

MACHINE LEARNING

Jérôme Lacaille
Expert émérite

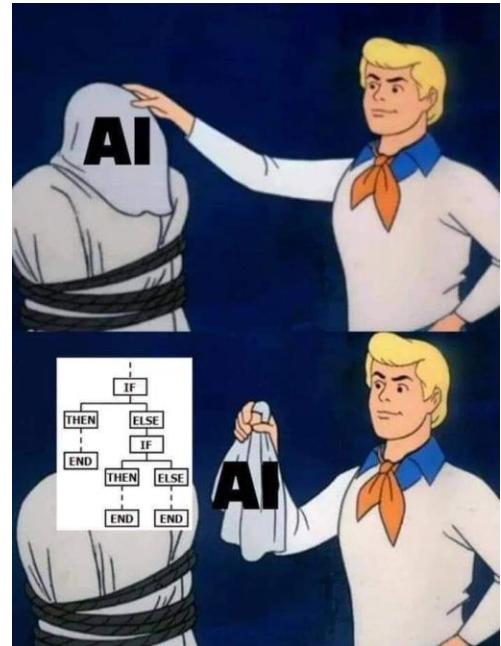
DataLab Safran Aircraft Engines
Formation DataClimber

2023



Objectifs de la formation

- Quelle est l'importance de la data dans le ML ? Quelle méthodologie mettre en place pour créer un modèle viable ?
- Exposé *non exhaustif* d'un panel algorithmes sans présentation des preuves mathématiques
- Quel(s) algorithme(s) choisir ? Comment le(s) choisir ? Comment le(s) valider ?





Sommaire

0. C'EST QUOI UN ALGORITHME DE MACHINE LEARNING?

1. PRÉPARATION DES DONNÉES

2. LES DIFFÉRENTS ALGORITHMES DE MACHINE LEARNING

3. VALIDATION D'UN MODÈLE

4. DES MOOCS POUR ALLER PLUS LOIN

0

C'EST QUOI UN ALGO DE ML ?

I.A. VS Machine Learning VS Deep Learning

INTELLIGENCE ARTIFICIELLE

Ensemble des techniques permettant à une machine de reproduire un comportement humain



1950's

1960's

1970's

1980's

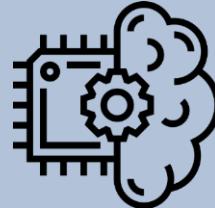
1990's

2000's

2010's

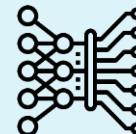
MACHINE LEARNING

Toute technique permettant à une machine d'apprendre sans avoir été explicitement programmée



DEEP LEARNING

Sous-ensemble du ML basé principalement sur l'utilisation de réseaux de neurones artificiels



Le principe du Machine learning

ON SOUHAITE APPRENDRE À UNE MACHINE À RECONNAÎTRE DES CHATS



Approche traditionnelle

On donne des instructions à la machine sous forme d'un programme qu'elle appliquera pas à pas pour labéliser l'image qu'on lui présente



Approche ML

On donne à la machine beaucoup d'images de chats et de non-chats; elle apprend d'elle-même à faire la différence



Principal pré-requis :

Une connaissance poussée de la sortie

Principal pré-requis

Beaucoup de données de bonne qualité et représentatives de la sortie

0 – C'est quoi un algo de ML ?

■ Fonction

- > $Y = a^*x + b$

■ Algorithme

- > If ... then ... else

■ Algorithme de Machine Learning

- > Composé généralement de 2 phases
 1. Apprentissage : création d'un « modèle » à partir d'un jeu de données
 2. Exécution : utilisation du modèle pour prédire / classifier à partir de nouvelles entrées
- > Algorithme de Machine Learning = algorithme produisant des fonctions / algorithmes basés sur des données et non sur un ensemble de règles déterminées à l'avance

0 – Prédire ou comprendre ?



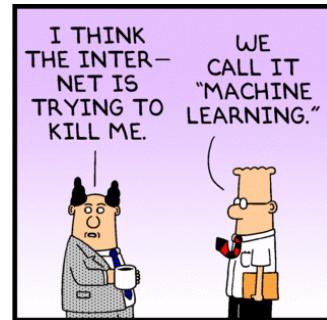
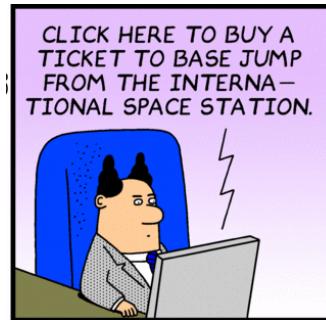
La complexité de certains algorithmes de prédiction en fait souvent des boîtes noires que l'on ne peut en général pas interpréter

- > La notion de **modèle** diffère alors du sens communément établi : il ne s'agit plus d'une **théorie scientifique** mais seulement d'une **technique de prévision**
- > Si le problème est **uniquement de prédire**, une méthode doit être jugée du point de vue de son **efficacité** et de sa **robustesse**

0 – Prédire ou comprendre ?

Peut-on prédire sans comprendre ?

- > Le progrès des outils de calcul semble bien montrer que oui
- > De nombreuses applications ne nécessitent pas de disposer d'une théorie qui serait d'ailleurs bien difficile à élaborer (prévisions de stocks, détection de défauts, prédiction de prix,...)
- > **Tout dépend du cas d'utilisation et des enjeux !** (enjeu de sécurité...)

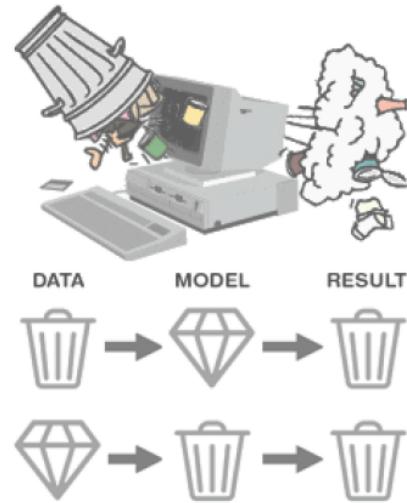


1

PRÉPARATION DES DONNÉES

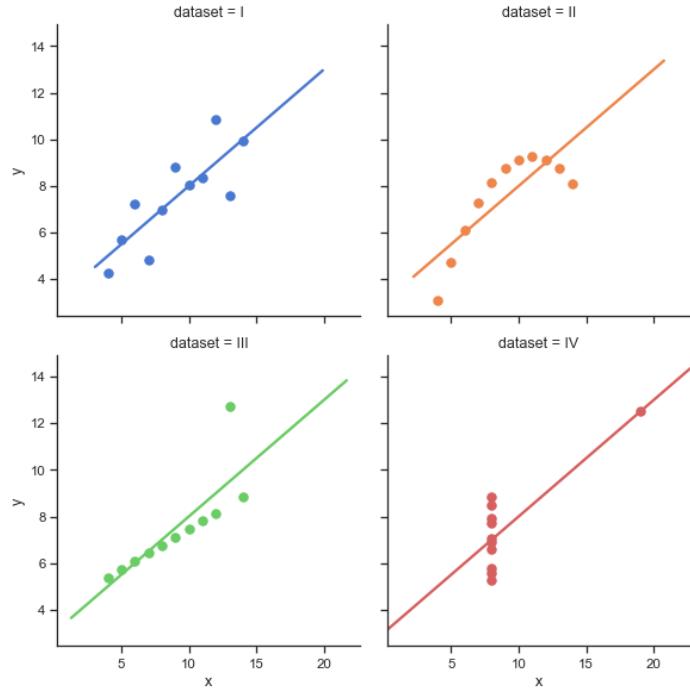
1 – Qualité de la donnée – Problématique

- Pourquoi s'attarder sur la qualité de la donnée ?
- Qualité pauvre de la donnée → résultat biaisé
→ mauvaise décision
- Confiance dans la donnée
 - > Nettoyage des données: 30 à 80% du temps et du budget d'un projet
 - > Être proactif dans le nettoyage de la donnée: identifier les failles pour fiabiliser la donnée à l'avenir



1 – Qualité de la donnée – Illustration

■ Le quartet d'Anscombe



■ 4 jeux de données visuellement différents...

■ ...mais qui ont :

- > La même moyenne selon x
- > La même moyenne selon y
- > La même variance selon x
- > La même variance selon y
- > Le même coefficient de corrélation (0,816)
- > La même équation de la droite de régression linéaire

■ Illustration de problématiques différentes

- > Modèle non adapté (dataset II)
- > Présence d'outliers (dataset III)
- > Données mal réparties (dataset VI)

1 – Qualité de la donnée – Illustration

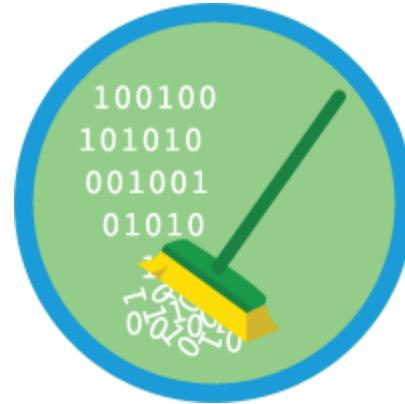
■ The Datasaurus Dozen



1 – Qualité de la donnée – Evaluation

■ Les critères

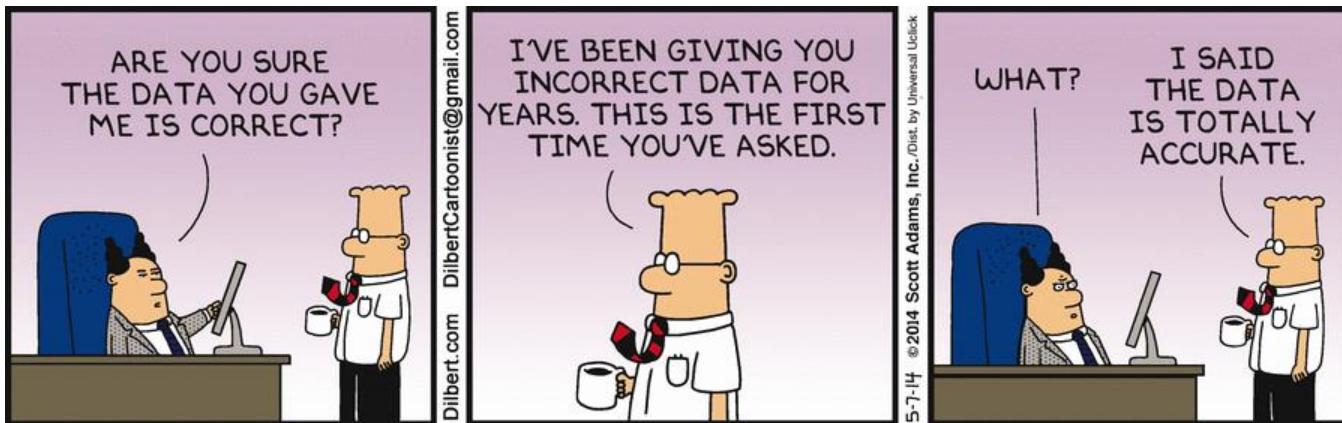
- > Accessibilité
 - ◆ Base de données
 - ◆ Formatage, erreur d'attribution, bonne infrastructure
- > Conformité
 - ◆ Peut-on facilement comprendre la donnée ?
- > Cohérence entre les données
 - ◆ Information conflictuelle
- > Exactitude
 - ◆ Valeurs aberrantes ?
 - ◆ Donnée incohérente ou périmée
- > Unicité
 - ◆ Doublons ?
- > Complétude
 - ◆ Valeurs manquantes?
 - ◆ Labélisation manquante, source de données indisponible, absence de mesures ...



1 – Qualité de la donnée

■ Mise en place de la gouvernance des données (*Data Quality Management*)

- > Métriques (Dashboard de suivi)
- > Cartographie des données
- > Mise en place de méthodes et d'outils de nettoyage automatique
- > Responsabilité collective



1 – Préparation de la donnée

Aucun algo de Machine Learning ne sait comment fonctionne un moteur. Vous éventuellement...

■ Choix des indicateurs

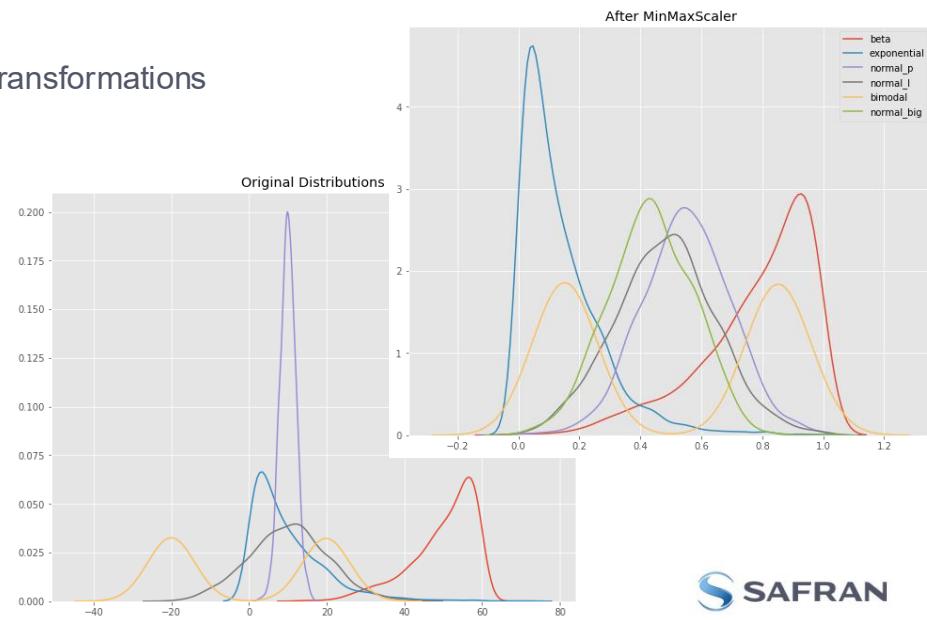
- Sélection « experte » / « métier » des données pertinentes

■ Construction d'indicateurs / « feature engineering » – Exemples

- Marge EGT plutôt que EGT
- N1 réduit plutôt que N1
- Produits de variables, cosinus d'une position angulaire, transformations pertinentes du point de vue « métier »

■ Normalisation / standardisation

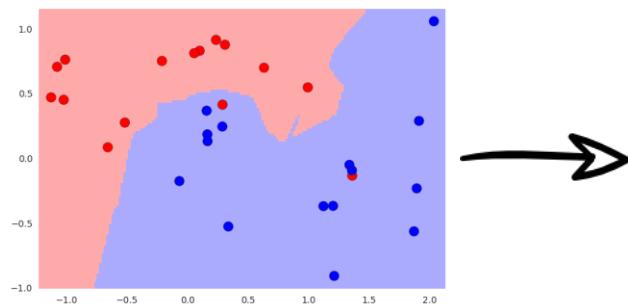
- Transformation permettant de ramener les indicateurs à des échelles comparables
- Utile voire indispensable selon les algorithmes de ML
- Plusieurs méthodes disponibles
- [Scale, Standardize, or Normalize with Scikit-Learn](#)
- [scikit-learn - scaler comparison](#)



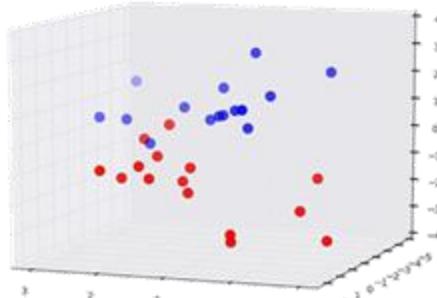
1 – Le fléau de la dimension

Que se passe-t-il lorsqu'on se retrouve avec un grand nombre de *features* relativement au nombre d'observations ?

Modèle 5-NN appliqué à un jeu de données



2 *features* en entrée permettant de décrire la classe rouge ou bleue



3 *features*, avec le même nombre de données

Avec 3 *features*,

- il y a moins de points d'exemples disponibles pour effectuer la prédiction relativement à la taille de l'espace à couvrir
- L'algorithme doit aller chercher plus loin les 5 plus proches voisins pour effectuer la prédiction

Le manque de données nécessaire à l'apprentissage du modèle explose très vite lorsqu'on augmente la dimension des données d'entrée...

Une solution : utiliser une technique de réduction de dimensions



2

ALGORITHMES DE MACHINE LEARNING

2 – Familles de machine learning

■ Apprentissage supervisé:

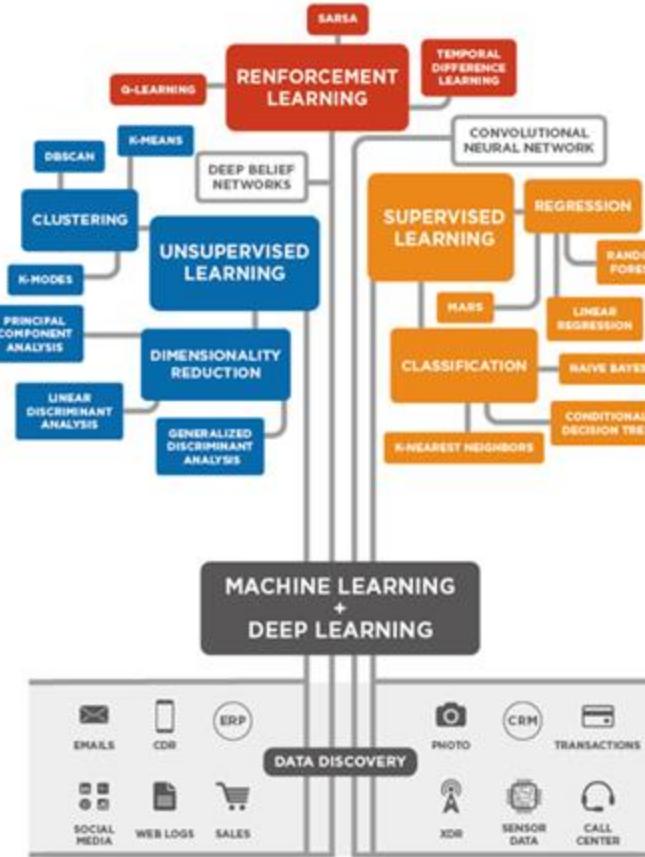
- > Etablir un modèle à partir de ce qu'on connaît (jeu d'entraînement)
 - ◆ Soit pour prédire
 - ◆ Soit pour classifier
- > $Y = f(X)$
 - ◆ Y: l'estimation de la valeur / de la classe
 - ◆ f: le modèle
 - ◆ X: la base d'apprentissage

■ Apprentissage non supervisé

- > Extraire des groupes avec des caractéristiques communes sans à priori sur les données
- > Réduire la dimension du problème
- > Méthode d'exploration et de compréhension des données

■ Apprentissage par renforcement:

- > Apprendre pour un agent (modèle, robot ...) les actions à prendre
- > Apprentissage à partir d'une expérience et par découverte petit à petit

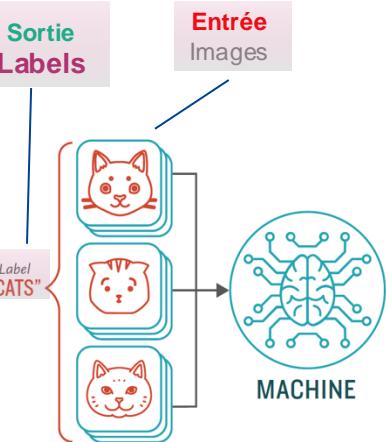


Voir aussi : [scikit-learn algorithm cheat sheet](#)

