Chapitre 5. Méthodes d'agrégation

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

2025-2026

Overview

Bootstrap et ré-échantillonnage

Bagging

Stacking

Comparaison des méthodes

Bootstrap et ré-échantillonnage

Bootstrap : Efron, 1979

• Créer des échantillons aléatoires permettant de simuler une loi...
... quand on n'a pas beaucoup d'échantillons disponibles.

• Principe : générer des échantillons qui ressemblent à l'échantillon de départ.

• Rudolph Raspe : les aventures du baron de Münchhausen (1785).

• « to pull oneself up by one's bootstraps ».
$\Rightarrow$ Se hisser en tirant sur les languettes de ses bottes.
$\Rightarrow$ Se sortir seul d'un situation difficile.

Rappels sur les estimateurs « plug-in » -1-

- $\mathcal{D}_n = \{Z_1, ..., Z_n\}$ n-échantillon de v.a.i.i.d. $Z_i = (X_i, Y_i)$.
- X_i observations issues d'une v.a. X : données, variables explicatives.
- Y_i issues d'une v.a. Y, catégories des X_i : étiquettes ou labels.
- $X \in \mathbb{X}$, $Y \in \mathbb{Y}$.
- $\mathbb{P}_\theta$ proba sur $\mathcal{E} = \mathbb{X} \times \mathbb{Y}$: loi inconnue de (X, Y) et (X_i, Y_i).
- $\theta = \theta(\mathbb{P})$ paramètre inconnu dépendant de $\mathbb{P}$.
- $T = T_n = T(X_1, ..., X_n) = T(\mathbb{P})$ estimateur de θ.
- Comme la loi de X est inconnue, on utilise la mesure empirique :

$$\mathbb{P}_n = \frac{1}{n} \sum_{i=1}^{n} \delta_{X_i}$$

On sait (th. de Glivenko-Cantelli, th. limite centrale fonctionnel et th. de Kolmogorov-Smirnov) que $\mathbb{P}_n$ est fortement et uniformément consistant pour $\mathbb{P}$.

Rappels sur les estimateurs « plug-in » -2-

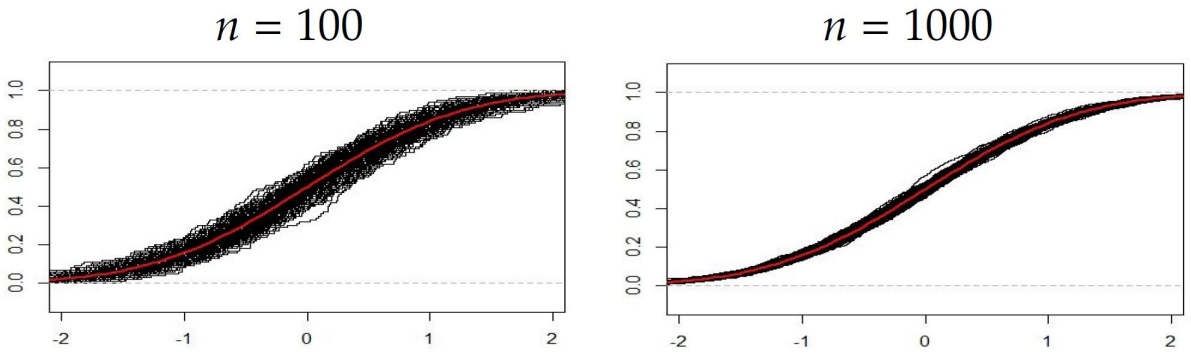
- Estimateur plug-in : remplacer $\mathbb{P}$ par $\mathbb{P}_n$ dans l'expression de $\theta = T(\mathbb{P})$

$$\widehat{\theta} = T(\mathbb{P}_n) = T(X_1, ..., X_n)$$

- Exemple : $\theta = \mathbb{E}[X] = \int x\mathbb{P}(dx) \Rightarrow \widehat{\theta} = \overline{X}_n = \frac{1}{n}\sum_i X_i = \int x\mathbb{P}_n(dx)$

- $\mathbb{P}_n$ est la mesure discrète qui sélectionne de façon uniforme chaque observation X_i avec probabilité $1/n$.

- Les estimateurs plug-in ont de bonnes propriétés :
$||\mathbb{P}_n - \mathbb{P}||_\infty = \sup_t |F_n(t) - F(t)| \longrightarrow 0$ p.s.



