Algorithmique des données
Classification

Charlotte Pelletier
MCF Univ. Bretagne Sud – IRISA Vannes

19 mars 2020

Plan du cours

- Partie I. Introduction
  - CM0. Introduction
  - CM1. Rappels en algèbre linéaire et probabilités
- Partie II. Apprentissage non-supervisé
  - CM2. Analyse par Composantes Principales
  - CM3. k-Means
- Partie III. Apprentissage supervisé : régression
  - CM4. Régression linéaire
  - CM5. Régression logistique
  - CM6. Compromis biais-variance et techniques de régularisation
- Partie IV. Apprentissage supervisé : classification
  - CM7. Algorithmes de classification
  - CM8. Sélection de modèles

Plan du cours

- Partie I. Introduction
  - CM0. Introduction
  - CM1. Rappels en algèbre linéaire et probabilités
- Partie II. Apprentissage non-supervisé
  - CM2. Analyse par Composantes Principales
  - CM3. k-Means
- Partie III. Apprentissage supervisé : régression
  - CM4. Régression linéaire
  - CM5. Régression logistique
  - CM6. Compromis biais-variance et techniques de régularisation
- **Partie IV. Apprentissage supervisé : classification**
  - CM7. Algorithmes de classification
  - **CM8. Sélection de modèles**

Sommaire

Rappel

- **Apprentissage supervisé** : dans les données observées, on connaît la "vraie" valeur de la variable de sortie et on cherche à comprendre / prédire le lien supposé entre les variables d'entrée et de sortie
- **Variable à expliquer/prédire**, notée Y
  - quantitative : régression
  - qualitative : classification binaire / multiclasses $\mathcal{C}$
- Variables explicatives, notées $X^1, X^2, \cdots, X^d$?
  - qualitatives et/ou quantitatives
  - plusieurs = de quelques dizaines à plusieurs (dizaines de) milliers $\Rightarrow$ sélection de variables

Données d'apprentissage

Échantillons

- $\mathbf{x} \in \mathbb{R}^d$ est une observation de d caractéristiques réelles (d variables)
- l'ensemble d'apprentissage est défini par les observations $\{\mathbf{x}_i\}_{i=1}^m$ où m est le nombre de données d'apprentissages (observations)
- d et m définissent la dimensionnalité du problème d'apprentissage
- les données sont mises sous la forme d'une matrice $\mathbf{X} \in \mathbb{R}^{m \times d}$ définie par $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2 \ldots, \mathbf{x}_m]^\top = [X^1, X^2, \ldots, X^d]$ contenant les exemples d'apprentissage en lignes et les variables en colonnes

Données d'apprentissage

Échantillons

- $\mathbf{x} \in \mathbb{R}^d$ est une observation de d caractéristiques réelles (d variables)
- l'ensemble d'apprentissage est défini par les observations $\{\mathbf{x}_i\}_{i=1}^m$ où m est le nombre de données d'apprentissages (observations)
- d et m définissent la dimensionnalité du problème d'apprentissage
- les données sont mises sous la forme d'une matrice $\mathbf{X} \in \mathbb{R}^{m \times d}$ définie par $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2 \ldots, \mathbf{x}_m]^\top = [X^1, X^2, \ldots, X^d]$ contenant les exemples d'apprentissage en lignes et les variables en colonnes

Étiquettes

- à chaque observation $\mathbf{x}_i$ une valeur à prédire $y_i \in \mathcal{Y}$ est associée (étiquette)
- les valeurs à prédire peuvent être concaténées en un vecteur $\mathbf{y} \in \mathcal{Y}^m$
- L'espace des valeurs à prédire $\mathcal{Y}$ sera :
  - $\mathcal{Y} = \mathbb{R}$ pour la régression
  - $\mathcal{Y} = \{-1, 1\}$ ou $\mathcal{Y} = \{0, 1\}$ pour la classification binaire
  - $\mathcal{Y} = \{1, \ldots, \mathcal{C}\}$ pour la classification multiclasses ($\mathcal{C}$ classes)

Apprentissage supervisé

Système d'apprentissage

1. **Phase d'apprentissage** : apprendre un modèle (règle de décision)
2. **Phase de prédiction** : prédire la classe de nouvelles observations (classification) ou donner une estimation de la réponse pour de nouvelles observations (régression)

Apprentissage supervisé

Système d'apprentissage

1. **Phase d'apprentissage** : apprendre un modèle (règle de décision)
2. **Phase de prédiction** : prédire la classe de nouvelles observations (classification) ou donner une estimation de la réponse pour de nouvelles observations (régression)

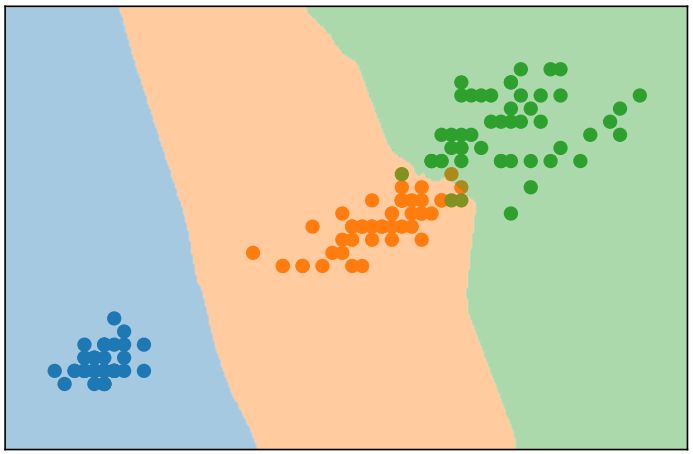
Comment sélectionner / choisir

- l'algorithme d'apprentissage supervisé ?
- la valeur des hyperparamètres de l'algorithme sélectionné ?

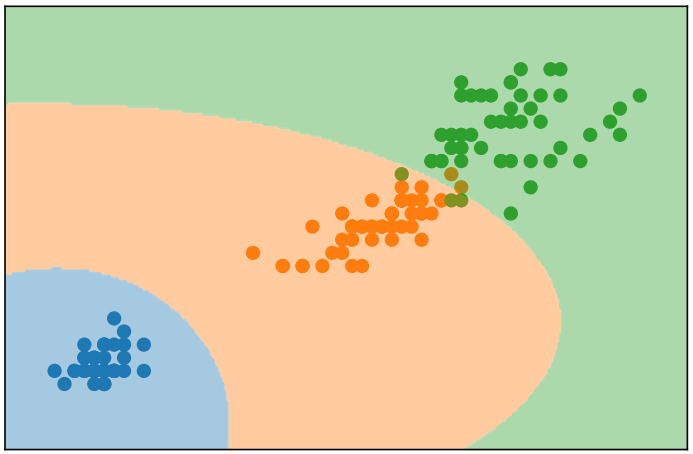
Choix des algorithmes

Comment sélectionner le meilleur algorithme possible pour un problème donné ?

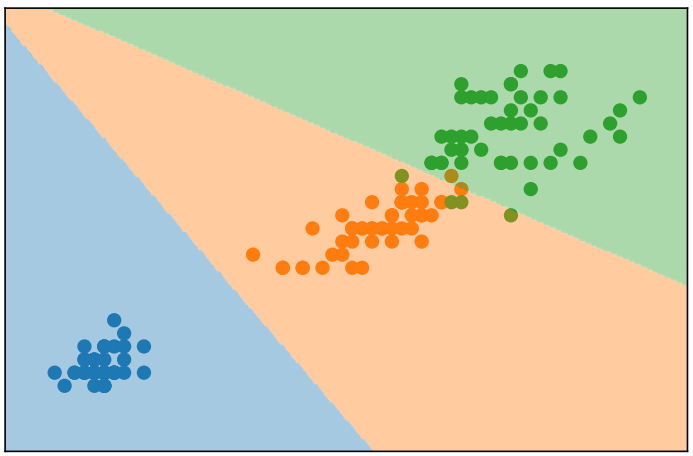
5-PPV



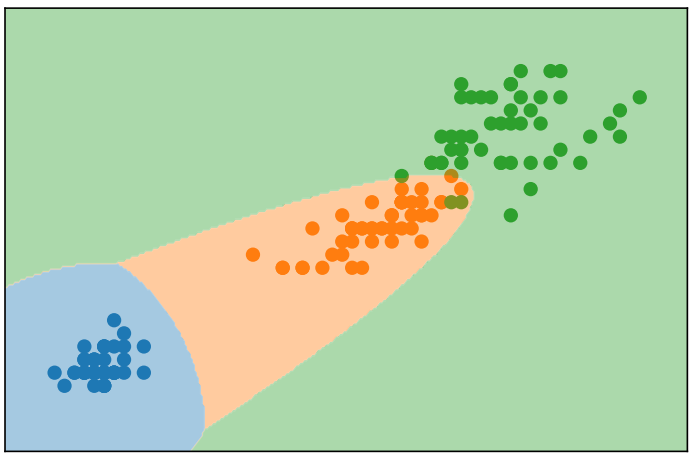
Classifieur bayésien naïf



LDA



QDA

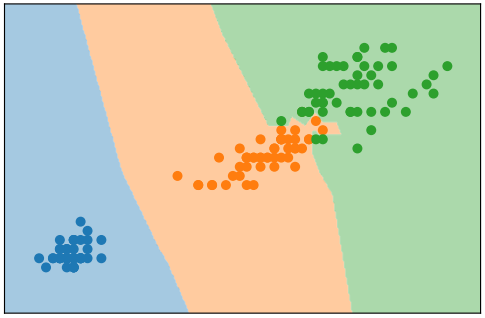


Jeu de données Iris (CM07 et TP07) :

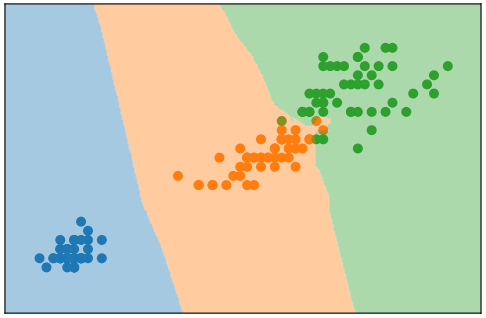
$$\text{largeur de la pétale (cm)} = f\left(\text{longueur de la pétale (cm)}\right)$$

k-Plus Proches Voisins : Quelle est la valeur optimale de k ?