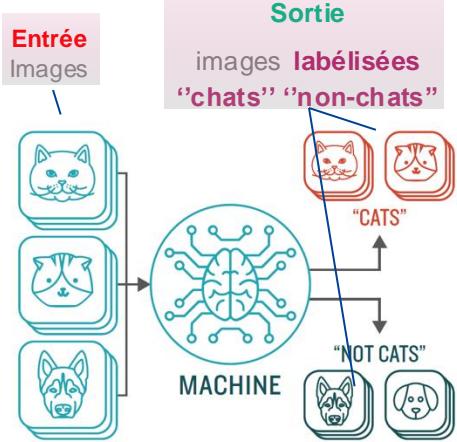
Supervisé Vs non Supervisé

Supervisé

Apprentissage supervisé

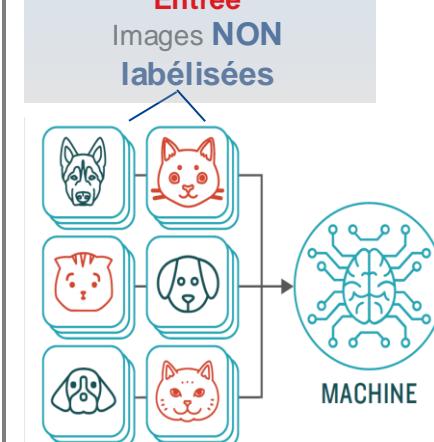


Test et validation

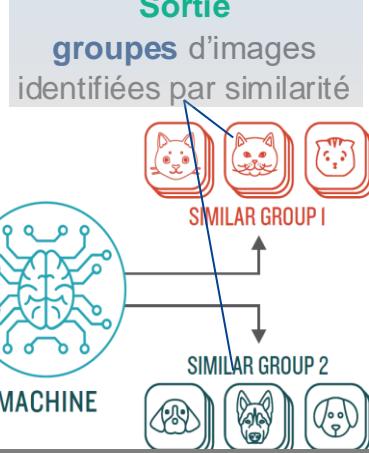


Non supervisé

Apprentissage non supervisé



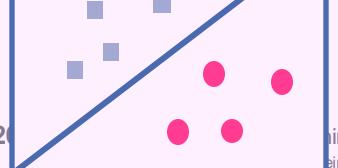
Test et validation



Types de problèmes supervisés

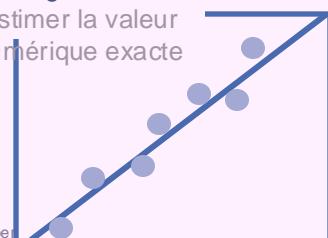
Classification

Classer les points dans catégories



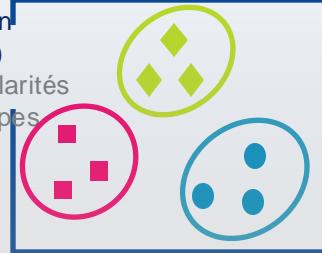
Régression

Estimer la valeur numérique exacte



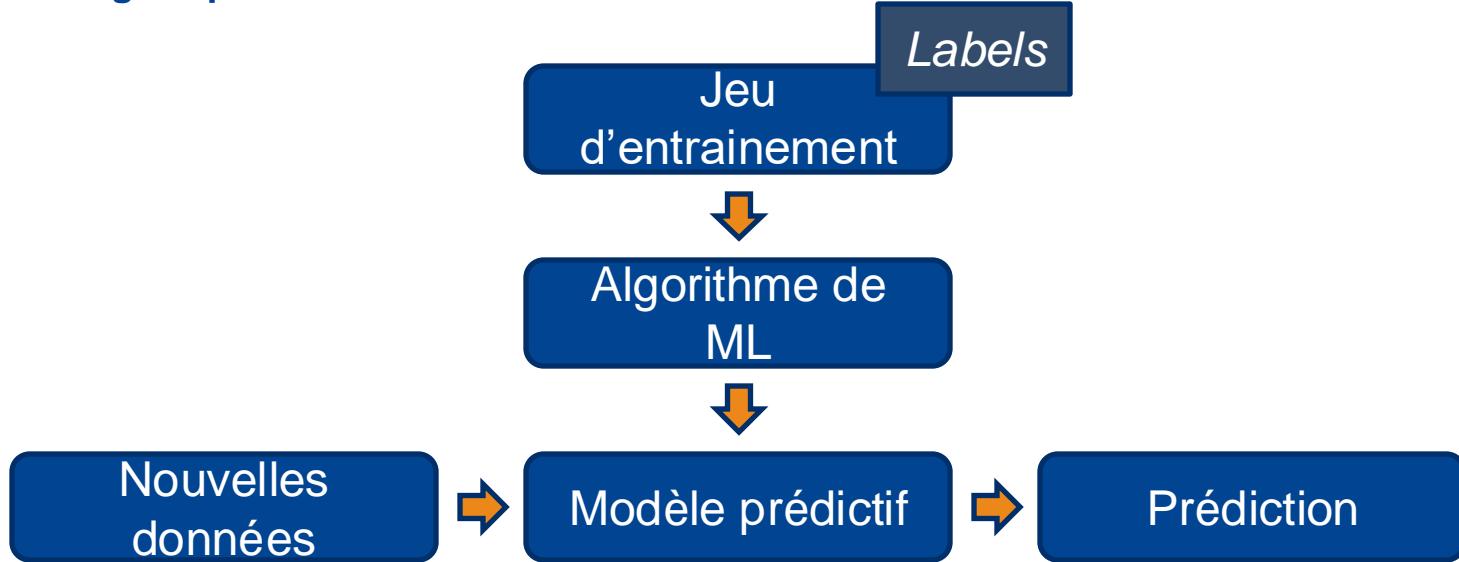
Classification (Clustering)

Identifier les similarités dans des groupes

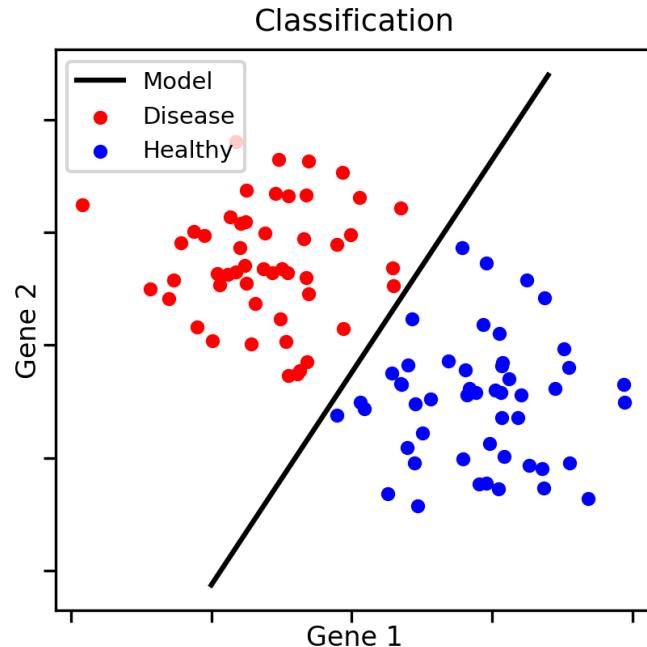


2 – Familles de machine learning

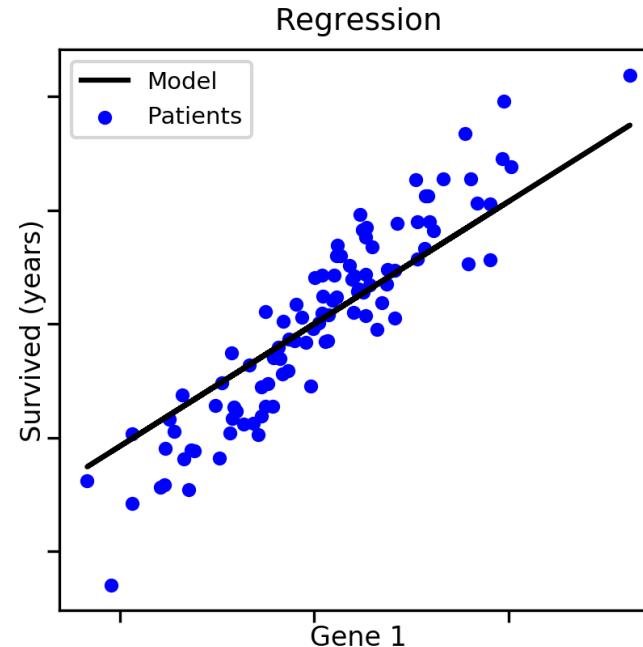
■ Apprentissage supervisé



2 – Apprentissage supervisé : Deux besoins différents



Classification: regrouper les données
Objectif: prédire l'appartenance d'une nouvelle instance à un groupe



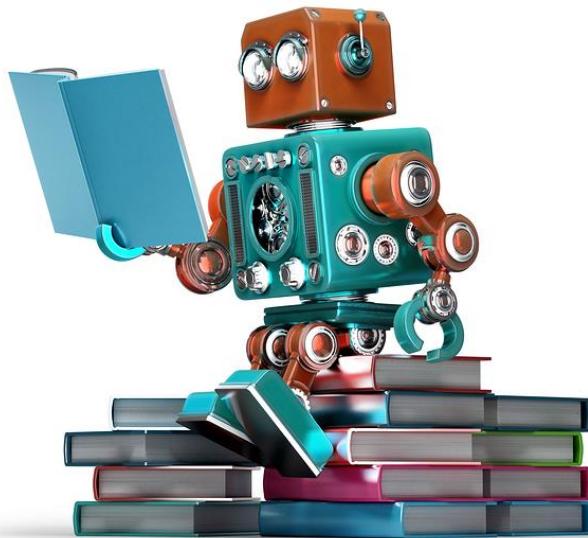
Régression: trouver le modèle qui relie les données
Objectif: prédire la prochaine valeur

2 – Apprentissage supervisé – Exemples

| Classification | Régression |
|--|---|
| <ul style="list-style-type: none">- Binaire (chien vs chat, oui vs non, spam vs non spam ...)- Classification multi-labels (caractères manuscrits, génétiques, profils clients ...) | <ul style="list-style-type: none">- Cours de la bourse- Dégradation de l'EGT- ... |

2 – Apprentissage supervisé

■ De la théorie à la pratique...



2 – Apprentissage en pratique

TD – Scikit learn en quelques mots

■ <https://scikit-learn.org/>

■ [In Depth: Parameter tuning for SVC | Medium](#)

■ Les grandes étapes de mise en place d'un algorithme

- > Avoir importer le jeu de donnée et identifier les variables cibles/explorative
- > Importer un algorithme (*estimator*) et le paramétriser:

```
>>> from sklearn import svm  
>>> clf = svm.SVC(gamma=0.001, C=100.)
```

- > Faire l'apprentissage (fit) du modèle:

```
>>> clf.fit(digits.data[:-1], digits.target[:-1])
```

- > Faire une prédiction à partir de nouvelles données:

```
>>> clf.predict(digits.data[-1:])  
[...]
```

- > Sauvegarder/Charger votre modèle:

```
>>> import pickle  
>>> s = pickle.dumps(clf)  
>>> clf2 = pickle.loads(s)
```

2 – Apprentissage en pratique

TD – Scikit learn en quelques mots

■ Séparation entre jeux d'apprentissage et de test:

```
# from sklearn.cross_validation import train_test_split # Version 0.17.1
from sklearn.model_selection import train_test_split # version 0.18.1
# split the data with 50% in each set
data_test = train_test_split(data, target
                             , random_state=0
                             , train_size=0.5)
data_train, data_test, target_train, target_test = data_test
```

■ Obtenir la précision d'un algorithme:

- > Dans le cas de la classification

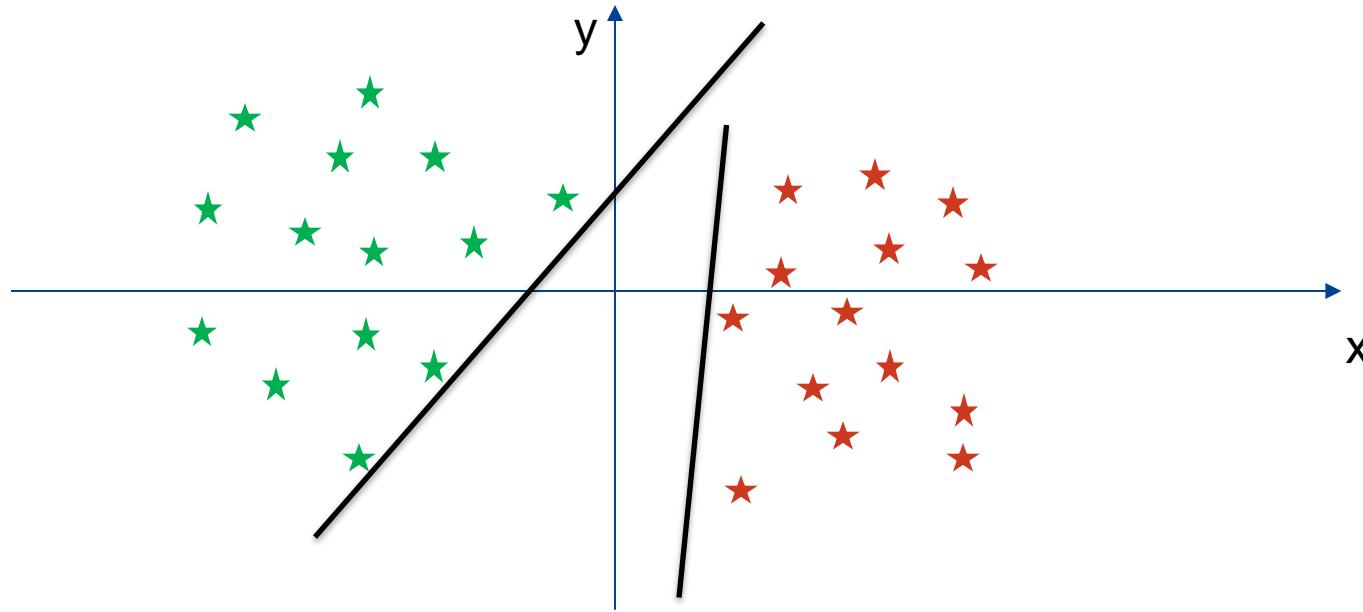
```
from sklearn.metrics import accuracy_score
accuracy_score(result, target) # 96% de réussite
```

- > Dans le cas de la régression:

```
>>> from sklearn.metrics import r2_score
>>> y_true = [3, -0.5, 2, 7]
>>> y_pred = [2.5, 0.0, 2, 8]
>>> r2_score(y_true, y_pred)
0.948...
```

2 – Apprentissage supervisé > Classification *Support Vector Machine (SVM)*

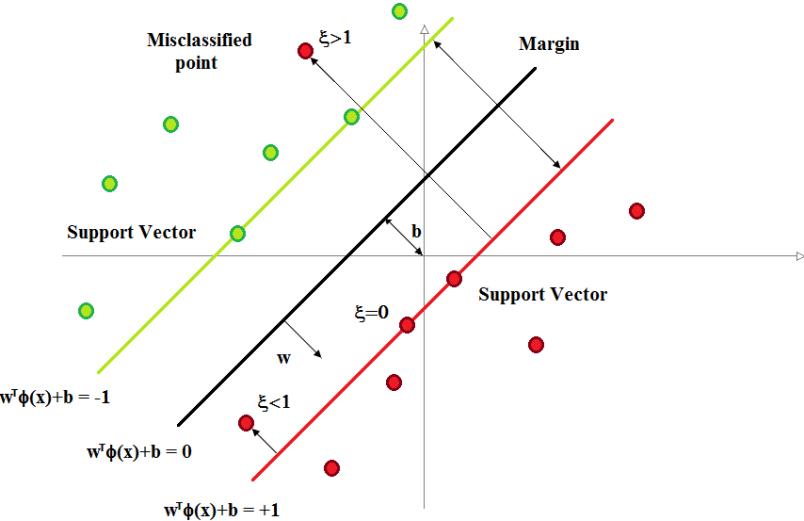
■ Question: comment séparer les groupes de points verts et rouges?



2 – Apprentissage supervisé > Classification *Support Vector Machine (SVM)*

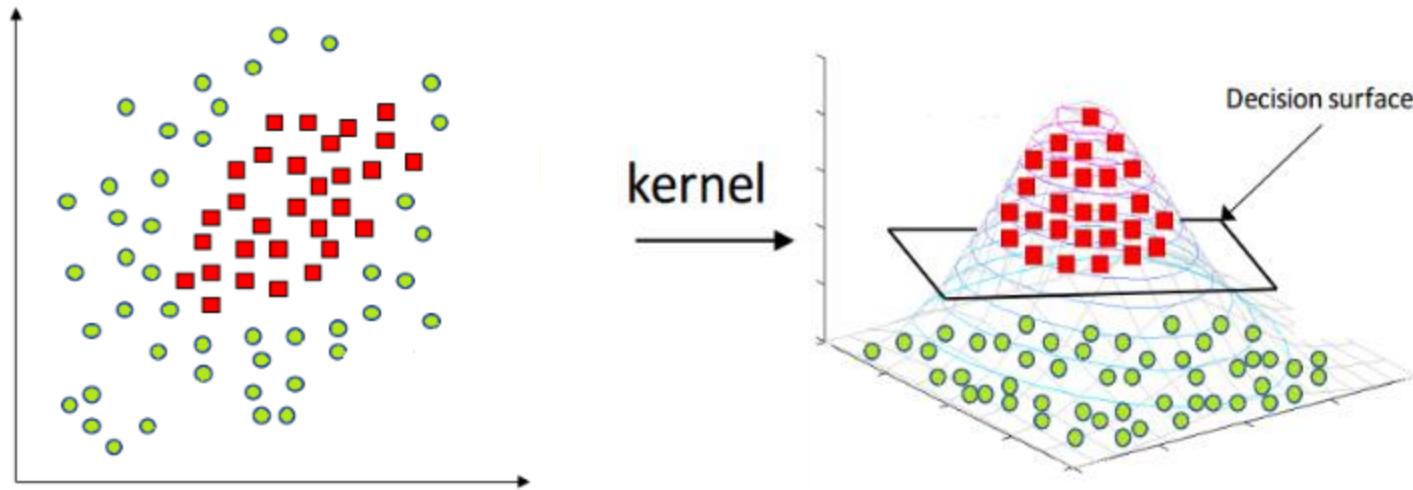
■ Résumé:

- On se sert des points pour construire l'hyperplan H.
H n'étant pas unique, on cherchera à l'optimiser
- Algorithme d'optimisation:
 - ◆ Classification: maximiser la marge entre les vecteurs supports
- On peut utiliser aussi les SVM pour faire des régressions



2 – Apprentissage supervisé SVM: l'astuce du noyau

- Objectif: transformer l'espace d'entrée via une fonction noyau (kernel) pour pouvoir trouver une séparation linéaire



2 – Apprentissage supervisé > Classification SVM

■ Avantages des SVM:

- > Simplicité de la méthode
- > Capacité à traiter de grandes dimensionnalités
- > Traitement des problèmes non linéaires
- > Non paramétrique
- > Méthode robuste

■ Inconvénients:

- > Pas de modèle explicite pour les noyaux non linéaires
- > Difficulté d'interprétation
- > Difficulté à traiter un grand nombre de données

■ <https://scikit-learn.org/stable/modules/svm.html>

SCR : MAJ LOI PCORE



■ CONTEXTE

- La pression interne au core compartiment du Silvercrest est nécessaire aux calculs de ventilation d'ensemble et de logique active du LPTACC. Cette pression ne peut être mesurée en vol. Il est donc nécessaire de **définir un modèle prédictif de cette**



METHODE

- Apprentissage supervisé d'une fonction de prédiction de la Pcore à partir de données d'essais → régression par machines à vecteurs supports
- Détermination d'une table Pcore/P13 vs. Mach et PCN12R pour implémenter le modèle prédictif dans le FADEC (contraintes de développement logiciel)



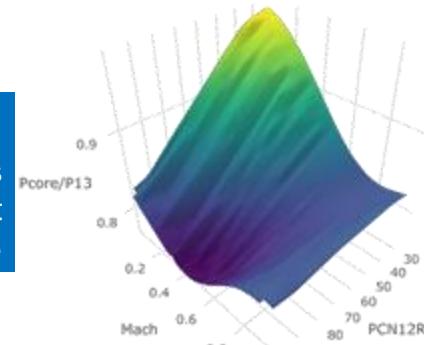
RESULTATS

- Modèle $P_{core}/P_{13} = f(PCN12R, Mach)$
- Table permettant d'obtenir la valeur de P_{core}/P_{13}



GAINS

- Modèle prédictif 2 fois plus précis que la loi actuelle
- Perspectives : implémentation du modèle dans le FADEC



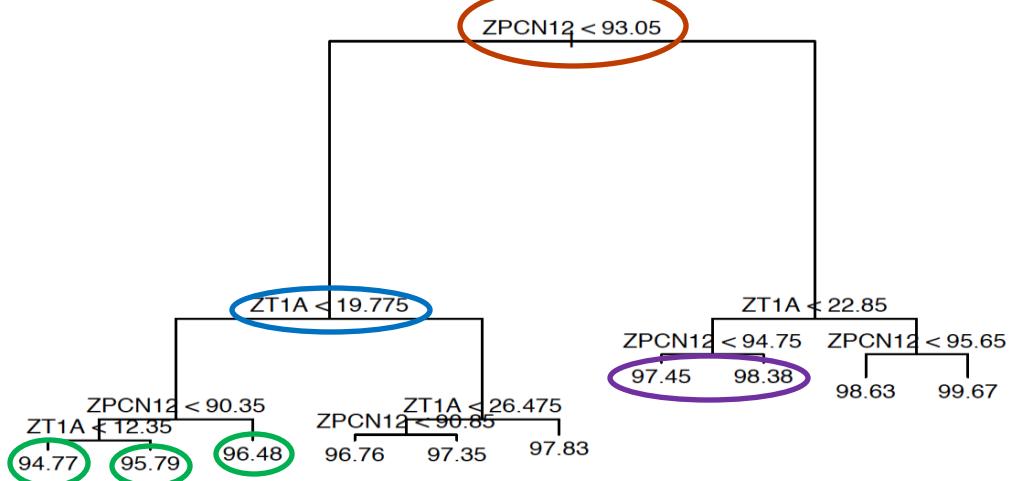
2 – Apprentissage supervisé > Classification

Arbres de décision

■ Principe:

- > Les méthodes de classifications/régressions basées sur des arbres de décision définissent des règles logiques en sélectionnant des variables et en découplant de manière progressive l'espace de recherche.
- > Racine : le premier nœud de l'arbre.
- > Feuille : un nœud terminal.
- > Règle : l'expression entre deux nœuds.
- > Régions : les ensembles définis par les feuilles formant la partition de l'espace des observations.