Rappels sur les estimateurs « plug-in » -3-

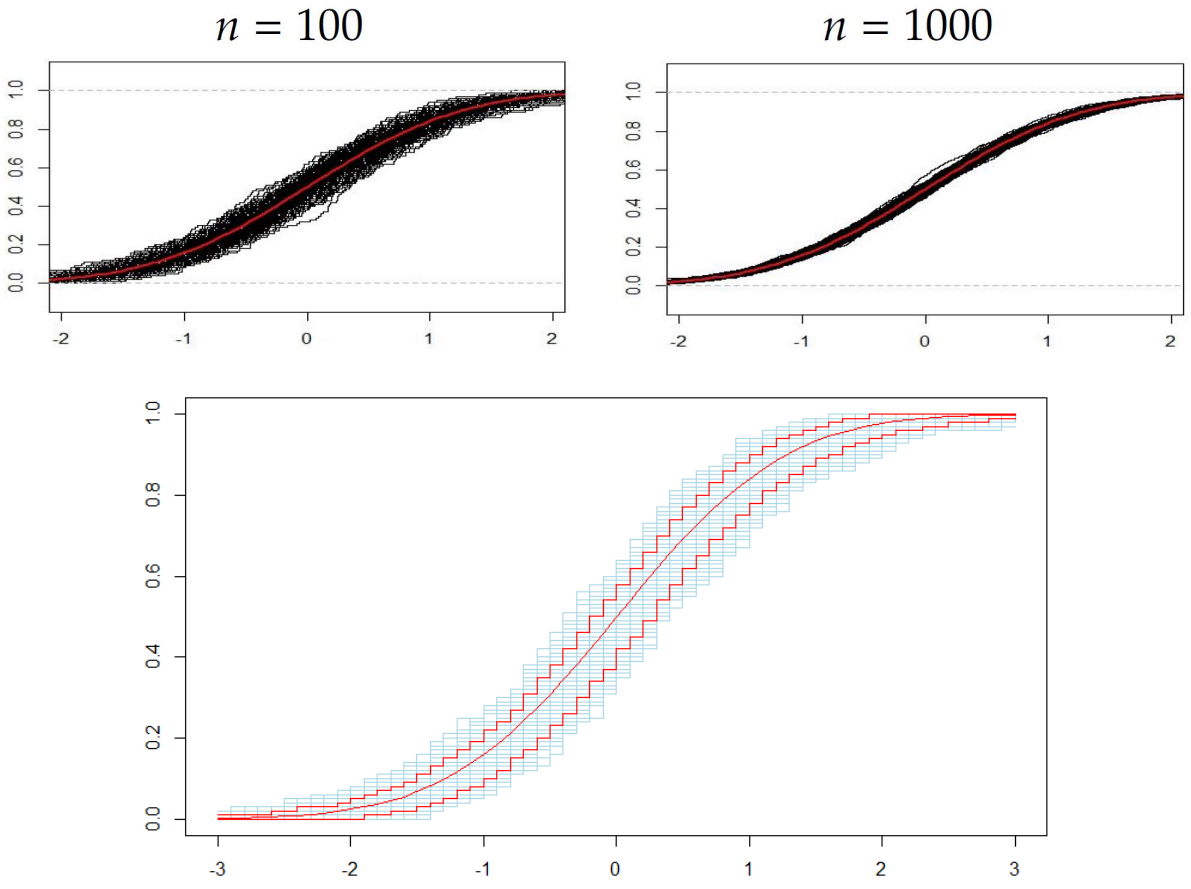


Figure 1 – Th. de Glivenko-Cantelli : $||\mathbb{P}_n - \mathbb{P}||_\infty = \sup_{t\in\mathbb{R}} |F_n(t) - F(t)| \xrightarrow[n\to+\infty]{\text{p.s.}} 0.$
Illustration Arthur Charpentier (Freakonometrics).

Principe du Bootstrap -1-

- On remplace $\mathbb{P}$ par $\mathbb{P}_n$.... et on ré-échantillonne à partir de $\mathbb{P}_n$.
- $(X_1^\star, ..., X_n^\star)$ échantillon bootstrap issu de $(x_1, ..., x_n)$.
- Bootstrap naïf : les $X_i^\star$ sont tirés de façon uniforme et avec remise parmi les $(x_1, ..., x_n)$ ($\sim$ tirages avec remise dans une urne).
- En quelque sorte, on échantillonne l'échantillon initial $(X_1, ..., X_n)$.
- Dans le monde réel, on estime θ par $\widehat{\theta} = T(\mathbb{P}_n) = T(X_1, ..., X_n)$.
- Dans le monde bootstrap, on estime θ par $\theta^\star = T(X_1^\star, ..., X_n^\star)$.
- Valide si la loi de $\theta^\star$ approche bien celle de $\widehat{\theta}$, c.-à-d. si $\mathcal{L}(\theta^\star | X_1, ..., X_n)$ et $\mathcal{L}(\widehat{\theta})$ ont même limite.

Principe du Bootstrap -2-

• Conditions de validité dépendent de la régularité des fonctions en jeu :

- La loi limite de $T(\mathbb{P}_n)$ doit dépendre continûment de $\mathbb{P}$.
- Convergence uniforme en $\mathbb{P}$ dans un voisinage d'une mesure.
- Conditions de régularité (continuité, différentiabilité) dans l'espace des mesures de probabilités (la variable est une mesure de probabilité).
- Fonctionne bien si lois bien approchées par des gaussiennes.
- Fonctionne mal si lois extrêmes (min, max, quantiles) et/ou fonction irrégulière de $\mathbb{P}$.

• Ex : Pour $T(X_1, ..., X_n) = \overline{X}$, si $\mathbb{E}[X^2] < \infty$, $\overline{X}^\star$ CV bien vers $\mathbb{E}[X]$.

• Ex : Pour $T(X_1, ..., X_n) = \max(X_1, ..., X_n)$ si $X \sim \mathcal{E}$ loi exponentielle, les échantillons bootstrap ne convergent pas.

$\Rightarrow$ Toujours vérifier les moments d'ordre 2, toujours vérifier par des simulations le comportement bootstrap.

Principe du Bootstrap -3-

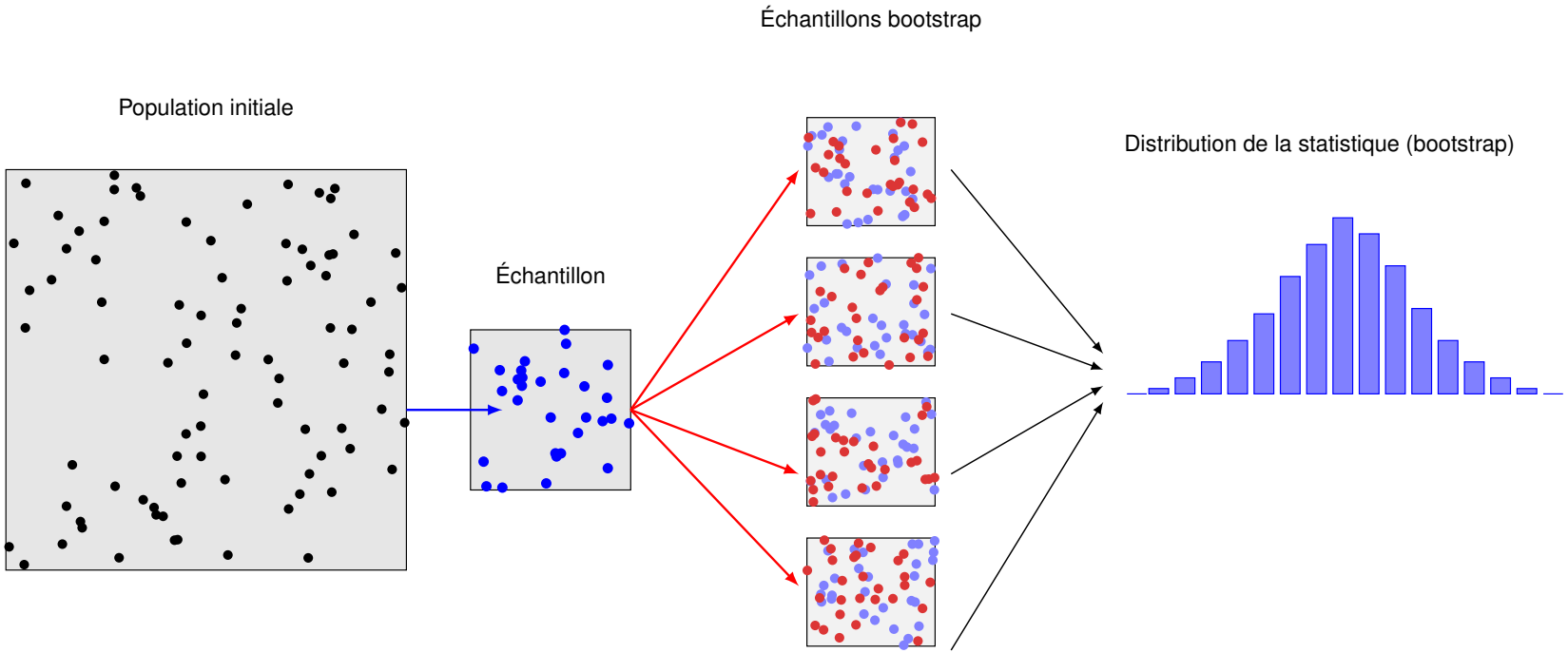
- On peut construire plusieurs échantillons bootstrap de façon à simuler plusieurs tirages : n^n échantillons bootstrap $\neq$ possibles.
- En pratique nombre B de tirages entre B = 200 et B = 1000.
- Échantillons non indépendants, mais i.i.d. conditionnellement à $\mathcal{D}_n$.