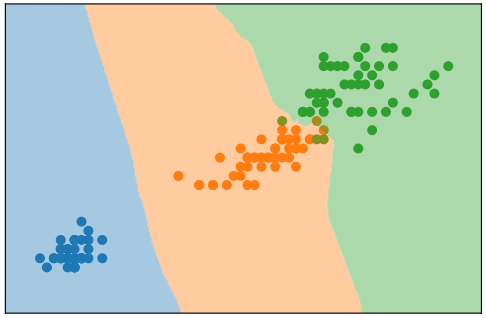
$k = 1$



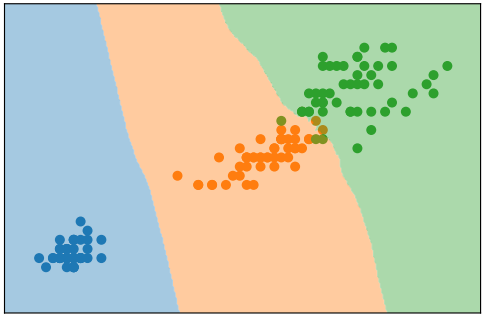
$k = 3$



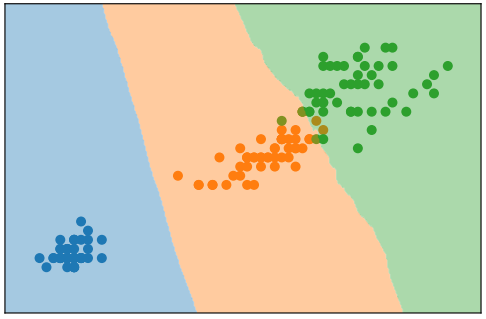
$k = 5$



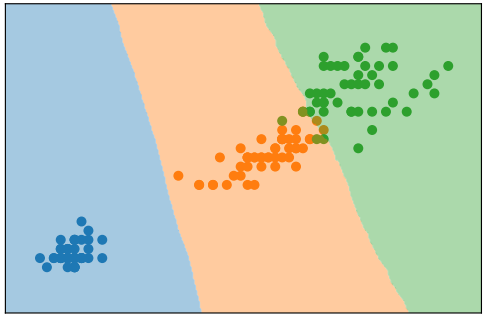
$k = 10$



$k = 30$



$k = 50$



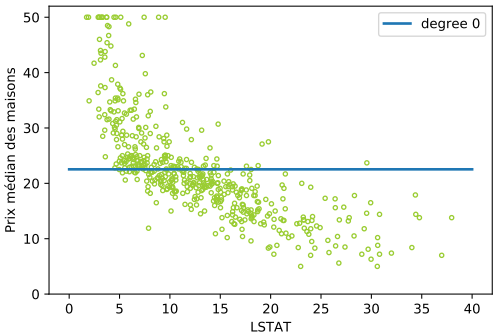
Jeu de données Iris (CM07 et TP07) :

$$\text{largeur de la pétale (cm)} = f\big(\text{longueur de la pétale (cm)}\big)$$

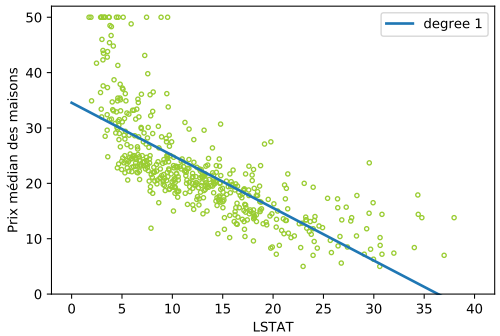
Régression polynomiale :

Quelle est la valeur optimale du degré du polynôme ?

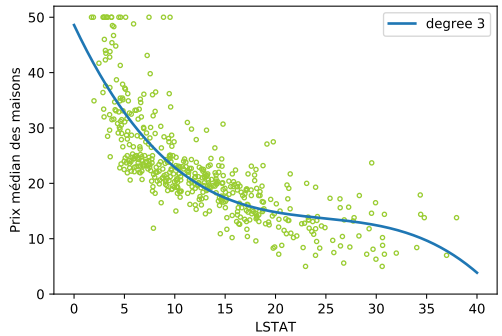
degré $= 0$



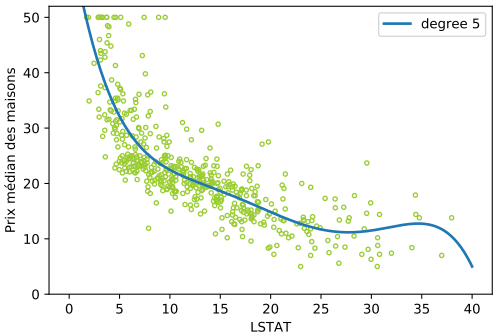
degré $= 1$



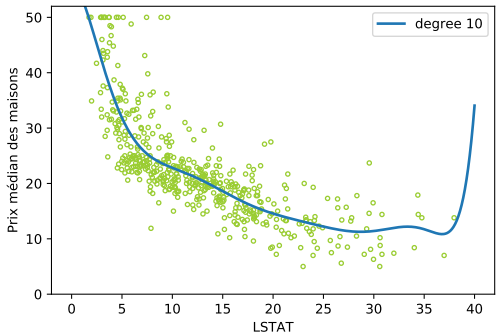
degré $= 3$



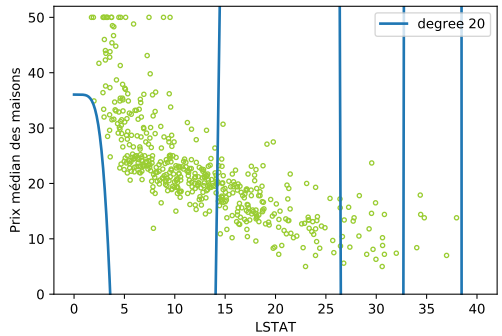
degré $= 5$



degré $= 10$



degré $= 20$



Jeu de données Boston (TP06).

- LSTAT est le pourcentage de ménages dont la catégorie socio-professionnelle est peu élevée
- y est la valeur médiane des maisons dans $m = 506$ quartiers aux alentours de Boston ($\times 1000$)

Objectifs

1. Sélectionner le meilleur algorithme possible pour un problème donné.
2. Sélectionner une valeur optimale pour chaque hyperparamètre de l'algorithme sélectionné.

Sommaire

Espace des hypothèses

Ensemble des données : $\{\mathbf{x}_i, y_i\}_{i=1}m$ avec $\mathbf{x}_i \in \mathbb{R}^d$

Ensemble des données

Espace des hypothèses

Ensemble des données : $\{\mathbf{x}_i, y_i\}_{i=1}m$ avec $\mathbf{x}_i \in \mathbb{R}^d$

Ensemble des données

Espace des hypothèses

- l'espace des fonctions $\mathcal{F}$ qui décrit les conditions de modélisations considérées
- cet espace est choisi en fonction de notre connaissance (et nos *convictions*) du problème d'apprentissage supervisé

Espace des hypothèses

Ensemble des données : $\{\mathbf{x}_i, y_i\}_{i=1}m$ avec $\mathbf{x}_i \in \mathbb{R}^d$

Ensemble des données

Espace des hypothèses

- l'espace des fonctions $\mathcal{F}$ qui décrit les conditions de modélisations considérées
- cet espace est choisi en fonction de notre connaissance (et nos *convictions*) du problème d'apprentissage supervisé

Exemple : $\mathbf{x}_i \in R$ $(d = 1)$

Si l'on choisit d'utiliser la régression linéaire,
$\mathcal{F}$ est l'ensemble des droites du plan.

Espace des hypothèses

Si l'on suppose que les données $\{\mathbf{x}_i, y_i\}_{i=1}m$ ont été générées par une fonction Φ, la tâche d'apprentissage automatique consiste à déterminer $f \in \mathcal{F}$ tel que f soit le plus proche possible de Φ, soit $f(\mathbf{x}) \approx \Phi(\mathbf{x})$.

Espace des hypothèses

Si l'on suppose que les données $\{\mathbf{x}_i, y_i\}_{i=1}m$ ont été générées par une fonction Φ, la tâche d'apprentissage automatique consiste à déterminer $f \in \mathcal{F}$ tel que f soit le plus proche possible de Φ, soit $f(\mathbf{x}) \approx \Phi(\mathbf{x})$.