*Un arbre de décision
établissant la relation
entre la vitesse du corps HP
d'un turbofan, les
commandes et les
conditions extérieures.*



2 – Apprentissage supervisé > Classification

Arbres de décision

■ Comment mesurer la précision d'un découpage ?

■ Indice de Gini:

- > Mesure statistique permettant de rendre compte de la répartition d'une variable au sein d'une population
- > Gini = 0 = nœud pur = tous les éléments dans le nœud sont de la même classe

$$G_i = 1 - \sum_{k=1}^n p_{i,k}^2$$

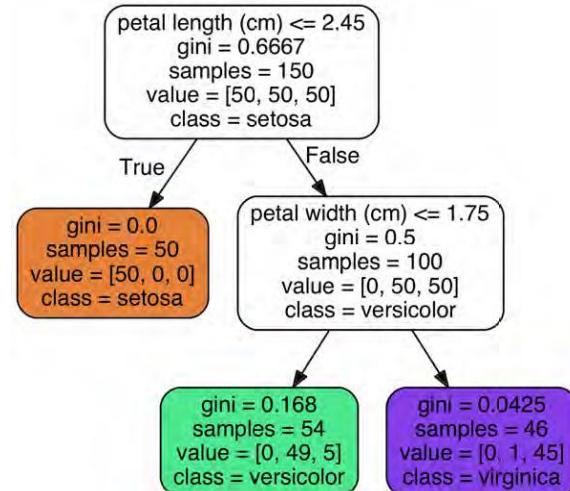
- > $p_{i,k}$ la proportion d'éléments appartenant à la classe i présent dans l'ensemble k

■ Mesure de l'entropie

- > Même principe qu'en thermodynamique: mesure du désordre

$$H_i = - \sum_{\substack{k=1 \\ p_{i,k} \neq 0}}^n p_{i,k} * \log(p_{i,k})$$

- > Entropie = 0 = tous les éléments dans le nœud sont de la même classe



2 – Apprentissage supervisé > Classification Arbres de décision

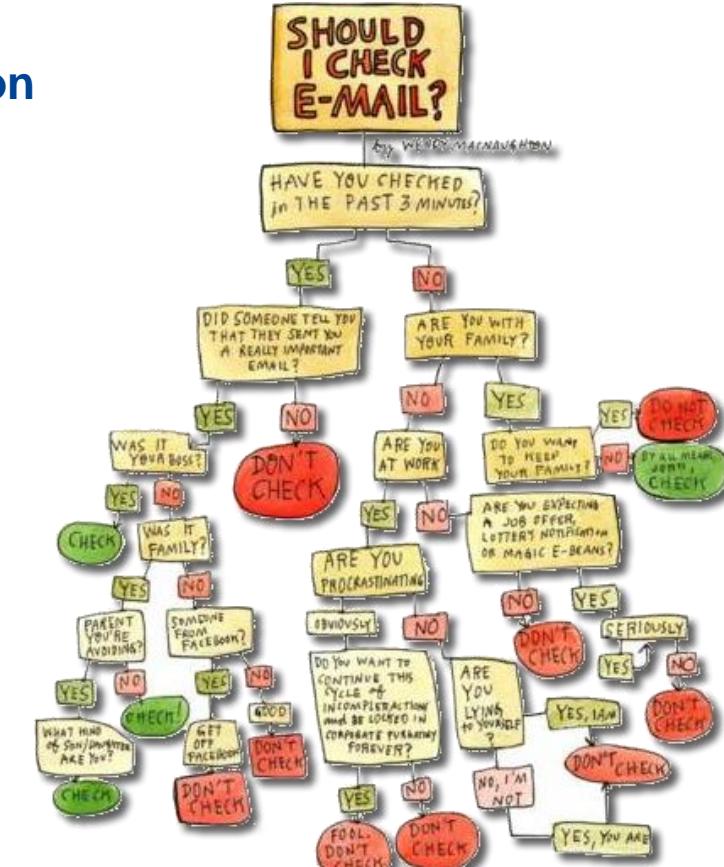
■ Avantages:

- > Facilement interprétables
- > Accepte les valeurs manquantes
- > Rapide d'implémentation
- > S'adapte à des données complexes

■ Inconvénients

- > Difficilement exploitable sur d'autres échantillons
- > Difficulté à extrapoler (notion d'*overfitting*)
- > Beaucoup de solutions possibles

■ <https://scikit-learn.org/stable/modules/tree.html>



THIS PUBLIC SERVICE ANNOUNCEMENT WAS BROUGHT TO YOU BY DELL.

Exemples Safran Aircraft Engines

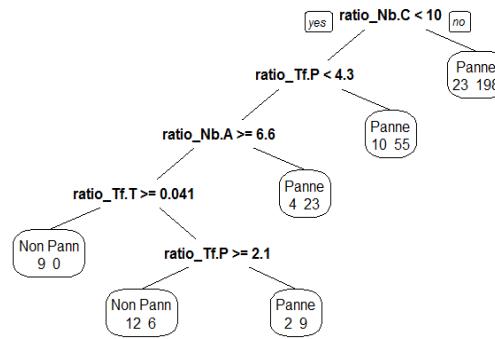
OBJECTIF : PRÉVOIR LES ANOMALIES DE BHMP sur le moteur M88



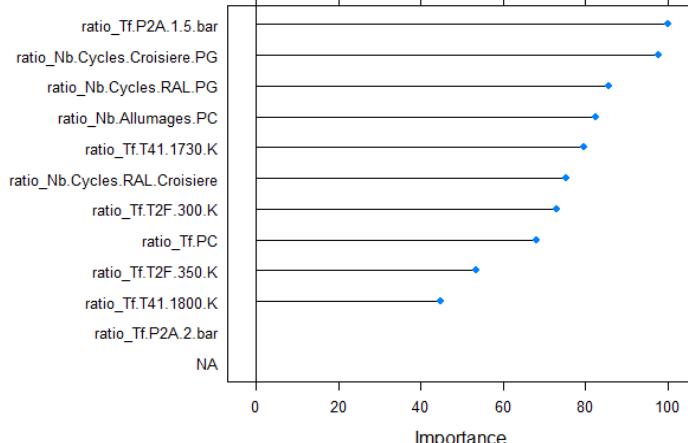
DONNEES D'ENTREE

- ✓ 415 BHMP avec panne
- ✓ 86 BHMP sans panne
- ✓ Paramètres d'entrée du modèle :
 - Indicateurs calculés sur les historiques de vols des BHMP

Arbre de décision



Random Forest

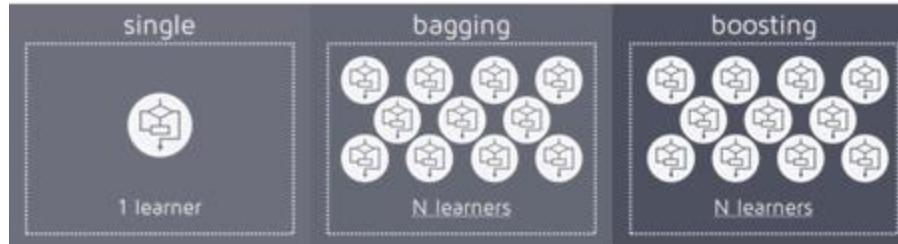


| Pourcentage de bien classé | Pourcentage de bonnes détections (POD) | Pourcentage de fausses alarmes (PFA) |
|----------------------------|--|--------------------------------------|
| 85% | 98% | 14% |

2 – Apprentissage supervisé

Les méthodes ensemblistes: Bagging et boosting

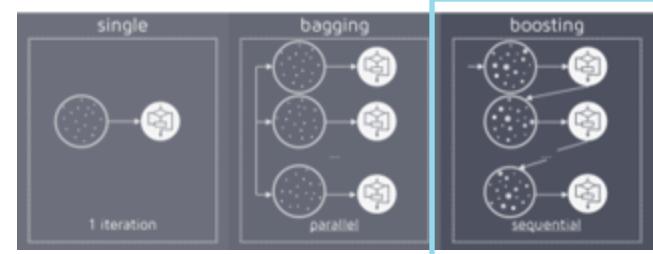
■ Génération des estimateurs



■ Partitionnement aléatoire du jeu d'entrée avec remise (*bootstrap*)



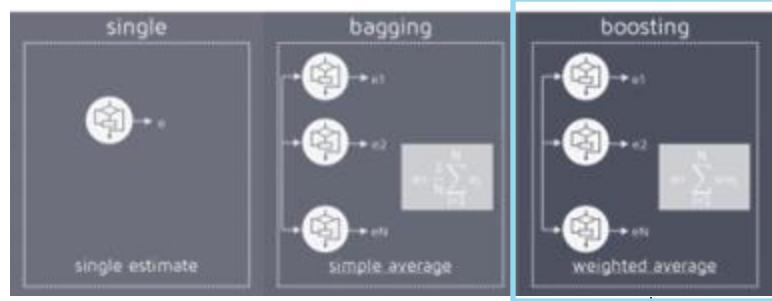
Calcul des poids



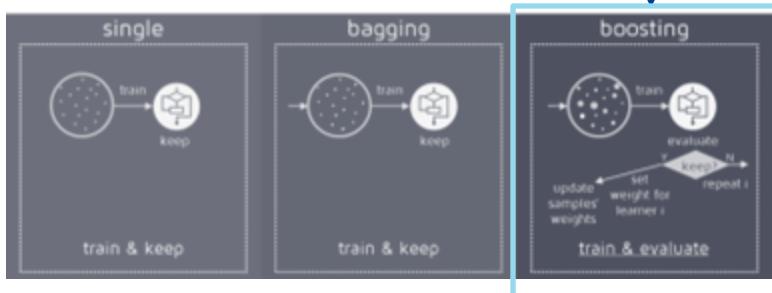
2 – Apprentissage supervisé

Bagging et boosting – Estimation de la réponse

■ Calcul de l'estimation globale/moyenne:



Calcul des poids



2 – Apprentissage supervisé *Bagging et boosting*

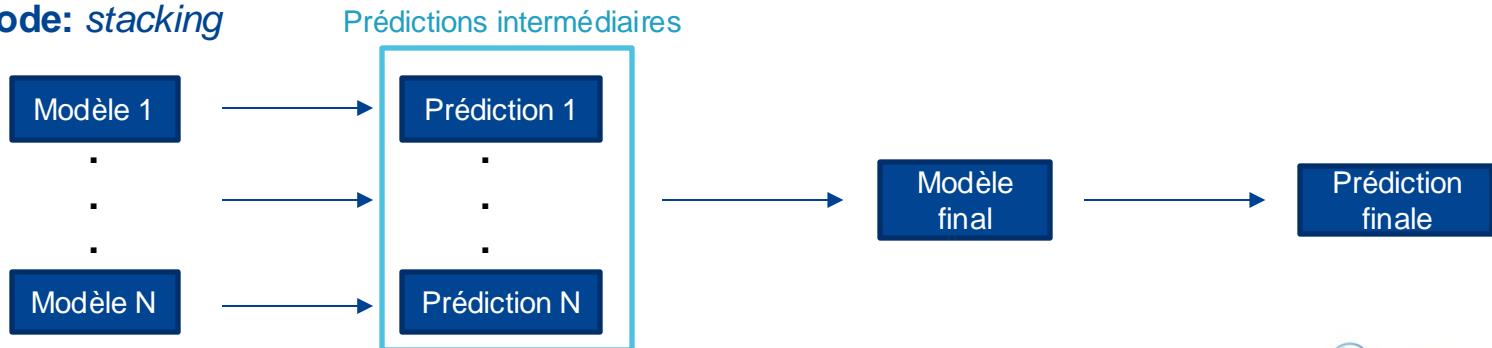
■ Principe de base:

- Minimiser le bruit, le biais et la variance de la prédiction
- On peut combiner plusieurs algorithmes aux résultats ‘faibles’ pour obtenir un meilleur algorithme final
- Exemples d’algorithmes de boosting: AdaBoost, LPBoost, XGBoost, GradientBoost ...
- Exemple d’algorithme de bagging: combiner plusieurs arbres de décision → Random Forest

■ Le bagging aura tendance à réduire l’over-fitting

■ Le boosting réduira le biais, mais peu augmenter le risque d’over-fitting

■ Autre méthode: stacking



2 – Apprentissage supervisé

Bagging et boosting

■ **Bagging:**

- <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.BaggingClassifier.html>

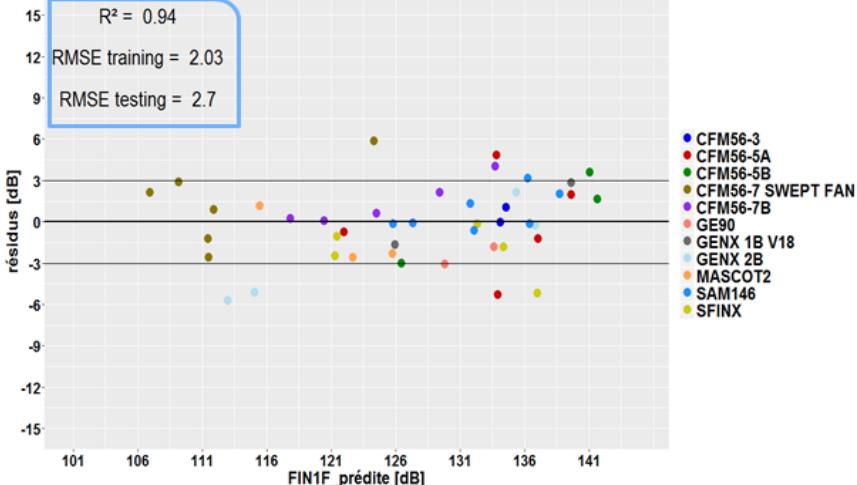
■ **Boosting:**

- <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.AdaBoostClassifier.html>

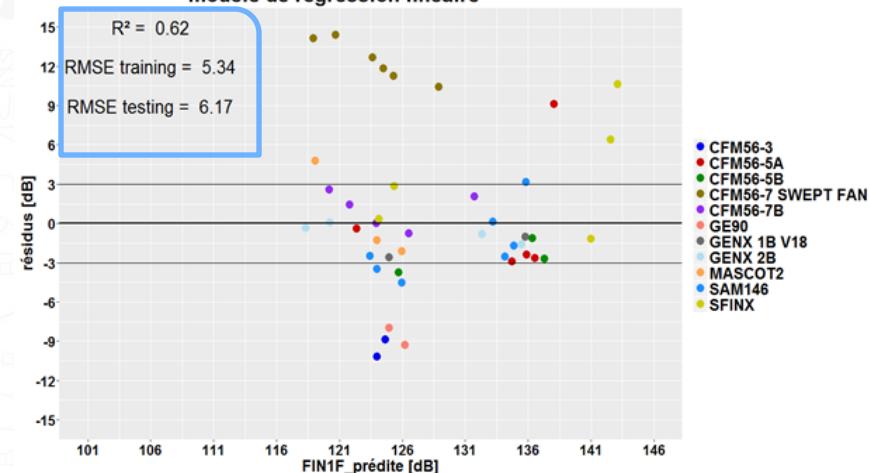
Exemples Safran Aircraft Engines

OBJECTIF : PRÉDIRE UNE PUISSEANCE ACOUSTIQUE AVEC DES CARACTÉRISTIQUES GÉOMÉTRIQUES POUR DES NOUVELLES ARCHITECTURES MOTEUR (AVANT PROJET)

modèle de régression GBM



modèle de régression linéaire



POUR DES ARCHITECTURES CONNUES, LE MODÈLE DES ARBRES DE RÉGRESSION **GBM “GRADIENT BOOSTING MODEL”** DONNE DE MEILLEURS RÉSULTATS QUE LE MODÈLE TRADITIONNELLEMENT UTILISÉ PAR LES MÉTIERS

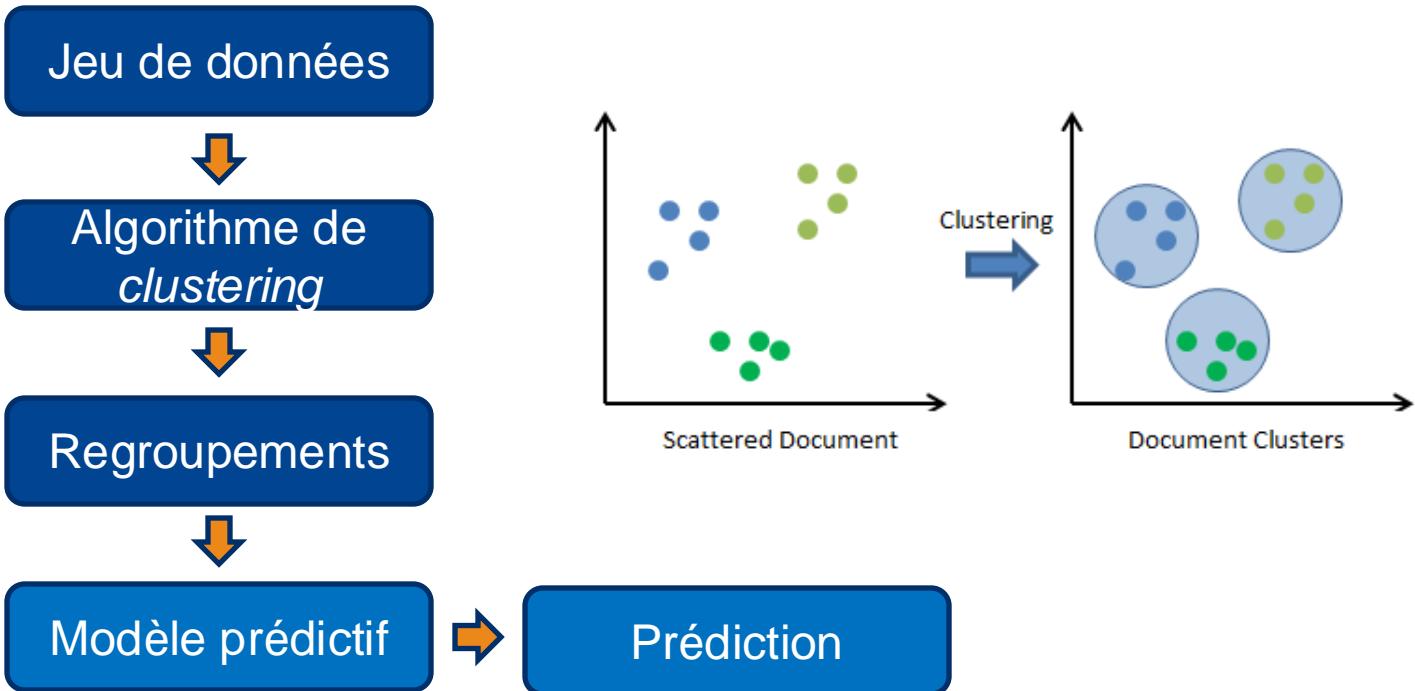
2 – Familles de machine learning

■ Les autres types de machine learning...