$$\begin{cases} \theta_1^\star = T(X_{11}^\star, ..., X_{1n}^\star) \\ ... \\ \theta_b^\star = T(X_{b1}^\star, ..., X_{bn}^\star) \\ ... \\ \theta_B^\star = T(X_{B1}^\star, ..., X_{Bn}^\star) \end{cases}$$

• Les vecteurs $\theta_b^\star$ pour b = 1, ..., B, sous $\mathbb{P}_n$, sont des réplications bootstrap de $\widehat{\theta} = T(X_1, ..., X_n)$.

• Double approximation : on approche $\mathbb{P}$ et θ par $\mathbb{P}_n$ et $\widehat{\theta}$, puis par $\mathbb{P}_n^\star$ et $\theta^\star$. Sachant $\mathcal{D}_n$, $\mathbb{P}_n^\star$ est la loi du tirage uniforme avec remise dans une urne contenant $\{x_1, ..., x_n\}$.

Principe du Bootstrap -4-



Bagging

Principe du Bagging

- Bagging = Bootstrap Aggregating (Breiman, 1996).
- B estimateurs bootstrap agrégés en calculant leur moyenne empirique.
- Modèle de régression : $Y = \eta(X) + \epsilon$ avec $\eta(x) = \mathbb{E}[Y|X = x]$.
- Pour chaque échantillon bootstrap $\mathcal{D}_{nb}$ on ajuste un régresseur $\eta_b(x)$. Le régresseur bagging est

$$\widehat{\eta}_B(x) = \frac{1}{B}\sum_{b=1}^{B} \eta_b(x)$$

Figure 2 – Copyright Peter Jackson, from the Lord of the Rings.

Algorithme Bagging pour la régression

- Entrées : échantillon $\mathcal{D}_n$ + régresseur ou classifieur η + valeur du nombre B.
- Pour b = 1, ..., B
  - Tirer un échantillon bootstrap $\mathcal{D}_{nb}$ dans $\mathcal{D}_n$.
  - Ajuster un régresseur ou classifieur $\eta_b(x)$ sur cet échantillon.
- Sortie : Estimateur bagging :

$$\widehat{\eta}_B(x) = \frac{1}{B}\sum_{b=1}^{B} \eta_b(x)$$

 Attention ! $\eta_b(x)$ v.a. car dépend de $\mathcal{D}_{nb} \subset \mathcal{D}_n$ (aléatoirement).

- $\eta_b(x)$ non indépendants. Mais conditionnellement à $\mathcal{D}_n$: i.i.d.

Bagging, biais et variance -1-

$$\mathbb{E}[\widehat{\eta}_B(x)] = \mathbb{E}[\eta_b(x)]$$

Mais en général, à cause de la dépendance entre les $\eta_b(x)$,

$$\mathbb{V}(\widehat{\eta}_B(x)) \neq \frac{1}{B}\mathbb{V}(\eta_b(x))$$

$\Rightarrow$ Malgré tout, le tirage bootstrap atténue la dépendance en introduisant une nouvelle source d'aléa.

- On passe de $\mathcal{D}_n$ à $\mathcal{D}_{nb}$ par tirage uniforme avec remise. Un vecteur aleatoire d'indices $I_b = (I_{b1}, ..., I_{bn}) \subset [\![1..n]\!]^n$ est généré.

$$\eta_b(x) = \eta_b(x, \mathcal{D}_{nb}, \mathcal{D}_n) = \eta_b(x, I_b, \mathcal{D}_n)$$

- Sachant $\mathcal{D}_n$, les $\eta_b(x)$ sont i.i.d. (mais dépendant de I_b).

$$\lim_{B \to +\infty} \widehat{\eta}_B(x) = \mathbb{E}_I[\eta(x, I)|\mathcal{D}_n]$$

- 😱 Escroquerie ! où est passé le b dans I_b ?

Bagging, biais et variance -2-

$$\lim_{B \to +\infty} \widehat{\eta}_B(x) = \mathbb{E}_I[\eta(x, I) | \mathcal{D}_n]$$

- 😱 Escroquerie ! où est passé le b dans I_b ?

- Sachant $\mathcal{D}_n$, seuls les I_b sont aléatoires (mais sont i.i.d.). Les $Z_b = \eta_b(x, I_b, \mathcal{D}_n)$ sont aussi i.i.d.

- D'après la LFGN (conditionnellement à $\mathcal{D}_n$) :

$$\lim_{B \to +\infty} \widehat{\eta}_B(x) = \lim_{B \to +\infty} \frac{1}{B} \sum_{b=1}^{B} Z_b = \mathbb{E}_I[Z_1 | \mathcal{D}_n]$$

- Chaque I_b est un tirage indépendant de la même v.a. I_1, donc sa loi ne dépend pas de b : $\eta_b(x, I_b, \mathcal{D}_n) \sim \eta_1(x, I_1, \mathcal{D}_n) \sim \eta(x, I, \mathcal{D}_n)$ qui est une copie générique de la v.a. η_b.

Bagging, biais et variance -3-

$$\mathbb{V}\left(\widehat{\eta}_B(x)\right) = \rho(x)\mathbb{V}(\eta_b(x)) + \frac{1-\rho(x)}{B}\mathbb{V}(\eta_b(x))$$

avec $\rho(x) = \text{cov}(\eta_b(x), \eta_{b'}(x))$. Quand $B \to +\infty$, de façon peu rigoureuse, on admet que :

$$\lim_{B\to+\infty} \mathbb{V}\left(\widehat{\eta}(x)\right) = \rho(x)\mathbb{V}(\eta_b(x))$$

- Plus la corrélation $\rho(x)$ est faible, plus la variance diminue.
- C'est l'objectif du bootstrap : le hasard fait bien les choses.
- $\Rightarrow$ agréger des estimateurs sensibles aux perturbations de $\mathcal{D}_n$.
- $\Rightarrow$ Les arbres ont ces propriétés.

Forêts aléatoires (Breiman, 2000)

- Forêt aléatoire (random forest) = bagging sur des arbres de décision.

$$\widehat{T}_B(x) = \frac{1}{B}\sum_{b=1}^{B} T_b(x)$$

- On sélectionne aléatoirement m variables de l'arbre parmi les d variables initiales.
- On cherche à diminuer la corrélation entre les arbres.
- 2 nouvelles sources d'aléa : le tirage bootstrap de l'échantillon et le choix des m variables sur les arbres.
- Profondeur typique : 5 en régression, 1 en classification.
- Valeur typique de m : $d/3$ en régression, $\sqrt{d}$ en classification.
- Attention au biais : le biais ne diminue pas en bagging.

