Chapitre 3. Méthodes à base de partition

Claude Petit, Insee et université de Rennes - claude.petit@univ-rennes.fr

2025-2026

Overview

1. Principe des algorithmes à base de partition

2. Algorithme des k plus proches voisins

3. Arbres de décision et algorithme CART

1. Principe des algorithmes à base de partition

Introduction -1-

- $\mathcal{F}$ ensemble des fonctions de $\mathbb{X}$ dans $\mathbb{Y}$.
- $\mathcal{G} \subset \mathcal{F}$ dictionnaire.

1 méthode d'apprentissage supervisé $=$ 1 choix de dictionnaire $\mathcal{G}$.

Choix populaire : fonction constante / morceaux sur 1 partition de $\mathbb{X}$.

$\Rightarrow$ algorithme des plus proches voisins k-pp ou k-NN (pour k Nearest Neighbors), arbres de décision et de régression avec l'algorithme CART (CIAssification and Regression Tree).

$\mathcal{A} = (A_1, ..., A_M)$ partition de $\mathbb{X}$:

$$\mathbb{X} = \bigcup_{m=1}^{M} A_m \tag{1}$$

$\mathcal{G}(\mathcal{A}) \subset \mathcal{F}(\mathbb{X}, \mathbb{Y})$ ensemble des fonctions constantes sur chaque $A_m \in \mathcal{A}$.

Introduction -2-

La partition peut être aléatoire, déterminée par $\mathcal{D}_n$.

Soit $\widehat{g}_n = \widehat{g}_n(., \mathcal{A})$ le minimiseur du risque empirique sur le dictionnaire $\mathcal{G}(\mathcal{A})$ et N_m le nombre d'exemples appartenant à A_m :

$$\widehat{g}_n \in \underset{g \in \mathcal{G}(\mathcal{A})}{\arg\min}\, R_n(g) = \underset{g \in \mathcal{G}(\mathcal{A})}{\arg\min}\, \frac{1}{n} \sum_{i=1}^{n} l(Y_i, g(X_i)) \text{ et } N_m = \sum_{i=1}^{n} \mathbb{1}_{A_m}(X_i)$$

Classifieur binaire optimal

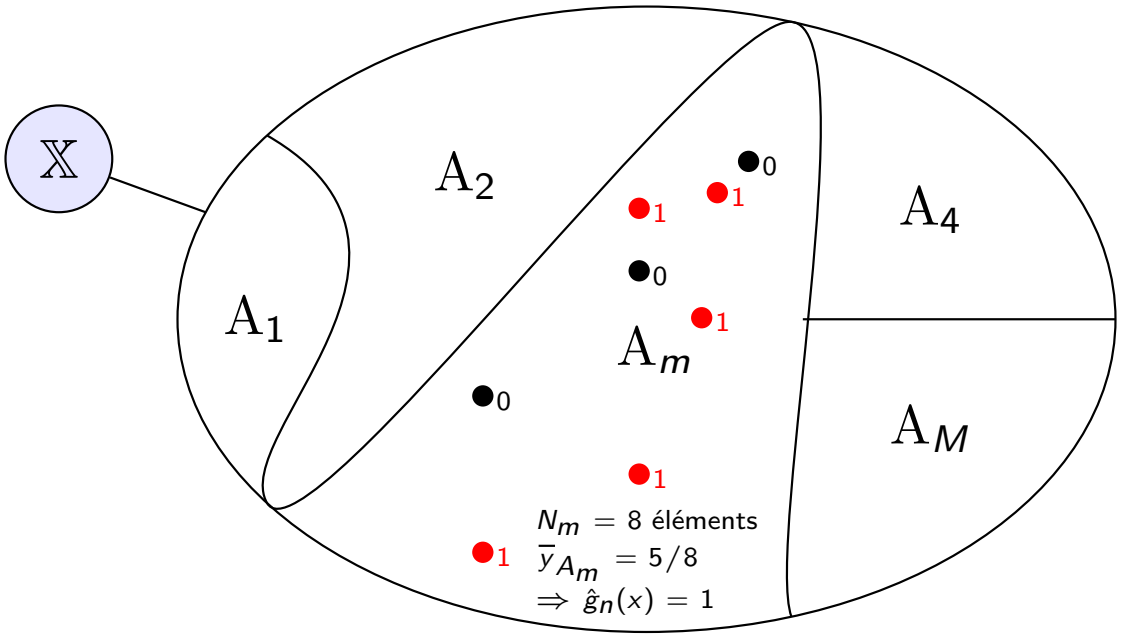
$$\forall m = 1..M, \forall x \in A_m,\ \widehat{g}_n(x, \mathcal{A}) = \begin{cases} 1 & \text{si } \overline{Y}_{A_m} > 1/2 \\ a_m & \text{si } \overline{Y}_{A_m} = 1/2 \\ 0 & \text{si } \overline{Y}_{A_m} < 1/2 \end{cases} \quad (2)$$

Minimiseur du risque empirique pour la régression MC

$$\widehat{g}_n(x, \mathcal{A}) = \sum_{m=1}^{M} \overline{Y}_{A_m} \mathbb{1}_{A_m}(x) = \sum_{m=1}^{M} \left(\frac{1}{N_m} \sum_{i=1}^{n} Y_i \mathbb{1}_{A_m}(X_i) \right) \mathbb{1}_{A_m}(x) \quad (3)$$

Démonstration en exercice !

Introduction -3-



Méthodes à base de partition -1-

Soit $\mathcal{A} = (A_1, ..., A_m)$ une partition de $\mathbb{X}$ et $\mathcal{G} = \mathcal{G}(\mathcal{A})$ classe des fonctions constantes sur chaque élément de la partition :

$$\mathcal{G} = \{g : \mathbb{X} \longrightarrow \mathbb{Y} : \ \forall m = 1, ..., M, \ g \text{ constante sur } A_m\} \tag{4}$$

Le minimiseur du risque empirique associé à cette partition est la fonction

$$\widehat{g}_n = \widehat{g}_n(., \mathcal{A}) = \underset{g \in \mathcal{G}(\mathcal{A})}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} l(Y_i, g(X_i)) \tag{5}$$

Notons N_m le nombre d'observations X_i de l'échantillon qui se trouvent dans l'élément A_m de la partition :

$$N_m = \sum_{i=1}^{n} \mathbb{1}_{A_m}(X_i) \tag{6}$$

Méthodes à base de partition -2-

et notons $\overline{Y}_m$ la moyenne des étiquettes observées dans l'élément A_m :

$$\overline{Y}_m = \sum_{i=1}^{n} Y_i \mathbb{1}_{A_m}(X_i) \tag{7}$$

Alors le minimiseur du risque empirique pour la classification binaire est, si $x \in A_m$,

$$\widehat{g}_n(x) = \mathbb{1}_{[\overline{Y}_m > 1/2]} \tag{8}$$

$\forall m = 1, ..., M$. Autrement dit :

$$\widehat{g}_n(x) = \sum_{m=1}^{M} \mathbb{1}_{[\overline{Y}_m > 1/2]} \mathbb{1}_{A_m}(x) \tag{9}$$

Méthodes à base de partition -3-

On généralise facilement de classifieur binaire à une situation multiclasses avec $\mathbb{Y} = \{1, ..., K\}$. $\forall x \in \mathbb{X}$,

$$\widehat{g}_n(x) = \underset{k=1,...,K}{\arg\max} \sum_{m=1}^{M} \left(\frac{1}{N_m} \sum_{i=1}^{n} \mathbb{1}_{[Y_i=k]} \mathbb{1}_{A_m}(X_i) \right) \mathbb{1}_{A_m}(x) \tag{10}$$

$$= \underset{k=1,...,K}{\arg\max} \sum_{m=1}^{M} w_{m,k} \mathbb{1}_{A_m}(x) \tag{11}$$

avec $w_{m,k} = N_m(k)/N_m$ proportion des observations dans A_m pour lesquelles $Y = k$.