2 – Familles de machine learning

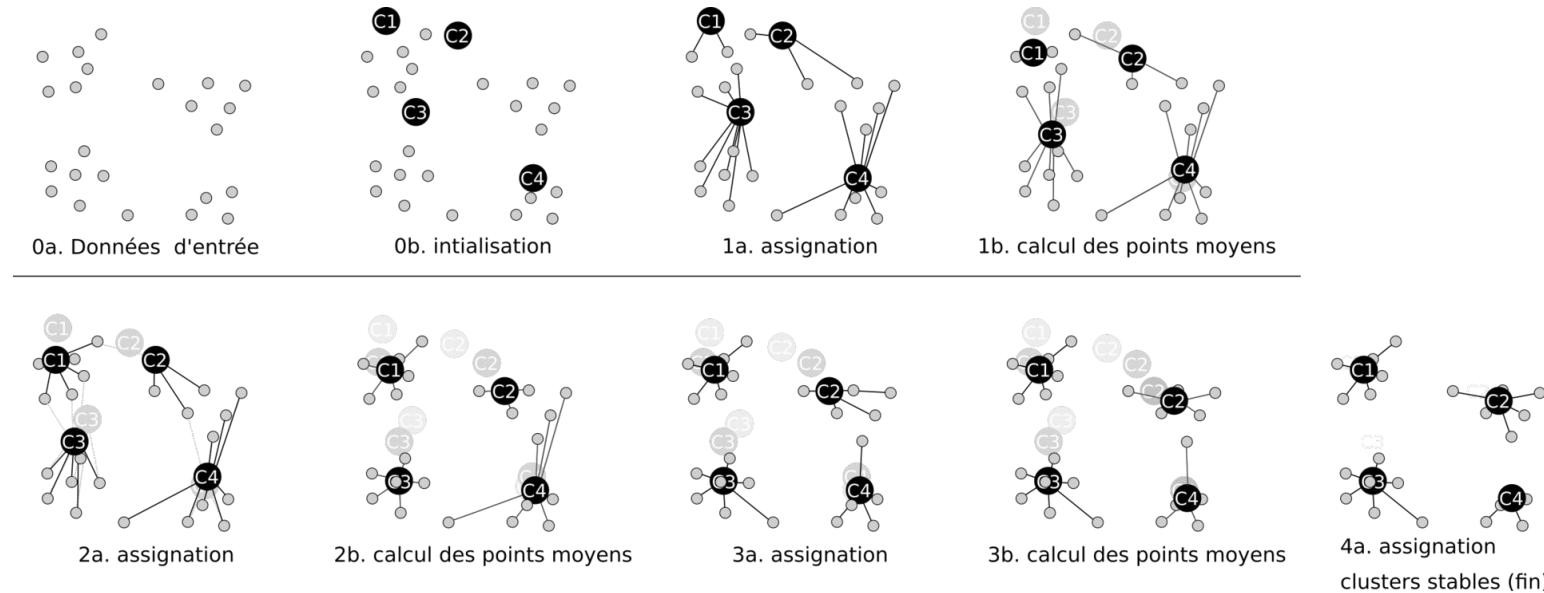
■ Apprentissage non-supervisé



2 – Apprentissage **NON** supervisé

K-Means

- **Objectif:** partitionner N points en k sous-ensembles en minimisant la distance entre les points à l'intérieur de chaque partition



- Les clusters vont dépendre de l'initialisation et de la distance choisie

■ <https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html>

Exemples Safran Aircraft Engines

OBJECTIF : CLUSTERING DES PROFILS DE DESCENTE VIETNAM AIRLINES (SFCO2)

DONNÉES D'ENTRÉE

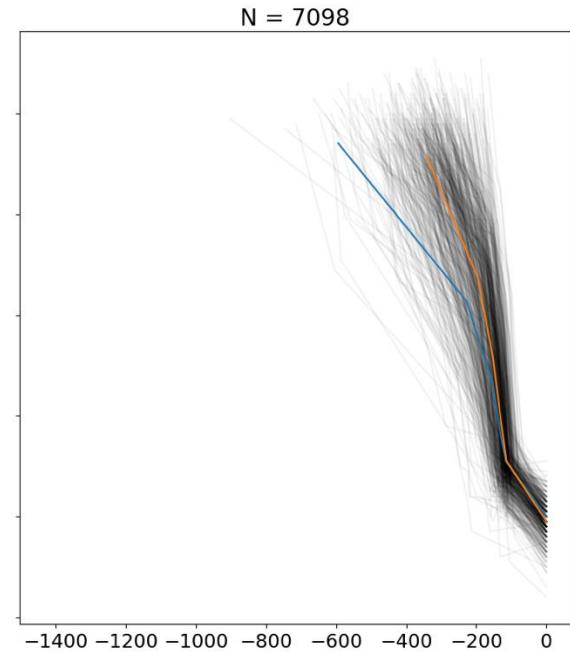
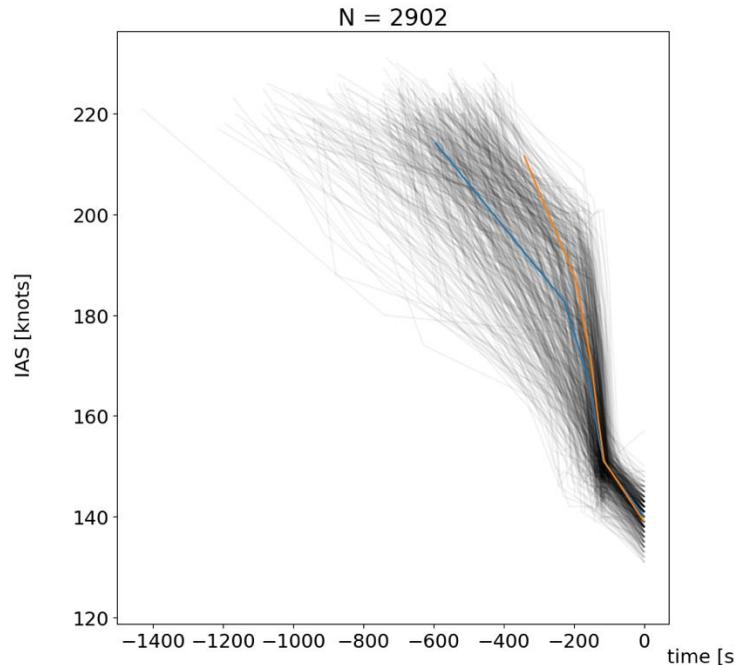
- Trajectoires d'approche

NETTOYAGE

- Filtre des valeurs aberrantes
- Recalcul d'indicateurs métiers

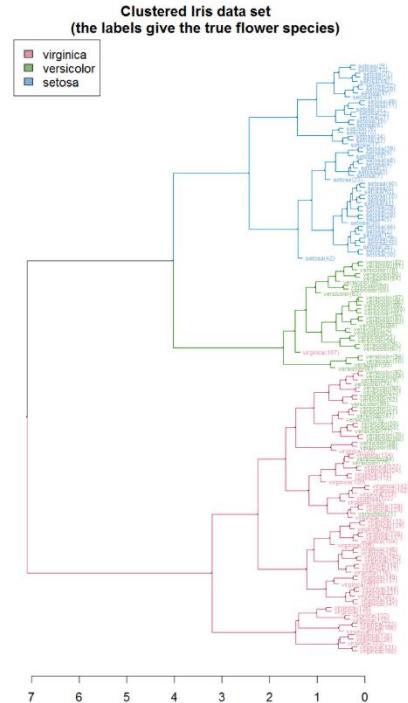
ANALYSE

- Clustering K-Means multivarié sur des séries temporelles
- Caractérisation des profils de descentes les plus courants chez Vietnam Airlines
- Utilisation de ces caractéristiques pour les algorithmes d'optimisation



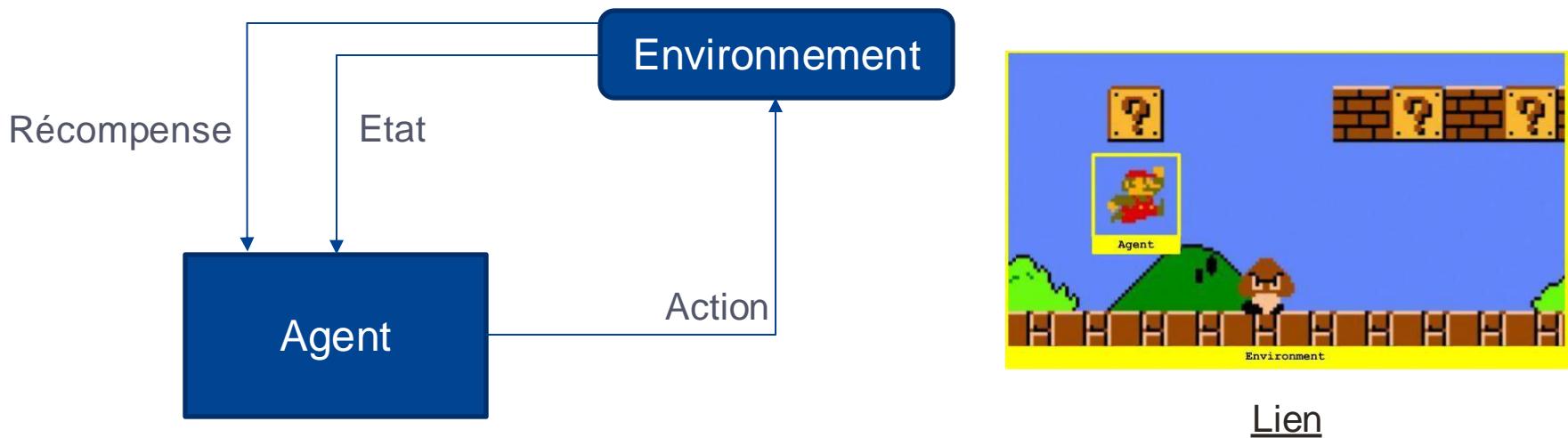
2 – Apprentissage *NON* supervisé *Regroupement hiérarchique*

- Construire une hiérarchie de clusters
- Chaque élément est regroupé de manière itérative 2 à 2 (agglomération)
- Tous les éléments sont dans le même cluster, puis on sépare récursivement les éléments (division)
- Plusieurs métriques peuvent être utilisées pour déterminer la distance entre 2 éléments (distance euclidienne, euclidienne au carré, Manhattan ...). Le choix de la métrique déterminera la forme des clusters



2 – Familles de machine learning

■ L'apprentissage par renforcement (*Reinforcement Learning*)



2 – Apprentissage supervisé

Les réseaux de neurones

■ Définition:

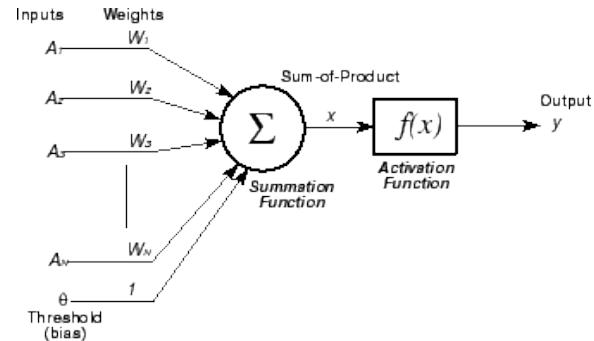
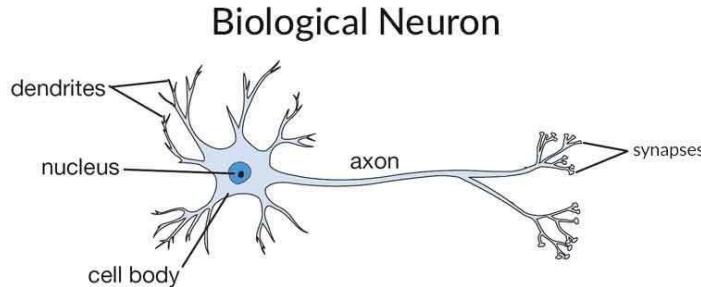
« Un réseau de neurones artificiels, ou réseau neuronal artificiel, est un système dont la conception est à l'origine schématiquement inspirée du fonctionnement des neurones biologiques, et qui par la suite s'est rapproché des méthodes statistiques »

- > 1943: le neurone formel, par Warren McCulloch et Walter Pitts
- > 1958: le perceptron de Rosenblatt
- > 1990: applications industrielles

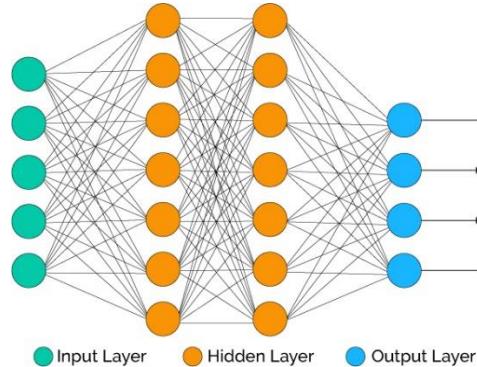
■ 1ere application industrielle: début des années 1960, suppression des échos dans une ligne téléphonique (par une variante du Perceptron: l'ADALINE, ADAptive LINear Element)

2 – Apprentissage supervisé Les réseaux de neurones

■ Principe du réseau de neurones:

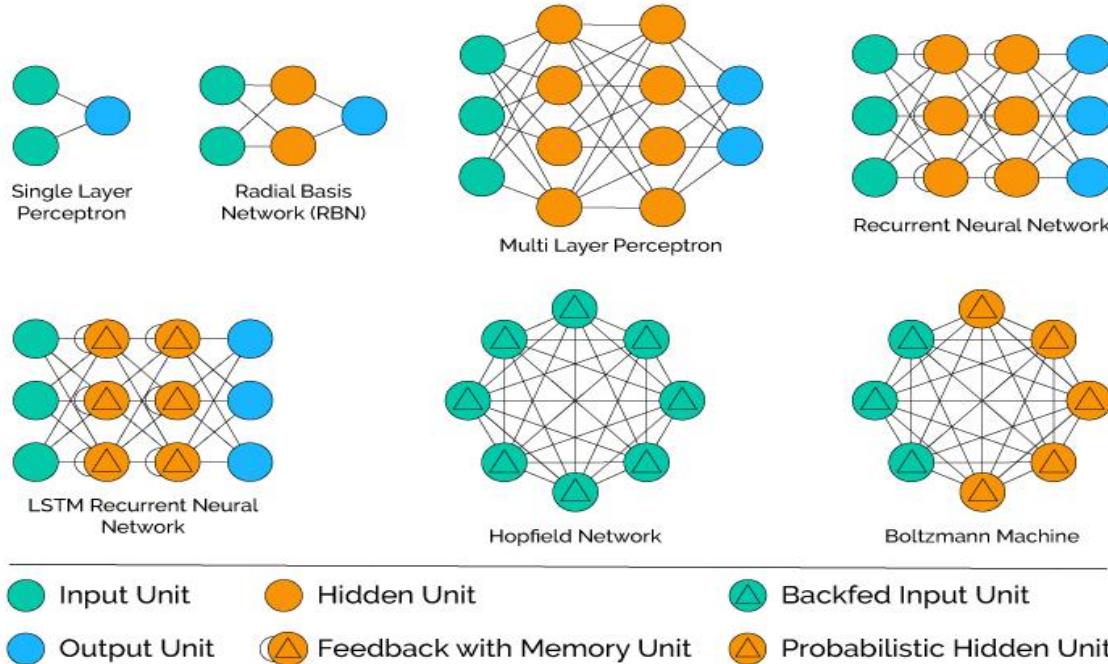


→ Assemblage des neurones



2 – Apprentissage supervisé Les réseaux de neurones

■ Différentes architectures pour différents types de NN:



2 – Apprentissage supervisé

Les réseaux de neurones

■ Avantages

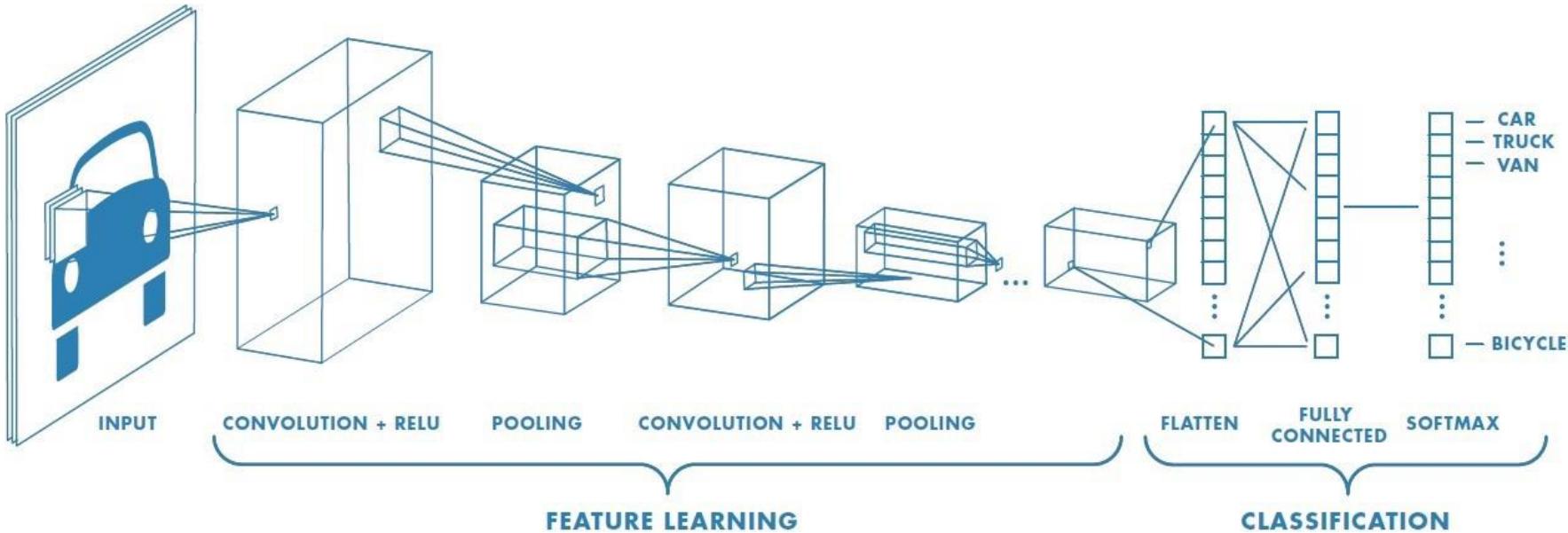
- > S'adapte à beaucoup de problèmes
- > Est très puissant lorsque bien paramétré