Il faut alors

- quantifier la qualité d'une fonction de décision $h \in \mathcal{F}$
  - fonction de coût ℓ (fonction perte, fonction d'erreur ou en anglais *loss function* ou *cost function*) (CM4)
- chercher la fonction de décision f optimale dans $\mathcal{F}$
  - f est optimale si elle minimise la fonction de coût (dans ce cours)

Minimisation du risque empirique

Risque

- espérance de la fonction de coût ℓ :

$$\mathcal{R}(h) = \mathbb{E}_{\mathbf{x}}\Big[\ell\big(h(\mathbf{x}), y\big)\Big]$$

- on cherche la fonction f optimale qui minimise ce risque :

$$f = \operatorname*{argmin}_{h \in \mathcal{F}} \mathcal{R}(h) = \operatorname*{argmin}_{h \in \mathcal{F}} \mathbb{E}_{\mathbf{x}}\Big[\ell\big(h(\mathbf{x}), y\big)\Big]$$

Minimisation du risque empirique

Risque

- espérance de la fonction de coût ℓ :

$$\mathcal{R}(h) = \mathbb{E}_{\mathbf{x}}\Big[\ell\big(h(\mathbf{x}), y\big)\Big]$$

- on cherche la fonction f optimale qui minimise ce risque :

$$f = \underset{h \in \mathcal{F}}{\operatorname{argmin}}\, \mathcal{R}(h) = \underset{h \in \mathcal{F}}{\operatorname{argmin}}\, \mathbb{E}_{\mathbf{x}}\Big[\ell\big(h(\mathbf{x}), y\big)\Big]$$

Risque empirique

- risque calculée pour les m observations de la base de données (erreur moyenne) :

$$\mathcal{R}_{emp}(h) = \frac{1}{m}\sum_{i=1}^{m} \ell\big(h(\mathbf{x}_i), y_i\big)$$

- minimisation du risque empirique

$$f = \underset{h \in \mathcal{F}}{\operatorname{argmin}}\, \mathcal{R}_{emp}(h) = \underset{h \in \mathcal{F}}{\operatorname{argmin}}\, \frac{1}{m}\sum_{i=1}^{m} \ell\big(h(\mathbf{x}_i), y_i\big)$$

Minimisation du risque empirique

Risque

- espérance de la fonction de coût ℓ :

$$\mathcal{R}(h) = \mathbb{E}_{\mathbf{x}}\Big[\ell\big(h(\mathbf{x}), y\big)\Big]$$

- on cherche la fonction f optimale qui minimise ce risque :

$$f = \underset{h\in\mathcal{F}}{\operatorname{argmin}}\, \mathcal{R}(h) = \underset{h\in\mathcal{F}}{\operatorname{argmin}}\, \mathbb{E}_{\mathbf{x}}\Big[\ell\big(h(\mathbf{x}), y\big)\Big]$$

Risque empirique

- risque calculée pour les m observations de la base de données (erreur moyenne) :

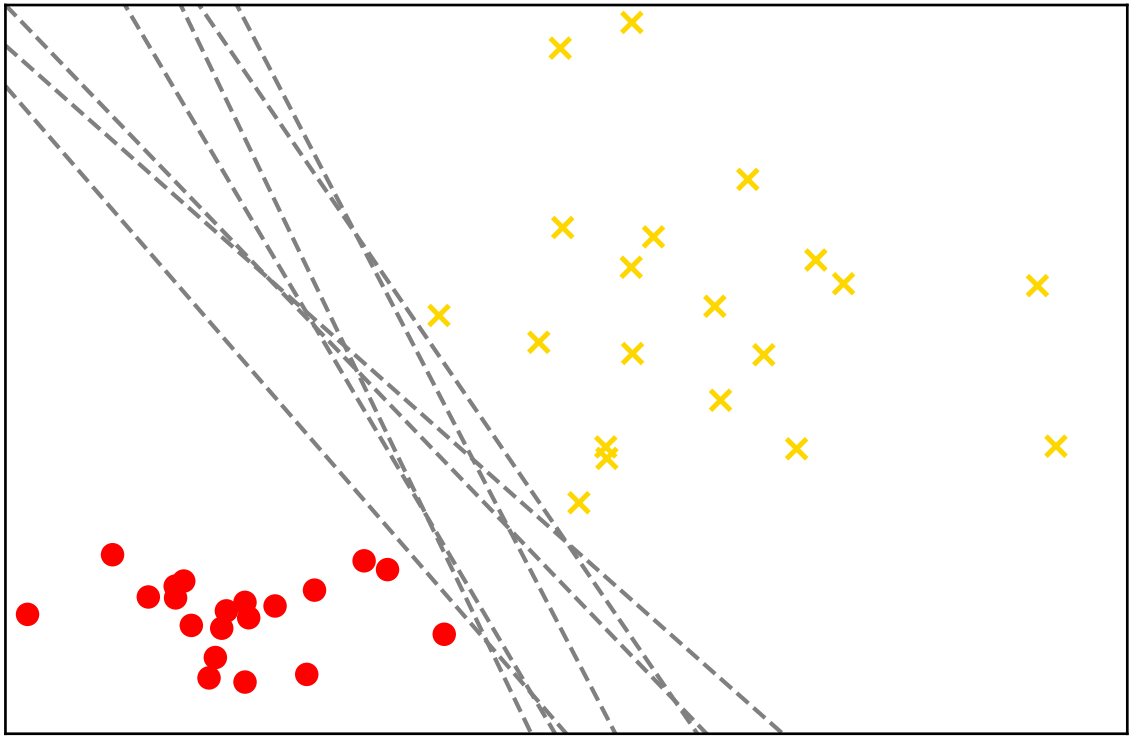
$$\mathcal{R}_{emp}(h) = \frac{1}{m}\sum_{i=1}^{m} \ell\big(h(\mathbf{x}_i), y_i\big)$$

- minimisation du risque empirique

$$f = \underset{h\in\mathcal{F}}{\operatorname{argmin}}\, \mathcal{R}_{emp}(h) = \underset{h\in\mathcal{F}}{\operatorname{argmin}}\, \frac{1}{m}\sum_{i=1}^{m} \ell\big(h(\mathbf{x}_i), y_i\big)$$

Minimisation du risque empirique

La minimisation du risque empirique est un problème **mal-posé*** :



Toutes les frontières de décision h (droites grisées en pointillés) minimisent le risque empirique ($\mathcal{R}_{emp}(h) = 0$). Il existe donc un nombre infini de solutions qui minimisent le risque empirique à zéro.

* Un problème est bien posé au sens de Hammard si : (1) une solution existe, (2) la solution est unique, et (3) la solution dépend de façon continue des données dans le cadre d'une topologie raisonnable. **Source** : Wikipédia

Objectif de l'apprentissage supervisé

Un système d'apprentissage supervisé doit être capable de **généraliser** :

- capacité d'un modèle à faire des prédictions correctes sur de nouvelles données qui n'ont pas été utilisées pour construire le modèle
  - prédire l'étiquette d'une nouvelle observation (classification)
  - estimer la variable réponse d'une nouvelle observation (régression)

Objectif de l'apprentissage supervisé

Un système d'apprentissage supervisé doit être capable de **généraliser** :

- capacité d'un modèle à faire des prédictions correctes sur de nouvelles données qui n'ont pas été utilisées pour construire le modèle
  - prédire l'étiquette d'une nouvelle observation (classification)
  - estimer la variable réponse d'une nouvelle observation (régression)

Évaluer un modèle sur les données qui ont servi à le construire ne permet pas de savoir comment le modèle se comporte sur de nouvelles données.

➜ Autrement dit, la capacité de généralisation d'un modèle ne peut pas être évaluée en observant le risque empirique.

➜ Il existe un lien avec le **compromis biais-variance** (CM06), et donc les problèmes de sur- et sous-apprentissage.

Objectif de l'apprentissage supervisé

Un système d'apprentissage supervisé doit être capable de **généraliser** :

- capacité d'un modèle à faire des prédictions correctes sur de nouvelles données qui n'ont pas été utilisées pour construire le modèle
  - prédire l'étiquette d'une nouvelle observation (classification)
  - estimer la variable réponse d'une nouvelle observation (régression)

Évaluer un modèle sur les données qui ont servi à le construire ne permet pas de savoir comment le modèle se comporte sur de nouvelles données.

➜ Autrement dit, la capacité de généralisation d'un modèle ne peut pas être évaluée en observant le risque empirique.

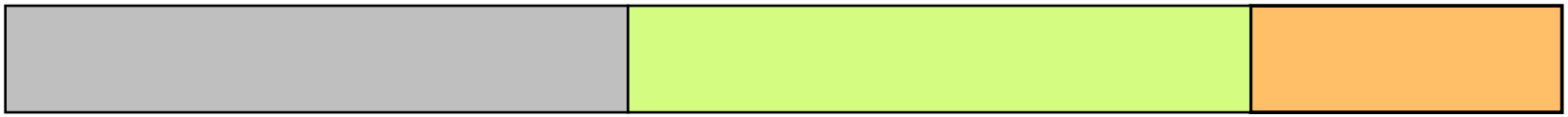
➜ Il existe un lien avec le **compromis biais-variance** (CM06), et donc les problèmes de sur- et sous-apprentissage.

