CY TECH

Machine Learning

Méthodes de classification supervisée :

- Arbres de décision
- Forêts aléatoires

CERGY PARIS UNIVERSITÉ

Définition

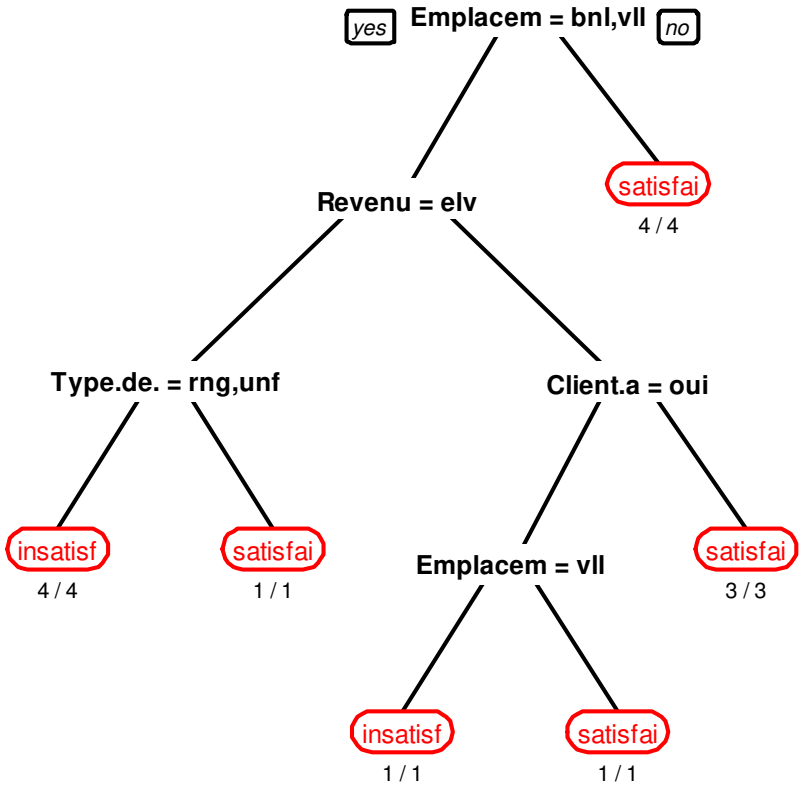
Un arbre de décision est une méthode d'apprentissage supervisée très intuitive. Son résultat est un graphe où chaque ***nœud*** intermédiaire représente un ***test*** et chaque nœud final (***feuille***) représente une décision (***classe***)

Attribut/variable — Cible

Instances/Observations

Emplacement	Type de maison	Revenu	Client anterieur?	Resultat
banlieue	jumelée	bas	non	satisfait
banlieue	rangée	bas	oui	satisfait
banlieue	unifamiliale	élevé	non	insatisfait
banlieue	unifamiliale	élevé	oui	insatisfait
banlieue	rangée	élevé	non	insatisfait
rural	jumelée	bas	oui	satisfait
rural	unifamiliale	bas	non	satisfait
rural	unifamiliale	élevé	non	satisfait
rural	rangée	élevé	oui	satisfait
ville	jumelée	bas	non	satisfait
ville	jumelée	bas	oui	insatisfait
ville	rangée	bas	non	satisfait
ville	jumelée	élevé	non	satisfait
ville	rangée	élevé	oui	satisfait

adapté de http://www.decisiontrees.net/tutorial/1_intro.html



On parle d'arbre ***binaire*** lorsque chaque nœud se divise en deux classes

Principe de construction

La démarche est récursive. Il s'agit de diviser le plus efficacement possible l'ensemble d'apprentissage par des tests sur les observations jusqu'à obtenir des sous-ensembles ne contenant que (ou presque) des observations d'une même classe.

```
Algorithme générique
Etape 0. Initialiser l'arbre à l'arbre vide et le nœud à la racine
Etape 1.
si nœud est terminal
    lui affecter une classe
sinon
    choisir une variable et un test
    pour  chaque valeur du test
            ➢ créer un sous-ensemble d'apprentissage avec les
          observations correspondantes à la valeur du test
            ➢ répéter l'étape 1 avec le sous-ensemble
    finpour
finsi
```

Il existe plusieurs algorithmes de construction (arbres CART, arbres de QUINLAN,…).
Chacun se distingue par sa façon de répondre aux deux questions :

- Décider si un nœud est terminal?
- Choisir un attribut pour le partage du nœud?