2. Algorithme des k plus proches voisins

Méthode des k-ppv (k-NN)

Soit $k \leq n$. $\forall x \in \mathbb{R}^d,\ \forall i = 1, ..., n,\ d_i(x) = ||X_i - x||$
Soient $r_i(x)$ indice du $i$ème ppv de x parmi $X_1, ..., X_n$:

$$r_1(x) = j \iff \begin{cases} d_j(x) = \min\limits_{i=1,...,n} d_i(x) \\ d_j(x) < \min\limits_{1 \leq i < j} d_i(x) \end{cases} \tag{12}$$

(13)

et par récurrence sur $k \geq 1$,

$$r_k(x) = j \iff \begin{cases} d_j(x) = \min\limits_{i=1..n; i \neq r_1, .., r_{k-1}} d_i(x) \\ d_j(x) < \min\limits_{1 \leq i < j; i \neq r_1, .., r_{k-1}} d_i(x) \end{cases} \tag{14}$$

(15)

Partition définie par les rangs

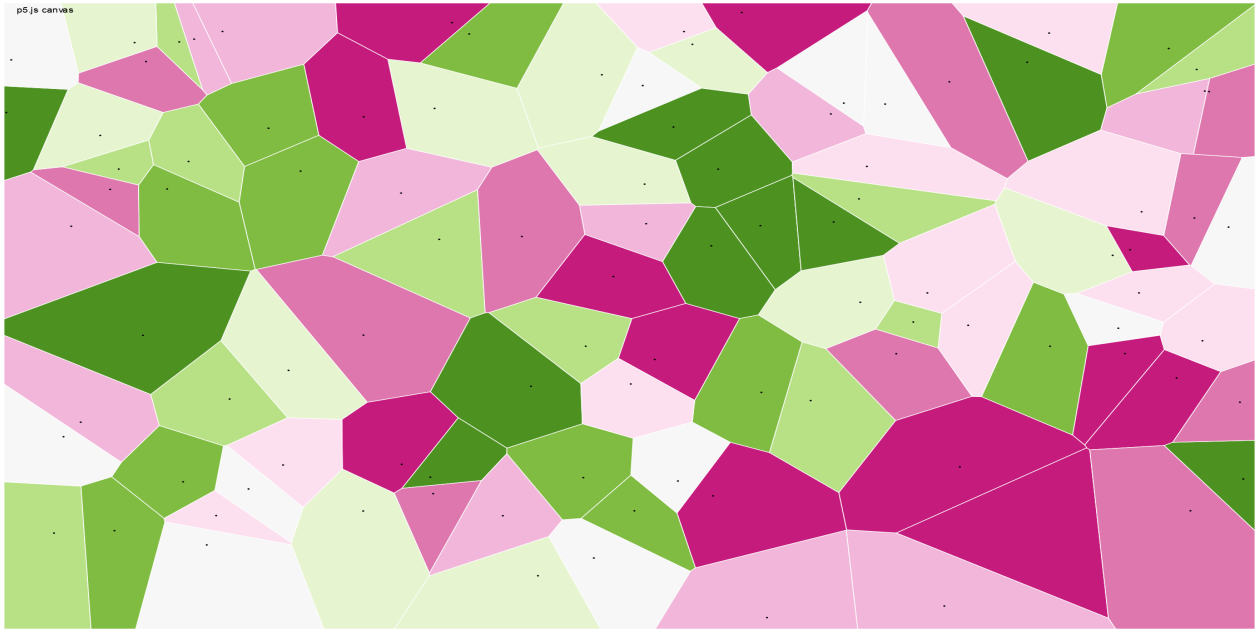
$\forall k \in [\![1, .., n]\!]$, les $d_i(x)$ définissent une partition $\mathcal{A}_k = (A_{m,k})_m$ telle que, $\forall m = 1, ..., M$:

$$A_m = \{x \in \mathbb{X} = \mathbb{R}^d : \ C_m = (r_1(x), ..., r_k(x))\} \tag{16}$$

C_m combinaison de k éléments parmi n et $M = \binom{n}{k}$. Les A_m sont les parties de $\mathbb{X}$ pour lesquelles $x \mapsto (r_1(x), ..., r_k(x))$ est constante : la partition forme des zones caractérisées par un choix de k observations parmi n et les points de cette zone sont les plus près des k observations caractérisant la zone. Pour $x, x' \in A_m$, les k-ppv de x et x' parmi $X_1, ..., X_n$ sont les mêmes. Si $x \in A_m$ et $x' \in A_{m'}$ avec $m \neq m'$, les k -ppv de x sont $\neq$ des k-ppv de x'.

Diagrammes de Voronoi -1-

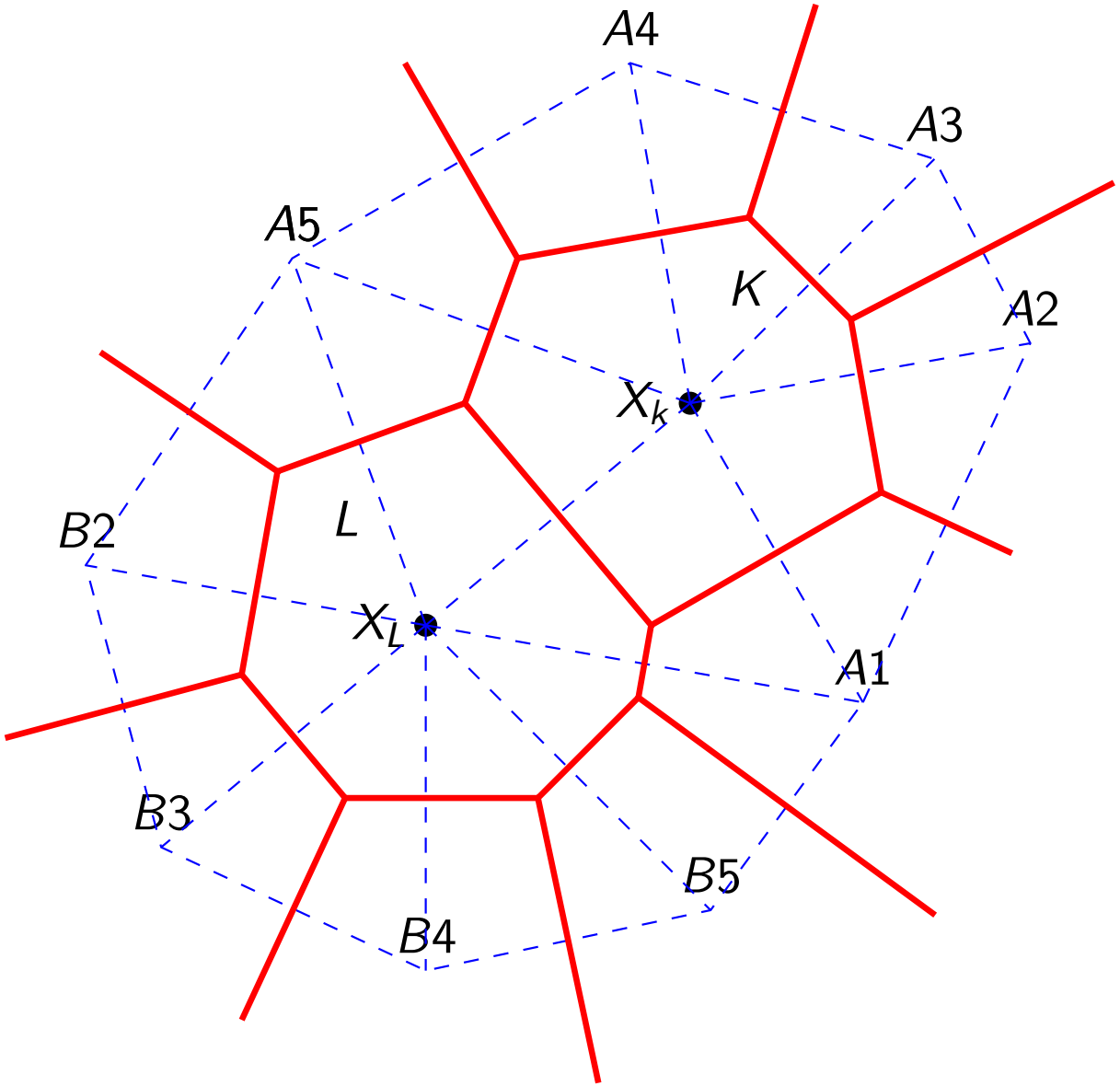
Ces partitions forment des diagrammes de Voronoi. Chaque zone est une cellule de Voronoi. Les figures illustrent des partitions ppv pour $k = 1$.



