

# Machine Learning

## Cours 1: Introduction

Stéphanie Bricq  
stephanie.bricq@u-bourgogne.fr

Université de Bourgogne

09/2022

# Module Outils de l'IA

## Organisation

- 4 CM (8h)
- 4 TD (8h)
- 4 TP (8h)

## Evaluation

- QCM/ CR TP/projet
- Examen : 1 feuille A4 recto-verso manuscrite autorisée

## Ressources

- en ligne sur la plateforme PLUBEL

# Bibliographie

- [Cornuéjols18] Apprentissage artificiel : deep learning, concepts et algorithmes, A. Cornuéjols, L. Miclet, V. Barra, Paris, Eyrolles, 2018
- [Mueller18] Le Machine Learning avec Python, A. Mueller, S. Guido, First Interactive, 2018
- [Géron17] Hands-On Machine Learning with Scikit-Learn and TensorFlow : Concepts, Tools, and Techniques to Build Intelligent Systems, A. Géron, O'Reilly Media, Inc, USA, 2017
- [Goodfellow16] Deep Learning, I. Goodfellow, Y. Bengio and A. Courville, MIT Press, 2016
- [Lantz19] Machine learning with R, Expert techniques for predictive modeling, B. Lantz, 3rd Edition, Packt Publishing, 2019
- [Mathivet21] Machine Learning, Implémentation en Python avec Scikit-learn, V. Mathivet, Eni Editions, 2021.
- MOOC INRIA Scikit-learn  
<https://inria.github.io/scikit-learn-mooc/>
- Cours d'Eric Leclercq

# Contenu

- Différentes approches de l'intelligence artificielle
- Principales techniques de machine learning
- Introduction au deep learning

# Plan

- 1 Intelligence artificielle
  - Machine learning
  - Deep learning
- 2 Machine learning
  - Introduction
  - Types de systèmes
  - Difficultés
  - Test et validation
- 3 Critères pour évaluer la performance

# Introduction

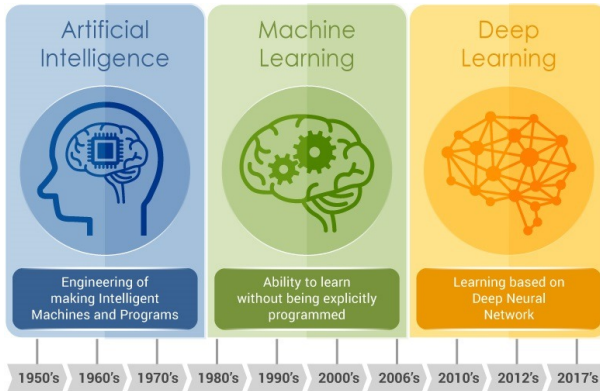
- volume de données important
- données de types plus variés (nombres, textes, images, vidéos, ...)
- grande quantité de données nécessite des méthodes automatisées d'analyse des données.

⇒ apprentissage automatique

# Intelligence artificielle

- développement du domaine de l'intelligence artificielle
- Certains problèmes difficiles pour l'homme ont été résolus par un ordinateur
- Mais d'autres tâches qui peuvent être très faciles à faire pour les humains mais difficiles à décrire formellement sont plus difficiles à résoudre avec une machine.

# Intelligence artificielle



Source : <https://www.viatech.com/>



# Apprentissage automatique (Machine learning)

- certains systèmes ont besoin de pouvoir acquérir leurs propres connaissances en extrayant des modèles à partir de données brutes
- s'appuient sur des observations passées pour apprendre de l'expérience

⇒ **apprentissage automatique** (*Machine learning*)

# Apprentissage automatique (Machine learning)

- a permis aux ordinateurs d'aborder des pbs impliquant la connaissance du monde réel
- et de prendre des décisions qui semblent subjectives

# Apprentissage automatique (Machine learning)

- a permis aux ordinateurs d'aborder des pbs impliquant la connaissance du monde réel
- et de prendre des décisions qui semblent subjectives
- Exemples
  - **régression logistique** pour déterminer s'il faut recommander ou non une césarienne [Mor-Yosef90]
  - algo **bayésien naïf** pour séparer les courriels légitimes des spams

# Apprentissage automatique (Machine learning)

- a permis aux ordinateurs d'aborder des pbs impliquant la connaissance du monde réel
- et de prendre des décisions qui semblent subjectives
- Exemples
  - **régression logistique** pour déterminer s'il faut recommander ou non une césarienne [Mor-Yosef90]
  - algo **bayésien naïf** pour séparer les courriels légitimes des spams
- performance de ces algos dépend de la **représentation** des données qui leur sont transmises
- **caractéristique** (*feature*, ou variable)

# Machine learning

- performances liées à la représentation des données
- nécessité de définir des variables (ou *caractéristiques*)

# Machine learning

- performances liées à la représentation des données
- nécessité de définir des variables (ou *caractéristiques*)
- ex : compagnie d'assurance

# Machine learning

- performances liées à la représentation des données
- nécessité de définir des variables (ou *caractéristiques*)
- ex : compagnie d'assurance
- L'algorithme apprend comment chacune des caractéristiques est en corrélation avec certains résultats, mais il ne peut pas influencer la manière dont les caractéristiques sont définies.

# Deep learning

## Apprentissage profond

- introduit des représentations qui s'expriment en termes d'autres représentations plus simples
- concepts complexes à partir de concepts plus simples

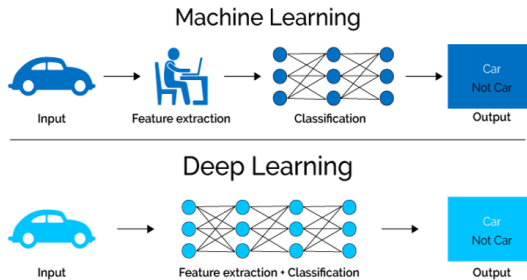


# Deep learning

## Apprentissage profond

- introduit des représentations qui s'expriment en termes d'autres représentations plus simples
- concepts complexes à partir de concepts plus simples
- Exemple : Comment représenter le concept d'image d'une personne ?

# Machine Learning vs Deep learning



# Deep learning

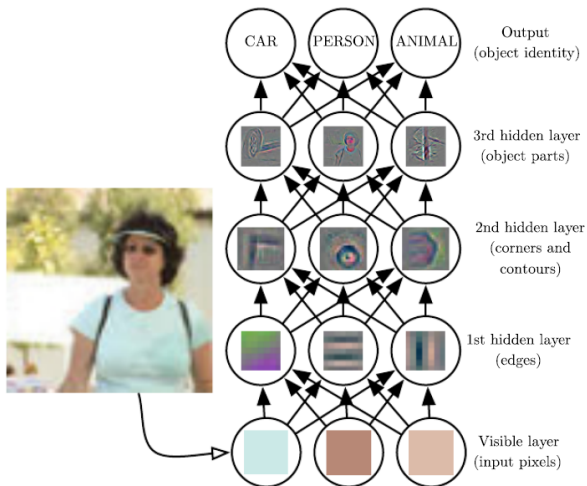


Fig. Illustration d'un modèle d'apprentissage profond [Goodfellow]