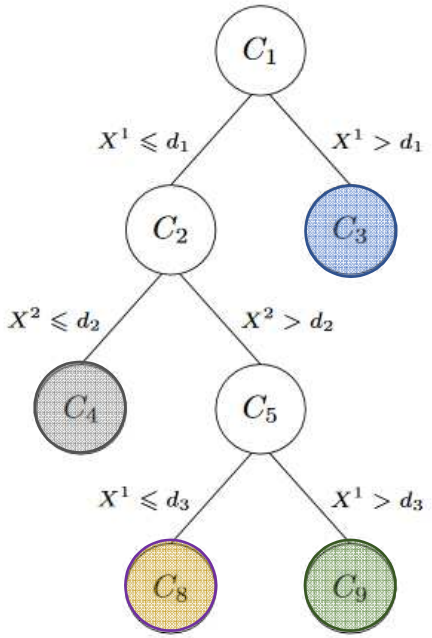
Le partage d'un nœud

Le partage

- Si l'attribut X est catégoriel alors on partage l'ensemble d'apprentissage suivant les modalités (ou regroupement de modalités) de l'attribut
- Si l'attribut X est quantitatif alors on partage l'ensemble d'apprentissage suivant un test $X<a$ ou $X \geq a$ (fonction split). On teste « toutes » les valeurs intermédiaires entre x_i et x_{i+1}.

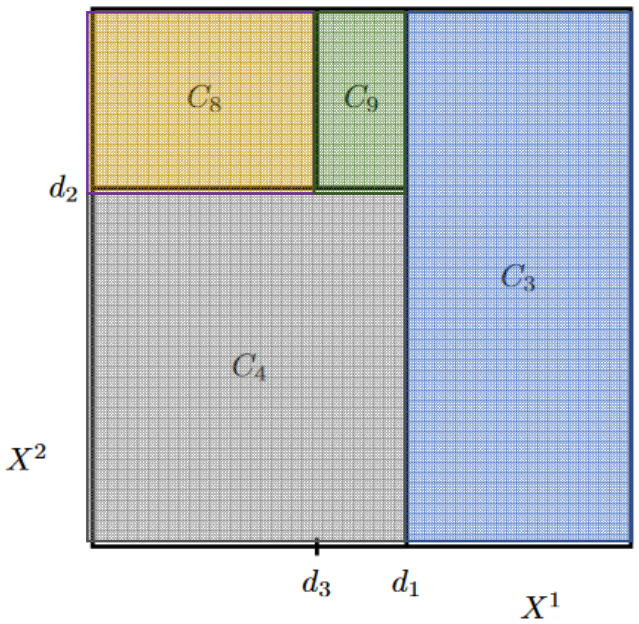
Exemple : Classification avec deux attributs quantitatifs

Arbre de décision

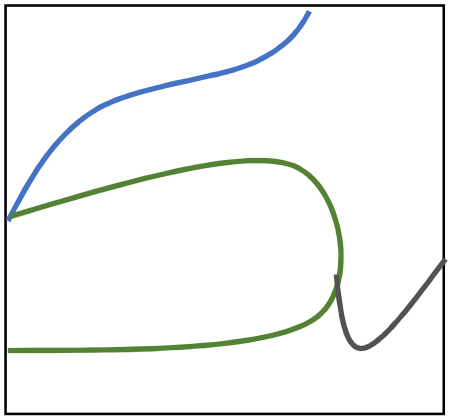


Genuer et Poggi : https://arxiv.org/pdf/1610.08203.pdf

Classification obtenue dans l'espace de variation des variables explicatives X_1 et X_2



Ne permet pas de modéliser des classes du type « patatoïdes »



Critères de partage d'un noeud

L'idée est de faire converger l'arbre le plus rapidement possible vers des feuilles qui caractérisent une classe de la variable cible, c.a.d. des feuilles contenant majoritairement une seule classe de la variable cible.

On préfère une feuille avec 15 satisfaits et 2 non satisfaits qu'une feuille avec 9 satisfaits et 8 non satisfaits.

Il existe deux critères usuels pour mesurer l'homogénéité (uniformité) de la répartition.