Cellules de Voronoi pour $k = 1$.

https://strongriley.github.io/d3/ex/voronoi.html

Diagrammes de Voronoi -2-



Prédicteur et classifieur des k-ppv

Prédicteur k-NN pour la régression aux moindres carrés

$$\widehat{\eta}_{n,k}(x) = \sum_{m=1}^{M} \overline{Y}_{A_m} \mathbb{1}_{A_m}(x) \tag{17}$$

où $(A_m)_m$ partition définie à partir des rangs.

Classifieur binaire k-NN

$$\widehat{g}_{n,k}(x) = \mathbb{1}_{[\widehat{\eta}_{n,k}(x) > 1/2]} = \sum_{m=1}^{M} \mathbb{1}_{]1/2,1]}(\overline{Y}_{A_m}) \mathbb{1}_{A_m}(x) \tag{18}$$

où $(A_m)_m$ partition définie à partir des rangs.

- k-NN $=$ k-ppv !

Consistance de l'algorithme k-NN

Pour que k-NN soit consistant, il faut choisir $k = k_n$ fonction croissante de la taille de l'échantillon (cf. point sur le sur-apprentissage évoqués dans le chapitre précédent concernant la taille de $\mathcal{G}$).

Théorème admis : consistance de k-NN

Si k_n tend vers $+\infty$ lorsque $n \to +\infty$ moins vite que n, c'est à dire $k/n \to 0$, alors le prédicteur k-NN est consistant. Il en est de même du classifieur k-NN.

En exercice, nous démontrerons que k-NN pour $k = 1$ n'est pas consistant. La démonstration générale de la consistance de k-NN est assez difficile.

Exemple des iris de Fisher -1-

- Exemple incontournable : classification des iris de Fisher.
- Utilisé par Ronald Fisher en 1936 (données d'Edgar Anderson).
- 50 échantillons des 3 espèces d'iris (setosa, virginica et versicolor). 4 variables statistiques : longueur et largeur des sépales et des pétales.

Figure 1 – Les 3 espèces d'Iris (de gauche à droite versicolor, setosa, virginica), Crédit photo : Frank Mayfield et Kosaciec Szczecinkowaty, Wikipédia.

Exemple des iris de Fisher -2-

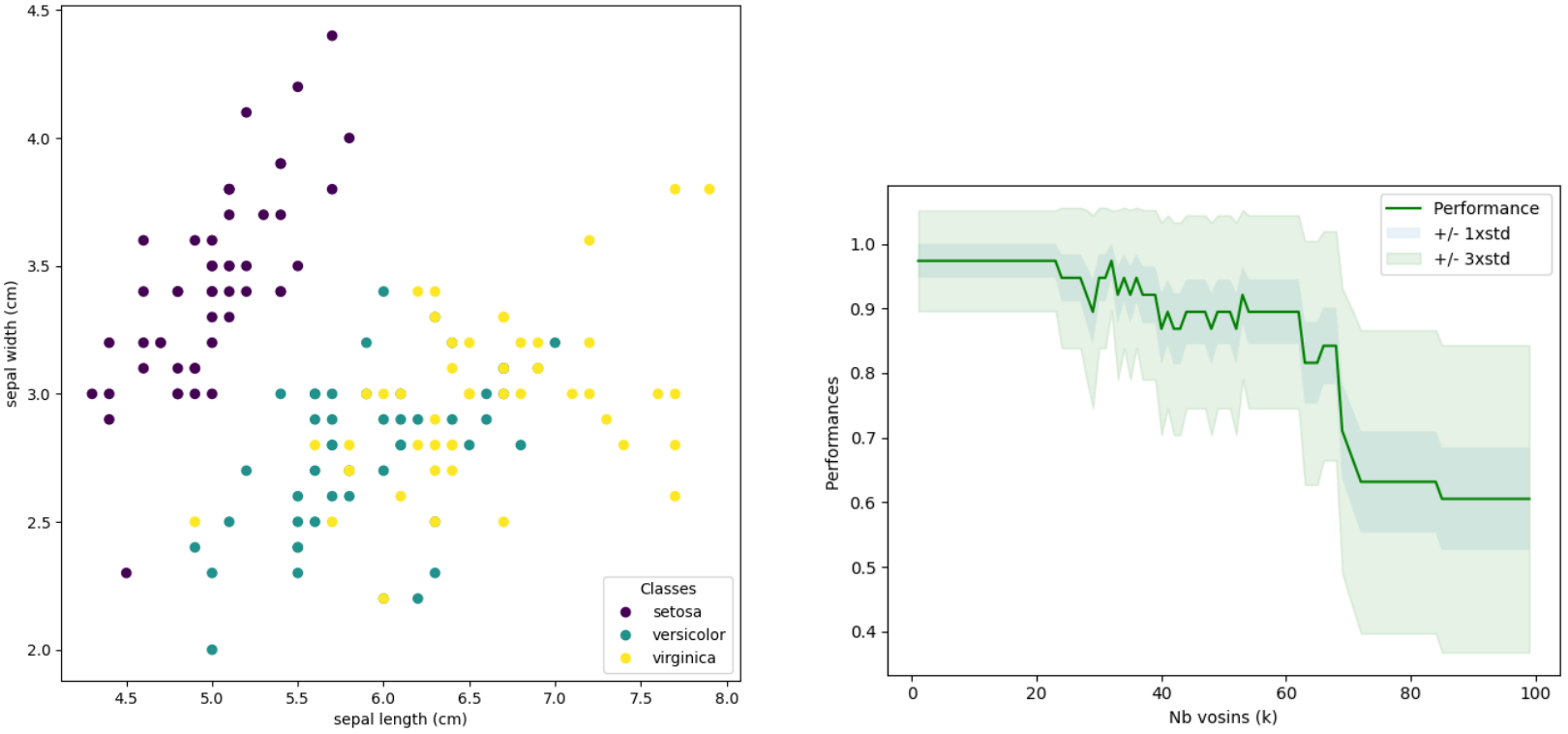


Figure 2 – À gauche, nuage de points représentant les iris de Fisher (une espèce par couleur) en fonction de la longueur et de la largeur de leur sépale. À droite, une courbe indiquant les performances de k-ppv sur la tâche de classification des iris, en fonction du nombre k.