# Deep learning

## Exemple : perceptron multicouche (PMC)

- PMC ou réseau de neurones profond à propagation avant
- fonction mathématique qui associe un ensemble de valeurs d'entrée à des valeurs de sortie
- fonction formée par la composition de nb fonctions + simples

# Plan

- 1 Intelligence artificielle
  - Machine learning
  - Deep learning
- 2 Machine learning
  - Introduction
  - Types de systèmes
  - Difficultés
  - Test et validation
- 3 Critères pour évaluer la performance

# Apprentissage automatique ? ?

## Définitions

- "Discipline donnant aux ordinateurs la capacité d'apprendre sans qu'ils soient explicitement programmés" (Arthur Samuel, 1959)

# Apprentissage automatique ? ?

## Définitions

- "Discipline donnant aux ordinateurs la capacité d'apprendre sans qu'ils soient explicitement programmés" (Arthur Samuel, 1959)
- "Etant donnée une tâche  $T$  et une mesure de performance  $P$ , on dit qu'un programme informatique apprend à partir d'une expérience  $E$  si les résultats obtenus sur  $T$ , mesurés par  $P$ , s'améliorent avec l'expérience  $E$ " (Tom Mitchell, 1997)

# Apprentissage automatique ? ?

## Définitions

- "Discipline donnant aux ordinateurs la capacité d'apprendre sans qu'ils soient explicitement programmés" (Arthur Samuel, 1959)
- "Etant donnée une tâche  $T$  et une mesure de performance  $P$ , on dit qu'un programme informatique apprend à partir d'une expérience  $E$  si les résultats obtenus sur  $T$ , mesurés par  $P$ , s'améliorent avec l'expérience  $E$ " (Tom Mitchell, 1997)

Exemple : filtre anti-spam :

- programme d'apprentissage automatique pouvant apprendre à identifier les emails frauduleux à partir d'ex de spam et de message normaux (*ham*)



# Apprentissage automatique ? ?

## Définitions

- **jeu d'entraînement** (*training set*) : exemples utilisés par le système pour son apprentissage
- **échantillon** ou observation d'entraînement

# Apprentissage automatique ??

## Définitions

- **jeu d'entraînement** (*training set*) : exemples utilisés par le système pour son apprentissage
- **échantillon** ou observation d'entraînement

Exemple : filtre anti-spam :

# Apprentissage automatique ? ?

## Définitions

- **jeu d'entraînement** (*training set*) : exemples utilisés par le système pour son apprentissage
- **échantillon** ou observation d'entraînement

Exemple : filtre anti-spam :

- tâche T : identifier les emails frauduleux parmi les nouveaux emails

# Apprentissage automatique ? ?

## Définitions

- **jeu d'entraînement** (*training set*) : exemples utilisés par le système pour son apprentissage
- **échantillon** ou observation d'entraînement

Exemple : filtre anti-spam :

- tâche T : identifier les emails frauduleux parmi les nouveaux emails
- expérience E : données d'entraînement

# Apprentissage automatique ? ?

## Définitions

- **jeu d'entraînement** (*training set*) : exemples utilisés par le système pour son apprentissage
- **échantillon** ou observation d'entraînement

Exemple : filtre anti-spam :

- tâche T : identifier les emails frauduleux parmi les nouveaux emails
  - expérience E : données d'entraînement
  - mesure de performance P doit être définie
    - par ex : % de courriels correctement classés
- ⇒ mesure de performance appelée exactitude (*accuracy*) souvent utilisée dans les tâches de classification

# Apprentissage automatique : Pourquoi ?

## Approche traditionnelle

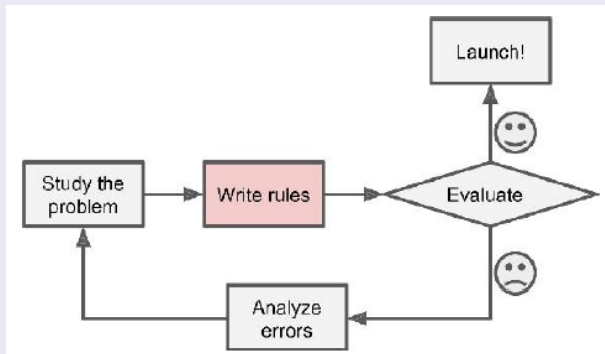


Fig. Approche traditionnelle [Géron17]

# Apprentissage automatique : Pourquoi ?

## Approche Machine Learning

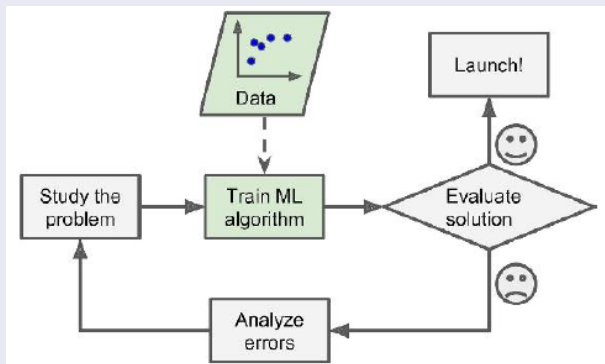


Fig. Approche Machine Learning [Géron17]

⇒ pgm plus court, plus facile à maintenir

# Apprentissage automatique : Pourquoi ?

## Approche Machine Learning

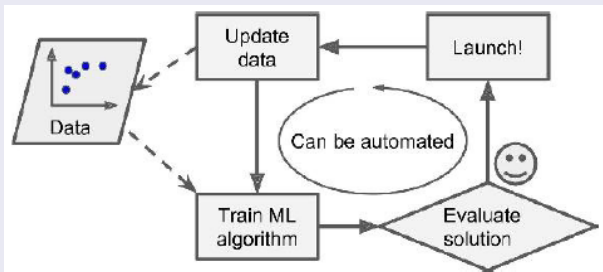


Fig. Adaptation automatique aux évolutions [Géron17]



# Apprentissage automatique : Pourquoi ?

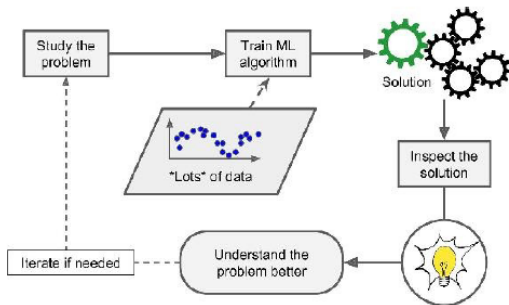


Fig. Apprentissage automatique peut aider les humains à apprendre [Géron17]

- peut révéler des corrélations insoupçonnées ou de nouvelles tendances, permettant ainsi d'avoir une meilleure compréhension du pb

# Apprentissage automatique : Pourquoi ?

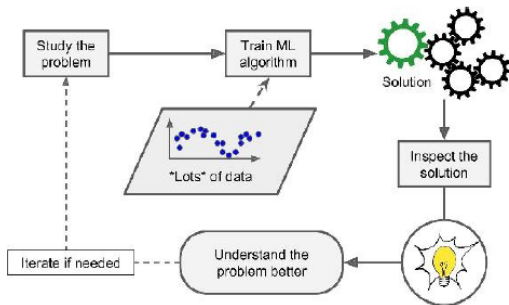


Fig. Apprentissage automatique peut aider les humains à apprendre [Géron17]