On divise donc les données en sous-ensemble !

Découpage des données

Cas #1 : m est grand

Séparation des données en trois sous-ensembles :



Partition pour l'apprentissage

Partition pour la validation

Partition pour le test

Découpage des données

Cas #1 : m est grand

1. **Données d'apprentissage** : apprentissage des paramètres du modèles. Par exemple,
   - les paramètres $\{\beta_j\}_{j=0}^{d}$ pour la régression linéaire
   - les moyennes et variances (μ_y^j, σ_y^j) pour le classifieur bayésien naïf $(1 \leq y \leq \mathbb{C}$ et $1 \leq j \leq d)$

Découpage des données

Cas #1 : m est grand

1. **Données d'apprentissage** : apprentissage des paramètres du modèles. Par exemple,
   - les paramètres $\{\beta_j\}_{j=0}^{d}$ pour la régression linéaire
   - les moyennes et variances (μ_y^j, σ_y^j) pour le classifieur bayésien naïf $(1 \leq y \leq \mathbb{C}$ et $1 \leq j \leq d)$
2. **Données de validation** : estimation objective de l'erreur de généralisation du modèle **et** estimation des hyperparamètres. Par exemple,
   - l'hyperparamètre λ, coefficient de régularisation, pour les modèles de régression linéaire et logistique
   - l'hyperparamètre k, nombre de plus proches voisins (PPV) à considérer, pour l'algorithme des k-PPV

Découpage des données

Cas #1 : m est grand

1. **Données d'apprentissage** : apprentissage des paramètres du modèles. Par exemple,
   - les paramètres $\{\beta_j\}_{j=0}^{d}$ pour la régression linéaire
   - les moyennes et variances (μ_y^j, σ_y^j) pour le classifieur bayésien naïf $(1 \leq y \leq \mathbb{C}$ et $1 \leq j \leq d)$
2. **Données de validation** : estimation objective de l'erreur de généralisation du modèle **et** estimation des hyperparamètres. Par exemple,
   - l'hyperparamètre λ, coefficient de régularisation, pour les modèles de régression linéaire et logistique
   - l'hyperparamètre k, nombre de plus proches voisins (PPV) à considérer, pour l'algorithme des k-PPV
3. **Données de test** : estimation de l'erreur de prédiction sur des données non-observées (risque réel)
   - comparaison de différents algorithmes de classification
   - **important** : les données de test ne sont **jamais** utilisées pour l'estimation des paramètres et des hyperparamètres des modèles

Découpage des données

Cas #1 : m est grand

Comment choisir les proportions de chaque ensemble ?

Découpage des données

Cas #1 : m est grand

Comment choisir les proportions de chaque ensemble ?

- Il n'y a pas de « recettes miracles ».
- Généralement,
  - 60 % / 10 % / 30 % ou 70 % / 10 % / 20 % pour des algorithmes d'apprentissage automatique traditionnels
  - 95 % / 2 % / 3 % ou 98 % / 1 % / 1 % pour des algorithmes d'apprentissage profond (si m est très grand)

Découpage des données

Cas #1 : m est grand

Comment choisir les proportions de chaque ensemble ?

- Il n'y a pas de « recettes miracles ».
- Généralement,
  - 60 % / 10 % / 30 % ou 70 % / 10 % / 20 % pour des algorithmes d'apprentissage automatique traditionnels
  - 95 % / 2 % / 3 % ou 98 % / 1 % / 1 % pour des algorithmes d'apprentissage profond (si m est très grand)

Remarque : Dans beaucoup de problèmes d'apprentissage automatique, il y a peu de données étiquetées (m est petit).

Découpage des données

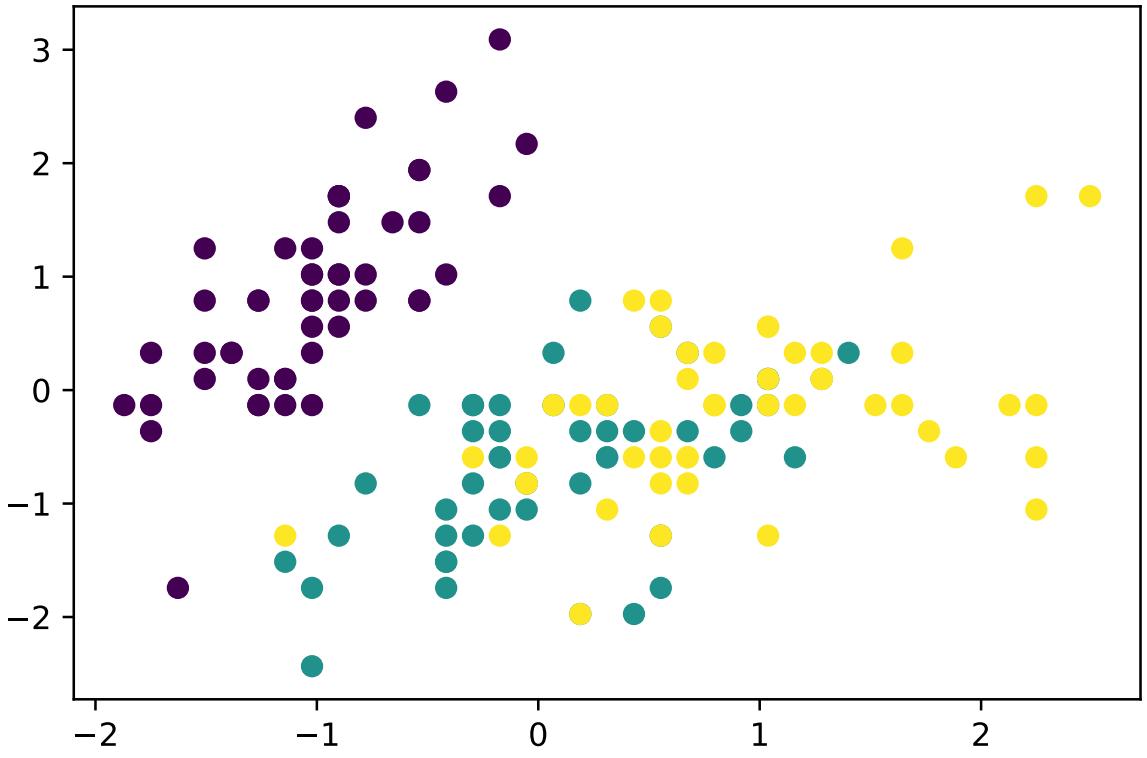
Cas #1 : m est grand

En pratique,

- Choisir un algorithme
- Définir une grille de recherche pour la valeur des hyperparamètres de l'algorithme sélectionné
- Séparer les données en sous-ensembles d'apprentissage, validation et test
- Pour chaque valeur sur la grille de recherche
  1. apprendre un modèle sur les données d'apprentissage
  2. évaluer les performances du modèle avec les données de validation
- Sélectionner les valeurs des hyperparamètres qui maximisent la performance du modèle (ou minimisent la fonction de coût)
- Apprendre un modèle sur les données d'apprentissage + les données de validation pour les valeurs des hyperparamètres sélectionnées.
- Évaluer la capacité de généralisation du modèle, *i.e.* évaluer les performances sur les données test

Découpage des données

Exemple : jeu de données Iris (CM07)



Découpage des données

Exemple : jeu de données Iris (CM07)

- **Choisir un algorithme** : le k-Plus Proche Voisin

Découpage des données

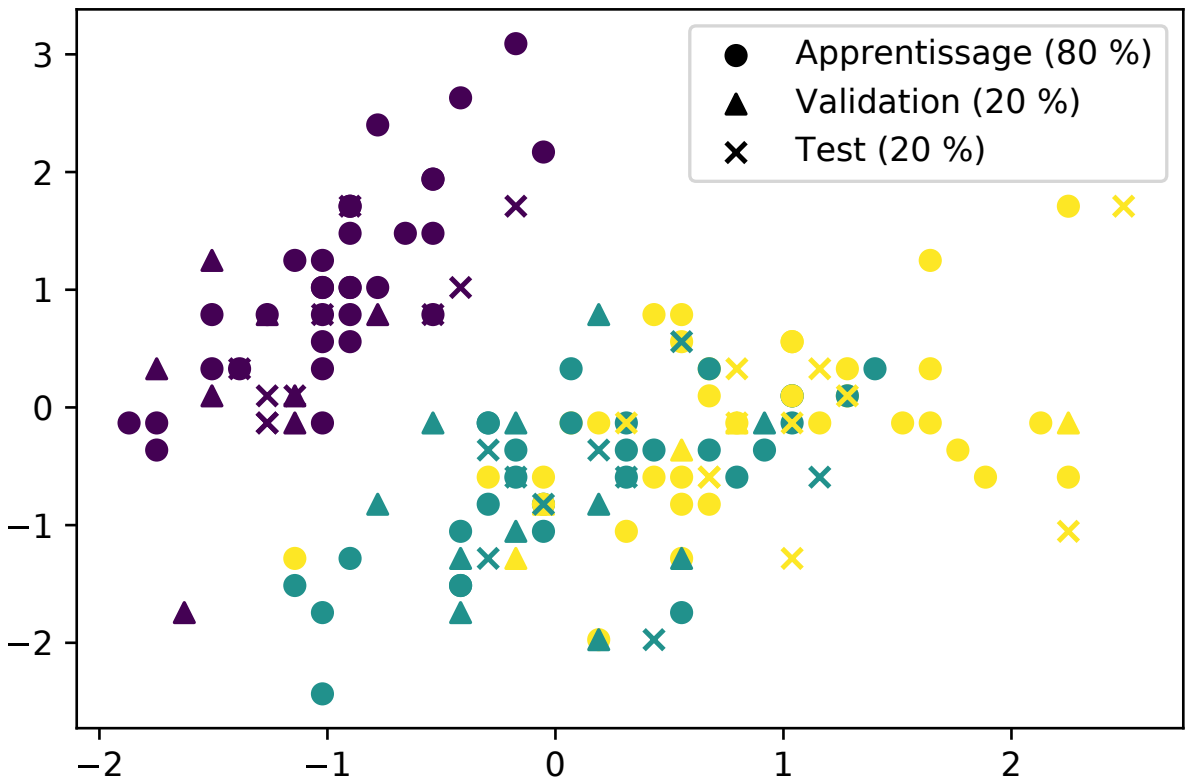
Exemple : jeu de données Iris (CM07)