Exemple des iris de Fisher -3-

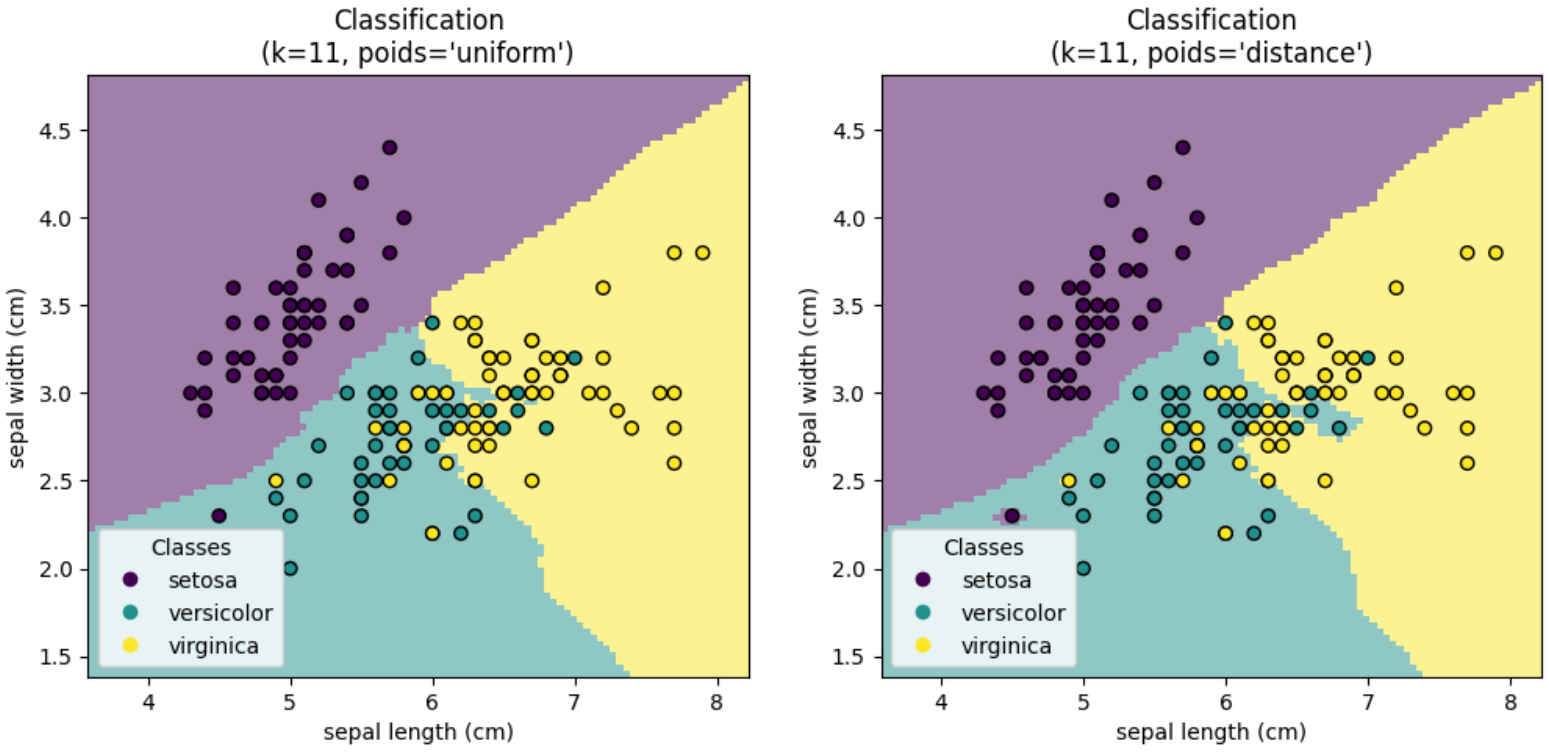


Figure 3 – Illustration d'un classifieur k-ppv pour $k = 11$, avec deux types de poids différents. Les zones de couleur représentent l'espèce qui sera affectée à une nouvelle observation selon la longueur et la largeur de son sépale. Simulations effectuées sous Python.

k-NN en régression

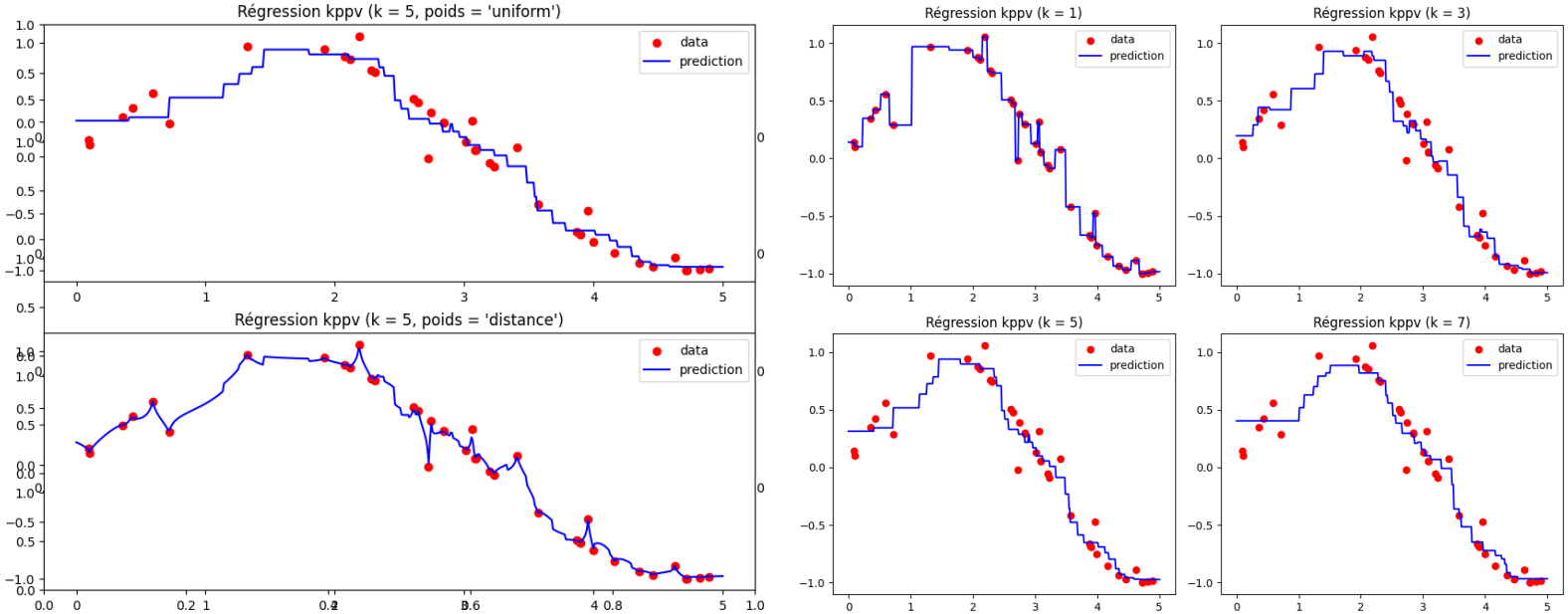


Figure 4 – Régression à l'aide de la méthode k-ppv. Un nuage de points (rouges) bruité est généré en perturbant aléatoirement les points d'une sinusoïde. La courbe bleue, issue de la méthode des k-ppv, doit reconstituer au mieux cette sinusoïde. Les deux courbes à gauche correspondent à $k = 5$ pour deux types de poids différents. Les 4 courbes de droite représentent la régression pour des valeurs différentes de $k = 1, 3, 5, 7$. Simulations effectuées sous Python.

3. Arbres de décision et algorithme CART

Arbre de décision -1-

Un graphe est un ensemble de nœuds reliés par des arêtes. Un arbre est un graphe sans cycle.

Dans un arbre de décision, chaque nœud correspond à un sous-ensemble $A \subset \mathbb{X}$ et un test statistique T (critère de segmentation, d'impureté) est appliqué sur les variables explicatives $x \in \mathbb{X}$. Si le test peut donner K résultats $1, ..., K$, alors le nœud (A, T) donne naissance à k nœuds fils, tel que l'ensemble A_k associé au k^{e} fils est $A_k = \{x \in A : T(x) = k\}$.

L'algorithme est initialisé à la racine correspondant à $A = \mathbb{X}$, puis on itère le procédé. Des branches sont élaguées si elles ne dégradent pas trop le taux d'erreur de l'arbre.

Différentes implémentations possibles selon :

- Les critères de segmentation choisis.
- Le critère d'arrêt.
- Le critère d'élagage.