- peut révéler des corrélations insoupçonnées ou de nouvelles tendances, permettant ainsi d'avoir une meilleure compréhension du pb
- appliquer des techniques d'apprentissage automatique pour explorer de gros volumes de données peut permettre d'y découvrir certains éléments de structuration qui n'étaient pas immédiatement apparents  
⇒ exploration de données (*data mining*)

# Apprentissage automatique

## En résumé, utile dans les cas suivants

- pour les pb pour lesquels les solutions existantes nécessitent bcp d'ajustements manuels ou de longues listes de règles

# Apprentissage automatique

## En résumé, utile dans les cas suivants

- pour les pb pour lesquels les solutions existantes nécessitent bcp d'ajustements manuels ou de longues listes de règles
- pour les pb complexes pour lesquels il n'existe aucune bonne solution si on adopte une approche traditionnelle

# Apprentissage automatique

## En résumé, utile dans les cas suivants

- pour les pb pour lesquels les solutions existantes nécessitent bcp d'ajustements manuels ou de longues listes de règles
- pour les pb complexes pour lesquels il n'existe aucune bonne solution si on adopte une approche traditionnelle
- pour les environnements fluctuants

# Apprentissage automatique

## En résumé, utile dans les cas suivants

- pour les pb pour lesquels les solutions existantes nécessitent bcp d'ajustements manuels ou de longues listes de règles
- pour les pb complexes pour lesquels il n'existe aucune bonne solution si on adopte une approche traditionnelle
- pour les environnements fluctuants
- pour l'exploitation des pb complexes et des gros volumes de données

# Quelques exemples d'applications

- Identification de spams dans les emails
- Segmentation du comportement des clients pour de la publicité ciblée
- Prévisions météorologiques
- Prédiction du résultat des élections
- Développement d'algorithmes pour le pilotage automatique de drones ou pour la conduite automatique de voitures
- Optimisation énergétique des maisons et immeubles
- Découverte de séquences génétiques liées à des maladies

# Plan

- 1 Intelligence artificielle
  - Machine learning
  - Deep learning
- 2 Machine learning
  - Introduction
  - **Types de systèmes**
  - Difficultés
  - Test et validation
- 3 Critères pour évaluer la performance



# Types de systèmes

## Classement en catégories

# Types de systèmes

## Classement en catégories

- apprentissage *supervisé*, *non supervisé*, *semi-supervisé* ou avec *renforcement*

# Types de systèmes

## Classement en catégories

- apprentissage *supervisé*, *non supervisé*, *semi-supervisé* ou avec *renforcement*
- apprentissage *en ligne* ou apprentissage *groupé*

# Types de systèmes

## Classement en catégories

- apprentissage *supervisé*, *non supervisé*, *semi-supervisé* ou avec *renforcement*
- apprentissage *en ligne* ou apprentissage *groupé*
- apprentissage *à partir d'observations* ou *à partir d'un modèle*

Critères non exclusifs, possibilité de les combiner

# Apprentissage supervisé/non supervisé

4 catégories majeures :

- apprentissage *supervisé*,
- apprentissage *non supervisé*,
- apprentissage *semi-supervisé*
- apprentissage avec *renforcement*

# Apprentissage supervisé

- données d'entraînement fournies à l'algo comportent les solutions désirées, appelées *étiquettes* (*labels*)

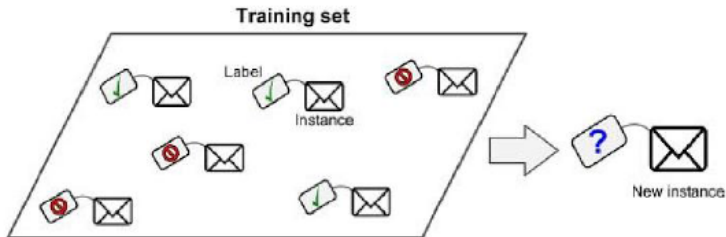


Fig. Jeu d'entraînement étiqueté pour apprentissage supervisé [Géron17]

# Apprentissage supervisé

## Exemple de tâche

- **classification**

# Apprentissage supervisé

## Exemple de tâche

- **classification**
- prédiction d'une valeur numérique cible (*target*) à partir des valeurs d'un certain nb d'*attributs* ou *variables* : **régression**. Ces valeurs sont appelées les caractéristiques d'une observation.

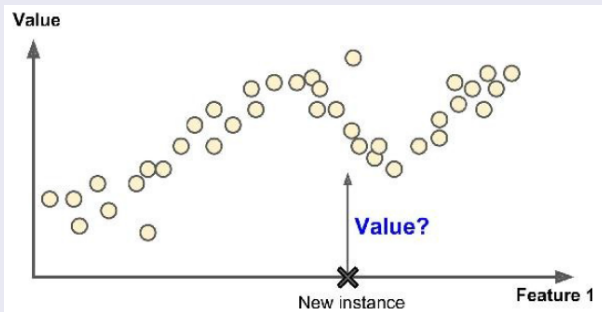


Fig. Régression [Géron17]



# Apprentissage supervisé

## Exemples

- K plus proches voisins
- Régression linéaire
- Régression logistique
- Machines à vecteurs de support
- Arbres de décisions et forêts aléatoires
- Réseaux neuronaux

# Apprentissage non supervisé

- les données d'apprentissage ne sont pas étiquetées

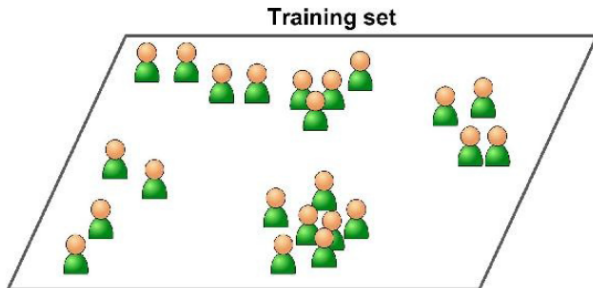


Fig. Jeu d'entraînement non étiqueté pour apprentissage non supervisé [Géron17]

# Apprentissage non supervisé

## Exemples

- Partitionnement
  - K-moyennes (*clustering*)
  - Partitionnement hiérarchique
- Visualisation et réduction de dimension
  - Analyse en composants principales
- Détection d'anomalies ou de nouveauté
  - One-class SVM
  - Isolation Forest
- Apprentissage par association de règles
  - A priori
  - Eclat

# Apprentissage non supervisé

## Exemples

- Partitionnement (*clustering*)

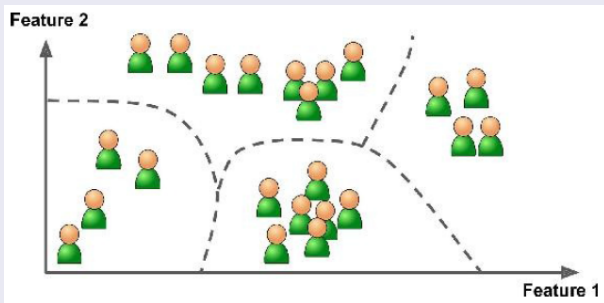


Fig. Partitionnement [Géron17]