Mesures de performance

• Comme pour les autres méthodes d'apprentissage statistique : erreur de prédiction en régression, probabilité d'erreur en classification.

• Utilisation par sous-ensemble d'apprentissage/validation ou par validation croisée.

• Le bootstrap permet une estimation de l'erreur par OOB (Out of Bag).

• Pour tout (X_i, Y_i) de $\mathcal{D}_n$, soit $\mathbb{J}_i$ le sous-ensemble des arbres qui ne contient pas l'observation i. La prévision de Y en X_i est

$$\widehat{Y}_i = \frac{1}{|\mathbb{J}_i|} \sum_{b \in \mathbb{J}_i} T(X_i, \mathcal{D}_{nb}).$$

• L'erreur de prédiction OOB est : $\frac{1}{n} \sum_{i=1}^{n} (\widehat{Y}_i - Y_i)^2$.

• La probabilité d'erreur OOB est : $\frac{1}{n} \sum_{i=1}^{n} \mathbb{1}_{[\widehat{Y}_i \neq Y_i]}$.

Performances -1-

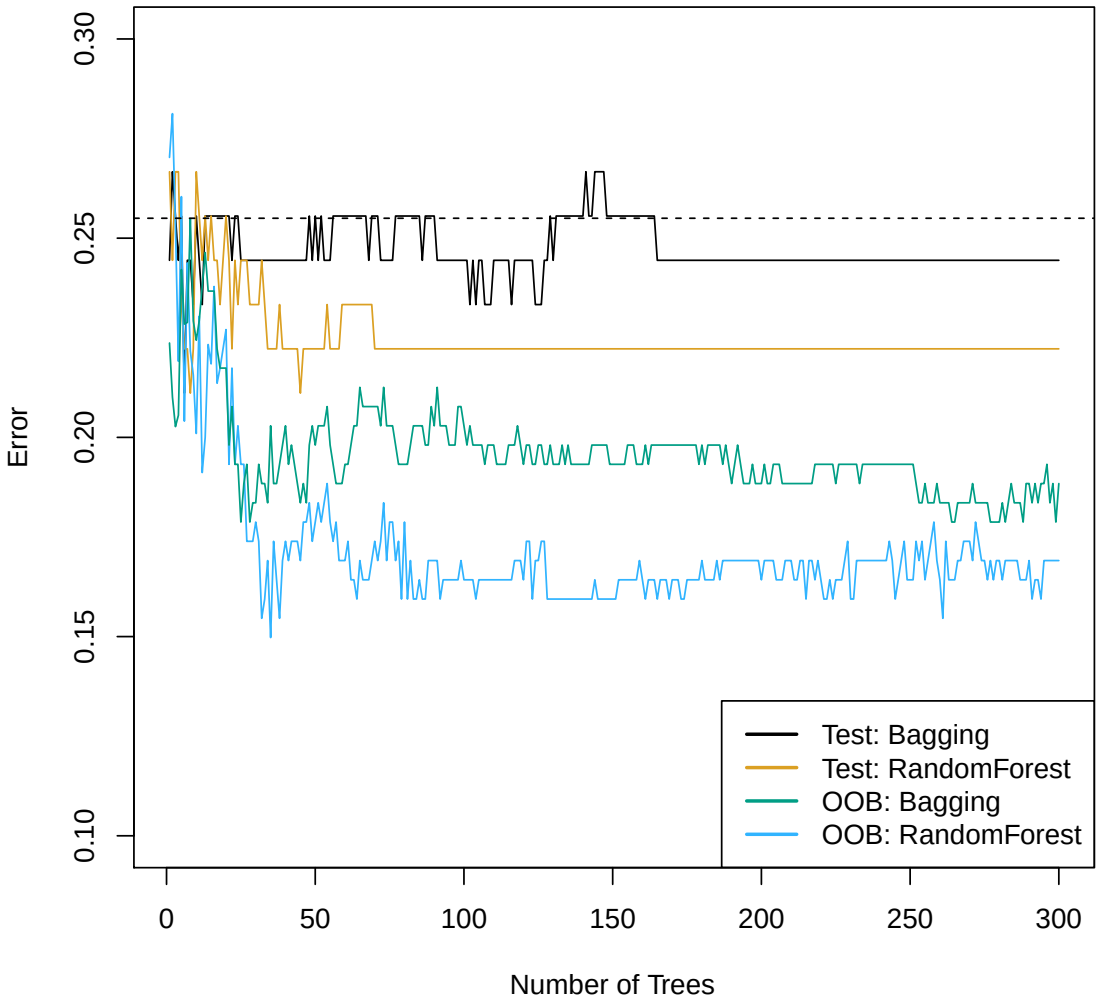


Figure 3 – Bagging and random forest for the Heart data. x-axis : the number B of boostraped training sets. Random forests are applied with m = $\sqrt{p}$. The dashed line indicates the test error from a single tree. From « an introduction to statistical learning », James and al. Springer.