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

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

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

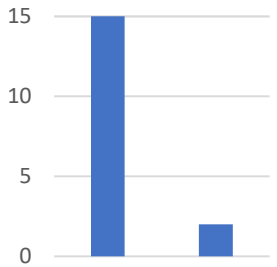
Entropie : $\text{Ent}(E) = -\sum_{i=1}^{k} p_i \times \log(p_i)$

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

$\Leftrightarrow p_i = \text{card}(E)/k$

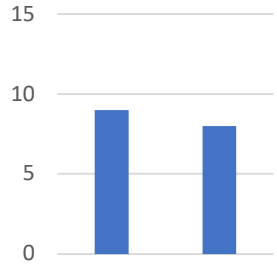
Dans l'exemple précédent, nous avons E_1 l'ensemble des satisfaits et E_2 l'ensemble des non satisfaits.

Feuille 1
15 satisfaits
2 non satisfaits



p_1=15/17
p_2=2/17
Gini = 0,21
Ent = 0,36

Feuille 2
9 satisfaits
8 non satisfaits



p_1=9/17
p_2=8/17
Gini = 0,50
Ent = 0,69

Choix de la variable de partage

Afin de faire converger l'arbre, à chaque nœud on va choisir la variable (attribut) de partage qui va maximiser le gain de l'indice de Gini (ou d'entropie), c.-à-d. la différence d'indices entre avant et après le partage (stratégie gloutonne),

$$\text{Gain} = \text{Gini}(E) - \sum_{i=1}^{k} p_i \times \text{Gini}(E_i) \qquad \text{Gain} = \text{Ent}(E) - \sum_{i=1}^{k} p_i \times \text{Ent}(E_i)$$

Le gain est maximal lors que le choix d'un attribut permet de classer correctement toutes les instances.

A la racine, on a

Gini(E)=(10/14)(1-10/14)+(4/14)(1-4/14)=0.41

L'indice se calcule par rapport à la variable cible !!!!

Si on choisit la variable de partage « Revenu », alors le gain est

*G(Revenu)=0.41-[p(bas)*Gini(bas)+p(élevé)*Gini(élevé)]=0.10*

cf. calculs →

Le gain ne permet pas de comparer des variables n'ayant pas le même nombre de modalités.

⇒
- Binariser les variables (solution souvent adoptée par logiciel)
- Pondérer le gain

Attention à construire des arbres équilibrés. Un arbre avec 4 feuilles dont une contiendrait 99% des observations ne serait pas prédictif.

Emplacement	Type de maison	Revenu	Client anterieur?	Resultat
banlieue	jumelée	bas	non	satisfait
banlieue	rangée	bas	oui	satisfait
banlieue	unifamiliale	élevé	non	insatisfait
banlieue	unifamiliale	élevé	oui	insatisfait
banlieue	rangée	élevé	non	insatisfait
rural	jumelée	bas	oui	satisfait
rural	unifamiliale	bas	non	satisfait
rural	unifamiliale	élevé	non	satisfait
rural	rangée	élevé	oui	satisfait
ville	jumelée	bas	non	satisfait
ville	jumelée	bas	oui	insatisfait
ville	rangée	bas	non	satisfait
ville	jumelée	élevé	non	satisfait
ville	rangée	élevé	oui	satisfait

E

L'indice se calcule par rapport à la variable cible !!!!

E_1

Emplacement	Type de maison	Revenu	Client anterieur?	Resultat
ville	jumelée	bas	non	satisfait
ville	jumelée	bas	oui	insatisfait
rural	jumelée	bas	oui	satisfait
banlieue	jumelée	bas	non	satisfait
ville	rangée	bas	non	satisfait
banlieue	rangée	bas	oui	satisfait
rural	unifamiliale	bas	non	satisfait

p(Revenu=bas)=7/14

Gini(Revenu=bas)=6/7(1-6/7)+1/7*(1-1/7)*

E_2

Emplacement	Type de maison	Revenu	Client anterieur?	Resultat
banlieue	unifamiliale	élevé	non	insatisfait
banlieue	unifamiliale	élevé	oui	insatisfait
rural	unifamiliale	élevé	non	satisfait
ville	jumelée	élevé	non	satisfait
banlieue	rangée	élevé	non	insatisfait
rural	rangée	élevé	oui	satisfait
ville	rangée	élevé	oui	satisfait

p(Revenu=élevé)=7/14

Gini(Revenu=élevé)=4/7(1-4/7)+3/7*(1-3/7)*

Critère d'arrêt

Décider si un nœud est terminal

- quand les instances associés au nœud sont quasiment tous dans la même classe
- quand un nœud à moins de x observations
- quand toutes les variables ont été testées (s'il n'y a que des variables qualitatives)