# Apprentissage non supervisé

## Exemples

- Partitionnement (*clustering*)

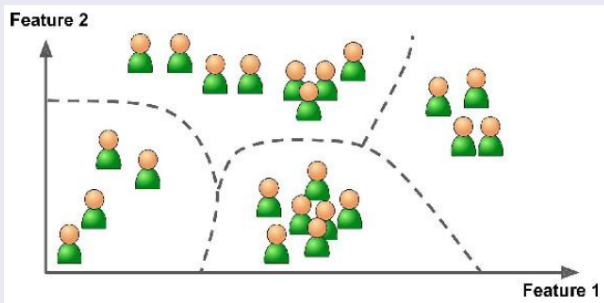


Fig. Partitionnement [Géron17]

- Partitionnement hiérarchique : chaque groupe en groupes plus petits

# Apprentissage non supervisé

## Exemples

- Détection d'anomalies

# Apprentissage non supervisé

## Exemples

- Détection d'anomalies

- entraînement du système avec des observations normales
- quand on lui fournit une nouvelle observation, le système peut dire si elle paraît normale ou si c'est vraisemblablement une anomalie.

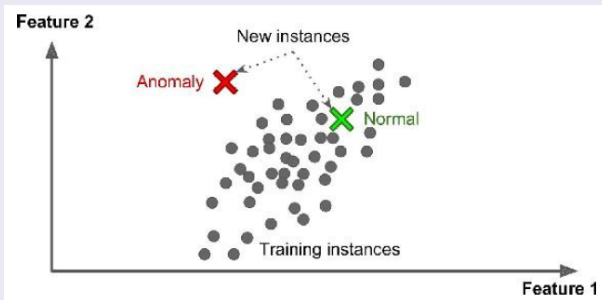


Fig. Détection d'anomalies [Géron17]

# Apprentissage non supervisé

## Exemples

- Apprentissage par association de règle
  - Objectif :
    - explorer de larges ensembles de données
    - pour découvrir des relations entre les variables
  - ex : gestion d'un supermarché



# Apprentissage semi-supervisé

- Données d'apprentissage partiellement étiquetées

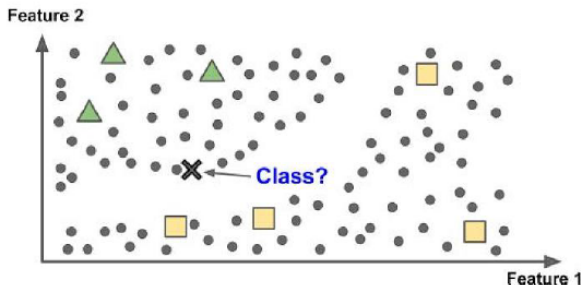


Fig. Apprentissage semi-supervisé [Géron17]

- ex : services d'hébergement d'images (Google Photos)
- souvent des combinaisons d'algos non supervisés et supervisés

# Apprentissage par renforcement

- $\neq$  des autres types
- dans ce contexte, système d'apprentissage appelé *agent*

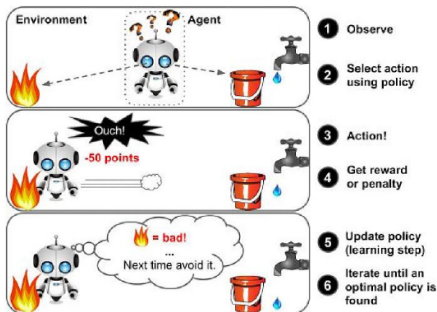


Fig. Apprentissage par renforcement [Géron17]

# Apprentissage par renforcement

- $\neq$  des autres types
- dans ce contexte, système d'apprentissage appelé *agent*

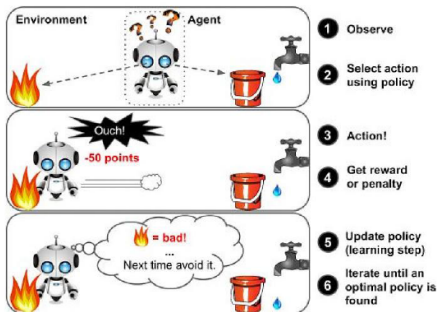


Fig. Apprentissage par renforcement [Géron17]

- ex : robots pour apprendre à marcher

# Apprentissage groupé et en ligne

- Autre critère pour classer les systèmes d'apprentissage automatique : peuvent-ils apprendre ou non progressivement à partir d'un flux de données entrantes

# Apprentissage groupé et en ligne

- Autre critère pour classer les systèmes d'apprentissage automatique : peuvent-ils apprendre ou non progressivement à partir d'un flux de données entrantes
  - apprentissage *groupé* (ou *batch*)
  - apprentissage *en ligne*

# Apprentissage groupé (batch)

- pas d'apprentissage progressif, système entraîné avec toutes les données dispos
- nécessite bcp de temps et de ressources informatiques

# Apprentissage groupé (batch)

- pas d'apprentissage progressif, système entraîné avec toutes les données dispos
- nécessite bcp de temps et de ressources informatiques

⇒ tâche effectuée en différé : **apprentissage hors-ligne**

# Apprentissage en ligne

- système entraîné progressivement en l'alimentant peu à peu avec des observations soit une à une soit par petits groupes (*mini-lots*, *mini-batches*)

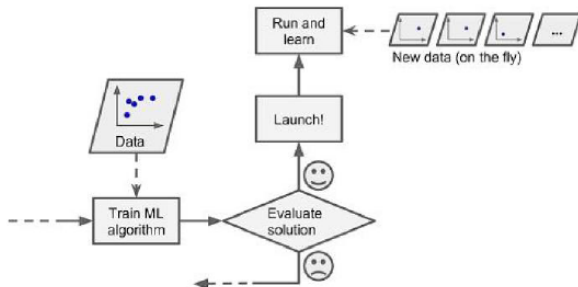


Fig. Apprentissage en ligne [Géron17]



# Apprentissage en ligne

- système entraîné progressivement en l'alimentant peu à peu avec des observations soit une à une soit par petits groupes (mini-lots, *mini-batches*)

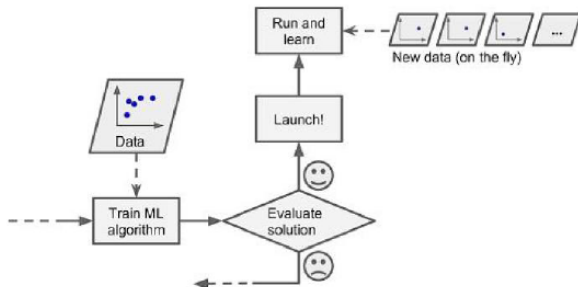


Fig. Apprentissage en ligne [Géron17]

- pour des systèmes recevant des données en flux continu
- ou dans le cas de ressources informatiques limitées

# Apprentissage en ligne

- pour l'entraînement sur des gros jeux de données ne pouvant tenir en mémoire principale