■ Inconvénients

- > Aspect boîte noire
- > Paramétrage du réseau
- > Puissance informatique

■ https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html

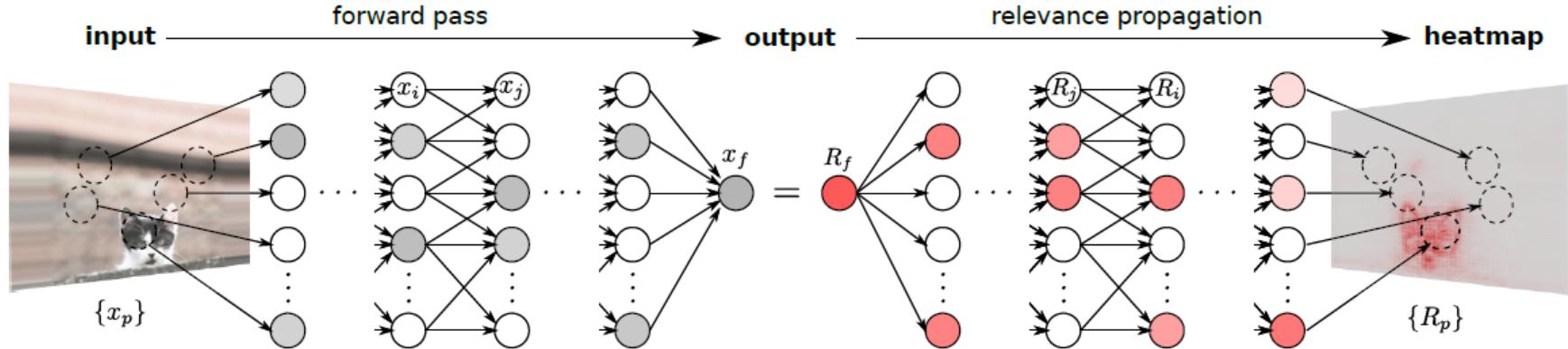
2 – Apprentissage supervisé Les réseaux de neurones convolutifs



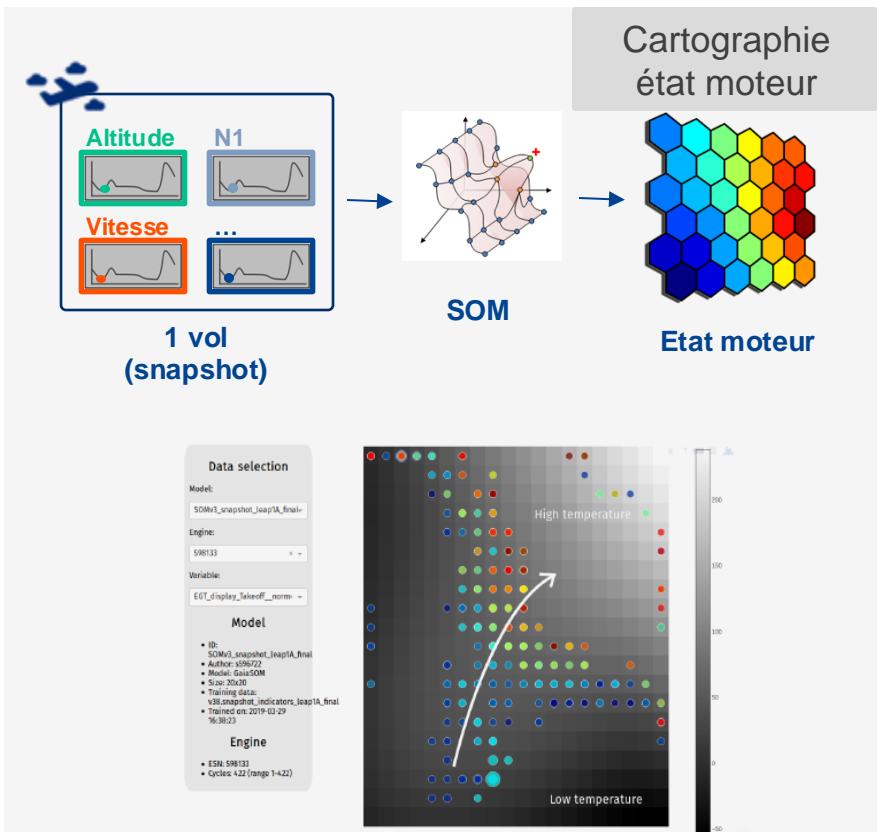
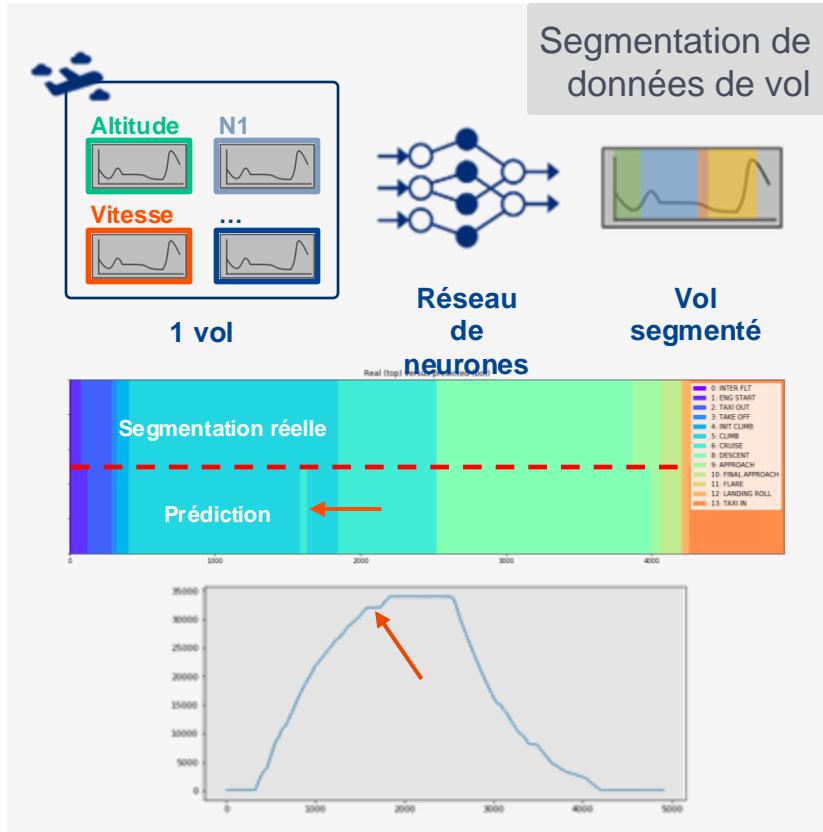
2 – Apprentissage supervisé Les réseaux de neurones

Quelle(s) features sont été le plus utilisés ?

Back-propagation du résultat:



Application Safran Aircraft Engines de *deep learning*



2 – Et le reste...



2 – Comment choisir le bon modèle...

■ Interprétabilité

- > Fait de pouvoir « lire » le modèle, « voir » les liens entre les entrées et les sorties
 - ◆ Lisible : modèle linéaire multivarié, arbre de décision
 - ◆ Peu lisible : réseau de neurones, random forest
- > A défaut, possibilité de quantifier l'influence de chaque variable d'entrée
- > Ne pas sous-estimer le besoin d'interprétabilité par l'utilisateur final / demandeur / expert métier

■ Performance

- > Besoin de précision du modèle
- > Peut s'opposer au besoin d'interprétabilité

■ Industrialisation / déploiement

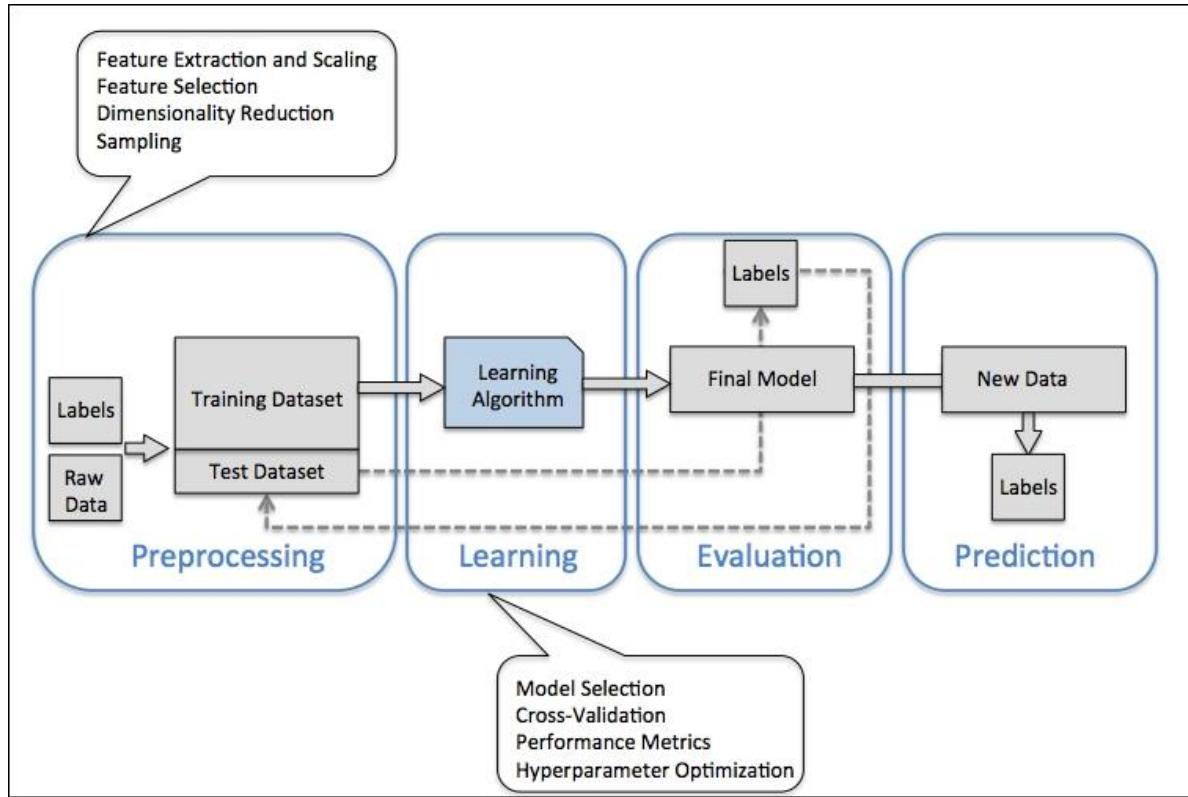
- > Prendre en compte le plus tôt possible les contraintes d'industrialisation

3

VALIDATION D'UN MODÈLE



3 – Validation d'un modèle



Validation d'un modèle : objectifs

- Créer un modèle est aujourd'hui très simple (packages python, R...)
- Avant de l'exploiter et de tirer des conclusions, on doit évaluer sa qualité.
- Le modèle a été créé pour quelle utilisation ?
 - ◆ Analyse d'influence des variables
 - ◆ Compréhension de la physique d'un problème
 - ◆ Utilisation d'algorithmes d'optimisation itératifs basés sur le modèle
 - ◆ Prédiction avec grande précision de nouveaux individus (traitement de dérogation)
 - ◆ ...

■ Des niveaux de validation différents en fonction de l'utilisation du modèle

■ Deux grands niveaux de validation possibles

- > Calculs de critères mathématiques liés au modèle (strict minimum)
- > Partitionnement du jeu de données initial pour l'apprentissage du modèle, puis pour sa validation

Validation d'un modèle :

■ Sans partitionnement du jeu de données, on utilise toutes les informations disponibles pour calculer un critère mathématique :

- ◆ R², R² ajusté
- ◆ Résidus standardisés

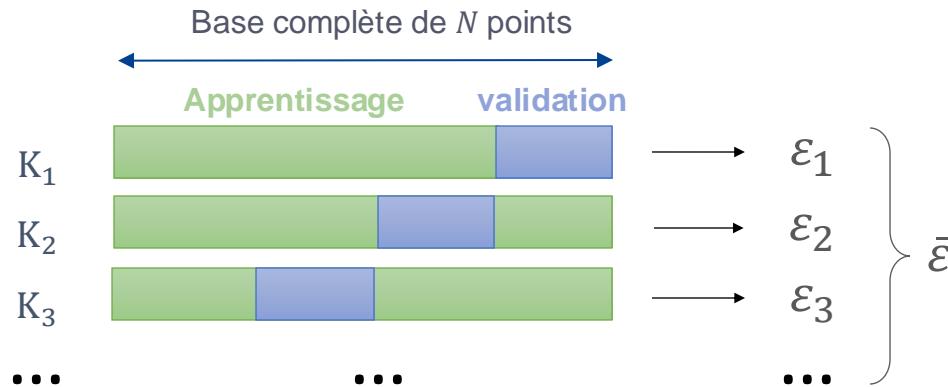
■ Avec partitionnement du jeu de données. Une partie est utilisée pour l'apprentissage du modèle, puis l'autre pour sa validation à partir des critères précédemment définis:

- ◆ K-fold
- ◆ Leave One Out Cross Validation (LOOCV) – Très utilisé dans le cas des modèles interpolant (Krigage, Radial Basis Functions...)
- ◆ Learning curves (déttection du sur-apprentissage)

Avec partitionnement du jeu de données : K-Fold

■ Partitionnement de la base de données initiale :

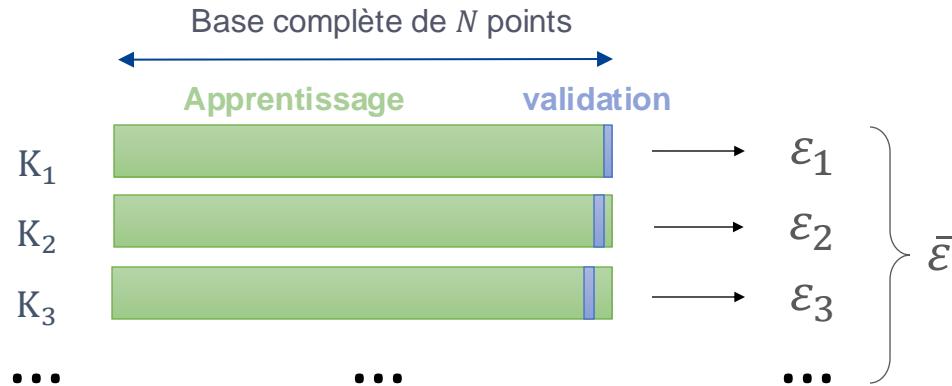
- ◆ Base d'apprentissage (~80%)
- ◆ Base de validation (~20%)
- ◆ Calcul d'un critère d'erreur \mathcal{E}_i (RMSE par exemple) pour K partitionnements différents sur la base de validation
(Dans la pratique, on réalise entre 5 et 10 partitionnements)
- ◆ Agrégation des erreurs \mathcal{E}_i en un critère global $\bar{\mathcal{E}}$



Avec partitionnement du jeu de données : Leave One Out Cross Validation

■ Identique au K-fold sur le principe, mais utilisation d'un seul point comme base de validation :

- ◆ Base d'apprentissage (N-1 points)
- ◆ Base de validation (1 point)
- ◆ Calcul d'un critère d'erreur \mathcal{E}_i pour N partitionnements différents sur la base de validation
- ◆ Agrégation des erreurs \mathcal{E}_i en un critère global $\bar{\mathcal{E}}$



Modèle de classification : matrice de confusion

- ◆ Dans le cas d'un modèle de classification, on utilise la base de test pour vérifier si les individus appartiennent bien à la bonne classe. On peut alors afficher une matrice de confusion.
- ◆ Exemple : Classement de 100 points dans deux catégories « A » ou « B ». Evaluation des points de la base de test pour vérifier les prédictions du modèle par rapport à la réalité

| | | Prédictions | |
|---------|---|-------------------|-------------------|
| | | A | B |
| Réalité | A | Vrai positif (VP) | Faux négatif (FN) |
| | B | Faux positif (FP) | Vrai négatif (VN) |

- ◆ Exemple: A: panne détectée, B panne non détectée

Modèle de classification : matrice de confusion

- ♦ Dans le cas d'un modèle de classification, on utilise la base de test pour vérifier si les individus appartiennent bien à la bonne classe. On peut alors afficher une matrice de confusion.
- ♦ Exemple : Classement de 100 points dans deux catégories « A » ou « B ». Evaluation des points de la base de test pour vérifier les prédictions du modèle par rapport à la réalité

| | | Prédictions | |
|---------|---|-------------|---------|
| | | A | B |
| Réalité | A | 45 (VP) | 5 (FN) |
| | B | 3 (FP) | 47 (VN) |

Critères :

$$\text{♦ Sensibilité} = \frac{VP}{VP+FN} = \frac{45}{45+5} = 90\% \text{ (Parmi les points réellement A, probabilité de classement correct)} = POD = 1 - \beta$$

Modèle de classification : matrice de confusion

- ◆ Dans le cas d'un modèle de classification, on utilise la base de test pour vérifier si les individus appartiennent bien à la bonne classe. On peut alors afficher une matrice de confusion.
- ◆ Exemple : Classement de 100 points dans deux catégories « A » ou « B ». Evaluation des points de la base de test pour vérifier les prédictions du modèle par rapport à la réalité

| | | Prédictions | |
|---------|---|-------------|---------|
| | | A | B |
| Réalité | A | 45 (VP) | 5 (FN) |
| | B | 3 (FP) | 47 (VN) |

Critères :

- ◆ Sensibilité = $\frac{VP}{VP+FN} = \frac{45}{45+5} = 90\%$ (Parmi les points réellement A, probabilité de classement correct) = POD = $1 - \beta$
- ◆ Spécificité = $\frac{VN}{VN+FP} = \frac{47}{47+3} = 94\%$ (Parmi les points réellement B, probabilité de classement correct) = $1 - \alpha$

Modèle de classification : matrice de confusion

- ◆ Dans le cas d'un modèle de classification, on utilise la base de test pour vérifier si les individus appartiennent bien à la bonne classe. On peut alors afficher une matrice de confusion.
- ◆ Exemple : Classement de 100 points dans deux catégories « A » ou « B ». Evaluation des points de la base de test pour vérifier les prédictions du modèle par rapport à la réalité

| | | Prédictions | |
|---------|---|-------------|---------|
| | | A | B |
| Réalité | A | 45 (VP) | 5 (FN) |
| | B | 3 (FP) | 47 (VN) |

Critères :

- ◆ Sensibilité = $\frac{VP}{VP+FN}$
- ◆ Précision = $\frac{VP}{VP+FP} = \frac{45}{45+3} = 93,7\%$ (Probabilité qu'une prédition A soit correcte) = $1-PFA$
- ◆ Spécificité = $\frac{VN}{VN+FP}$

Modèle de classification : matrice de confusion

- ♦ Dans le cas d'un modèle de classification, on utilise la base de test pour vérifier si les individus appartiennent bien à la bonne classe. On peut alors afficher une matrice de confusion.
- ♦ Exemple : Classement de 100 points dans deux catégories « A » ou « B ». Evaluation des points de la base de test pour vérifier les prédictions du modèle par rapport à la réalité

| | | Prédictions | |
|---------|---|-------------|---------|
| | | A | B |
| Réalité | A | 45 (VP) | 5 (FN) |
| | B | 3 (FP) | 47 (VN) |

Critères :

- ♦ Sensibilité = $\frac{VP}{VP+FN}$
- ♦ Précision = $\frac{VP}{VP+FP} = \frac{45}{45+3} = 93,7\%$ (Probabilité qu'une prédition A soit correcte) = $1 - PFA$
- ♦ Spécificité = $\frac{VN}{VN+FP}$
- ♦ Rappel = $\frac{VP}{VP+FN} = \frac{47}{47+5} = 90\%$ (C'est aussi la sensibilité !) = $POD = 1 - \beta$