Arbre de décision -2-

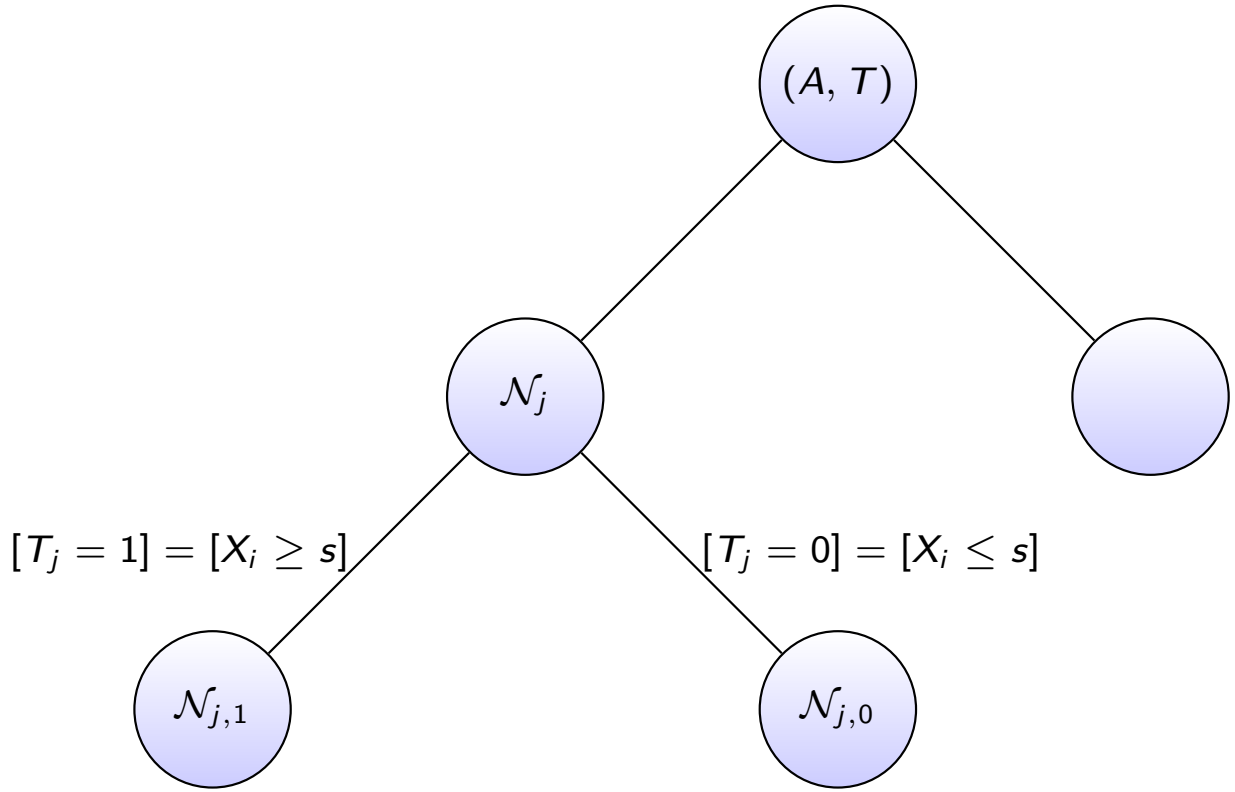


Figure 5 – Exemple d'arbre avec 5 nœuds et 4 arêtes. La racine A correspond à l'ensemble $\mathbb{X}$. À un nœud donné $\mathcal{N}_j$ de profondeur j est affecté un sous-ensemble de $\mathbb{X}$. Le test courant T_j est effectué sur les observations contenues dans $\mathcal{N}_j$, qui sont réparties dans les deux nœuds fils $\mathcal{N}_{j,0}$ et $\mathcal{N}_{j,0}$ en fonction du résultat de T_j.

Algorithme CART de Breiman, 1984

Algorithm 1: CART algorithm

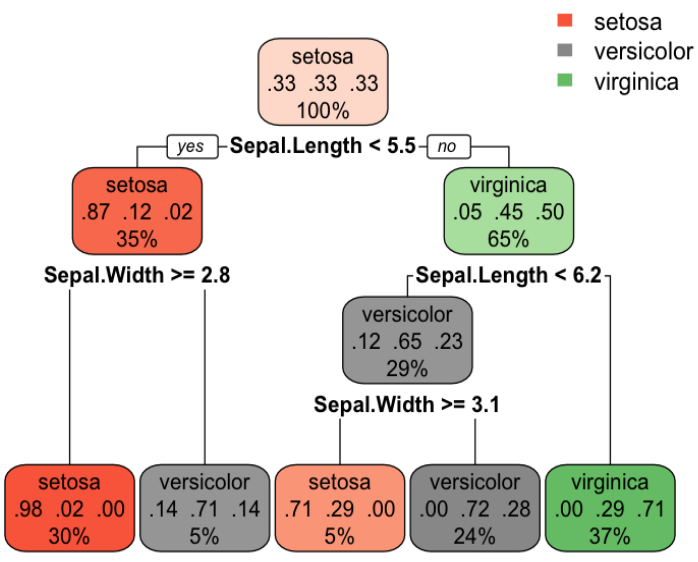
Input: Arbre = nœud racine

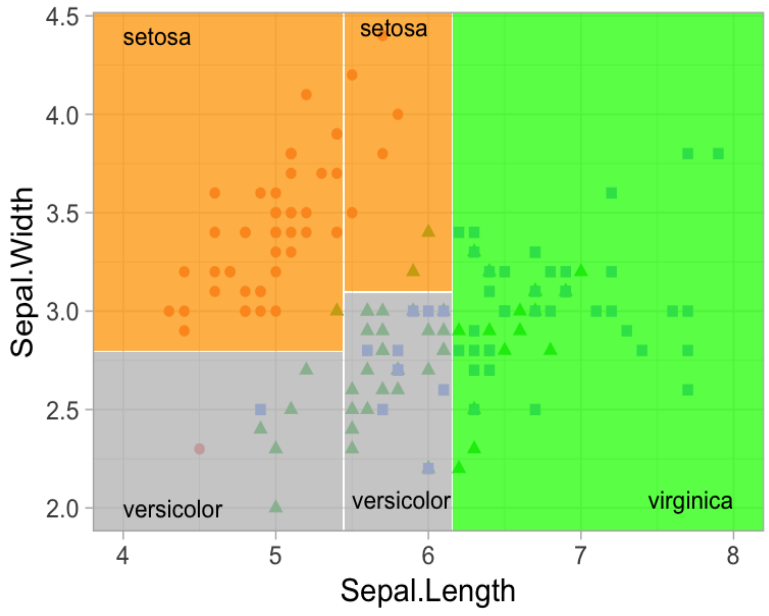
```
// Expansion
for chaque nœud n de Arbre do
    if n ≠ condition d'arrêt then
        Choisir critère de segmentation T
        Créer les nœuds fils
        Maj : Arbre = Arbre ∪ nœuds fils
    end
end
// Élagage
for chaque nœud n de Arbre do
    if n = condition d'élagage then
        Maj : Arbre = Arbre − nœuds fils et descendants
    end
end
```

CART exemple 1 : iris de Fisher (2 variables)

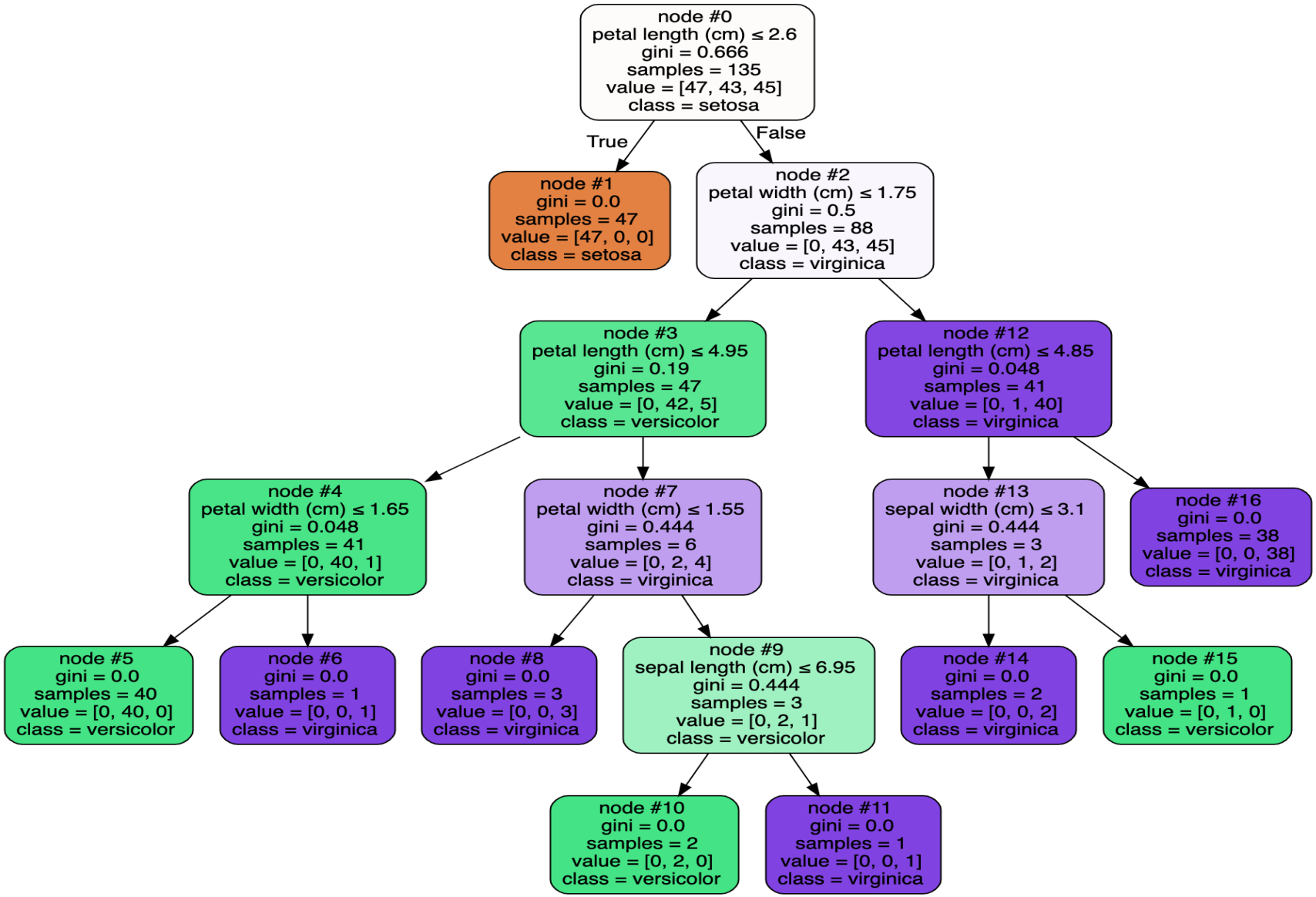
L'algorithme CART est appliqué à la base de données des iris de Fisher.