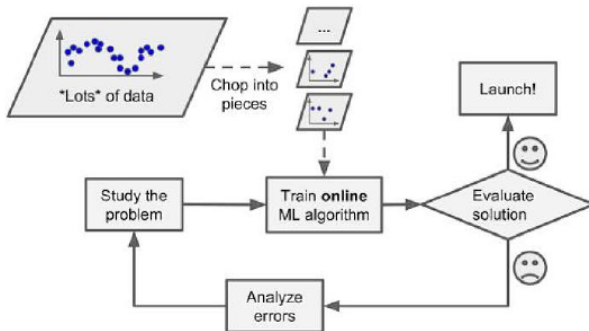


Fig. Apprentissage en ligne [Géron17]

# Apprentissage en ligne

- Paramètre : taux d'apprentissage (*learning rate*)
  - rythme auquel ils doivent s'adapter à l'évolution des données
  - si taux élevé

# Apprentissage en ligne

- Paramètre : taux d'apprentissage (*learning rate*)
  - rythme auquel ils doivent s'adapter à l'évolution des données
  - si taux élevé
    - ⇒ adaptation rapide aux nouvelles données
    - ⇒ oubli rapide des anciennes

# Apprentissage en ligne

- Paramètre : taux d'apprentissage (*learning rate*)
  - rythme auquel ils doivent s'adapter à l'évolution des données
  - si taux élevé
    - ⇒ adaptation rapide aux nouvelles données
    - ⇒ oubli rapide des anciennes
  - inversement, si taux faible, le système a une plus grande inertie
    - ⇒ apprentissage plus lent
    - ⇒ moins sensible aux parasites dans les nouvelles données

# Apprentissage en ligne

- Paramètre : taux d'apprentissage (*learning rate*)
  - rythme auquel ils doivent s'adapter à l'évolution des données
  - si taux élevé
    - ⇒ adaptation rapide aux nouvelles données
    - ⇒ oubli rapide des anciennes
  - inversement, si taux faible, le système a une plus grande inertie
    - ⇒ apprentissage plus lent
    - ⇒ moins sensible aux parasites dans les nouvelles données
- Difficulté :
  - en cas d'introduction de mauvaises données dans le système, dégradation progressive des résultats

# Apprentissage à partir d'observations ou à partir d'un modèle

- classement sur leur mode de généralisation
- à partir d'exemples d'apprentissage, le système doit pouvoir généraliser à des exemples non vus auparavant

# Apprentissage à partir d'observations ou à partir d'un modèle

- classement sur leur mode de généralisation
- à partir d'exemples d'apprentissage, le système doit pouvoir généraliser à des exemples non vus auparavant
- 2 approches :
  - Apprentissage à partir d'observations
  - Apprentissage à partir d'un modèle



# Apprentissage à partir d'observations

- système apprend les exemples
- généralisation en utilisant une **mesure de similarité**

⇒ **Apprentissage à partir d'observations**

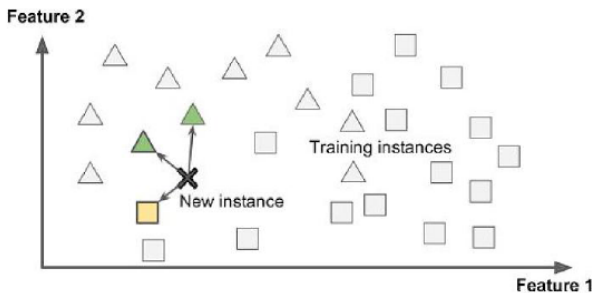


Fig. Apprentissage à partir d'observations [Géron17]

# Apprentissage à partir d'un modèle

- construire un modèle de ces exemples
- utilisation du modèle pour faire des **prédictions**

⇒ **Apprentissage à partir d'un modèle**

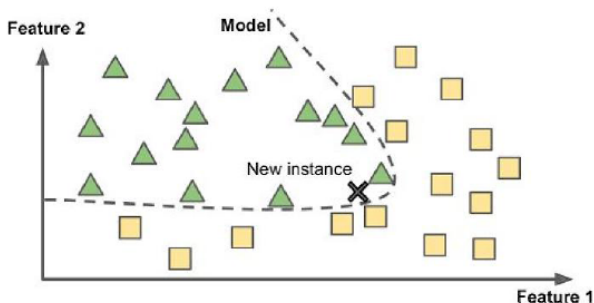


Fig. Apprentissage à partir d'un modèle [Géron17]

# Apprentissage à partir d'un modèle

## Etapes

- étude des données
- sélection d'un modèle
- entraînement du modèle sur des données d'entraînement
- application du modèle pour effectuer des prédictions sur des nouveaux cas  $\Rightarrow$  **inférence**

# Plan

- 1 Intelligence artificielle
  - Machine learning
  - Deep learning
- 2 Machine learning
  - Introduction
  - Types de systèmes
  - **Difficultés**
  - Test et validation
- 3 Critères pour évaluer la performance

# Principales difficultés de l'apprentissage automatique

- Tâche principale : sélectionner un algorithme d'apprentissage et l'entraîner sur certaines données
- 2 écueils
  - "Mauvais algorithme"
  - "Mauvaises données"

# Difficultés au niveau des données

- Données d'apprentissage en nb insuffisant
  - nécessité d'avoir un grand nb de données

# Difficultés au niveau des données

- Données d'apprentissage en nb insuffisant
- Données d'entraînement non représentatives
  - nécessité d'avoir un jeu d'entraînement représentatif
    - si l'échantillon est trop petit  $\Rightarrow$  **bruit d'échantillonnage** : données non représentatives résultant du hasard
    - échantillons très importants peuvent être non représentatifs en cas de méthode d'échantillonnage défectueuse  $\Rightarrow$  **biais d'échantillonnage**

# Difficultés au niveau des données

- Données d'apprentissage en nb insuffisant
- Données d'entraînement non représentatives
- Données de mauvaise qualité
  - jeu d'entraînement peut contenir des erreurs, des données aberrantes et du bruit
  - ⇒ moins bons résultats
  - ⇒ nettoyer les données d'apprentissage
    - si données aberrantes ⇒ suppression ou correction manuelle
    - si qq valeurs manquantes dans les observations , il faut décider si on ignore la variable ou ces observations, ou si on remplit la valeur manquante (par ex avec la moyenne des autres valeurs ...)