- Choisir un algorithme : le k-Plus Proche Voisin
- **Définir une grille de recherche** : on cherche à déterminer la meilleure valeur de k possible (1 seul hyperparamètre).
  Testons par exemple $k \in [1, 2, 3, 5, 10, 15, 30]$

Découpage des données

Exemple : jeu de données Iris (CM07)

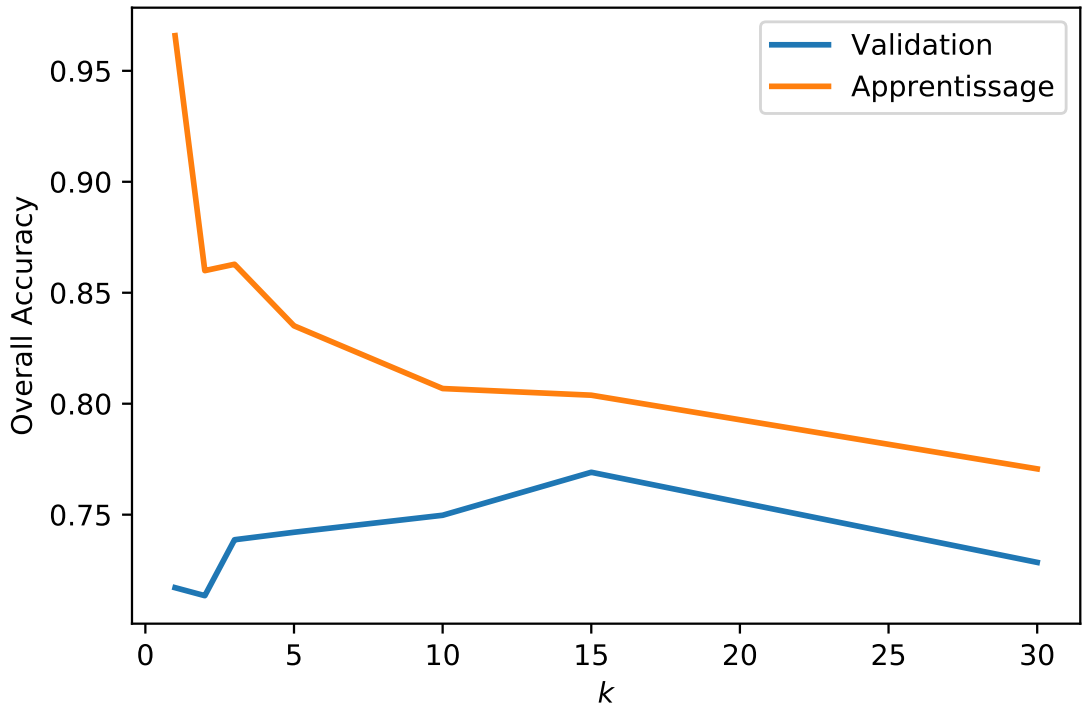
- Choisir un algorithme : le k-Plus Proche Voisin
- Définir une grille de recherche : on cherche à déterminer la meilleure valeur de k possible (1 seul hyperparamètre).
  Testons par exemple $k \in [1, 2, 3, 5, 10, 15, 30]$
- **Séparer les données**



Découpage des données

Exemple : jeu de données Iris (CM07)

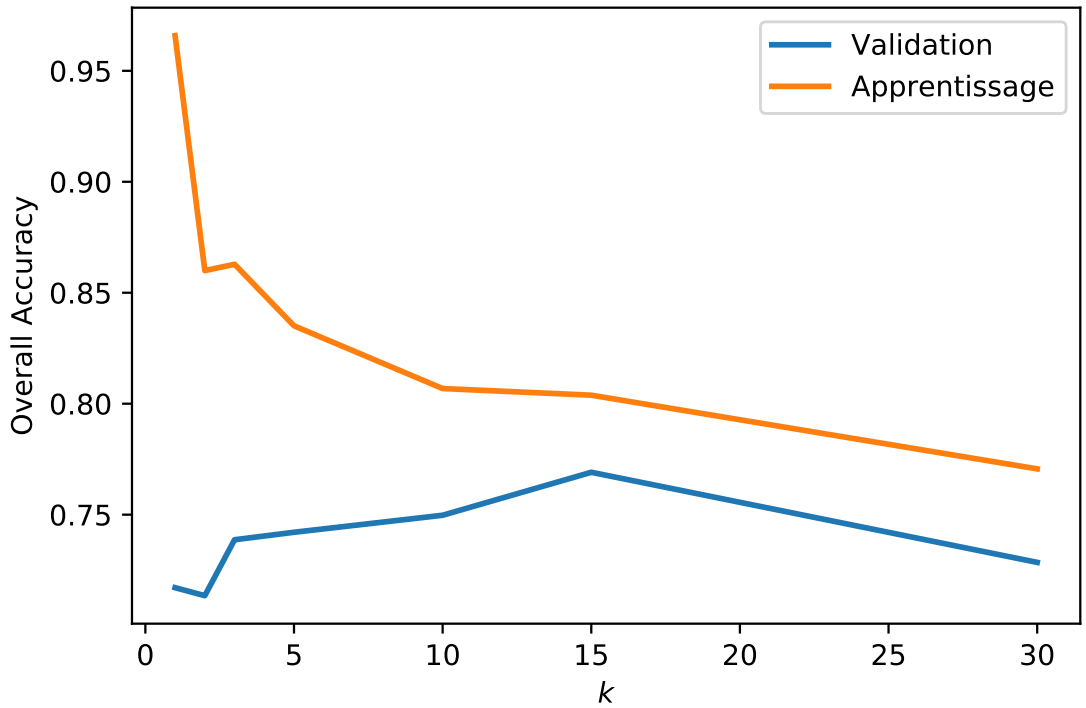
- **Évaluer les performances** sur les échantillons de validation pour les différentes valeurs des hyperparamètres.



Découpage des données

Exemple : jeu de données Iris (CM07)

- **Évaluer les performances** sur les échantillons de validation pour les différentes valeurs des hyperparamètres.

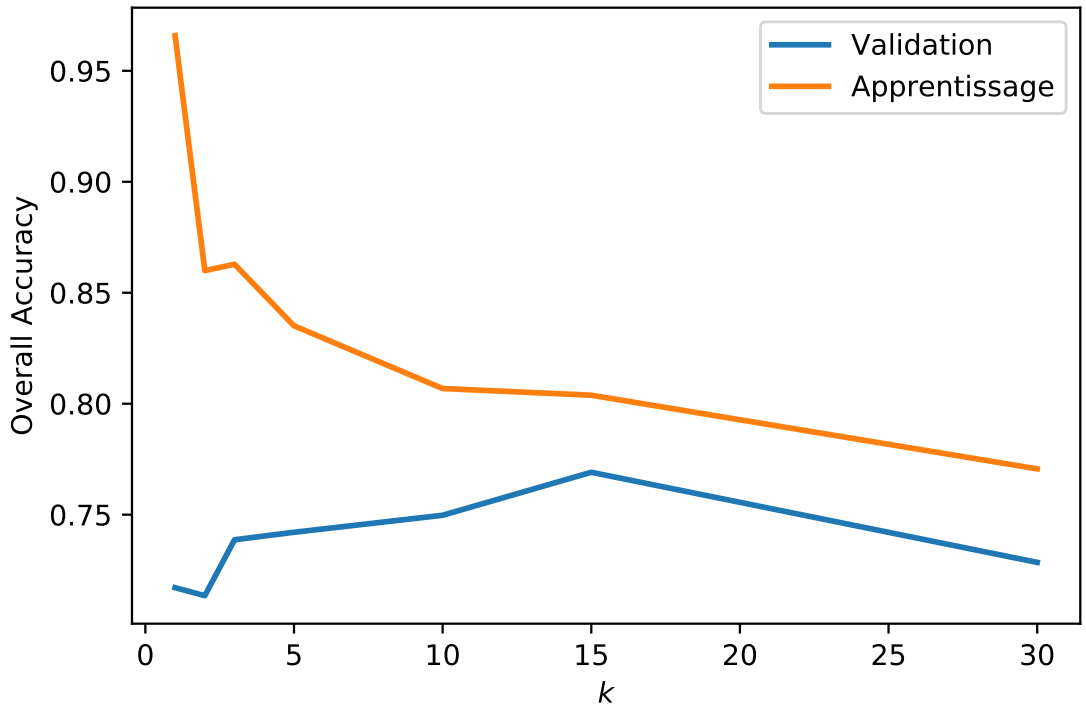


La courbe en orange montre (1−) le risque empirique, *i.e.*, évaluation du taux de bonne classification sur les données d'apprentissage.