• À gauche, l'arbre de décision construit sur les variables longueur et largeur de la sépale.
• À droite, la partition de l'ensemble $\mathbb{X} = \mathbb{R}^2$ en fonction des critères.

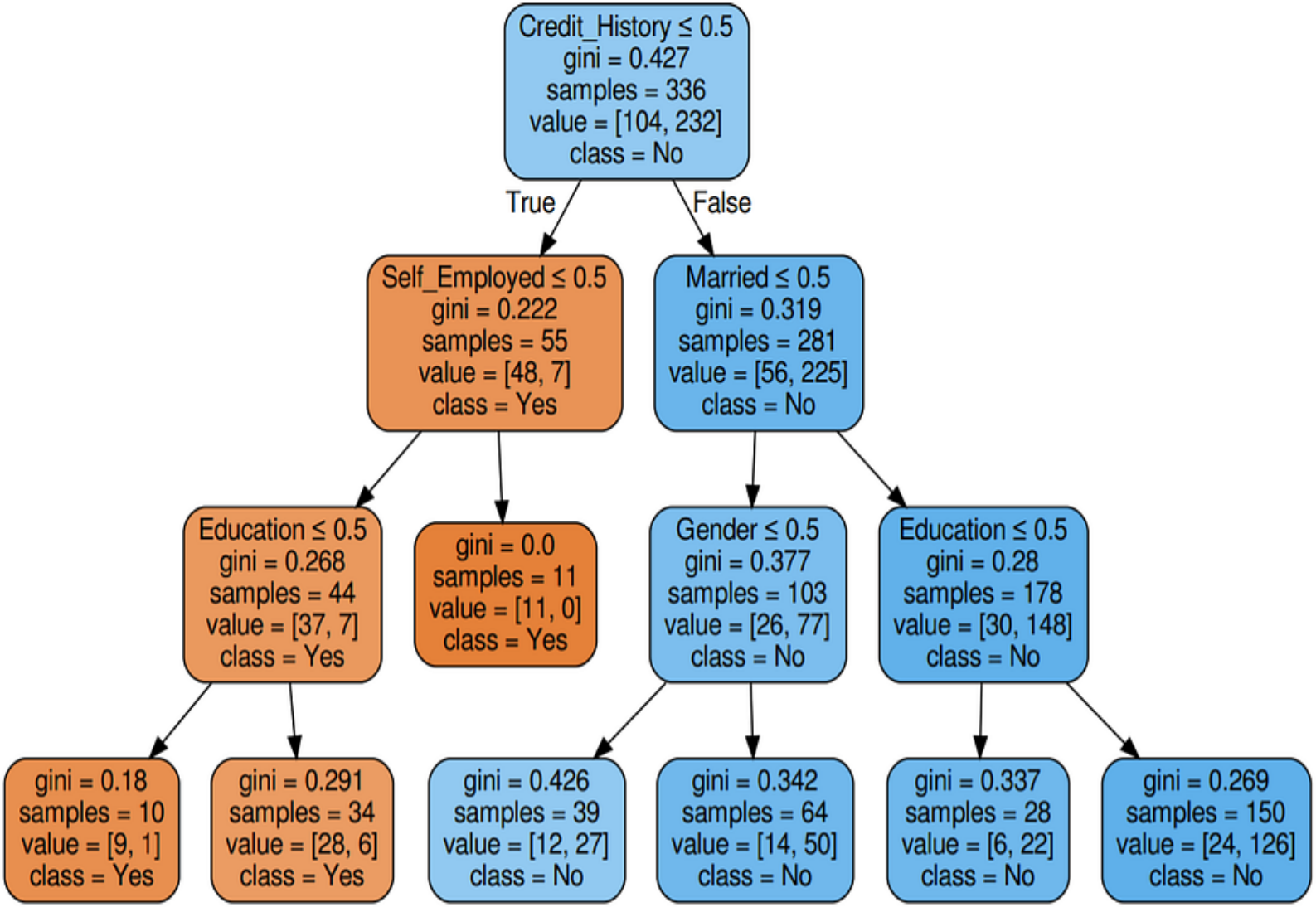




CART exemple 2 - Iris de Fisher (4 variables)



CART exemple 3 : base de données des défauts de paiement



Règle de classification

Une fois l'arbre construit, la règle de classification consiste simplement à parcourir l'arbre depuis la racine :

- Pour un $x \in \mathbb{X}$, on détermine la feuille (nœud terminal) qui le contient en parcourant l'arbre de haut en bas.
- En classification, on affecte à x l'étiquette y correspondant à la classe majoritairement représentée par les exemples x_i de cette feuille.
- En régression, on affecte à x la moyenne des étiquettes y_i correspondant aux exemples x_i de la feuille.

CART : segmentation, critère d'arrêt et élagage

Le critère d'arrêt vérifie l'une des conditions suivantes sur un seuil donné :

- Chaque feuille satisfait un seuil d'homogénéïté.
- La profondeur de l'arbre dépasse le seuil.
- Le nombre de feuilles dépasse le seuil.
- L'effectif du nœud est inférieur au seuil.

La segmentation peut produire de bonnes performances sur l'ensemble des données d'entrainement, mais est susceptible de provoquer du sur-apprentissage si l'arbre est trop complexe. Un arbre avec moins de branches aura une variance plus faible en contrepartie d'un biais un peu plus fort.

Un stratégie pour simplifier l'arbre est d'élaguer des branches qui ne contribuent pas trop à augmenter le risque estimé, c'est le processus d'élagage (« pruning »).

Critères de segmentation 1 : l'indice de Gini

- Indice de Gini ou entropie pour classification, variance pour régression.

Définition de l'indice de Gini

$$\forall A \subset \mathbb{X},\ G(A) = 1 - \overline{Y}_A^2 - (1 - \overline{Y}_A)^2 \tag{19}$$

- $G(A) = 0 \iff \overline{Y}_A = 0$ ou 1.

- Un nœud sera de bonne qualité (homogène, pur) si une très grande majorité des étiquettes des exemples associés à ce nœud sont identiques. Il est alors très discriminant et $G(A)$ est presque nul.

- Pour évaluer la qualité d'un critère de segmentation, on calcule le gain d'homogénéité lorsque A est segmenté en A_1 et A_2 :

$$I_G(A_1, A_2) = G(A) - qG(A_1) - (1 - q)G(A_2) \tag{20}$$

avec $q = N_{A_1}/A_{A_2}$ proportion des $x_i \in A$ qui se dirigent vers A_1. CART choisit la partition qui maximise I_G à chaque étape.

Critères de segmentation 2 : l'entropie

La quantité d'information de Shannon apportée par la réalisation d'un évènement A est $\mathbb{I}(A) = -\mathbb{P}(A)\log_2 \mathbb{P}(A)$. Elle mesure la vraisemblance de cet évènement.