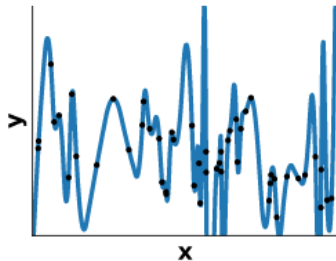
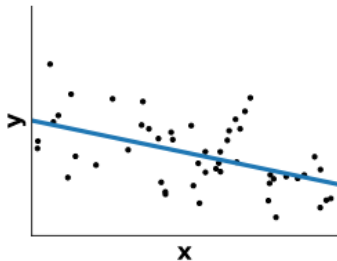
# Difficultés au niveau des données

- Données d'apprentissage en nb insuffisant
- Données d'entraînement non représentatives
- Données de mauvaise qualité
- Variables non pertinentes
  - nécessité de choisir un bon ensemble de variables sur lesquelles s'entraîner  $\Rightarrow$  **ingénierie des variables** (*feature engineering*)
    - sélection de variables
    - extraction de variables : combiner plusieurs variables existantes pour en produire une autre qui sera plus utile
    - introduction de nouvelles variables grâce à la collecte de nouvelles données

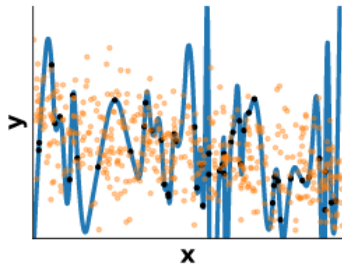
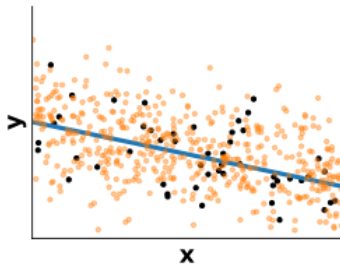
# Difficultés au niveau des algorithmes

- Surajustement des données d'entraînement (*overfitting*)
- Sous-ajustement des données d'entraînement (*underfitting*)

# Quel ajustement ?

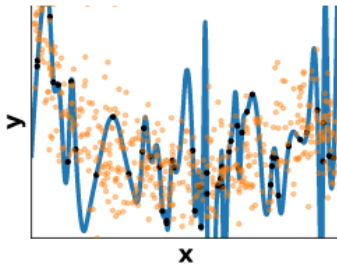
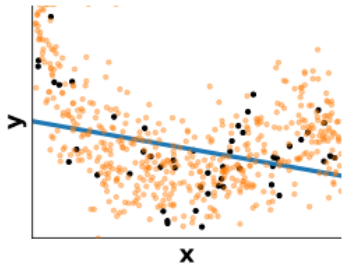


# Quel ajustement ?



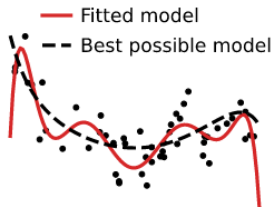
**On new data**

# Quel ajustement ?



A harder example

# Surajustement / Overfitting



- Modèle **trop complexe** pour les données :
  - Son meilleur ajustement possible se rapprocherait bien du processus génératif
  - Cependant, sa flexibilité capture le bruit
- pb rencontré quand :
  - pas assez de données
  - ou trop de bruit

# Difficultés au niveau des algorithmes

## Surajustement des données d'entraînement (*overfitting*)

- lorsque le modèle est trop complexe par rapport à la quantité de données d'apprentissage et au bruit qu'elles contiennent
- Plusieurs solutions :

# Difficultés au niveau des algorithmes

## Surajustement des données d'entraînement (*overfitting*)

- lorsque le modèle est trop complexe par rapport à la quantité de données d'apprentissage et au bruit qu'elles contiennent
- Plusieurs solutions :
  - simplifier le modèle
    - en sélectionnant moins de paramètres
    - en réduisant le nb d'attributs des données d'entraînement
    - ou en imposant des contraintes au modèle (régularisation)



# Difficultés au niveau des algorithmes

## Surajustement des données d'entraînement (*overfitting*)

- lorsque le modèle est trop complexe par rapport à la quantité de données d'apprentissage et au bruit qu'elles contiennent
- Plusieurs solutions :
  - simplifier le modèle
    - en sélectionnant moins de paramètres
    - en réduisant le nb d'attributs des données d'entraînement
    - ou en imposant des contraintes au modèle (régularisation)
  - rassembler davantage de données d'apprentissage

# Difficultés au niveau des algorithmes

## Surajustement des données d'entraînement (*overfitting*)

- lorsque le modèle est trop complexe par rapport à la quantité de données d'apprentissage et au bruit qu'elles contiennent
- Plusieurs solutions :
  - simplifier le modèle
    - en sélectionnant moins de paramètres
    - en réduisant le nb d'attributs des données d'entraînement
    - ou en imposant des contraintes au modèle (régularisation)
  - rassembler davantage de données d'apprentissage
  - réduire le bruit dans ces données
    - en corrigeant les erreurs
    - en supprimant les données aberrantes

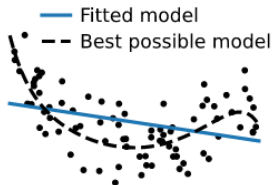
# Difficultés au niveau des algorithmes

## Surajustement des données d'entraînement (*overfitting*)

- lorsque le modèle est trop complexe par rapport à la quantité de données d'apprentissage et au bruit qu'elles contiennent
- Plusieurs solutions :
  - simplifier le modèle
    - en sélectionnant moins de paramètres
    - en réduisant le nb d'attributs des données d'entraînement
    - ou en imposant des contraintes au modèle (régularisation)
  - rassembler davantage de données d'apprentissage
  - réduire le bruit dans ces données
    - en corrigeant les erreurs
    - en supprimant les données aberrantes

niveau de régularisation peut être contrôlé par un **hyperparamètre** : param de l'algorithme d'apprentissage

# Sous-ajustement / underfitting



- Modèle **trop simple** pour les données :
  - Son meilleur ajustement ne se rapproche pas bien du processus génératif
  - mais capture peu de bruit
- pb rencontré quand :
  - il y a beaucoup de données par rapport à la complexité du modèle
  - ou dans des situations à faible bruit.

# Difficultés au niveau des algorithmes

## Sous-ajustement des données d'entraînement (*underfitting*)

- lorsque le modèle est trop simple pour découvrir la structure sous-jacente des données
- Plusieurs solutions :

# Difficultés au niveau des algorithmes

## Sous-ajustement des données d'entraînement (*underfitting*)

- lorsque le modèle est trop simple pour découvrir la structure sous-jacente des données
- Plusieurs solutions :
  - choisir un modèle plus puissant avec plus de paramètres
  - fournir de meilleures variables à l'algorithme d'apprentissage
  - réduire les contraintes sur le modèle

# En résumé

- apprentissage automatique
  - consiste à rendre une machine capable de mieux accomplir une tâche grâce à un entraînement sur des données
  - de nombreux types différents de systèmes d'apprentissage automatique
    - supervisés ou non
    - en différé ou en ligne
    - à partir de modèles ou à partir d'observations

# En résumé

- Etapes :



# En résumé

- Etapes :
  - Rassembler des données dans un jeu d'entraînement

# En résumé

- Etapes :
  - Rassembler des données dans un jeu d'entraînement
  - Alimenter l'algorithme d'apprentissage avec ce jeu

# En résumé

- Etapes :
  - Rassembler des données dans un jeu d'entraînement
  - Alimenter l'algorithme d'apprentissage avec ce jeu
  - Puis
    - Si l'apprentissage s'effectue à partir d'un modèle, l'algorithme ajuste ce modèle au jeu d'entraînement

# En résumé

- Etapes :

- Rassembler des données dans un jeu d'entraînement
- Alimenter l'algorithme d'apprentissage avec ce jeu
- Puis
  - Si l'apprentissage s'effectue à partir d'un modèle, l'algorithme ajuste ce modèle au jeu d'entraînement
  - Si l'apprentissage s'effectue à partir d'observations, il mémorise les exemples et utilise une mesure de similarité pour généraliser à de nouvelles observations

