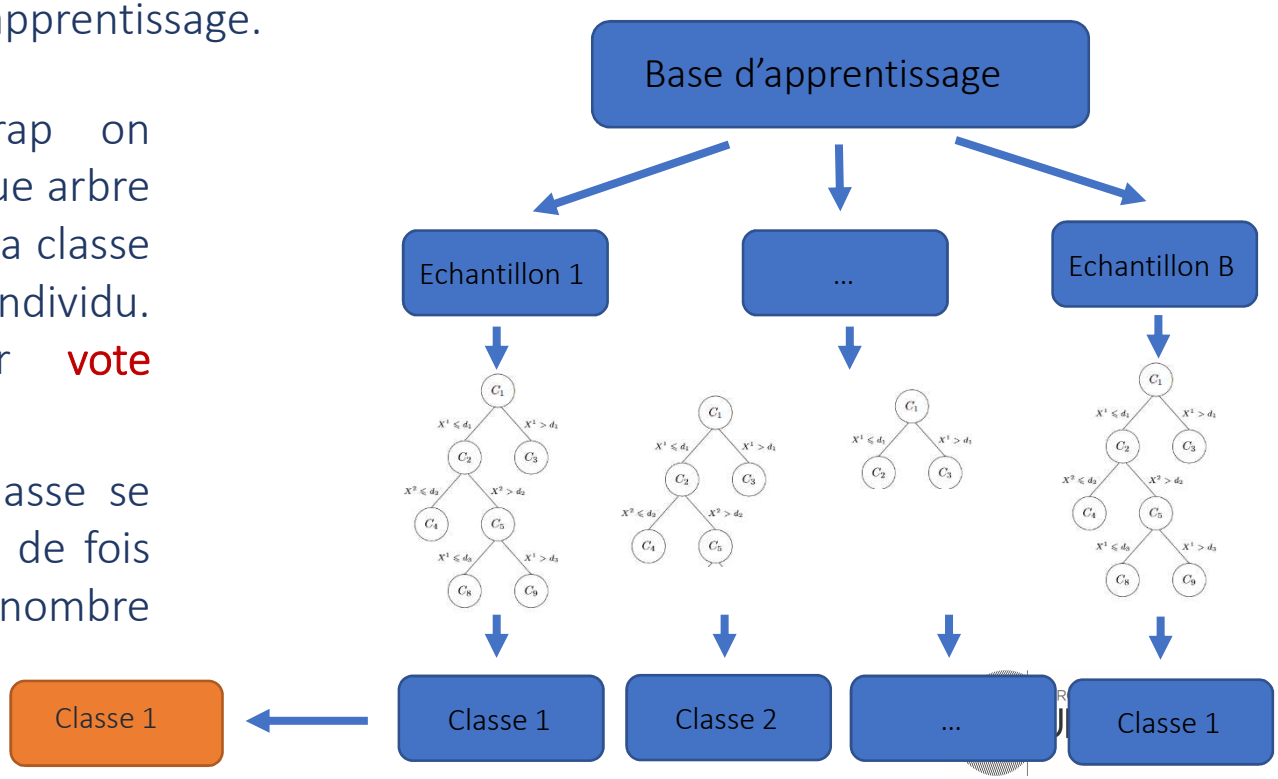
Un échantillon bootstrap est un tirage aléatoire d'éléments de la base d'apprentissage :

- soit de *n* éléments avec remise
- soit de *k*<*n* éléments (avec ou sans remise)

parmi le *n* observations de la base d'apprentissage.

Pour chaque échantillon bootstrap on construit un arbre de décision. Chaque arbre permet d'obtenir une estimation de la classe de la variable cible pour un nouvel individu. L'estimation finale se fait par vote majoritaire.

La probabilité d'appartenir à une classe se calcule en comptabilisant le nombre de fois où un arbre a prédit la classe sur le nombre d'arbres total.



Algorithme des random forests

On peut montrer que cet algorithme est d'autant plus performant que les estimations des arbres sont « décorrélées » les unes des autres. Pour ce faire on adopte la stratégie suivante :

- Construire beaucoup d'arbres mais de faible profondeur
- Pour chaque noeud de chaque arbre, sélectionner un petit sous-ensemble de variables pour le partage parmi les p variables explicatives ($\sim\sqrt{p}$)

Chacun des petits arbres est donc moins performant mais leur agrégation est performante.

Algorithme des random forests

E={$(x_1,y_1),\ldots,(x_n,y_n)$} base d'apprentissage avec p variables explicatives
Pour b=1,…,B (boucle sur B arbres)
    tirer un échantillon bootstrap E_b
    construire un arbre sur l'échantillon E_b tel que :
        À chaque nœud de l'arbre choisir le meilleur split
        sur un nombre restreint de variables tirées aléatoirement
            parmi les p variables
Fin pour b
Pour un nouvel individu défini par x,
    estimer sa classe pour chacun des B arbres
    choisir la classe majoritaire

Procédure Out Of Bag

Afin d'éviter le sur-ajustement, on calcule l'erreur de prévision à l'aide d'une procédure de validation croisée. Cette technique couteuse en temps de calcul est directement intégrée dans l'algorithme des forêts aléatoires. Il est possible de calculer une erreur de prévision car les échantillons bootstrap n'utilisent pas toutes les observations.

Pour chaque point de la base d'apprentissage (x_i, y_i) :

- On considère les échantillons bootstrap qui ne contiennent pas cette observation
- On construit une forêt aléatoire sur ces échantillons (on se restreint aux arbres n'ayant pas cette observation dans leur base d'apprentissage)
- On calcule $\hat{y}_i$ la prévision de y_i

L'erreur Out Of Bag (OOB) est donnée par $\frac{1}{n}\sum_{i=1}^{n} \mathbf{1}_{\hat{y}_i \neq y_i}$

Importance des variables

L'agrégation des arbres pour construire une forêt fait que l'on perd l'interprétabilité du modèle. Il n'y plus de visualisation sous forme de graphe. Il est difficile de déterminer quelles variables ont eu un rôle crucial dans la discrimination.

Une mesure permet alors de quantifier le rôle de chaque variable explicative dans le modèle. Elle est basée sur le fait que si une variable a un rôle négligeable alors le fait de perturber aléatoirement ses valeurs n'aura pas d'impact sur l'erreur OOB, et vice-versa.

Questions?