Performances -2-

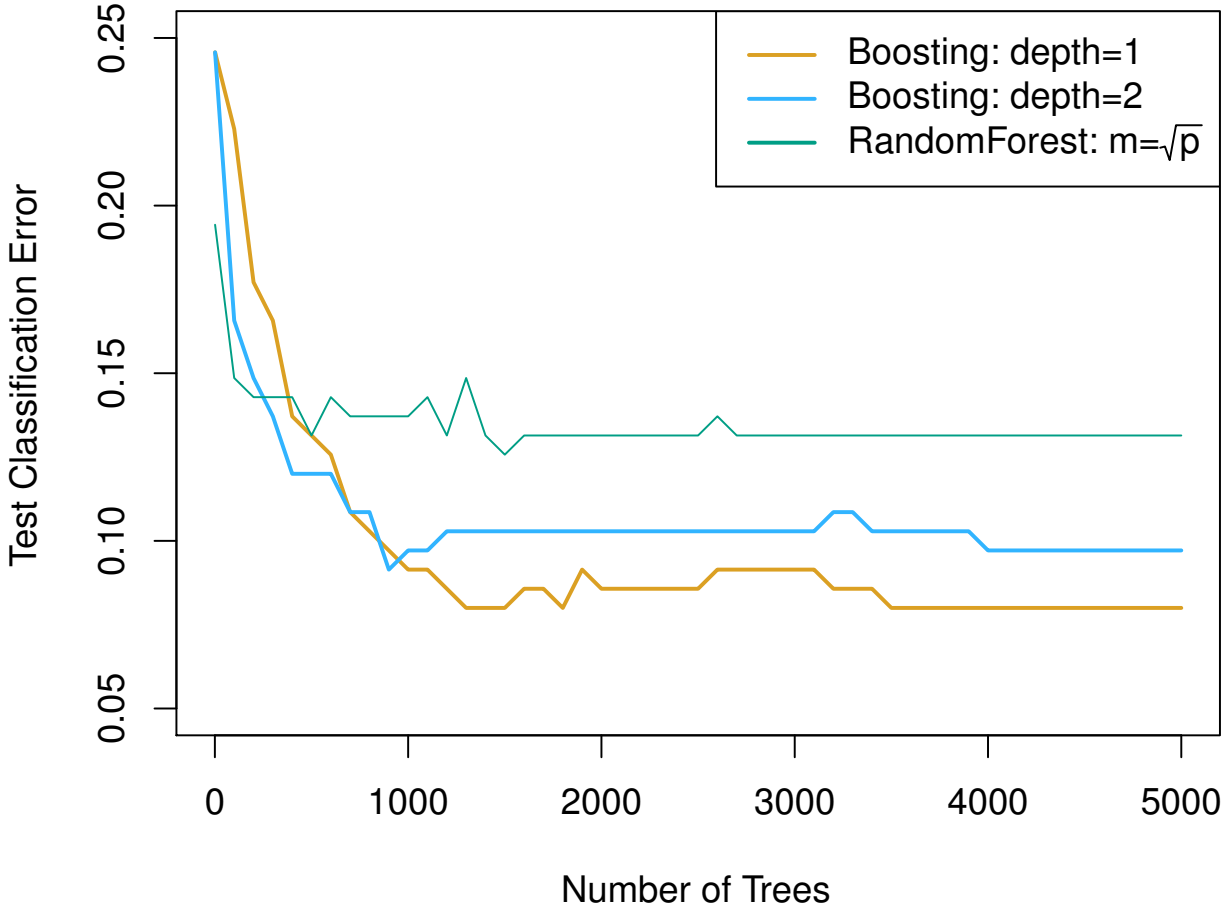


Figure 4 – Random forest with 500 predictors. Test error as a function of the number of trees. From « an introduction to statistical learning », James and al. Springer.

Stacking

Stacking : introduction

- Toujours une agrégation de classifieurs...
- ... mais en petit nombre,
- ... qui ne sont pas des classifieurs faibles...
- ... et sont éventuellement de nature différente.

Stacking : approche naïve

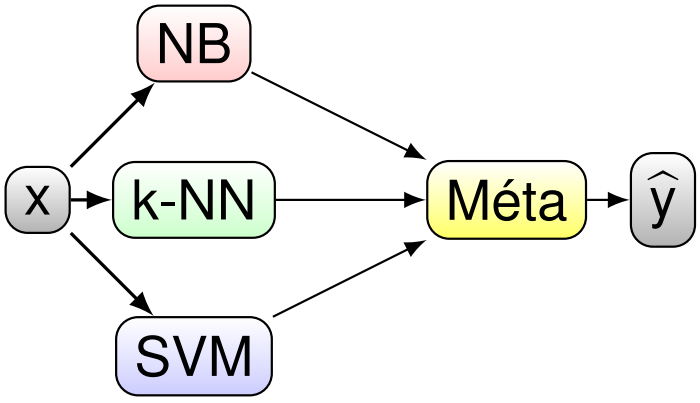


Figure 5 – Classification.

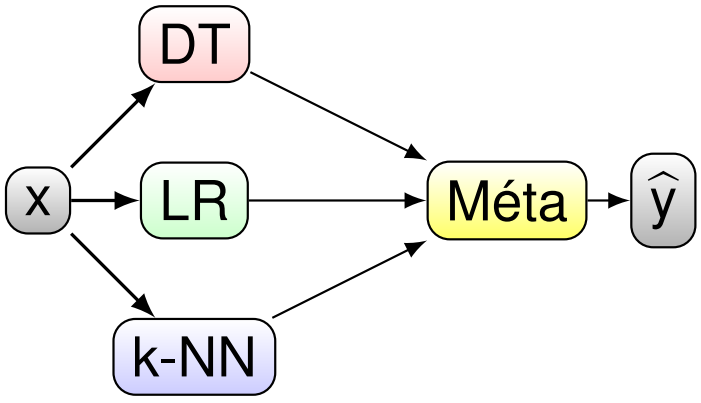


Figure 6 – Régression.

- Approche naïve : une combinaison linéaire (CL) pondérée de bons classifieurs est meilleure que chaque classifieur.
- C'est vrai si les poids minimisent l'erreur théorique (inconnue).
- Mais si on minimise l'erreur d'entraînement $\Rightarrow$ sur-apprentissage.
- Nécessité d'une structure à deux niveaux.
- CL performante si les classifieurs sont individuellement performants...
- ... tout en étant très différents les uns des autres.

Stacking : notion de méta-modèle

- Le premier niveau est formé de plusieurs modèles de base.
- Chaque modèle de base : entraîné séparément sur mêmes données.
- Chaque modèle donne des prédictions différentes à une donnée.
- Le second niveau est appelé méta-modèle.
- Les prédictions du niv.1 servent de données au niv.2.
- Le méta-modèle est entraîné sur ces prédictions...
- ... mais pas sur l'échantillon du niv.1 !
- Poids initiaux niv.2 : fonction des perf. de chaque modèle du niv.1.
- Ces poids sont modifiés lors de l'entraînement du méta-modèle.
- Le méta-modèle propose une prédiction finale.

Modèles de base : régression logistique, arbre de décision, k-ppv, SVM, réseaux de neurones, etc.

 Rappel important : le gain vient de la diversité des modèles.