# En résumé

- Etapes :
  - Rassembler des données dans un jeu d'entraînement
  - Alimenter l'algorithme d'apprentissage avec ce jeu
  - Puis
    - Si l'apprentissage s'effectue à partir d'un modèle, l'algorithme ajuste ce modèle au jeu d'entraînement
    - Si l'apprentissage s'effectue à partir d'observations, il mémorise les exemples et utilise une mesure de similarité pour généraliser à de nouvelles observations
- Pas de bons résultats
  - si jeu d'entraînement trop petit

# En résumé

- Etapes :
  - Rassembler des données dans un jeu d'entraînement
  - Alimenter l'algorithme d'apprentissage avec ce jeu
  - Puis
    - Si l'apprentissage s'effectue à partir d'un modèle, l'algorithme ajuste ce modèle au jeu d'entraînement
    - Si l'apprentissage s'effectue à partir d'observations, il mémorise les exemples et utilise une mesure de similarité pour généraliser à de nouvelles observations
- Pas de bons résultats
  - si jeu d'entraînement trop petit
  - si données
    - non-représentatives
    - entâchées de bruit
    - polluées par des variables non appropriées

# En résumé

- Etapes :

- Rassembler des données dans un jeu d'entraînement
- Alimenter l'algorithme d'apprentissage avec ce jeu
- Puis
  - Si l'apprentissage s'effectue à partir d'un modèle, l'algorithme ajuste ce modèle au jeu d'entraînement
  - Si l'apprentissage s'effectue à partir d'observations, il mémorise les exemples et utilise une mesure de similarité pour généraliser à de nouvelles observations

- Pas de bons résultats

- si jeu d'entraînement trop petit
- si données
  - non-représentatives
  - entâchées de bruit
  - polluées par des variables non appropriées
- si modèle trop simple (sous-ajustement) ou trop complexe (surajustement)

# En résumé

- Etapes :

- Rassembler des données dans un jeu d'entraînement
- Alimenter l'algorithme d'apprentissage avec ce jeu
- Puis
  - Si l'apprentissage s'effectue à partir d'un modèle, l'algorithme ajuste ce modèle au jeu d'entraînement
  - Si l'apprentissage s'effectue à partir d'observations, il mémorise les exemples et utilise une mesure de similarité pour généraliser à de nouvelles observations

- Pas de bons résultats

- si jeu d'entraînement trop petit
- si données
  - non-représentatives
  - entâchées de bruit
  - polluées par des variables non appropriées
- si modèle trop simple (sous-ajustement) ou trop complexe (surajustement)

⇒ Après entraînement, validation + réglages éventuels



# Plan

- 1 Intelligence artificielle
  - Machine learning
  - Deep learning
- 2 Machine learning
  - Introduction
  - Types de systèmes
  - Difficultés
  - **Test et validation**
- 3 Critères pour évaluer la performance

# Test et validation

- partage des données en 2 ensembles : **jeu d'entraînement** et **jeu de test** : en général 80% des données pour l'entraînement et 20% pour les tests

# Test et validation

- partage des données en 2 ensembles : **jeu d'entraînement** et **jeu de test** : en général 80% des données pour l'entraînement et 20% pour les tests
- erreur de généralisation : taux d'erreur sur le jeu de test
  - si erreur d'apprentissage faible mais erreur de généralisation élevée  $\Rightarrow$  modèle surajuste les données d'entraînement

# Test et validation

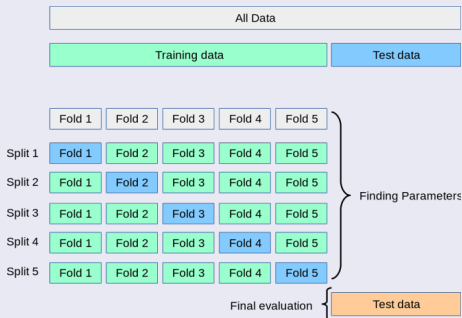
- partage des données en 2 ensembles : **jeu d'entraînement** et **jeu de test** : en général 80% des données pour l'entraînement et 20% pour les tests
- erreur de généralisation : taux d'erreur sur le jeu de test
  - si erreur d'apprentissage faible mais erreur de généralisation élevée  $\Rightarrow$  modèle surajuste les données d'entraînement
- comment choisir entre plusieurs modèles ou comment choisir les valeurs des hyperparamètres ?

# Test et validation

- partage des données en 2 ensembles : **jeu d'entraînement** et **jeu de test** : en général 80% des données pour l'entraînement et 20% pour les tests
- erreur de généralisation : taux d'erreur sur le jeu de test
  - si erreur d'apprentissage faible mais erreur de généralisation élevée  $\Rightarrow$  modèle surajuste les données d'entraînement
- comment choisir entre plusieurs modèles ou comment choisir les valeurs des hyperparamètres ?
- jeu de validation
  - entraînement de plusieurs modèles avec différents hyperparamètres sur le jeu d'entraînement
  - sélection du modèle et des hyperparamètres donnant les meilleurs résultats sur le jeu de validation
  - test final unique sur le jeu de test pour estimer l'erreur de généralisation

# Test et validation

## Validation croisée

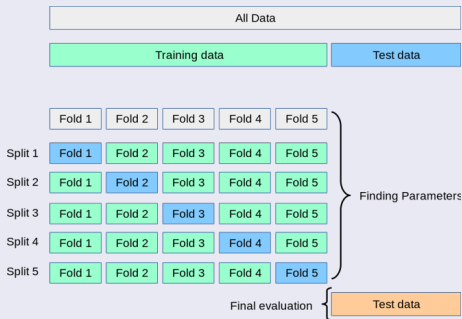


Source : scikit-learn.org

- jeu d'entraînement partagé en sous-ens complémentaires
- chaque modèle est
  - évalué sur une combinaison différente des ces sous-ens
  - puis validé sur les sous-ens restants

# Test et validation

## Validation croisée



Source : scikit-learn.org

- après sélection du type de modèle et des hyperparamètres, le modèle final utilisant ces hyperparamètres est entraîné sur le jeu d'entraînement complet
- erreur de généralisation mesurée sur le jeu de test

# Travailler avec des données réelles

- préférable de s'entraîner sur des données du monde réel
- jeux de données librement accessibles (*open datasets*)
  - <https://www.kaggle.com/> : plateforme de compétition
  - <https://www.openml.org>
  - <http://archive.ics.uci.edu/ml/> : entrepôt de Machine Learning de l'Université d'Irvine en Californie







# Plan










- 1 Intelligence artificielle
  - Machine learning
  - Deep learning
- 2 Machine learning
  - Introduction
  - Types de systèmes
  - Difficultés
  - Test et validation
- 3 Critères pour évaluer la performance

# Matrice de confusion

**Two Classes**

		Predicted Class	
		A	B
Actual Class	A		
	B		

**Three Classes**

		Predicted Class		
		A	B	C
Actual Class	A			
	B			
	C			

# Matrice de confusion : cas $2 \times 2$

		Predicted class	
		no	yes
Actual class	no	<div>TN True Negative</div>	<div>FP False Positive</div>
	yes	<div>FN False Negative</div>	<div>TP True Positive</div>