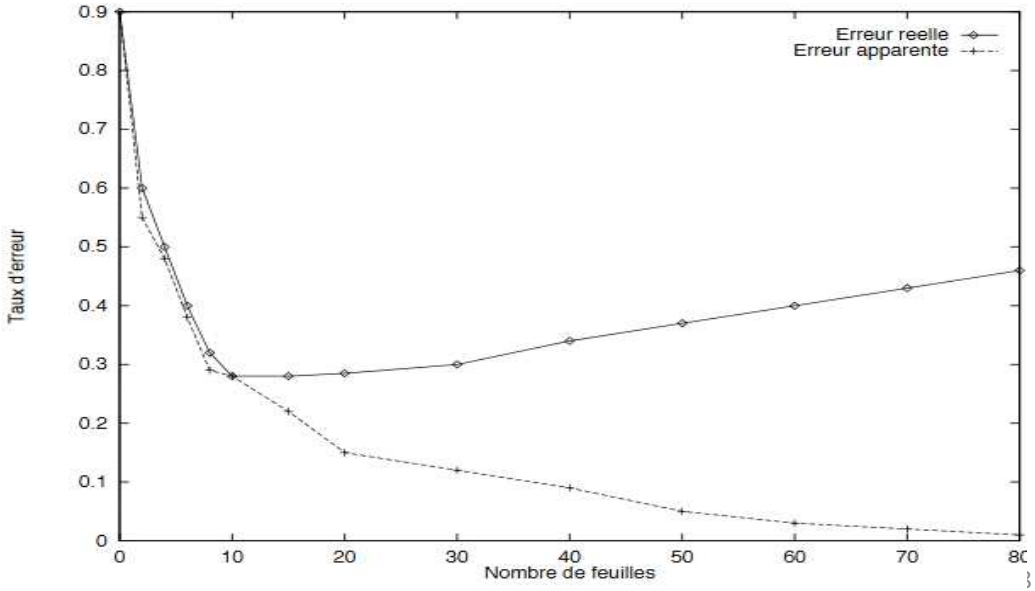
Le choix de la condition d'arrêt influe sur la profondeur de l'arbre et sur la précision de la prévision. Plus un arbre est profond, meilleure sera la prévision, mais plus il sera long à calculer et plus il y aura de risque de sur-apprentissage.
Exemple extrême d'un arbre ayant autant de feuilles que d'observations. Il n'a donc plus de capacité de généralisation.

Early stopping

Comme pour toute méthode de classification, pour éviter le sur-apprentissage, on utilise le taux d'erreur sur un échantillon test.

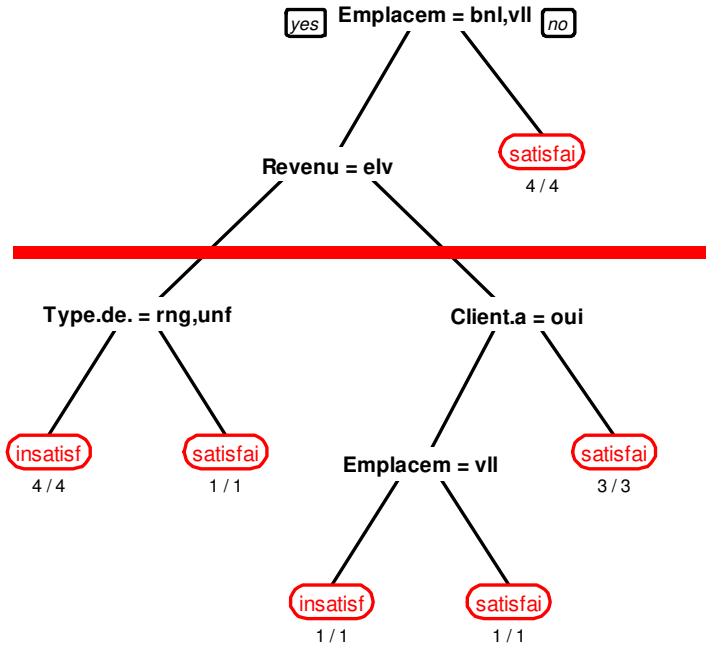
Comme pour le taux d'erreur d'apprentissage, le taux d'erreur de prédiction diminue lorsque le nombre de feuilles augmente, puis il stagne, puis il augmente. Au début, l'arbre apprend les caractéristiques générales de la population et à partir d'un certain moment, il n'apprend plus que les spécificités de l'échantillon.
Il faut donc arrêter la construction de l'arbre avant que le taux d'erreur de prévision n'augmente .