L'entropie d'une va mesure l'information moyenne sur l'ensemble des réalisations possibles. C'est le nombre moyen de questions binaires que l'on doit poser pour déterminer la valeur exacte de la variable aléatoire.

Entropie d'une va ou d'une mesure de probabilité

$$H(X) = -\mathbb{E}[\log_2 \mathbb{P}(X)] = -\sum_{i=1}^{n} p_i \log_2 p_i \tag{21}$$

On l'utilise comme critère de segmentation en calculant l'entropie de la mesure de probabilité empirique uniforme créée par la partition des données (p_i proportion de 1 (ou 0) dans chaque classe).

Critères de segmentation 3 : la variance en régression

• On rappelle que dans une tâche de régression, la valeur affectée à un nœud est la moyenne des observations appartenant à ce nœud.

• L'hétérogénéïté est mesurée par la variance du nœud. Si l'on note $\mathcal{N}$ le nœud et $\overline{y}$ la moyenne des valeurs y_i des observations se trouvant dans le nœud correspondant,

$$\mathbb{V}(\mathcal{N}) = \frac{1}{|\mathcal{N}|} \sum_{i:x_i \in \mathcal{N}} (y_i - \overline{y})^2. \tag{22}$$

• La segmentation privilégie le découpage en deux nœuds homogènes dont la variance sera la plus faible possible.

Critère d'élagage CART -1-

- Élaguer pour éviter le sur-apprentissage et garder un modèle simple.
- Complexité d'un arbre donnée par sa dimension de Vapnik (hors programme) ou bien son nombre de coupures ou sa profondeur.
- Tester tous les sous-arbres ? Trop coûteux. Méthode de Breiman : se limiter à une suite de sous-arbres emboités de taille raisonnable.
- Choisir un arbre de la suite par minimisation d'un risque d'ajustement.

Si T arbre dont les noeuds terminaux sont les $\mathcal{N}_m$. Deux risques d'ajustement classiques (respectivement régression et classification)

$$R_r(m) = \frac{1}{N_m} \sum_{i:x_i \in \mathcal{N}_m} (y_i - \overline{y}_m)^2 \text{ et } R_c(m) = \frac{1}{N_m} \sum_{i:x_i \in \mathcal{N}_m} \mathbb{1}_{[y_i \neq \overline{y}_m]} \quad (23)$$

Critère d'élagage CART -2-

- On définit un critère côut/complexité par

$$C_\alpha(T) = \sum_{m=1}^{M} N_m R_m(T) + M\alpha \tag{24}$$

- $M = |T|$ nombre de nœuds de T, $\alpha \geq 0$ paramètre d'ajustement.

- La somme représente l'erreur totale (le risque) sur l'arbre, le paramètre αM pénalise le nombre total de feuilles (la complexité).

- On cherche l'arbre T qui minimise $C_\alpha(T)$ pour α bien choisi ($\alpha = 0$ arbre entier, $\alpha = \infty$ racine uniquement).

- Approche CART de Breiman : régularise pour améliorer les performances en prédiction, évite toute l'exploration de l'espace des solutions, intègre une préférence pour la simplicité (règle de l'écart-type).

Critère d'élagage CART -3-

Théorème de Breiman, 1984

Il existe une suite finie $0 = \alpha_0 < ... < \alpha_M$ avec $M \leq |T|$ et une suite imbriquées de sous-arbres $(T_{\alpha_m})_m$ avec

$$T = T_{\alpha_0} \subseteq T_{\alpha_1} \subseteq ... \subseteq T_{\alpha_M} = \text{racine} \tag{25}$$

telle que $\forall \alpha \in [\alpha_m, \alpha_{m+1}[$,

$$T_m \in \underset{T_i \subset T}{\arg\min}\, C_\alpha(T_i) \tag{26}$$

• Choisir un arbre revient à choisir une valeur de α : sur le chemin de coût/complexité, en augmentant α, on trouve une succession d'arbres emboîtés de taille décroissante.

• Vidéo de Ricco Rakotomalala :
https://www.youtube.com/watch?v=if6QEtJP77E

Mesure d'importance d'un arbre

- La visualisation de l'arbre peut donner une idée sur l'importance des variables.
- Mais certaines variables possèdent une grande importance sans apparaitre explicitement dans l'arbre.
- Difficile de quantifier l'importance juste en regardant l'arbre !
- C'est le même problème que la significativité d'un régresseur dans une régression linéaire ou logistique.
- La mesure d'importance d'un arbre est basée sur le gain d'impureté des noeuds internes.

Plus de détails seront donnée en TP.

CART : exemple en classification

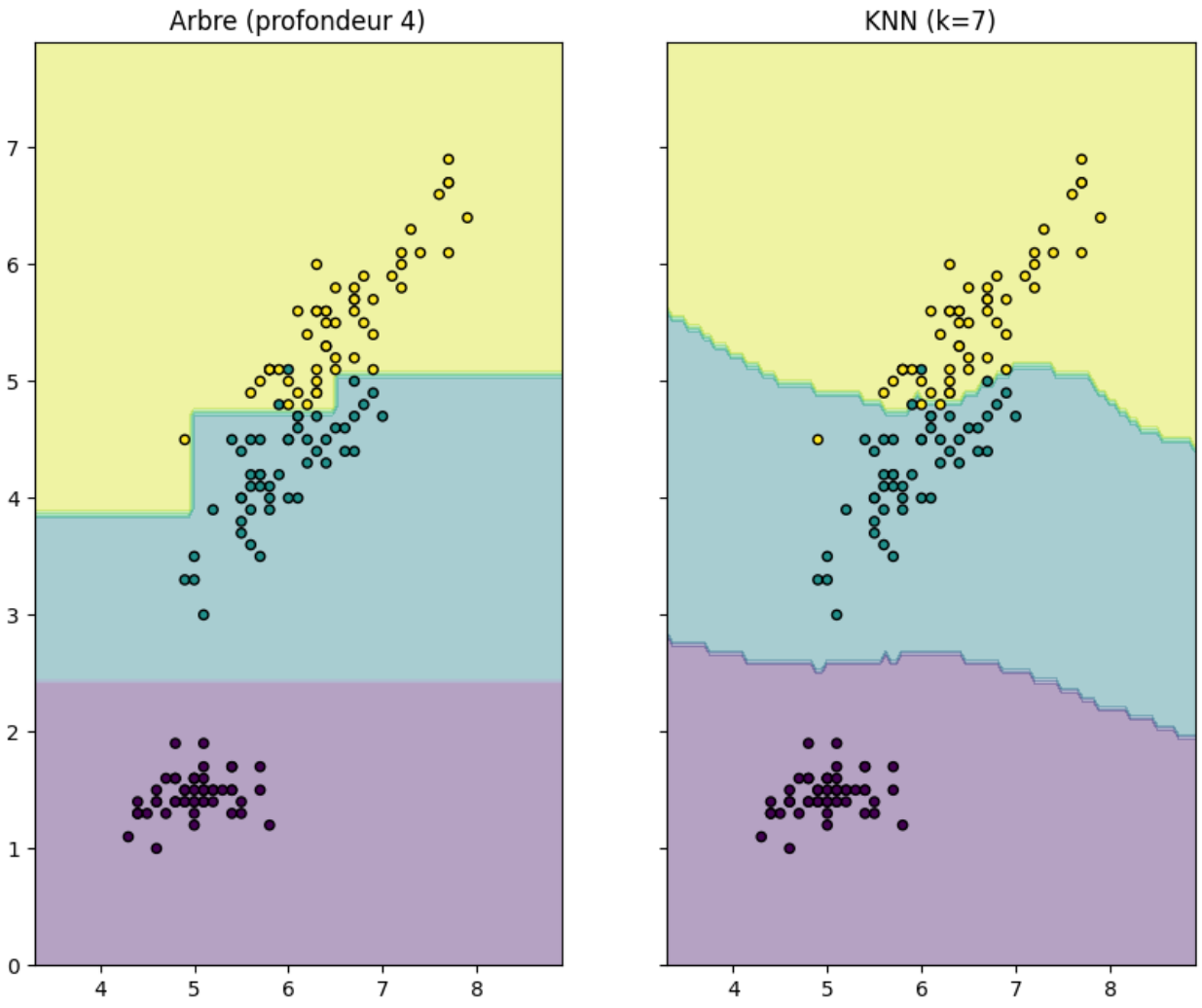


Figure 6 – Classification des iris de Fisher par arbre CART profondeur 4 (à gauche) et k-ppv pour $k = 7$ (à droite).