Découpage des données

Exemple : jeu de données Iris (CM07)

- Évaluer les performances sur les échantillons de validation pour les différentes valeurs des hyperparamètres.



La courbe en orange montre (1−) le risque empirique, *i.e.*, évaluation du taux de bonne classification sur les données d'apprentissage.

- **Sélectionner les valeurs optimale des hyperparamètres** : $k = 15$

Découpage des données

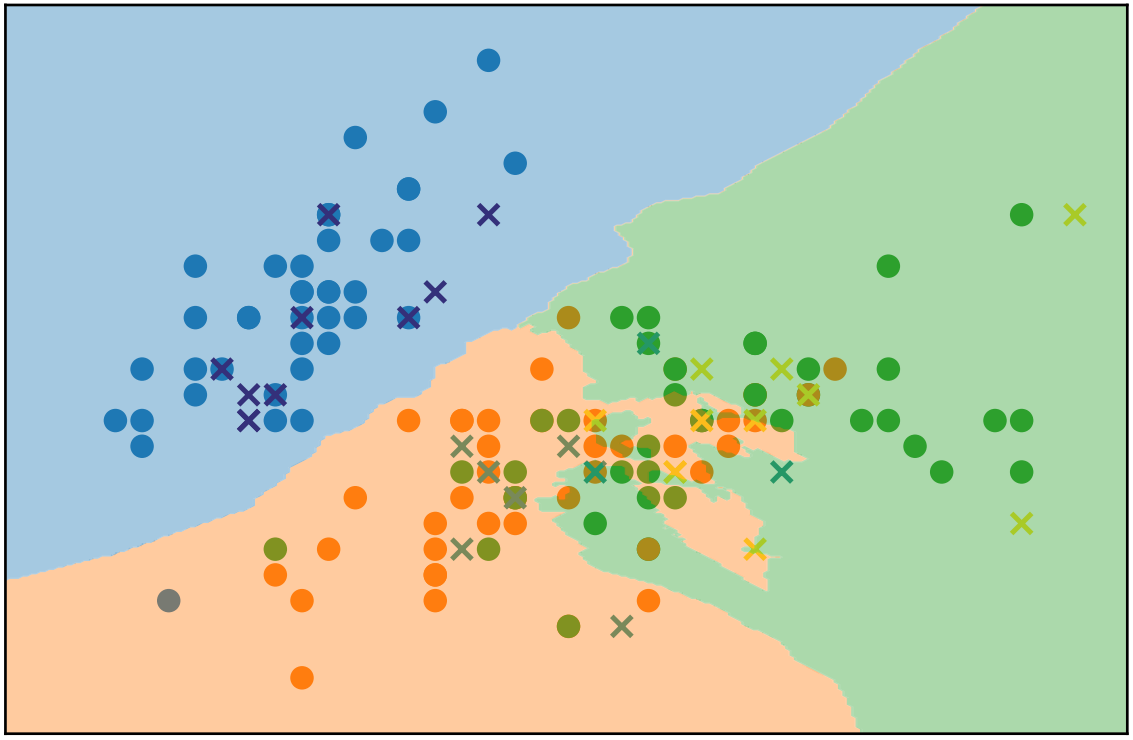
Exemple : jeu de données Iris (CM07)

- **Apprendre le modèle final sur les échantillons d'apprentissage et de validation**

Découpage des données

Exemple : jeu de données Iris (CM07)

- Apprendre le modèle final sur les échantillons d'apprentissage et de validation
- **Évaluer les performances sur les données de test**

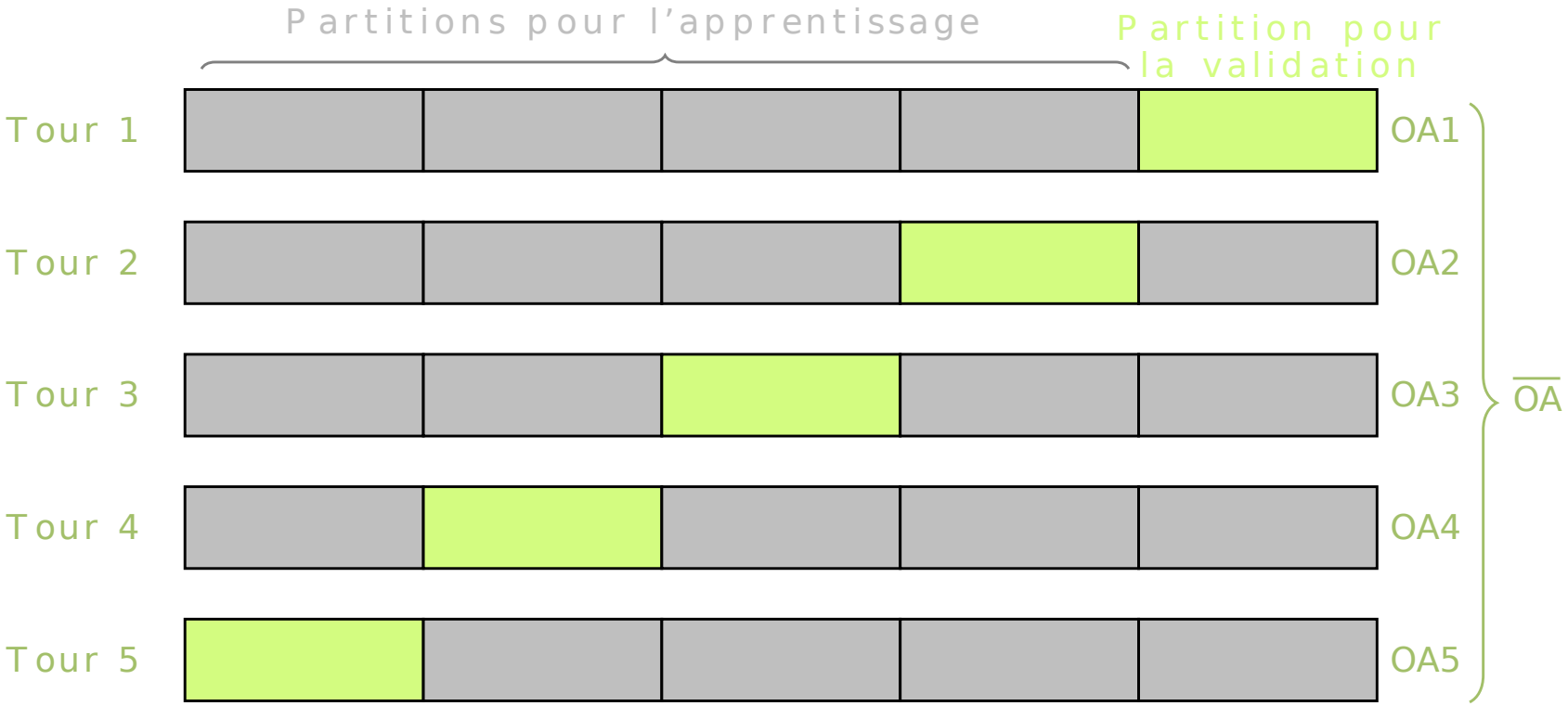


$OA = 76.6$ % (x : données de test)

Validation croisée

Cas #2 : *m* est petit

Validation croisée ou *k-fold cross-validation*



Validation croisée

Cas #2 : m est petit

Validation croisée ou *k-fold cross-validation*

1. Diviser l'ensemble des données d'apprentissage en k sous-ensembles de tailles égales
2. Répéter k fois (séquentiellement)
   - apprendre un modèle sur $k - 1$ sous-ensembles
   - évaluer sa performance (par exemple, le taux de bonne classification OA) sur le sous-ensemble restant

Validation croisée

Cas #2 : m est petit

Validation croisée ou *k-fold cross-validation*

1. Diviser l'ensemble des données d'apprentissage en k sous-ensembles de tailles égales
2. Répéter k fois (séquentiellement)
   - apprendre un modèle sur $k - 1$ sous-ensembles
   - évaluer sa performance (par exemple, le taux de bonne classification OA) sur le sous-ensemble restant

Question : Combien d'échantillons sont utilisés pour l'apprentissage de chaque modèle ?