Modèle de classification : matrice de confusion

- ♦ Dans le cas d'un modèle de classification, on utilise la base de test pour vérifier si les individus appartiennent bien à la bonne classe. On peut alors afficher une matrice de confusion.
- ♦ Exemple : Classement de 100 points dans deux catégories « A » ou « B ». Evaluation des points de la base de test pour vérifier les prédictions du modèle par rapport à la réalité

| | | Prédictions | |
|---------|---|-------------|---------|
| | | A | B |
| Réalité | A | 45 (VP) | 5 (FN) |
| | B | 3 (FP) | 47 (VN) |

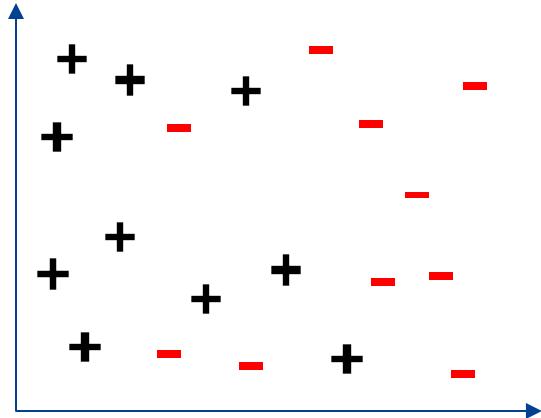
Critères :

$$\begin{aligned}\diamond \text{ Sensibilité} &= \frac{VP}{VP+FN} \\ \diamond \text{ Spécificité} &= \frac{VN}{VN+FP}\end{aligned}$$

$$\begin{aligned}\diamond \text{ Précision} &= \frac{VP}{VP+FP} \\ \diamond \text{ Rappel} &= \frac{VP}{VP+FN}\end{aligned}\left.\right\} \text{ Critère combiné : } F_1 = 2 * \frac{\text{Précision} * \text{Rappel}}{\text{Précision} + \text{Rappel}}$$

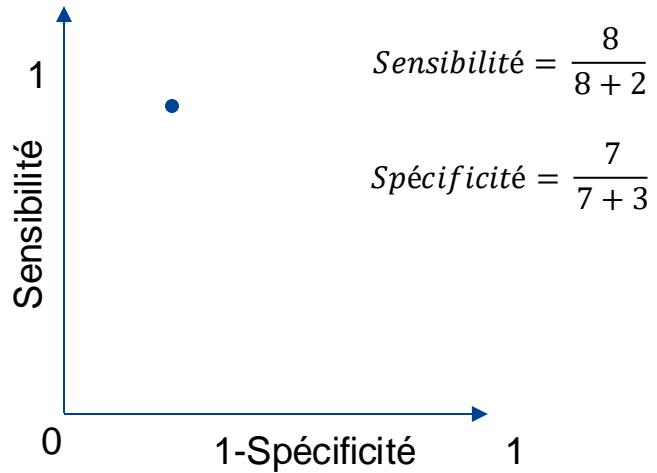
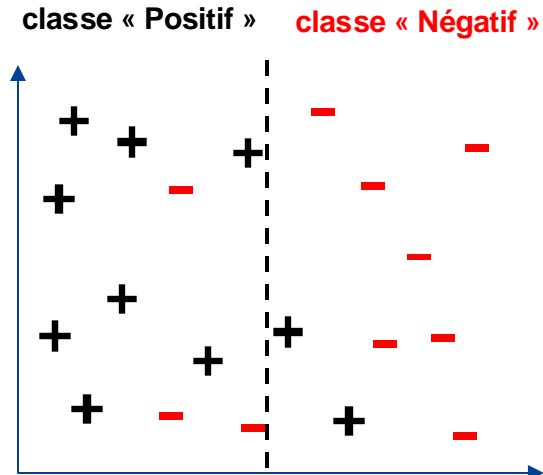
Modèle de classification : Courbe ROC (*receiver operating characteristic*)

- ◆ Visualisation des performances d'un modèle de classification pour une classe spécifique sur un graphique « Sensibilité en fonction de la spécificité »
- ◆ Exemple : Classement des points en 2 classes « Positif » et « Négatif »



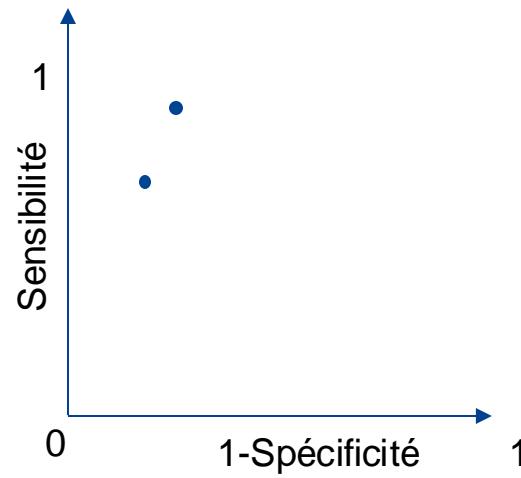
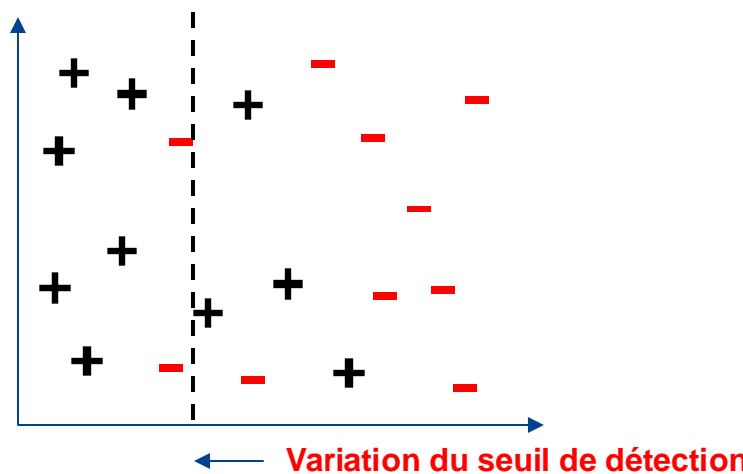
Modèle de classification : Courbe ROC (*receiver operating characteristic*)

- ◆ Visualisation des performances d'un modèle de classification pour une classe spécifique sur un graphique « Sensibilité en fonction de la spécificité »
- ◆ Exemple : Classement des points en 2 classes « Positif » et « Négatif »



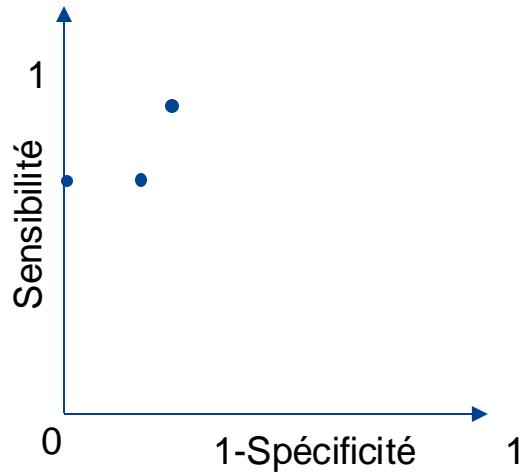
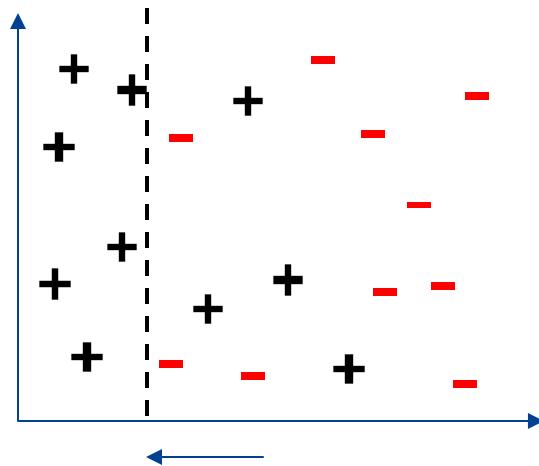
Modèle de classification : Courbe ROC (*receiver operating characteristic*)

- ◆ Visualisation des performances d'un modèle de classification pour une classe spécifique sur un graphique « Sensibilité en fonction de la spécificité »
- ◆ Exemple : Classement des points en 2 classes « Positif » et « Négatif »



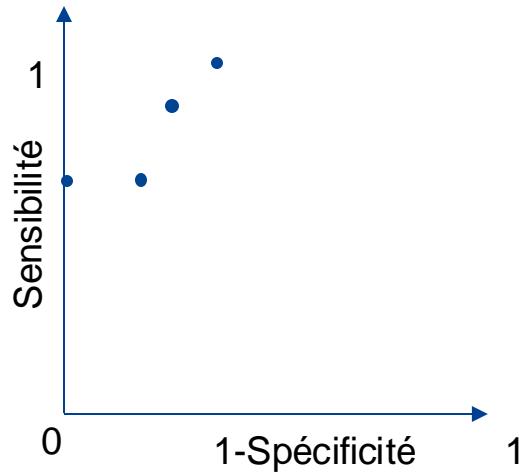
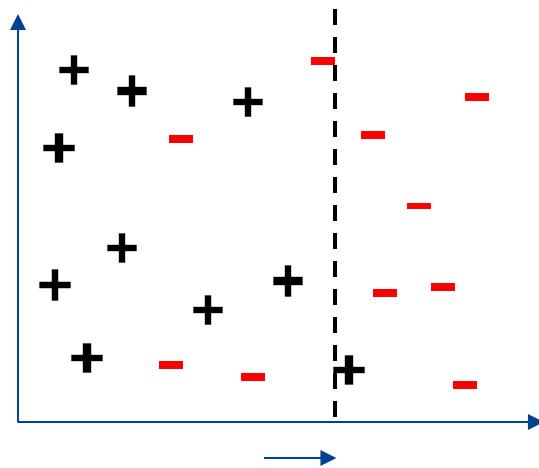
Modèle de classification : Courbe ROC (*receiver operating characteristic*)

- ◆ Visualisation des performances d'un modèle de classification pour une classe spécifique sur un graphique « Sensibilité en fonction de la spécificité »
- ◆ Exemple : Classement des points en 2 classes « Positif » et « Négatif »



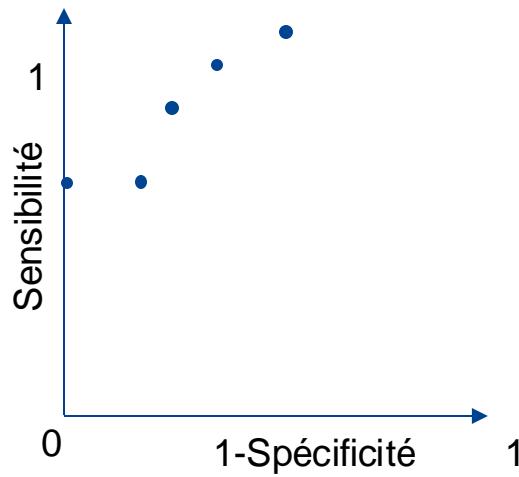
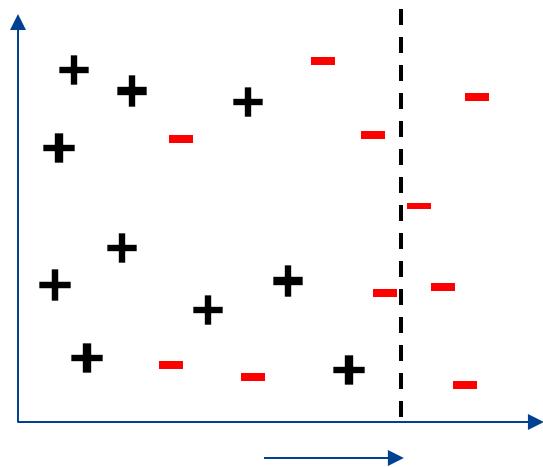
Modèle de classification : Courbe ROC (*receiver operating characteristic*)

- ◆ Visualisation des performances d'un modèle de classification pour une classe spécifique sur un graphique « Sensibilité en fonction de la spécificité »
- ◆ Exemple : Classement des points en 2 classes « Positif » et « Négatif »



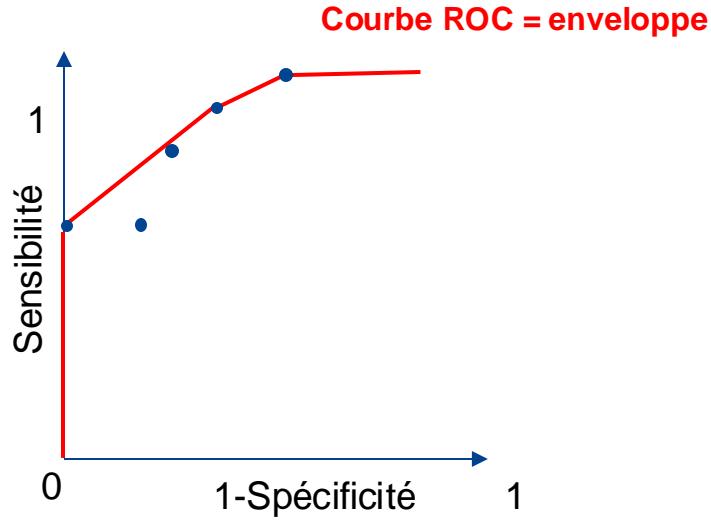
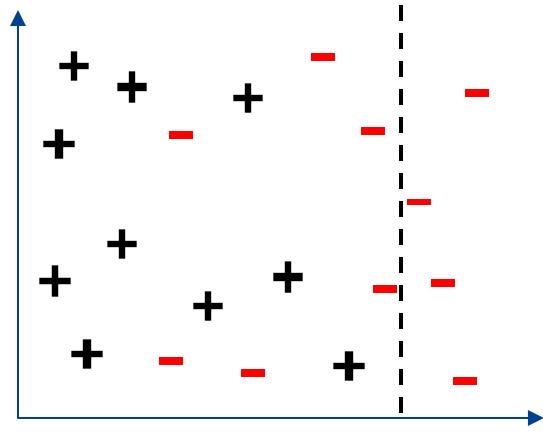
Modèle de classification : Courbe ROC (*receiver operating characteristic*)

- ◆ Visualisation des performances d'un modèle de classification pour une classe spécifique sur un graphique « Sensibilité en fonction de la spécificité »
- ◆ Exemple : Classement des points en 2 classes « Positif » et « Négatif »



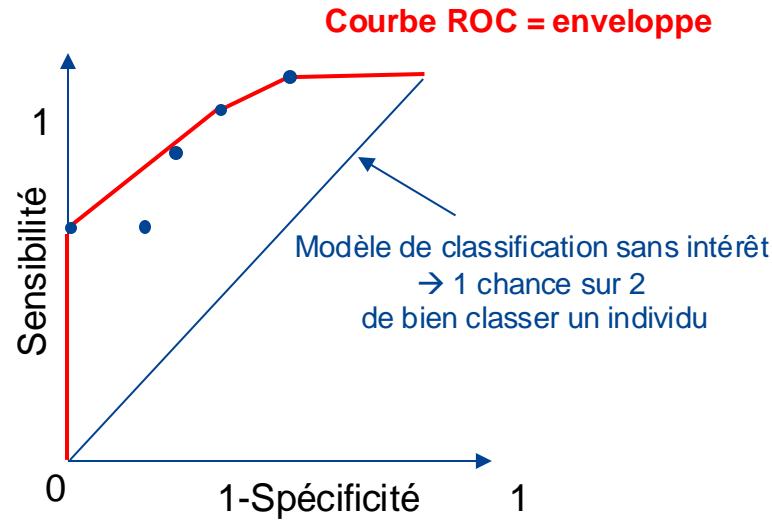
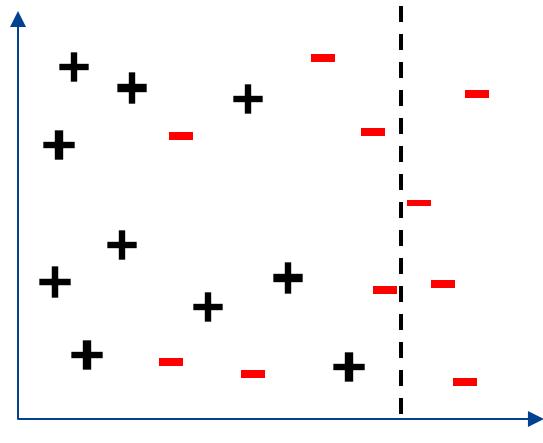
Modèle de classification : Courbe ROC (*receiver operating characteristic*)

- ◆ Visualisation des performances d'un modèle de classification pour une classe spécifique sur un graphique « Sensibilité en fonction de la spécificité »
- ◆ Exemple : Classement des points en 2 classes « Positif » et « Négatif »



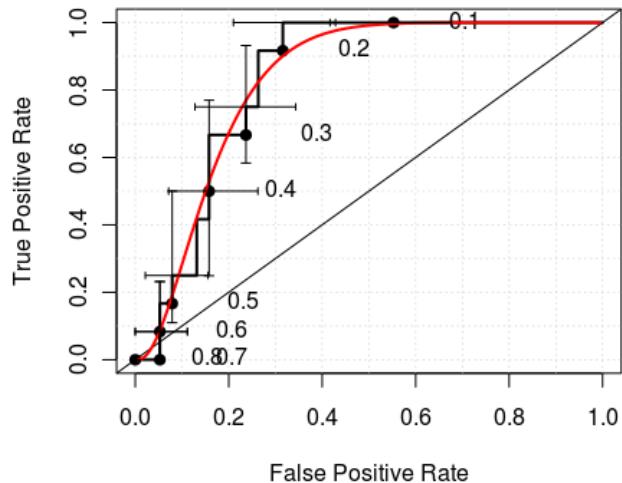
Modèle de classification : Courbe ROC (*receiver operating characteristic*)

- ◆ Visualisation des performances d'un modèle de classification pour une classe spécifique sur un graphique « Sensibilité en fonction de la spécificité »
- ◆ Exemple : Classement des points en 2 classes « Positif » et « Négatif »



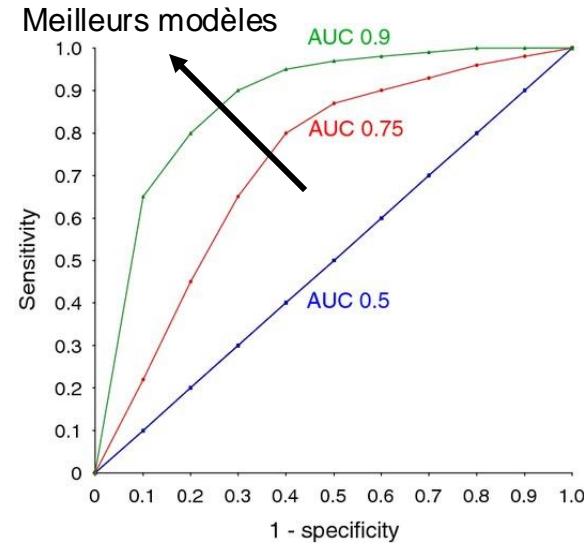
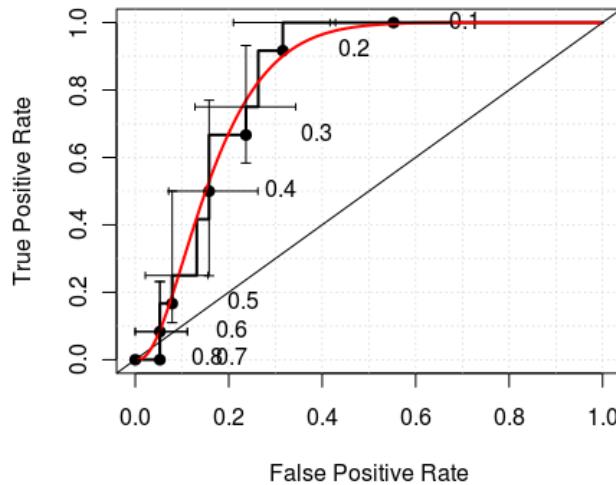
Modèle de classification : Courbe ROC (*receiver operating characteristic*)

- ◆ Visualisation des performances d'un modèle de classification pour une classe spécifique sur un graphique « Sensibilité en fonction de la spécificité »
- ◆ Avec des techniques de « bootstrap », on peut estimer un intervalle de confiance pour ces courbes ROC



Modèle de classification : Courbe ROC (*receiver operating characteristic*)

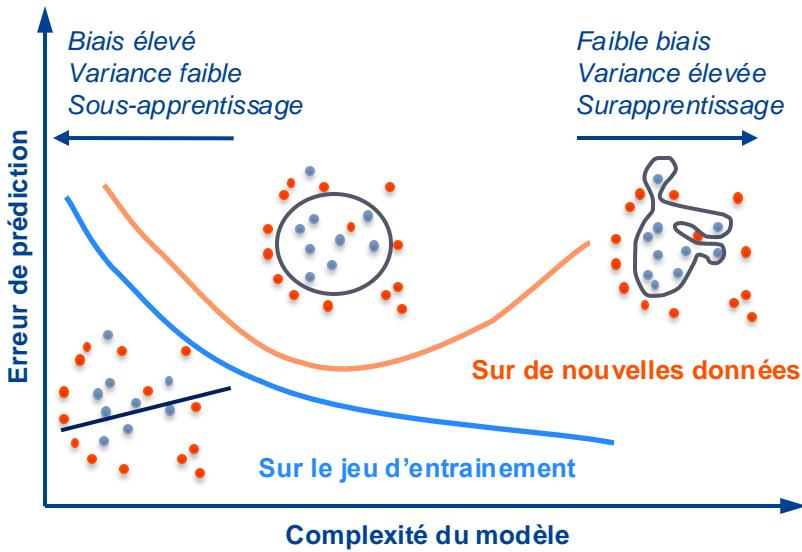
- ◆ Visualisation des performances d'un modèle de classification pour une classe spécifique sur un graphique « Sensibilité en fonction de la spécificité »
- ◆ Avec des techniques de bootstrap, on peut estimer un intervalle de confiance pour ces courbes ROC
- ◆ Ces courbes permettent notamment de comparer des modèles entre eux (AUC = Area Under Curve)



Analyse des résultats :

- Il revient toujours à l'ingénieur d'interpréter le critère final $\bar{\varepsilon}$ obtenu.
- Ce résultat est à mettre en perspective avec l'utilisation souhaitée du modèle :
 - > Dans le cas de la **prédiction** de nouveaux points pour le traitement de dérogations :
 - ◆ Une erreur de prédiction d'1% sur une contrainte mécanique peut être très satisfaisante, alors qu'une erreur d'1% sur la prédiction d'un rendement de compresseur est inconcevable.
 - ◆ Il convient de valider, avec une marge d'erreur choisie, l'ensemble du domaine de définition pour couvrir tous les nouveaux cas de dérogations possibles.
 - > Dans le cas d'une **analyse d'influence** (ANOVA par exemple), la précision de prédiction du modèle sur l'ensemble du domaine n'est pas nécessaire pour obtenir des premières tendances cohérentes.