⇒ **Elagage de l'arbre**

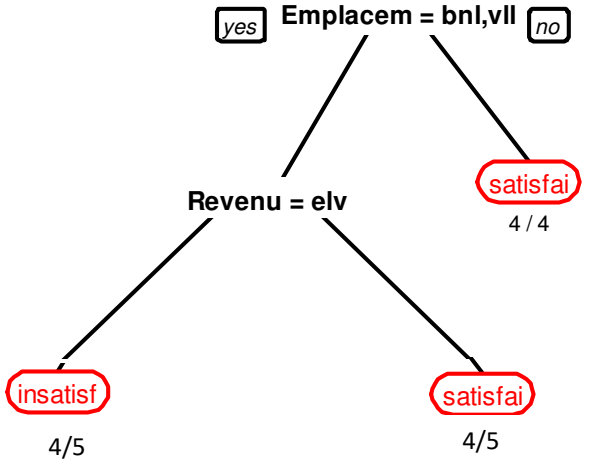


Elagage d'un arbre

L'élagage d'un arbre consiste à limiter la profondeur de l'arbre. L'objectif est d'éviter de les feuilles les plus profondes soient des cas particuliers.

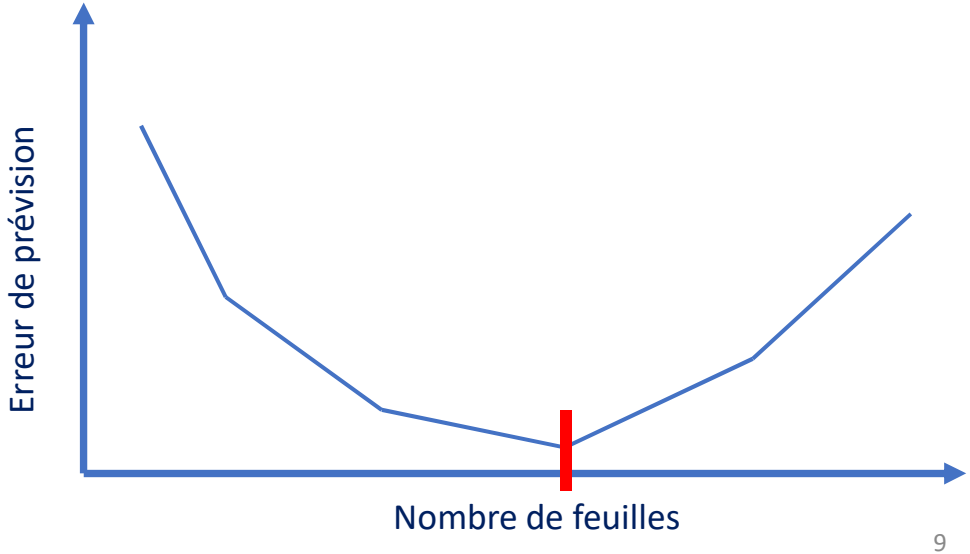


Elagage de l'arbre



Automatiquement l'élagage augmente l'erreur d'ajustement (car le modèle n'apprend plus les cas particuliers). En contre partie, il permet d'avoir une meilleure généralisation.