Validation croisée

Cas #2 : m est petit

Validation croisée ou *k-fold cross-validation*

1. Diviser l'ensemble des données d'apprentissage en k sous-ensembles de tailles égales
2. Répéter k fois (séquentiellement)
   - apprendre un modèle sur $k-1$ sous-ensembles
   - évaluer sa performance (par exemple, le taux de bonne classification OA) sur le sous-ensemble restant

Question : Combien d'échantillons sont utilisés pour l'apprentissage de chaque modèle ?

$$\frac{m(k-1)}{k}$$

Validation croisée

Cas #2 : m est petit

Validation croisée ou *k-fold cross-validation*
Remarques :

- Comment choisir la valeur de k ?
  - plus le nombre de données utilisées pour la phase d'apprentissage est grand, plus le modèle est précis

Validation croisée

Cas #2 : m est petit

Validation croisée ou *k-fold cross-validation*

Remarques :

- Comment choisir la valeur de k ?
  - plus le nombre de données utilisées pour la phase d'apprentissage est grand, plus le modèle est précis
  - Cas particulier $k = m$: *leave-one-out*
    Principe
    - apprentissage de m modèles différents
    - une seule observation est utilisée dans la phase test

    Inconvénients
    - $k = m \Rightarrow m$ modèles à apprendre $\Rightarrow$ temps calculatoire $\nearrow$
    - comme une seule observation change pour l'apprentissage de chaque modèle, les modèles appris sont très similaires

Validation croisée

Cas #2 : m est petit

Validation croisée ou *k-fold cross-validation*
Remarques :

- Comment choisir la valeur de k ?
  - plus le nombre de données utilisées pour la phase d'apprentissage est grand, plus le modèle est précis
  - Cas particulier $k = m$: *leave-one-out*
    Principe
    - apprentissage de m modèles différents
    - une seule observation est utilisée dans la phase test

    Inconvénients
    - $k = m \Rightarrow m$ modèles à apprendre $\Rightarrow$ temps calculatoire $\nearrow$
    - comme une seule observation change pour l'apprentissage de chaque modèle, les modèles appris sont très similaires
- Quelle mesure de performance utilisée (OA, Kappa, taux de faux positifs, taux de vrais négatifs, *etc.*) ?

Bootstrap

Cas #2 : m est petit

Bootstrap

- tirages aléatoires **avec** remise de m observations dans l'ensemble des données

Cas #2 : m est petit

Bootstrap

- tirages aléatoires **avec** remise de m observations dans l'ensemble des données

Pour aller plus loin
La probabilité de tirer k fois une observation lors de m tirages aléatoires avec remise est donnée par la loi binomiale suivante :

$$\mathbb{P}(X = k) = \binom{k}{m} p^k (1-p)^{m-k},$$

avec p la probabilité de tirer aléatoirement l'observation.
Chaque observation a la même probabilité d'être tiré au sort, donc $p = \frac{1}{m}$:

$$\mathbb{P}(X = k) = \binom{k}{m} \left(\frac{1}{m}\right)^k \left(1 - \frac{1}{m}\right)^{m-k},$$

Si m est grand on peut calculer que :

- 36.79 % des échantillons ne sont pas inclus
- 36.79 % des échantillons sont inclus une seule fois
- 18.39 % des échantillons sont inclus exactement deux fois
- 6.13 % des échantillons sont inclus exactement trois fois
- 1.53 % des échantillons sont inclus exactement quatre fois
- *etc.*

Pour aller plus loin

Définitions : on distingue généralement trois types d'analyse statistique

1. les analyses descriptives
   - « résumer » les données en utilisant des mesures statistiques (moyenne, pourcentage, *etc.*)
   - première étape avant d'effectuer une analyse inférentielle
2. les analyses inférentielles
   - test statistique
   - intervalle de confiance
3. les analyses prédictives (*i.e.*, l'apprentissage automatique)

Sommaire

Évaluation des modèles de régression

Rappel sur la régression :

- $\{\mathbf{x}_i, y_i\}_{i=1}^N$ les N échantillons test
  - $\mathbf{x}_i \in \mathbb{R}^d$ une observation représentée par d variables
  - $y_i \in \mathbb{R}$ l'étiquette associée à l'observation $\mathbf{x}_i$
- f la fonction de décision appris par le modèle de régression
  - par exemple $f(\mathbf{x}_i) = \beta_0 + \sum_{j=1}^d \beta_j \times x_i^j$

Évaluation des modèles de régression

Erreur quadratique moyenne (*Mean Squared Error*) : la moyenne des carrés des résidus :

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^{N} \big(y_i - f(\mathbf{x}_i)\big)^2$$

En régression, le résidu est l'écart entre la valeur à expliquer/prédire (y_i) et la valeur expliquée/prédite ($f(\mathbf{x}_i)$), soit les résidus $y_i - f(\mathbf{x}_i)$ pour tout i.

Évaluation des modèles de régression

Erreur quadratique moyenne (*Mean Squared Error*) : la moyenne des carrés des résidus :

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}\big(y_i - f(\mathbf{x}_i)\big)^2$$

En régression, le résidu est l'écart entre la valeur à expliquer/prédire (y_i) et la valeur expliquée/prédite ($f(\mathbf{x}_i)$), soit les résidus $y_i - f(\mathbf{x}_i)$ pour tout i.

D'autres mesures possibles (variantes) :

- *Root-Mean-Square Error*

$$\text{RMSE} = \sqrt{\text{MSE}} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\big(y_i - f(\mathbf{x}_i)\big)^2}$$

- *Mean Absolute Error*

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}|y_i - f(\mathbf{x}_i)|$$

Évaluation des modèles de classification

Rappel sur la classification :

- $\{\mathbf{x}_i, y_i\}_{i=1}^{N}$ les N échantillons test
  - $\mathbf{x}_i \in \mathbb{R}^d$ une observation représentée par d variables
  - $y_i \in \mathcal{Y}$ l'étiquette associée à l'observation $\mathbf{x}_i$
    - $\mathcal{Y} = \{-1, 1\}$ ou $\mathcal{Y} = \{0, 1\}$ pour la classification binaire
    - $\mathcal{Y} = \{1, \ldots, \mathcal{C}\}$ pour la classification multiclasses ($\mathcal{C}$ classes)

La matrice de confusion : $C = \{c_{ij}\}_{i,j=1}^{\mathcal{C}}$ pour $\mathcal{C}$ classes

Réelle \ Prédite	1	2	$\cdots$	$\cdots$	j	$\cdots$	$\mathcal{C}$
1	c_{11}	c_{12}	$\cdots$	$\cdots$	c_{1j}	$\cdots$	$c_{1\mathcal{C}}$
2	c_{21}	c_{22}	$\cdots$	$\cdots$	c_{2j}	$\cdots$	$c_{2\mathcal{C}}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
i	c_{i1}	c_{i2}	$\cdots$	$\cdots$	c_{ij}	$\cdots$	$c_{i\mathcal{C}}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\mathcal{C}$	$c_{\mathcal{C}1}$	$c_{\mathcal{C}2}$	$\cdots$	$\cdots$	$c_{\mathcal{C}j}$	$\cdots$	$c_{\mathcal{C}\mathcal{C}}$

Évaluation des modèles de classification

La matrice de confusion : $C = \{c_{ij}\}_{i,j=1}^{\mathcal{C}}$ pour $\mathcal{C}$ classes

Réelle \ Prédite	1	2	$\cdots$	$\cdots$	j	$\cdots$	$\mathcal{C}$
1	c_{11}	c_{12}	$\cdots$	$\cdots$	c_{1j}	$\cdots$	$c_{1\mathcal{C}}$
2	c_{21}	c_{22}	$\cdots$	$\cdots$	c_{2j}	$\cdots$	$c_{2\mathcal{C}}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
i	c_{i1}	c_{i2}	$\cdots$	$\cdots$	c_{ij}	$\cdots$	$c_{i\mathcal{C}}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\mathcal{C}$	$c_{\mathcal{C}1}$	$c_{\mathcal{C}2}$	$\cdots$	$\cdots$	$c_{\mathcal{C}j}$	$\cdots$	$c_{\mathcal{C}\mathcal{C}}$

- c_{ij} corresponds au nombre d'échantillons qui appartiennent à la classe i et pour lequel l'algorithme de classification a prédit la classe j
- les éléments diagonaux $\{c_{ii}\}_{i=1}^{j}$ correspondent donc aux échantillons dont la classe a correctement été prédite par l'algorithme
- $\sum_{i=1}^{\mathcal{C}} \sum_{j=1}^{\mathcal{C}} c_{ij} = N$ avec N le nombre d'observations test

Mesures d'évaluation

- Taux de bonne classification (en anglais *Overall Accuracy*) :

$$OA = \frac{\sum_{i=1}^{C} c_{ii}}{N}$$

- $0\ \% \leq OA \leq 100\ \%$ on cherche à maximiser le taux de bonne classification ($100\ \%$)

Mesure d'évaluation

- Le coefficient Kappa :

$$\text{Kappa} = \frac{OA - p_h}{1 - p_h}$$

avec $p_h = \frac{1}{N^2} \sum_{i=1}^{C} \left(\sum_{j=1}^{C} c_{ij} \right) \left(\sum_{j=1}^{C} c_{ji} \right)$ le pourcentage d'observations bien étiquetées attribué au hasard

- Le coefficient Kappa permet de s'affranchir du taux de bonne classification dû à l'aléatoire
- Référentiel de Landis et Koch pour interpréter la valeur de Kappa.

Interprétation	**Valeur de Kappa**
Excellente	$1.00 - 0.81$
Bonne	$0.80 - 0.61$
Faible	$0.60 - 0.41$
Négligeable	$0.20 - 0.00$
Mauvaise	< 0.00

Source : J. R. Landis and G. G. Koch. The measurement of observer agreement for categorical data. *Biometrics*. (1) :159 ?174, 1977.