Compromis biais-variance



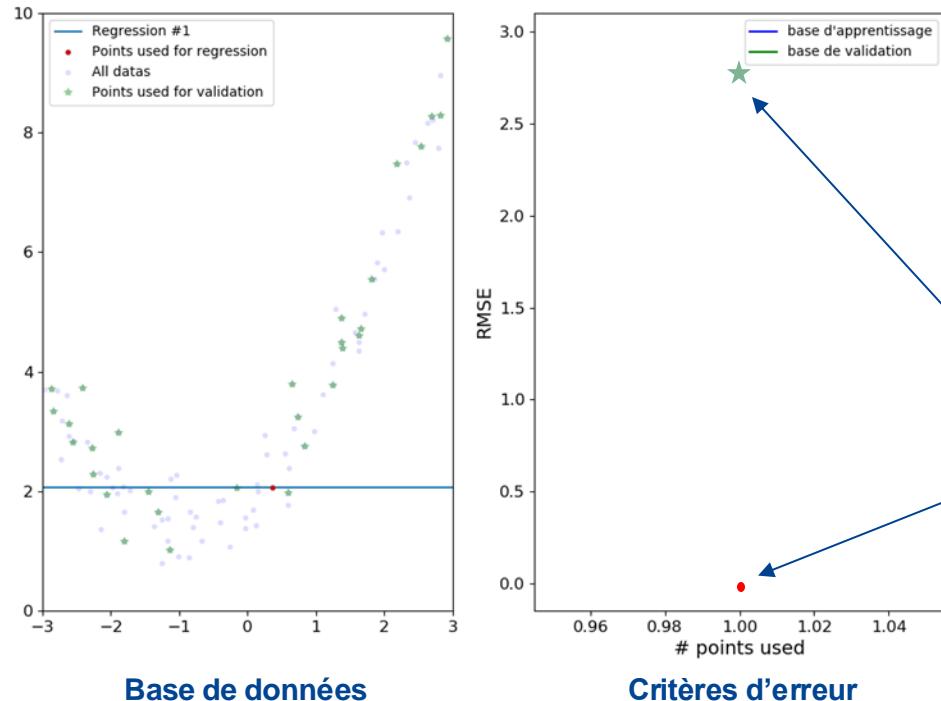
Un modèle simple (variance faible) risque le sous-apprentissage (biais élevé y compris sur les données d'entraînement)

Un modèle complexe (variance élevée) risque le sur-apprentissage (biais faible sur les données d'entraînement mais élevé sur de nouvelles données)

L'idéal est de trouver un modèle intermédiaire, vers le creux de la courbe orange, là où le biais de prédiction est le plus faible et la généralisation la meilleure

Détection du sur-apprentissage : Les « Learning curves »

Exemple sur un modèle de régression d'ordre 2 : itération 1



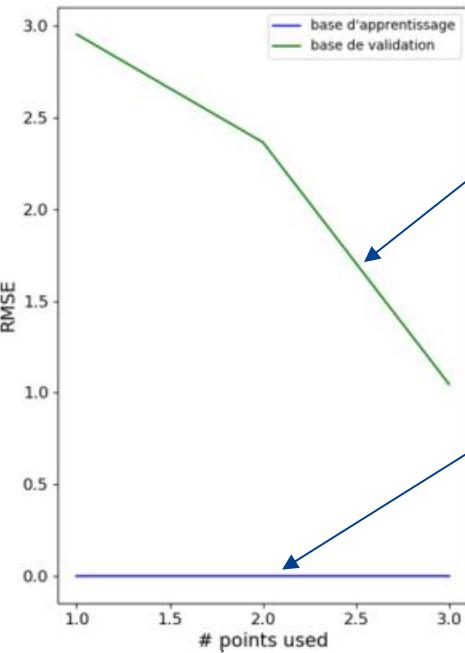
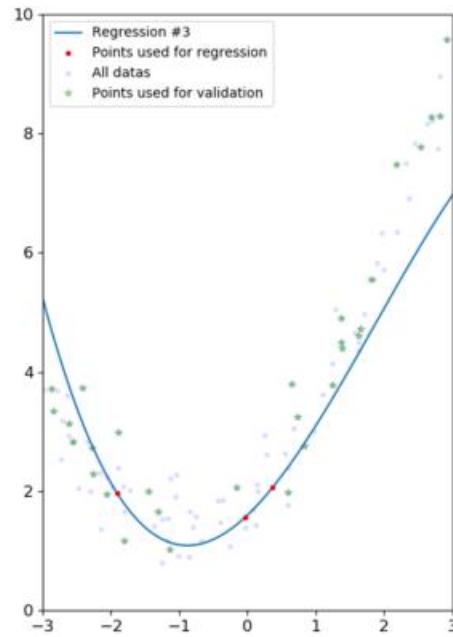
- Partitionnement du jeu de données en base d'apprentissage (points bleus) et de validation (étoiles vertes).
- Construction itérative d'un modèle de régression avec 1 point, puis 2, 3... et calcul du critère d'erreur à chaque itération

Critère d'erreur entre les points de validation et le modèle

Critère d'erreur entre les points d'apprentissage et le modèle

Détection du sur-apprentissage : Les « Learning curves »

Exemple sur un modèle de régression d'ordre 2 : itération 3



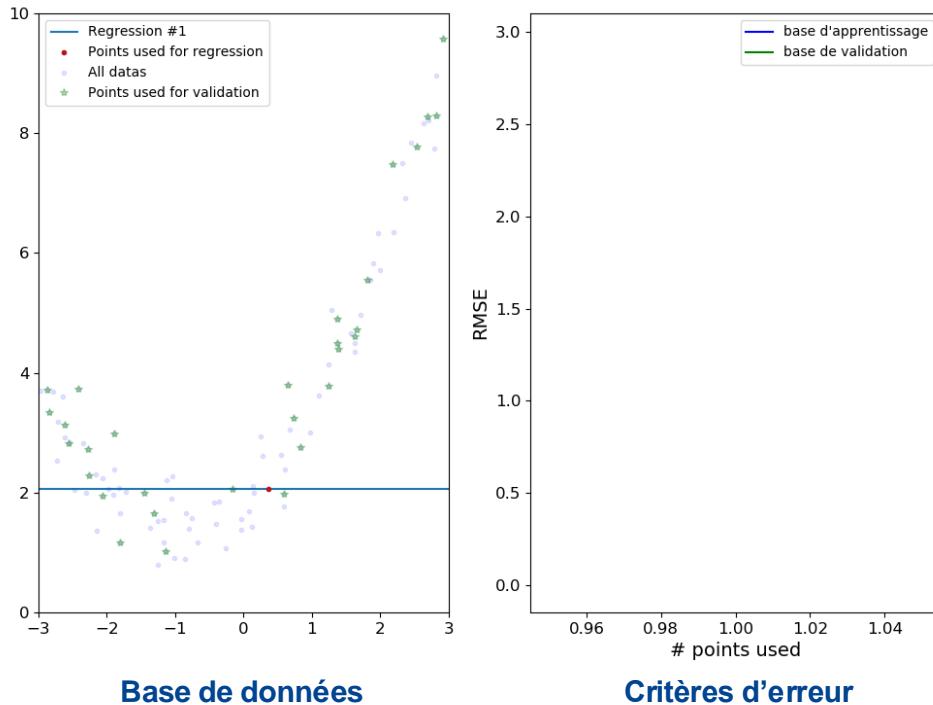
Critère d'erreur entre les points de validation et le modèle

Critère d'erreur entre les points d'apprentissage et le modèle

Base de données

Critères d'erreur

Détection du sur-apprentissage : Les « Learning curves »



Dans l'idéal, les critères d'erreur entre les points de validation et les points d'apprentissage convergent vers la même valeur.

Si ce n'est pas le cas, la base de validation n'est peut-être pas assez représentative de la variance des données. Ou bien le modèle créé est trop complexe pour le nombre de données disponibles.

Validation croisée – autres usages

■ Principe de la validation croisée pour choix des hyperparamètres

- > K-Fold pour chacune des N valeurs d'un hyperparamètre (valeur discrétisées)
- > Choix de la meilleure valeur de l'hyperparamètre à partir de la valeur moyenne du score

■ Principe de la validation croisée pour la détection d'outliers

- > Exemple : RANSAC
 - ◆ Apprentissage du modèle sur un échantillon aléatoire des données
 - ◆ Classification de chaque échantillon en « inlier » / « outlier » selon la distance à l'estimation du modèle
 - ◆ Répétition du process
- ◆ https://scikit-learn.org/stable/auto_examples/linear_model/plot_ransac.html
- ◆ https://scikit-learn.org/stable/modules/linear_model.html (Robustness regression)

3 – Le petit mot de la fin

- Les algorithmes de Machine Learning ne sont que des fonctions mathématiques → il n'y a d'intelligence que la complexité mathématique. Les algorithmes transforment vos données dans différents espaces afin de trouver une relation simple entre elles pour en tirer une conclusion.



4

LISTE DE MOOCS



4 – Liste de MOOCs



Coursera

Probably the most famous MOOC platform

Data Science

- Python for Everybody (Specialization - 5 courses) : University of Michigan - Pour débuter en Python
- Applied Data Science with Python (Specialization - 5 courses) : University of Michigan - Python + Pandas, Matplotlib, Scikit Learn

Machine Learning

- Andrew Ng's Machine Learning (the world's most popular ML course)
- Machine Learning Specialization (6 courses): University of Washington
- Probabilistic Graphical Models

Udacity

- Artificial Intelligence
- Data Analyst nanodegree
- Machine Learning course

DataCamp

- Machine Learning with R
- Big Data Analysis with Revolution R Enterprise

Dataquest

- Data Scientist track

Γ

**POWERED
BY TRUST**

5

BACK-UP

Exemple d'arbre de décision : Iris dataset



■ Données :

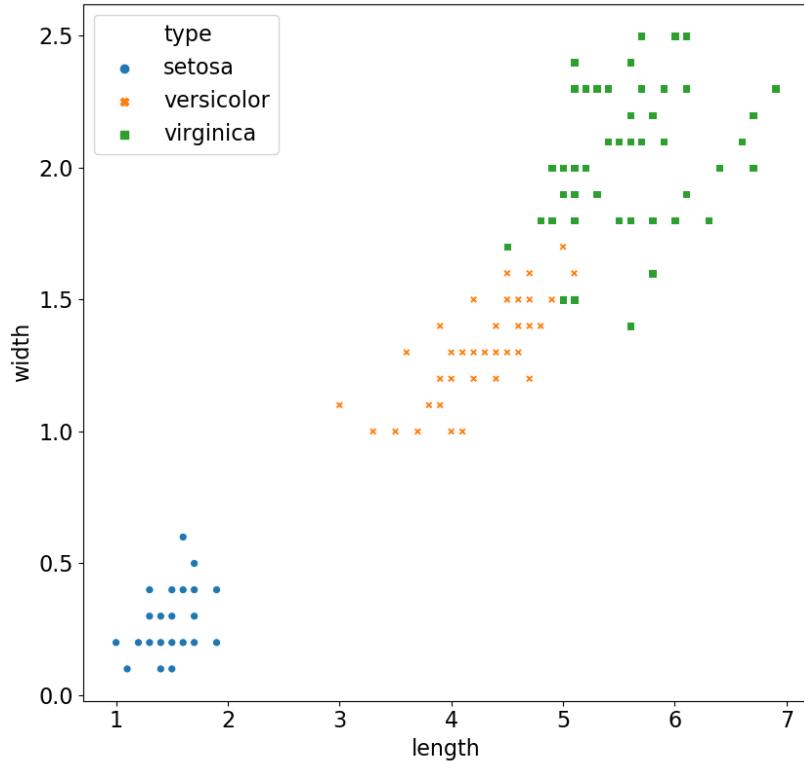
- > 150 fleurs dont on a mesuré la longueur et largeur des pétales, labélisées selon le type d'iris (Virginica, Setosa ou Versicolor)

■ Objectifs :

- > Créer un arbre de classification tel que, pour une nouvelle iris dont on mesure le pétale, on obtienne directement son type



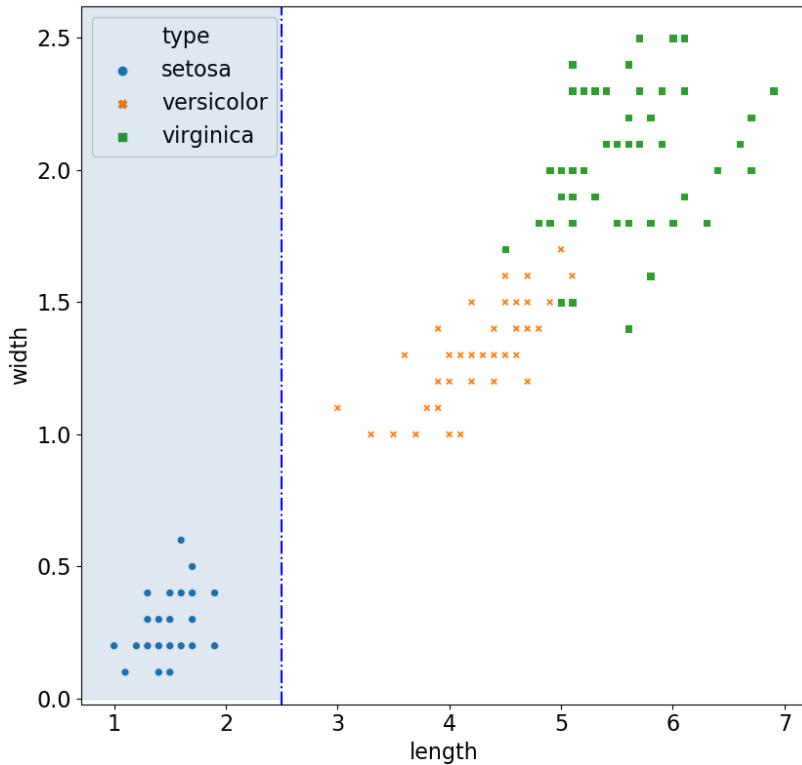
Iris Dataset



*Samples = 150
value = [50, 50, 50]
Petal length <= 2, 45 ?*



Iris Dataset



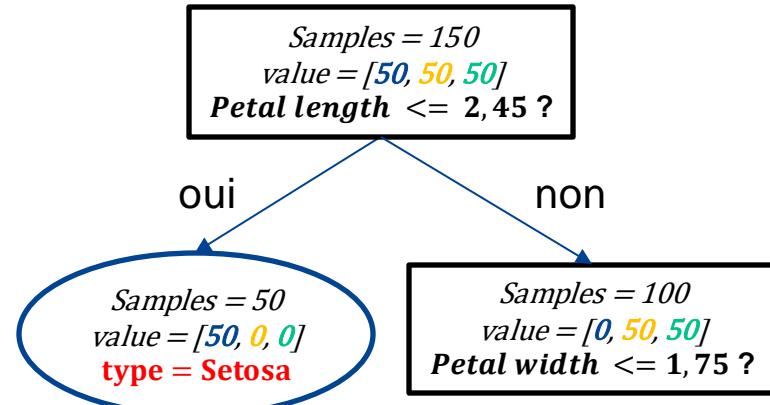
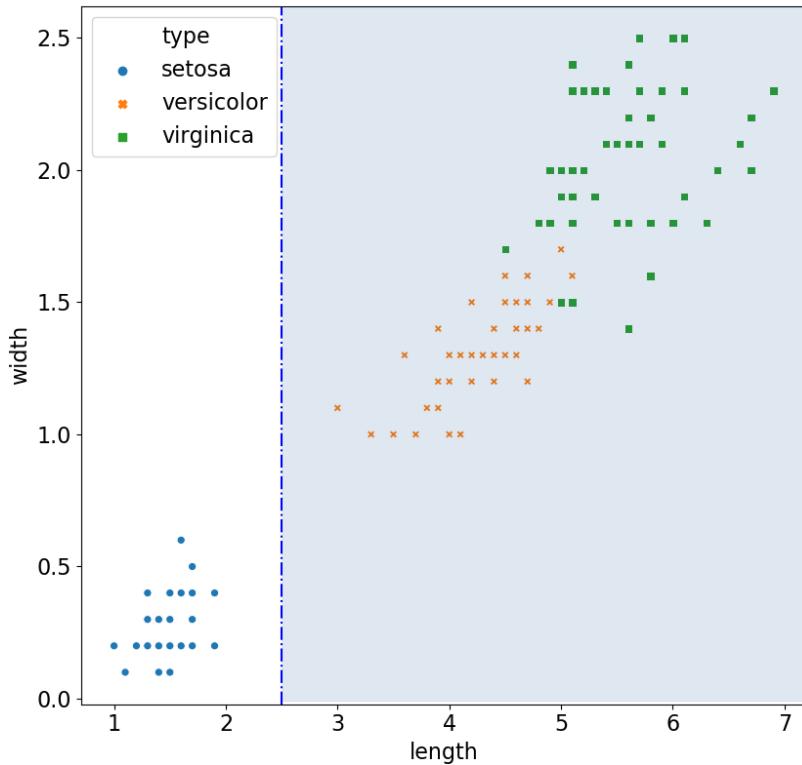
Samples = 150
value = [50, 50, 50]
Petal length <= 2.45 ?