Stacking : données d'entraînement et de test

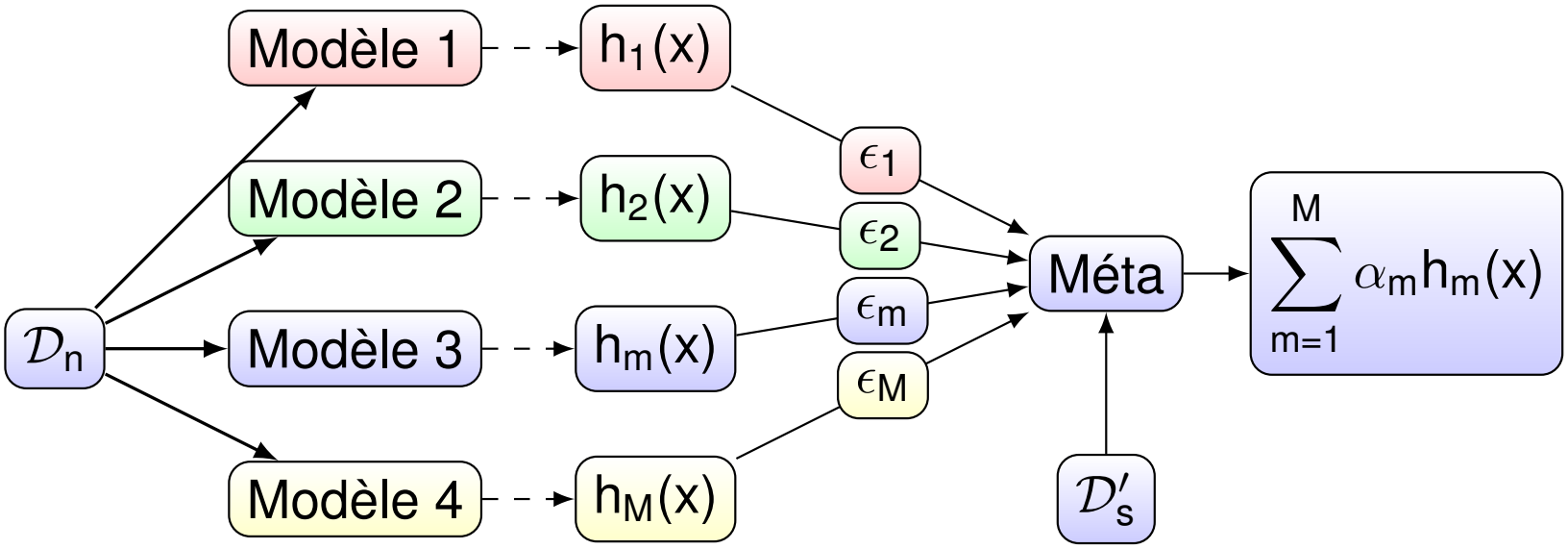
- Sur quelles données d'apprentissage entraîner les algorithmes ?
  - Bagging : versions modifiées du même échantillon (tirages avec remise).
  - Boosting : même données mais pondération des individus.
  - Stacking niv.1 : même échantillon à tous les classifieurs de niv.1.
  - Stacking niv.2 : autre échantillon pour entraîner le niv.2.

- Utiliser une validation croisée pour éviter le sur-apprentissage inter niveaux.

- Importance de garder peu de modèles pour ne pas avoir trop de paramètres ou de complexité.

- Le stacking peut réduire à la fois la variance et le biais.

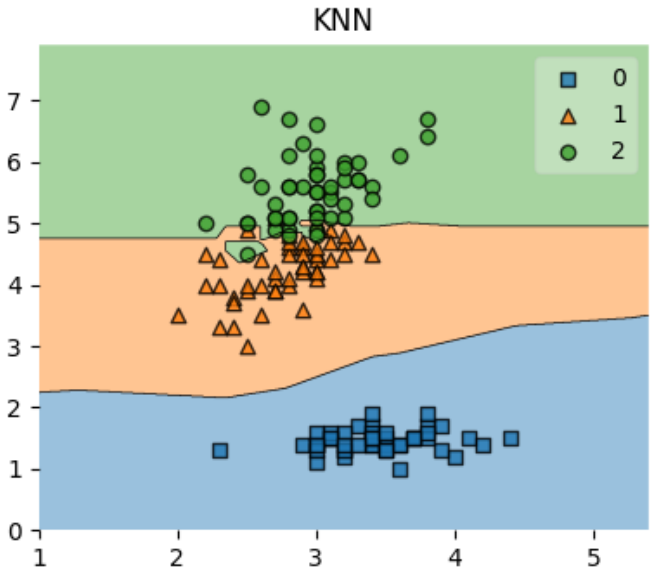
Stacking : figure du principe

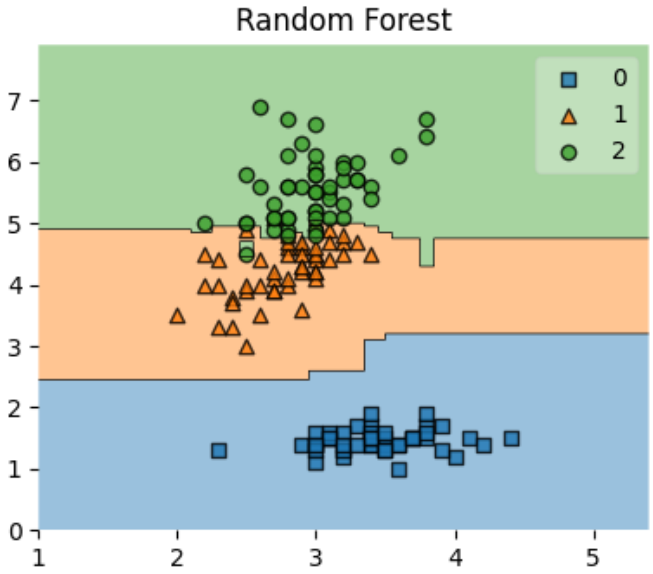


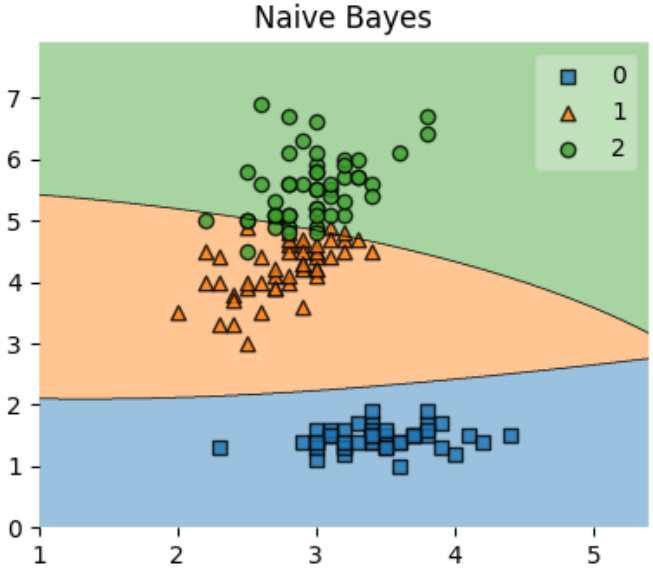
- Chaque classifieur niv.1 h_m fournit au méta-modèle son erreur de régression / classification ϵ_m.
- Le méta-modèle initialise les poids α_m à partir de ϵ_m.
- Il met à jour les poids lors de son entraînement sur $\mathcal{D}'_s$.