CART : exemple en régression

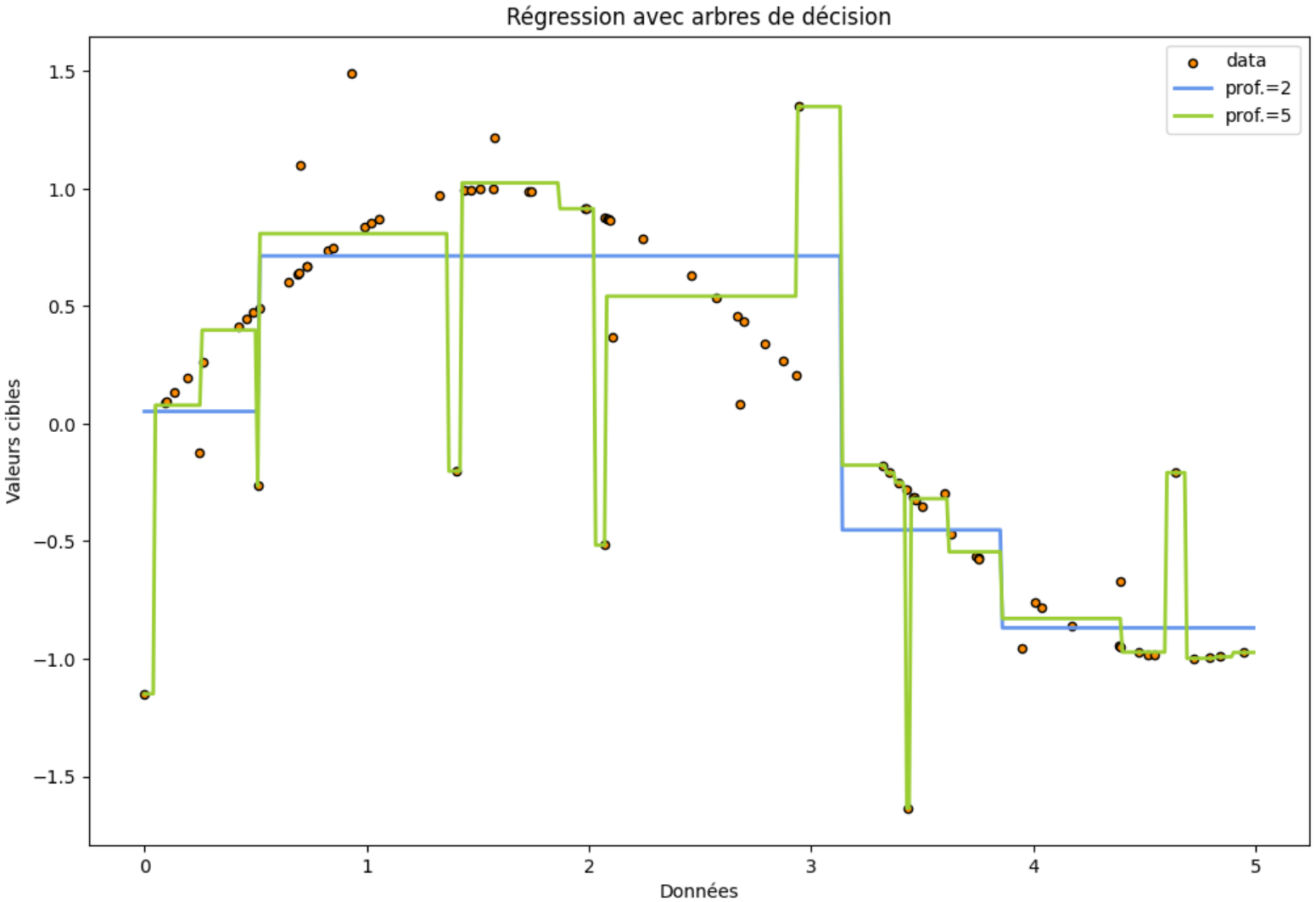
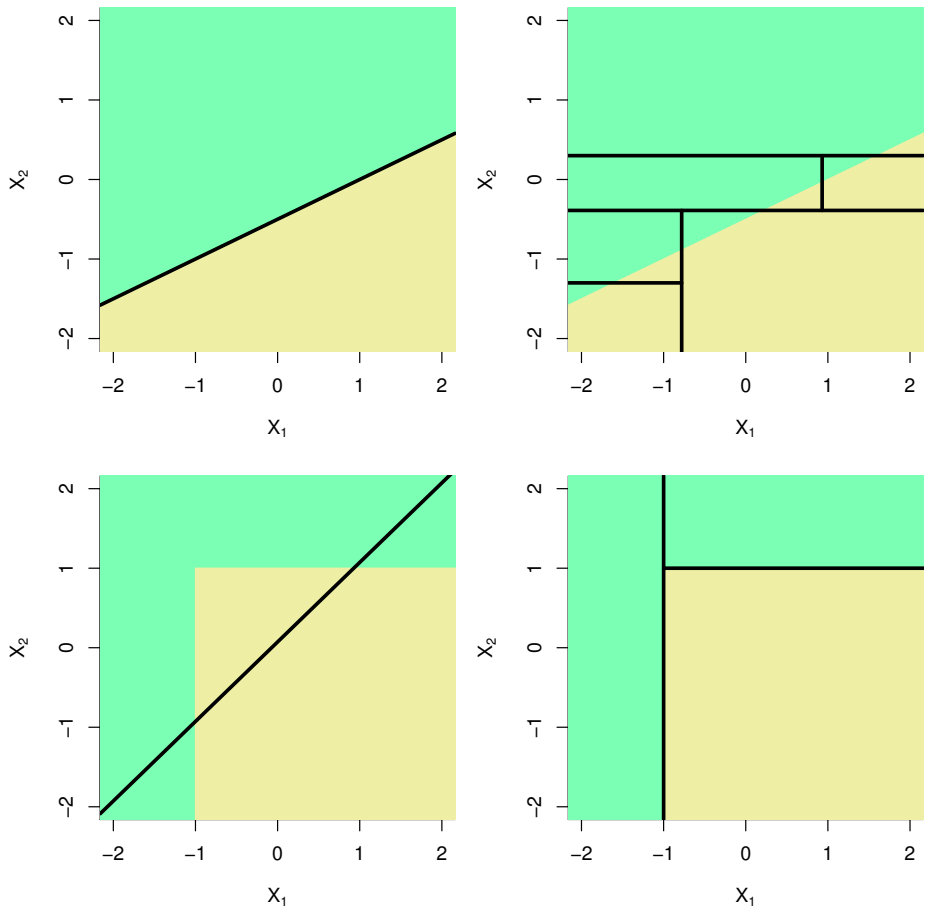


Figure 7 – Régression CART. Nuage de points (oranges) généré avec sinusoïde + bruit aléatoire. Objectif : reconstruire la courbe avec arbre de profondeur 2 (en bleu) ou 5 (en vert).

Arbres : une méthode non linéaire

En fonction des données, les arbres peuvent obtenir de meilleures performances (ou pas) que les méthodes linéaires.



Remarques finales

- Les arbres sont des modèles hiérarchiques, non linéaires, itératifs.
- Ils sont adaptés aux problèmes continus, discrets, multiclasses, multimodes.
- Ils prédisent par l'intermédiaire d'une stratification (partition) de l'espace des données.
- Les arbres proposent une méthode graphique intuitive et facile à comprendre.
- Ils ne sont pas performants !
- Ils ne sont pas robustes au bruit dans les données.
- Variantes diverses : CID3, C4.5, C5, CHAID, MARS, QUEST, etc.

Des méthodes plus robustes et ayant de meilleures performances peuvent se déduire des arbres de décision : les forêts aléatoires et le bagging, par exemple, dont nous parlerons dans un autre chapitre.