Le niveau d'élagage correspond à l'augmentation de l'erreur de prévision



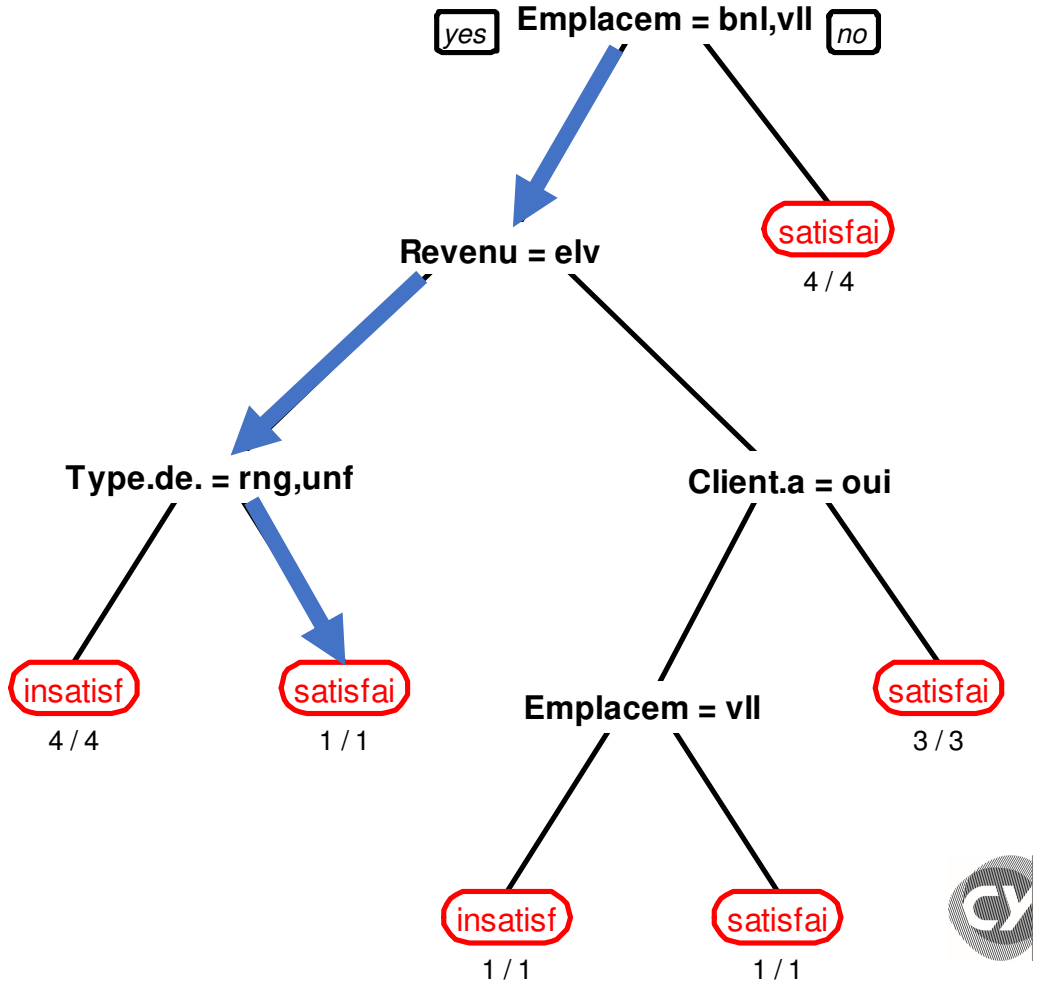
Comment faire une prévision?

Pour faire une prévision avec une nouvelle observation, il suffit de la faire « descendre » dans l'arbre jusqu'à une feuille.

Soit le nouvel individu défini par :

- *Emplacement=ville*
- *Revenu : Elevé*
- *Type de maison = jumelée*
- *Client antérieur = oui*