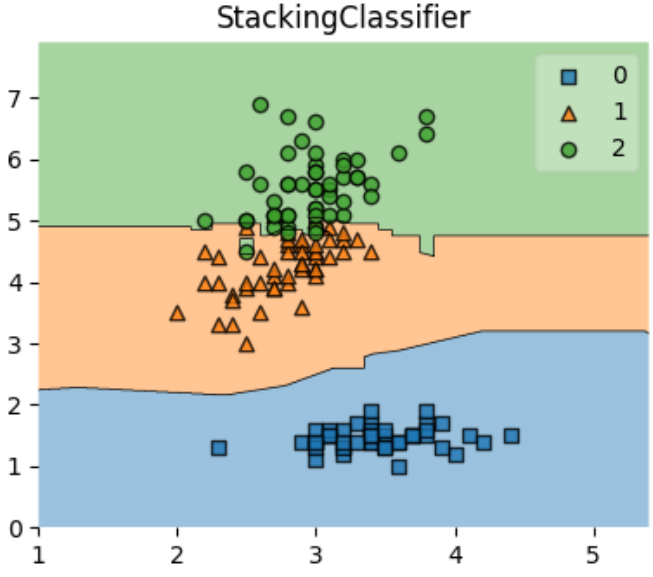
Comparaison des méthodes

Eternelles Iris de Fisher

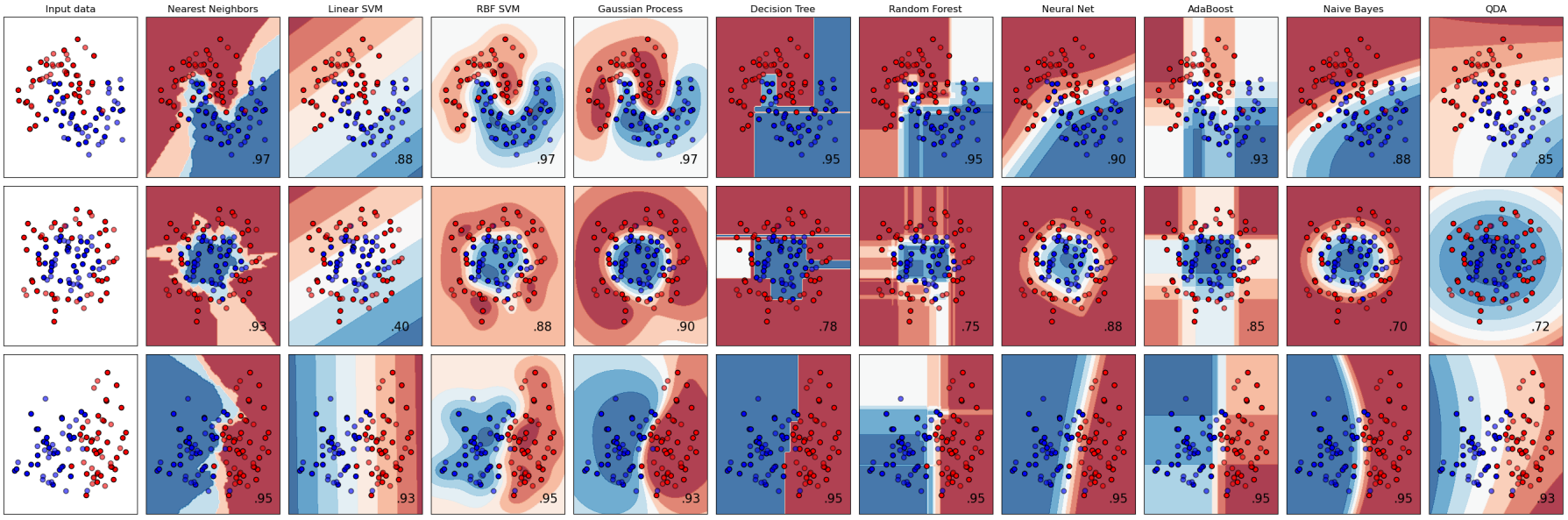






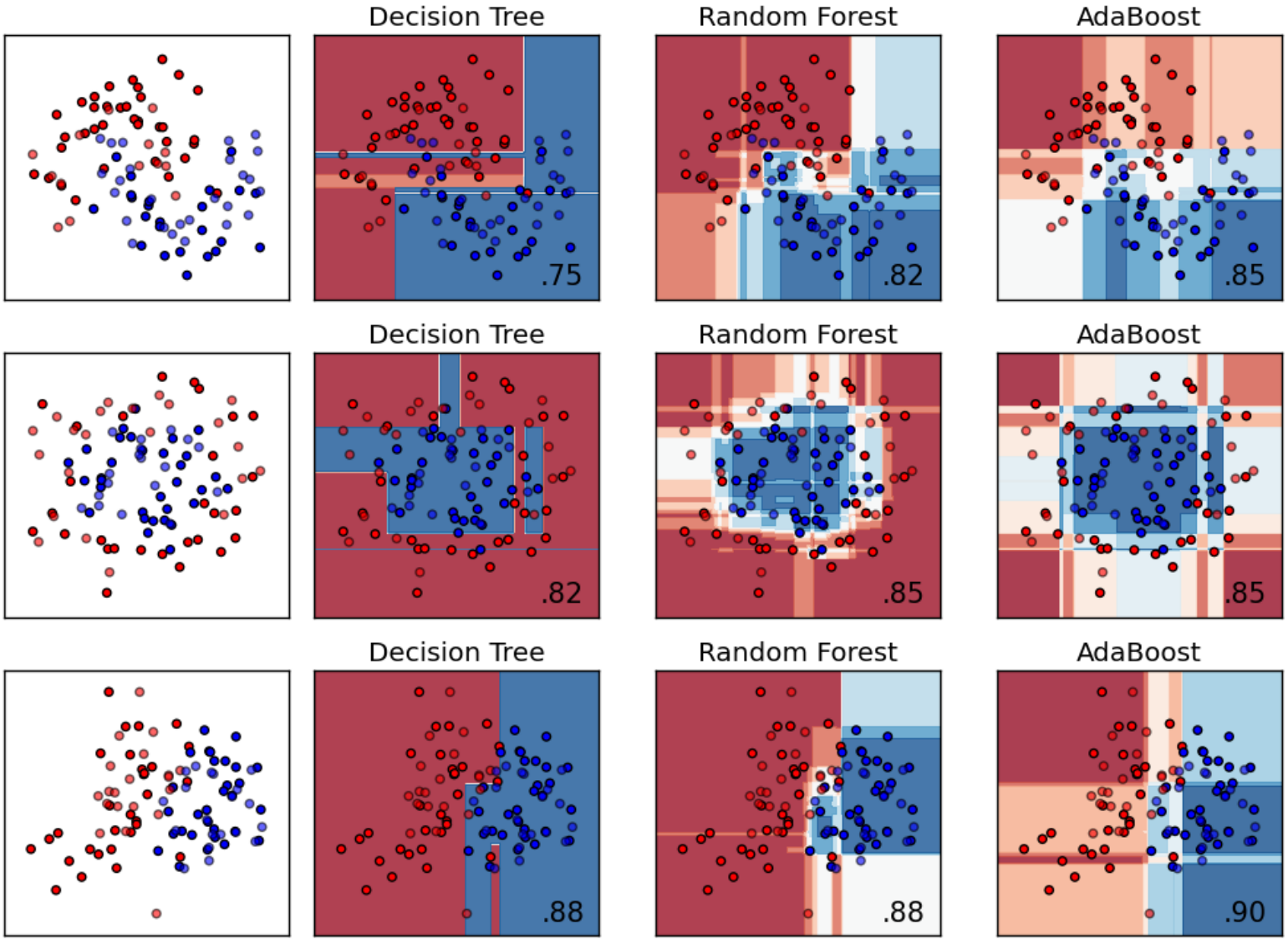


10 Classifieurs sur 3 jeux synthétiques



- Illustration de scikit-learn.org.