oui

Samples = 50
value = [50, 0, 0]
type = Setosa

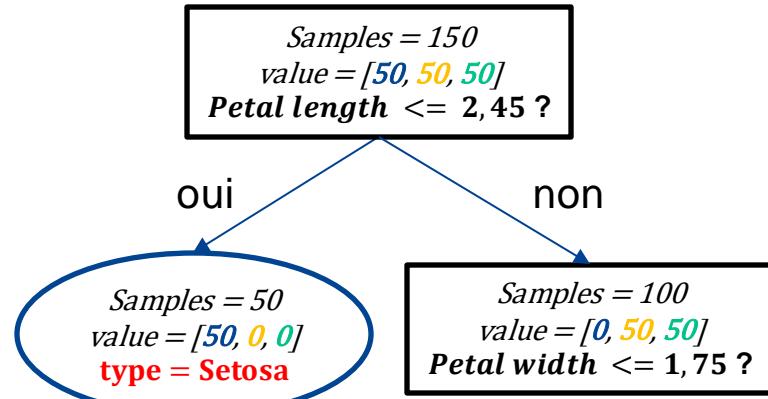
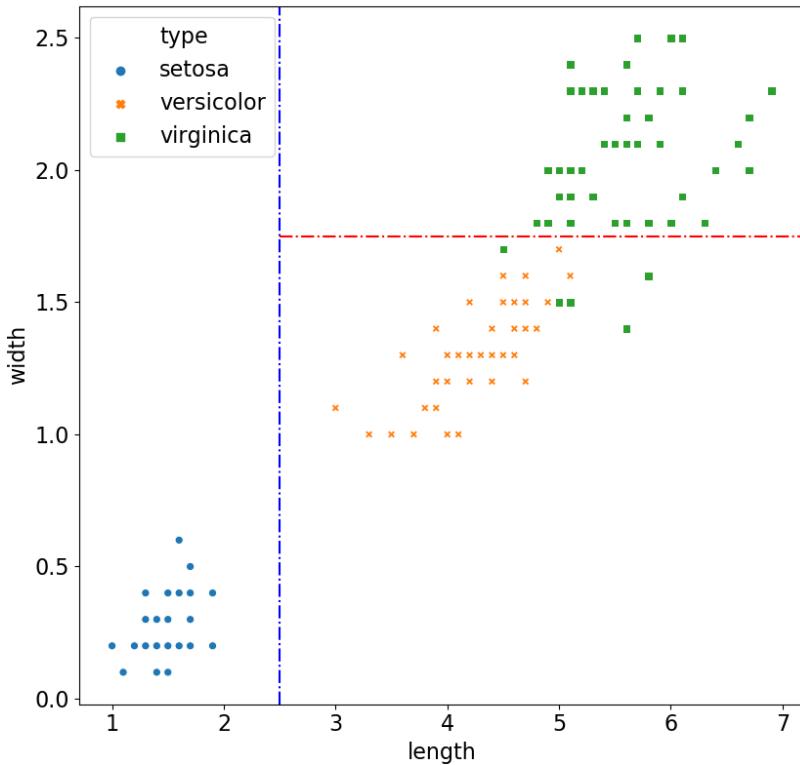


Iris Dataset



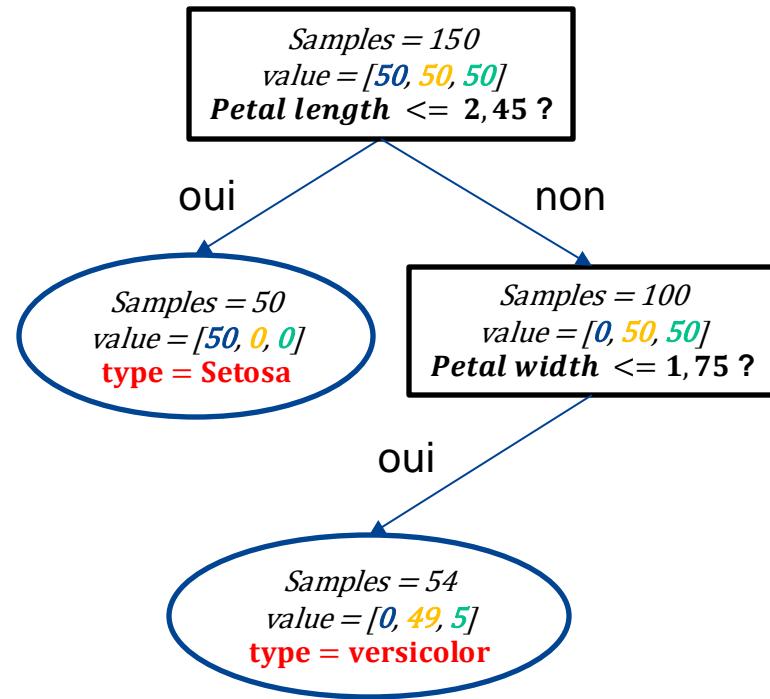
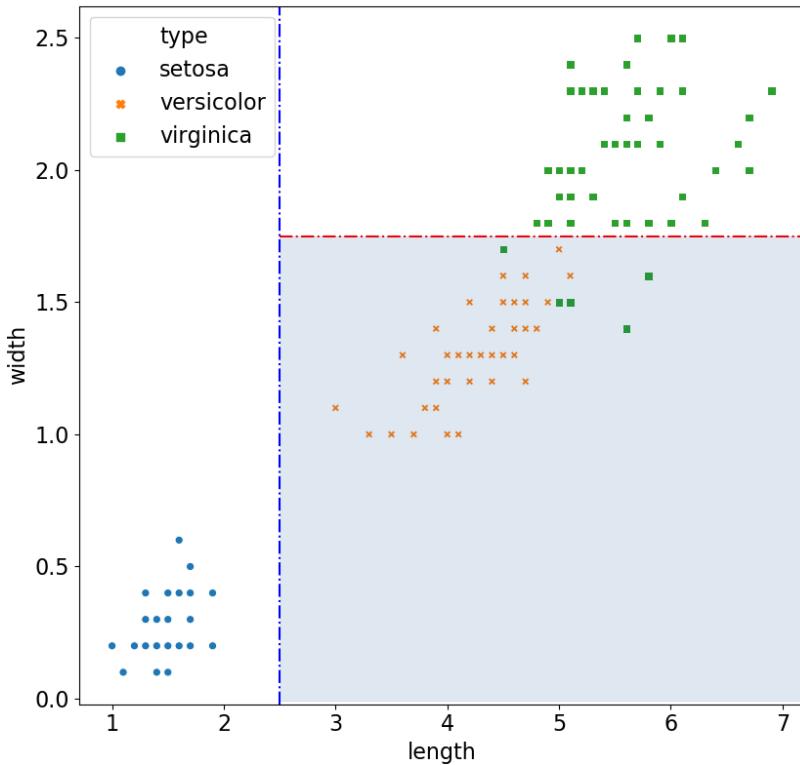


Iris Dataset

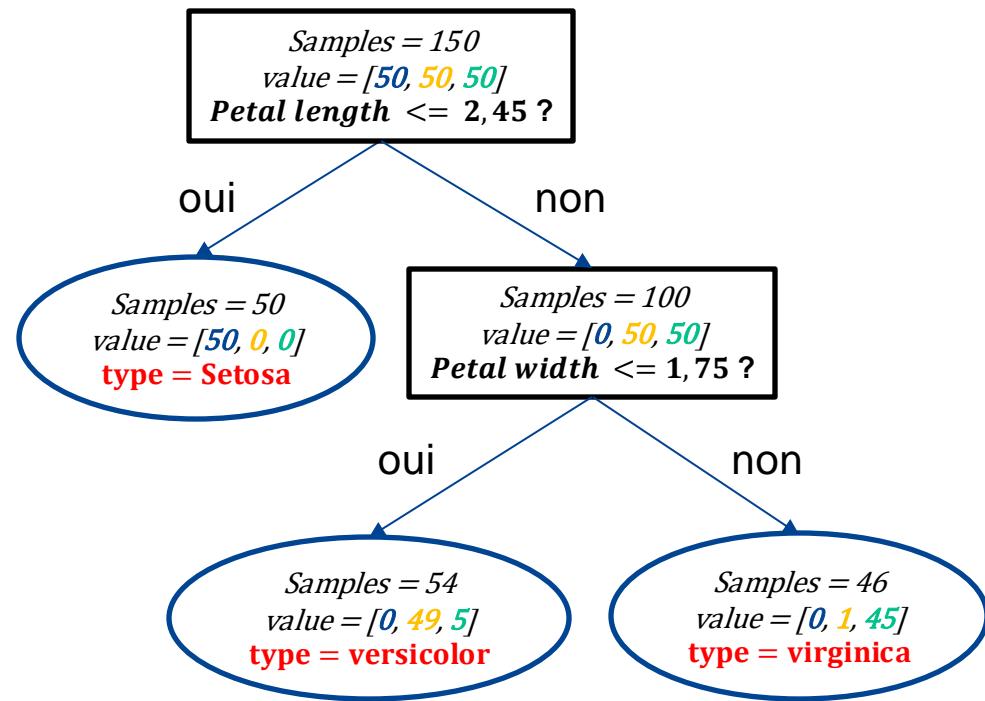
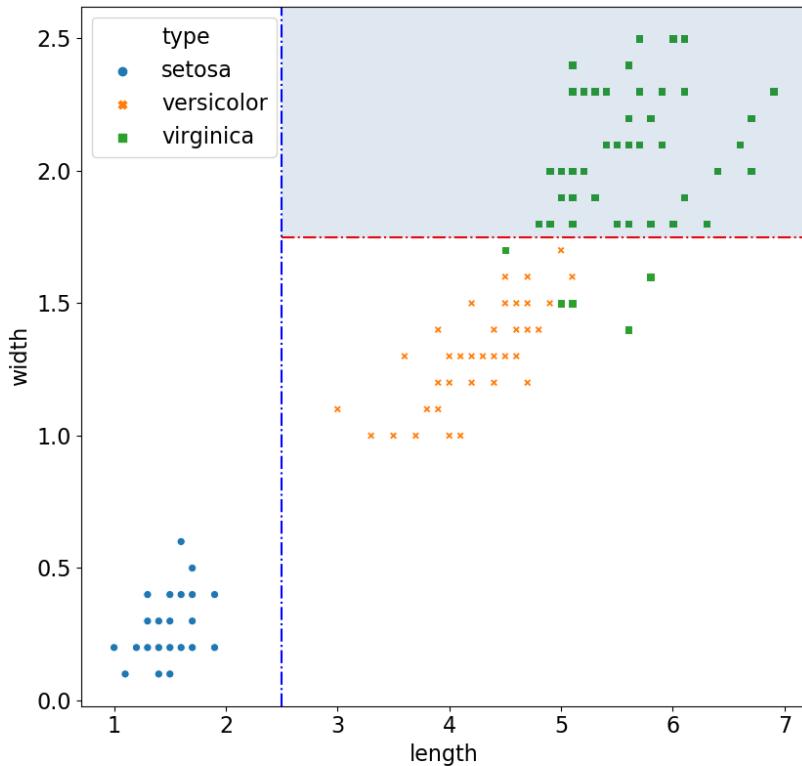




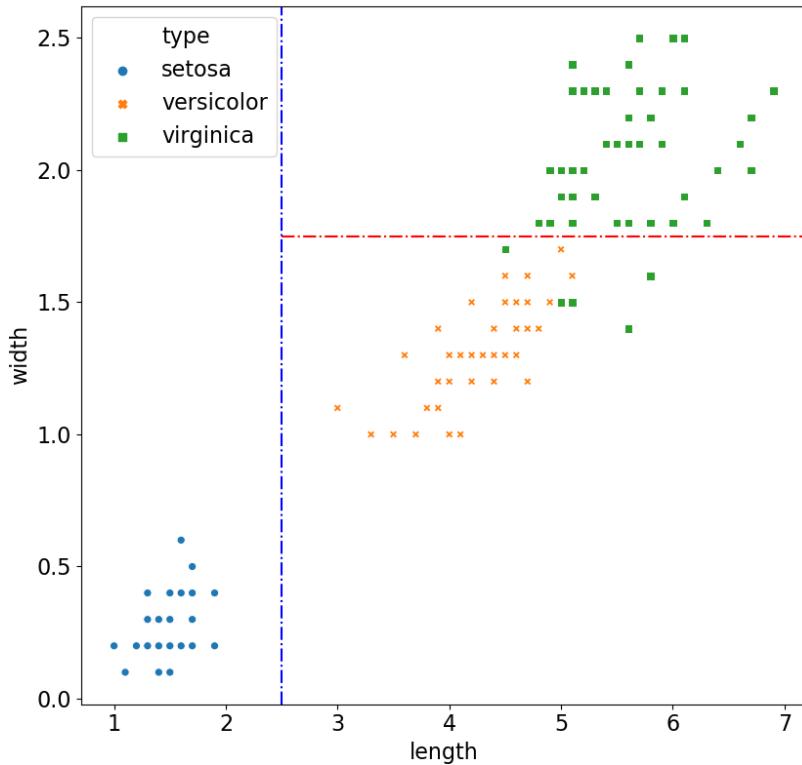
Iris Dataset



Iris Dataset



Critère d'impureté de Gini



■ Comment choisir les questions (délimitations) de l'arbre ?

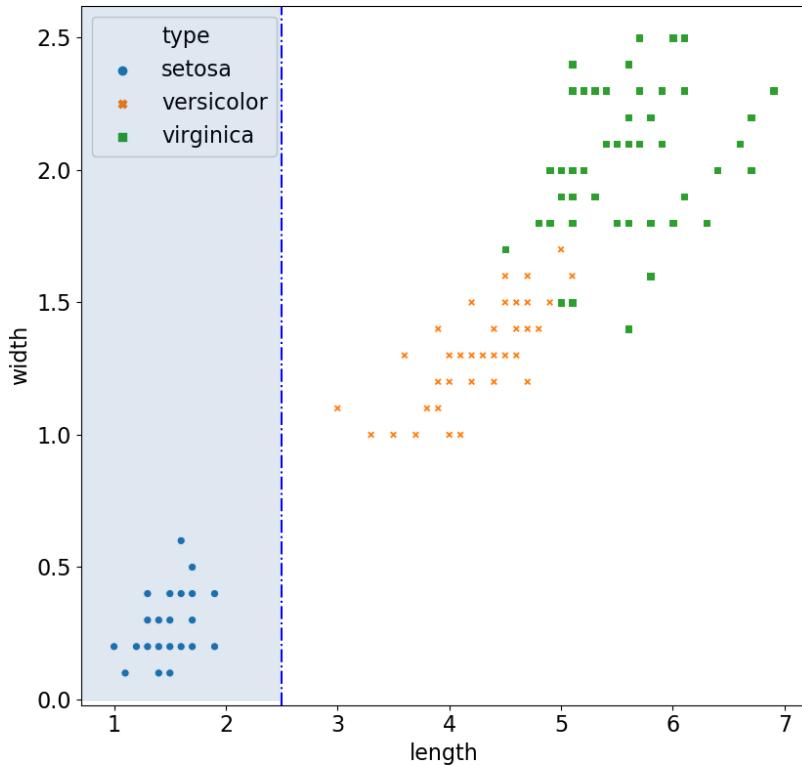
■ Optimisation du « Critère d'impureté de Gini » :

$$G_i = 1 - \sum_{k=1}^n p_{i,k}^2$$

Avec :

- k : le nombre de types de données à classer
- $p_{i,k}$: le ratio du nombre de données de la classe k sur le nombre de données du sous ensemble délimité

Exemple de calcul du critère d'impureté de Gini



Samples = 150
value = [50, 50, 50]
Petal length <= 2,45 ?

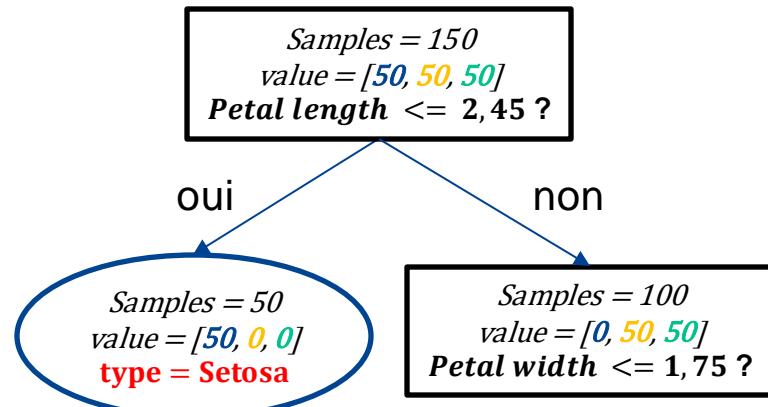
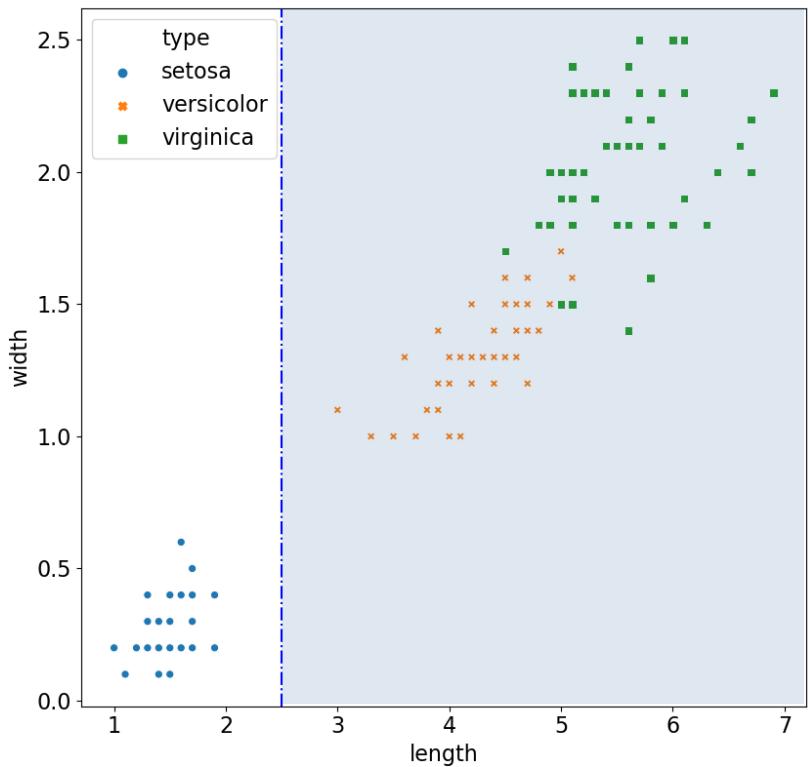
oui

Samples = 50
value = [50, 0, 0]
type = Setosa

$$G = 1 - \left(\frac{50}{50}\right)^2 - \left(\frac{0}{50}\right)^2 - \left(\frac{0}{50}\right)^2 = 0$$

→ Critère parfait. Le sous-ensemble est composé uniquement du même type de donnée

Exemple de calcul du critère d'impureté de Gini



$$G = 1 - \left(\frac{0}{100} \right)^2 - \left(\frac{50}{100} \right)^2 - \left(\frac{50}{100} \right)^2 = 0,5$$

→ Sous ensemble très hétérogène

Optimisation du critère d'impureté de Gini

■ Optimisation :

- > Utilisation d'un algorithme permettant de minimiser la fonction de coût suivante :

$$f = \frac{m_{left}}{m} G_{left} + \frac{m_{right}}{m} G_{right}$$

Avec :

$m_{left/right}$: le nombre de données dans le sous-ensemble de gauche ou droite

$G_{left/right}$: critère d'impureté de Gini dans le sous-ensemble de gauche ou droite

■ Autres critères d'impureté (Entropie), critère d'arrêt de l'algorithme d'optimisation, nombre de feuilles dans l'arbre (sur-apprentissage) ?