Satisfait

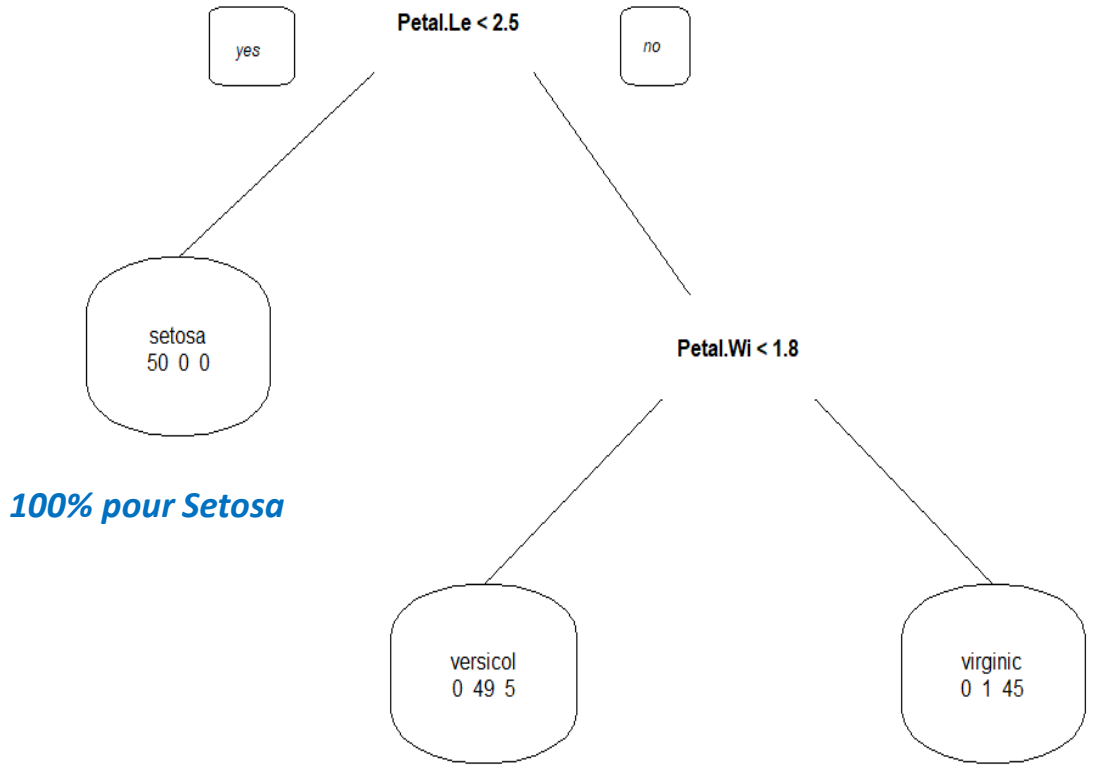


Probabilité d'appartenir à une classe

Le chemin dans l'arbre permet d'aboutir à une feuille. Dans une feuille, plusieurs classes de la variable cible peuvent être présentes.

Pour connaitre la classe prédite, on calcule la probabilité de chacune des classes dans la feuille et on choisit la plus probable.

Cette probabilité est importante pour connaitre la fiabilité de la prévision. La décision sera plus confortable si une nouvelle observation à 95% de chance d'appartenir à C_1 et 5% de chance d'appartenir à C_2, que si le pourcentage est de 51% pour C_1 et 49% pour C_2.



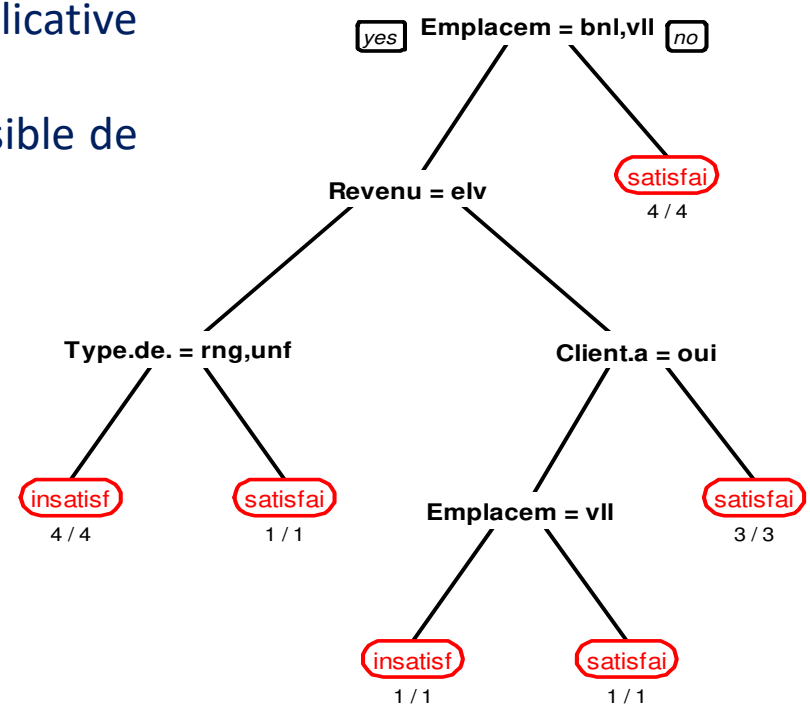
100% pour Setosa

90,74% pour Versicolor
9,26% pour Virginica

2,17pour Versicolor
97,83% pour Virginica

Gestion des données manquantes

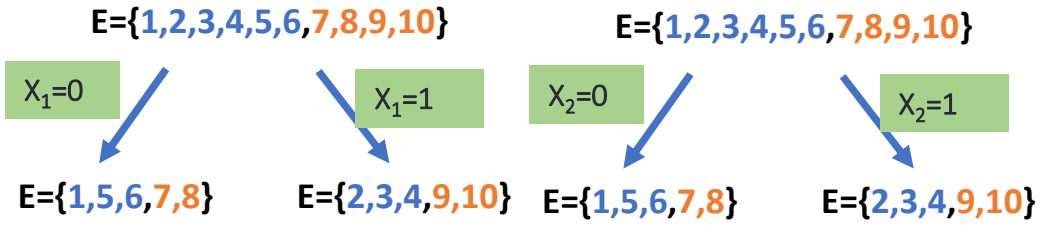
Supposons que pour un nouvel individu une variable explicative n'est pas renseignée.
Si cette variable intervient dans l'arbre alors il n'est pas possible de prédire la classe de ce nouvel individu.