Cas particulier de la classification binaire : $\mathcal{C} = 2$

Réelle \ Prédite	Positive	Négative
Positive	Vrais Positifs (TP)	Faux Negatifs (FN)
Negative	Faux Positifs (FP)	Vrais Négatifs (TN)

- Taux de bonne classification : $OA = \frac{TP+TN}{TP+FN+FP+TN}$
- Taux de faux positifs : $FPR = \frac{FP}{FP+TN}$ (erreur de type I)
- Taux de faux négatifs : $FNR = \frac{FN}{FN+TP}$ (erreur de type II)

Cas particulier de la classification binaire : $\mathcal{C} = 2$

Réelle \ Prédite	Positive	Négative
Positive	Vrais Positifs (TP)	Faux Negatifs (FN)
Negative	Faux Positifs (FP)	Vrais Négatifs (TN)

- Taux de bonne classification : $OA = \frac{TP+TN}{TP+FN+FP+TN}$
- Taux de faux positifs : $FPR = \frac{FP}{FP+TN}$ (erreur de type I)
- Taux de faux négatifs : $FNR = \frac{FN}{FN+TP}$ (erreur de type II)
- Rappel (*recall*, *sensitivity*) : Rappel $= \frac{TP}{TP+FN}$
- Spécificité (*specificity*) : Spécificité $= \frac{TN}{FP+TN} = 1 - FPR$

Courbe ROC

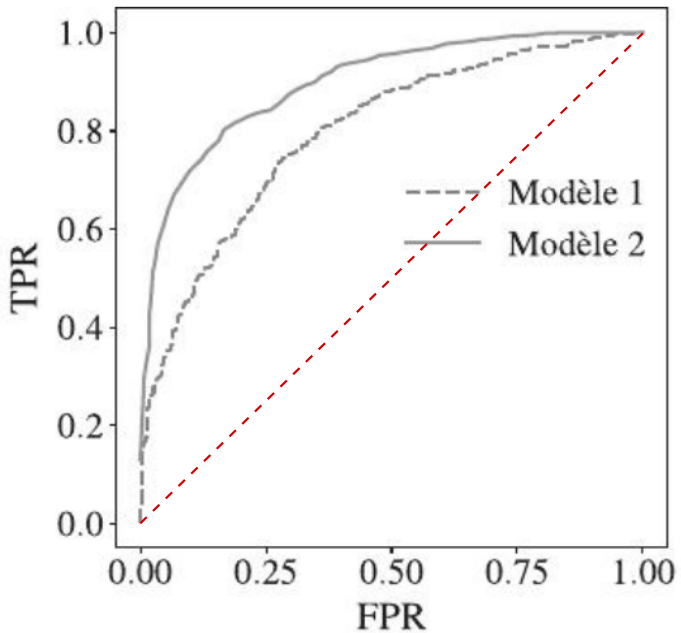
Courbe ROC (*Receiver-Operator Characteristic*)

- pour les méthodes de classification binaire dont la fonction de décision retourne un score qui doit être seuillée, et pas directement une classe

Courbe ROC

Courbe ROC (*Receiver-Operator Characteristic*)

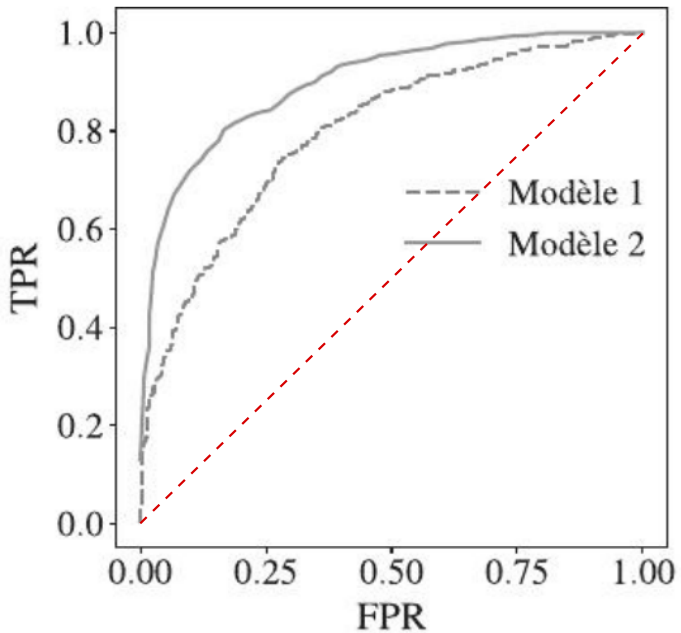
- pour les méthodes de classification binaire dont la fonction de décision retourne un score qui doit être seuillée, et pas directement une classe
- pour différentes valeurs de seuil : Rappel $= f(1 - \text{Spécificité}) = f(FPR)$



Courbe ROC

Courbe ROC (*Receiver-Operator Characteristic*)

- pour les méthodes de classification binaire dont la fonction de décision retourne un score qui doit être seuillée, et pas directement une classe
- pour différentes valeurs de seuil : Rappel $= f(1 - \text{Spécificité}) = f(FPR)$

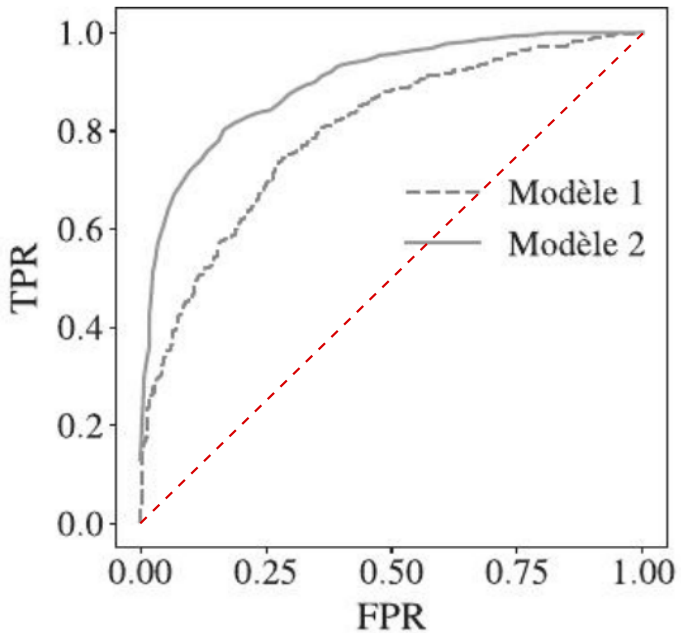


Si la courbe ROC est en-dessous de la ligne rouge en pointillés, le modèle fait moins bien que l'aléatoire

Courbe ROC

Courbe ROC (*Receiver-Operator Characteristic*)

- pour les méthodes de classification binaire dont la fonction de décision retourne un score qui doit être seuillée, et pas directement une classe
- pour différentes valeurs de seuil : Rappel $= f(1 - \text{Spécificité}) = f(FPR)$



Si la courbe ROC est en-dessous de la ligne rouge en pointillés, le modèle fait moins bien que l'aléatoire

- L'information de la courbe ROC peut être résumée par l'AUROC aire sous la courbe ROC : $0 \leq \text{AUROC} \leq 1$.

Source : Chloé-Agathe Azencott. Introduction au machine learning. *Dunod*, 2019.

Autres critères d'évaluation

Outre les mesures d'évaluation basées sur la capacité de généralisation du modèle, il peut être intéressant de considérer d'autres critères :

- **Complexité calculatoire**
  - temps d'exécution = temps d'apprentissage + temps de prédiction
  - espace mémoire utilisé

  On parle de passage à l'échelle (algorithme scalable ou *scalability* en anglais)

Autres critères d'évaluation

Outre les mesures d'évaluation basées sur la capacité de généralisation du modèle, il peut être intéressant de considérer d'autres critères :

- **Complexité calculatoire**
  - temps d'exécution = temps d'apprentissage + temps de prédiction
  - espace mémoire utilisé

  On parle de passage à l'échelle (algorithme scalable ou *scalability* en anglais)
- **Interprétabilité** : comprendre ce qui a mené un algorithme à prendre une décision
  - simplicité du modèle *versus* boîte noire

Autres critères d'évaluation

Outre les mesures d'évaluation basées sur la capacité de généralisation du modèle, il peut être intéressant de considérer d'autres critères :

- **Complexité calculatoire**
  - temps d'exécution = temps d'apprentissage + temps de prédiction
  - espace mémoire utilisé

  On parle de passage à l'échelle (algorithme scalable ou *scalability* en anglais)
- **Interprétabilité** : comprendre ce qui a mené un algorithme à prendre une décision
  - simplicité du modèle *versus* boîte noire
- Capacité d'adaptation du modèle
  - aux données manquantes ou aberrantes
  - aux données non pertinentes
  - aux données hétérogènes (par exemple présence de variables quantitatives et qualitatives)

Rasoir d'Occam

Rasoir d'Occam (Ockham) ou principe de simplicité :

- Pour des taux d'erreur comparables, le modèle de plus petite complexité est préférée

$\Rightarrow$ meilleure compréhension de la décision prise par le modèle d'apprentissage supervisé

$\Rightarrow$ interprétation plus simple du phénomène étudié

FIN