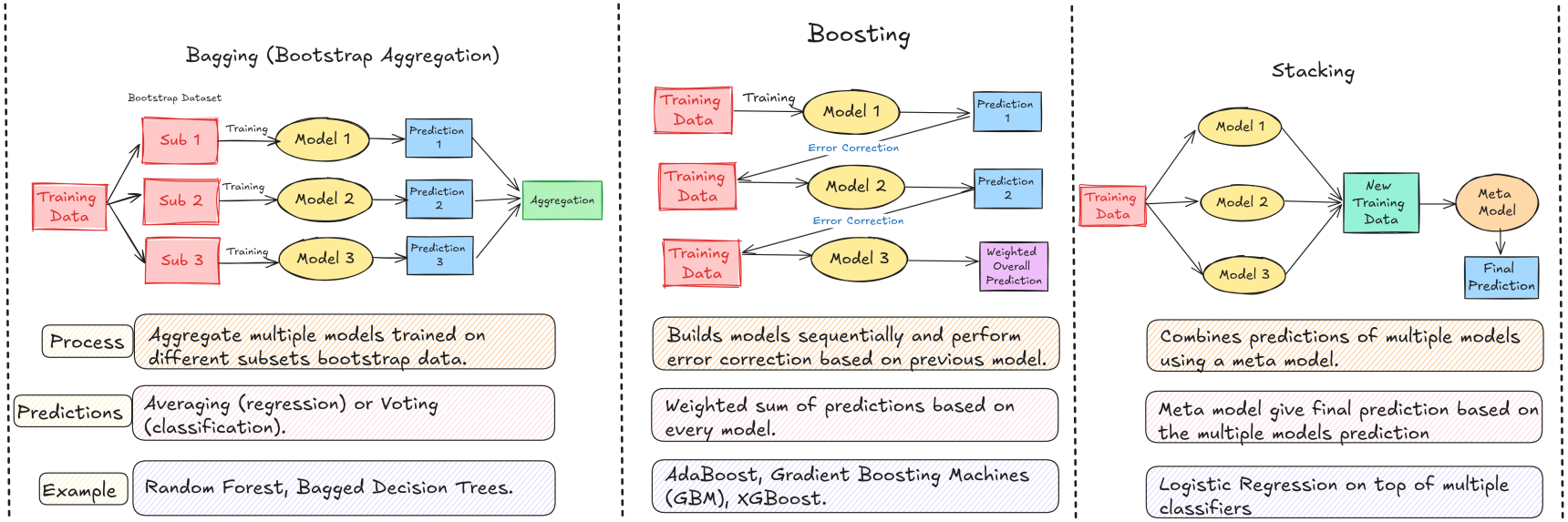
3 Classifieurs sur 3 jeux synthétiques



Différences Bagging, Boosting, Stacking -1-

	Bagging	Boosting	Stacking
Objectif	Réduire la variance	Réduire le biais	Performances ↑
Base classifieur (CB)	Homogène	Homogène	Hétérogène
Entraînement CB	Parallèle	Séquentiel	Méta modèle
Agrégation	Vote majoritaire ou moyenne	Moyenne pondérée	Moyenne pondérée

Différences Bagging, Boosting, Stacking -2-



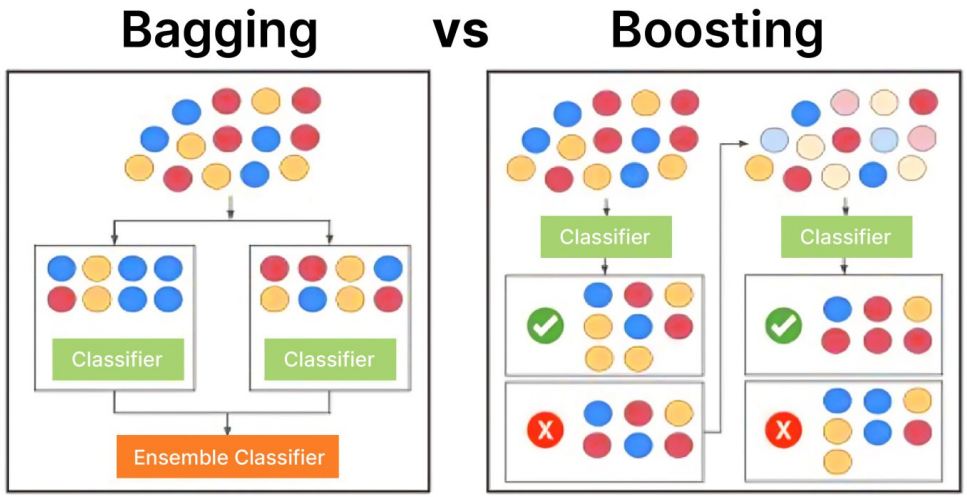


Tableau de synthèse -1-

Méthode	Inter-préta-tion	Perf.	Calibra-tion	Overfit.	Coût calc.
Régression linéaire	▲	▼	✗	▲	▲
Régression logistique	▲	▼	✗	▲	▲
SVM (kernel RBF)	✗	▲	▼	▼	▼
Réseau de neurones	✗	▲	✗	✗	✗
Arbres de décision	▲	✗	▲	✗	▲
Boosting	✗	▲	▲	▼	▼
Bagging (forêt)	✗	▲	▼	▲	▼

▲ : bon ▼ : moyen ✗ : faible ou problématique

• Ce tableau est une sorte de moyenne de ce que l'on trouve sur internet, il vaut ce qu'il vaut et ne donne qu'une idée générique. Le contenu des cases est finalement assez aléatoire. En fait, ne vous fiez pas du tout à ce type de tableau et quand vous traitez un problème de Machine Learning, essayez toutes les méthodes !

Blibliographie

- Bagging Predictors. Breiman. Machine Learning. 1996.
- Forêts aléatoires. Genuer. Thèse de doctorat. 2010.
- Stacked Generalization. D. Wolpert. Neural Networks. 1992.
- Super Learner. Van der Laan, Polley, Hubbard. Statistical Applications in Genetics. 2007.
- + Chapitre 5 (bootstrap p.212) Intro to Statistical Learning with Python.
- + Chapitre 8 (Bagging et random forests p.343) Intro to Statistical Learning with Python.
- + Chapitre 7 (Bagging p.88) du polycopié de Frédéric Sur.