Soit le nouvel individu défini par :

- *Emplacement=ville*
- *Revenu : Non renseigné*
- *Type de maison = jumelée*
- *Client antérieur = oui*

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 sont l'une des méthodes prédictives les plus utilisées car ils donnent des résultats remarquables en pratique et ils présentent beaucoup d'avantages :

- Prise en compte de variables mixtes (même si pour les quantitatives le choix du seuil reste problématique)
- Pas d'hypothèse sur la distribution des variables
- 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,…).

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,…

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.

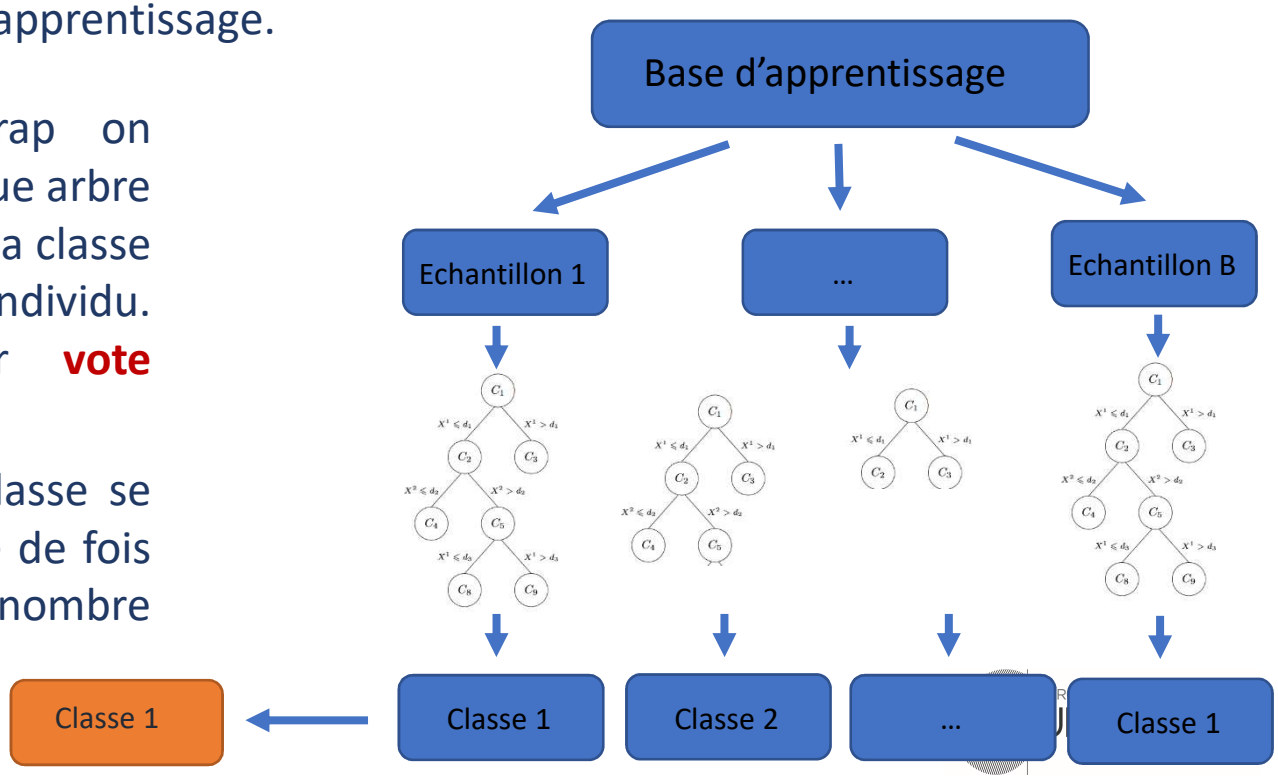
Un **échantillon bootstrap** est un tirage aléatoire d'éléments de la base d'apprentissage :

- 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 (~$\sqrt{p}$)

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?