- vrais positifs (TP) : bien classés comme la classe d'intérêt
- vrais négatifs (TN) : bien classés comme n'étant pas la classe d'intérêt
- faux positifs (FP) : mal classés comme la classe d'intérêt
- faux négatifs (FN) : mal classés comme n'étant pas la classe d'intérêt

# Métriques de classification

- Accuracy (*success rate*)

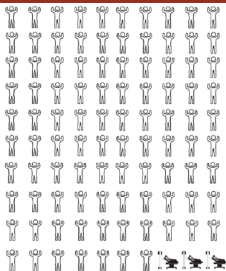
$$\text{accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

- taux d'erreur (*error rate*)

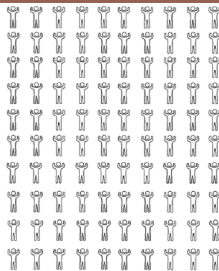
$$\text{error rate} = \frac{FP + FN}{TP + TN + FP + FN} = 1 - \text{accuracy}$$

# Accuracy

Reality



Prediction



**Imbalanced dataset**

# Sensibilité et spécificité

- Sensibilité (*sensitivity* ou *true positive rate*) : mesure la proportion de cas positifs correctement classés

$$\text{sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

# Sensibilité et spécificité

- Sensibilité (*sensitivity* ou *true positive rate*) : mesure la proportion de cas positifs correctement classés

$$\text{sensitivity} = \frac{TP}{TP + FN}$$

- spécificité (*specificity* ou *false positive rate*) : mesure la proportion de cas négatifs correctement classés

$$\text{specificity} = \frac{TN}{TN + FP}$$

# Precision et recall

- Précision ou valeur positive prédictive (*precision, positive predictive value*) : proportion de cas positifs qui sont réellement positifs

$$\text{precision} = \frac{TP}{TP + FP}$$



# Precision et recall

- Précision ou valeur positive prédictive (*precision, positive predictive value*) : proportion de cas positifs qui sont réellement positifs

$$\text{precision} = \frac{TP}{TP + FP}$$

- Rappel (*recall*)

$$\text{recall} = \frac{TP}{TP + FN}$$

# F-Mesure

- F1-score : moyenne harmonique de la précision et du rappel

$$\text{F-mesure} = \frac{2 \times \text{precision} \times \text{recall}}{\text{recall} + \text{precision}} = \frac{2 \times \text{TP}}{2 \times \text{TP} + \text{FP} + \text{FN}}$$

# Exemple

Test clinique : frottis de dépistage du cancer du col de l'utérus

	Cancer	Pas de cancer	Total
Frottis +	190	210	400
Frottis -	10	3590	3600
Total	200	3800	4000

Calculer rappel, spécificité et précision.

# Exemple

Test clinique : frottis de dépistage du cancer du col de l'utérus

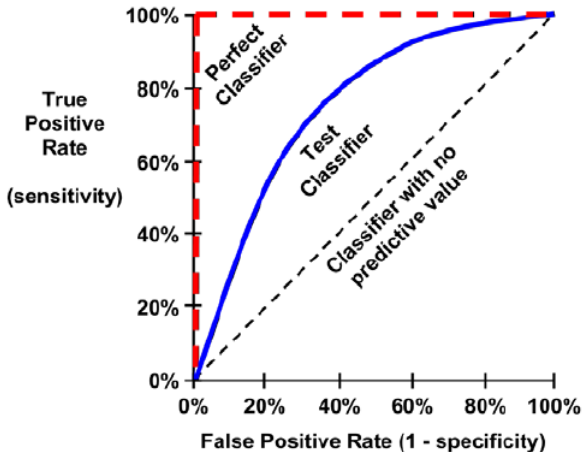
	Cancer	Pas de cancer	Total
Frottis +	190	210	400
Frottis -	10	3590	3600
Total	200	3800	4000

Calculer rappel, spécificité et précision.

Interpréter les résultats obtenus : ce test est-il un bon outil de dépistage ? un bon outil diagnostique ?

# Visualisation de la performance : courbe ROC

- courbe ROC (*receiver operating characteristic curve*)
  - pour visualiser l'efficacité des modèles de machine learning
  - décrit la l'évolution de la sensibilité en fonction de 1 - spécificité



# Courbe ROC

- Construction

- on prend pour seuil les valeurs successives de la fonction de décision sur notre jeu de données
- à chaque valeur de seuil, une observation que l'on prédisait précédemment négative change d'étiquette
  - si cette observation est effectivement positive, la sensibilité augmente de  $1/n_p$  (où  $n_p$  est le nb d'exemples positifs)
  - sinon c'est le taux de faux positifs qui augmente de  $1/n_n$  ( $n_n$  nb d'exemples négatifs)

⇒ courbe en escalier

# Courbe ROC

- Construction

- on prend pour seuil les valeurs successives de la fonction de décision sur notre jeu de données
- à chaque valeur de seuil, une observation que l'on prédisait précédemment négative change d'étiquette
  - si cette observation est effectivement positive, la sensibilité augmente de  $1/n_p$  (où  $n_p$  est le nb d'exemples positifs)
  - sinon c'est le taux de faux positifs qui augmente de  $1/n_n$  ( $n_n$  nb d'exemples négatifs)

⇒ courbe en escalier

- AUC (*Area Under the Roc*) : aire sous la courbe

- plus la valeur est proche de 1, meilleur est le classifieur

# Segmentation d'images

- Attribution d'une classe à chaque pixel
- Comment évaluer les performances ?



# Segmentation d'images

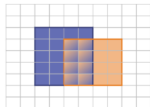
- Attribution d'une classe à chaque pixel
- Comment évaluer les performances ?
  - Recouvrement (*overlap*) entre la prédiction et la vérité terrain
  - Qualité de la frontière

# Segmentation d'images

- Attribution d'une classe à chaque pixel
- Comment évaluer les performances ?
  - Recouvrement (*overlap*) entre la prédiction et la vérité terrain
  - Qualité de la frontière
- On peut considérer la labelisation de chaque pixel de l'image comme un pb de classification
  - possibilité d'utiliser les métriques de classification
  - !! Déséquilibre !

# Overlap

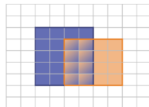
(a) DSC



■ A ■ B ■  $A \cap B$

$$\begin{aligned} DSC(A,B) &= \frac{2 \cdot |A \cap B|}{|A| + |B|} \\ &= \frac{2 \cdot |A \cap B|}{|A| + |B|} \end{aligned}$$

(b) IoU



■ A ■ B ■  $A \cap B$

$$\begin{aligned} IoU(A,B) &= \frac{|A \cap B|}{|A| + |B| - |A \cap B|} \\ &= \frac{|A \cap B|}{|A \cup B|} \end{aligned}$$

- Dice Similarity Coefficient (DSC)
- Intersection over Union (IoU) (ou Index Jaccard)

$$DSC = \frac{2|A \cap B|}{|A| + |B|} \quad \text{or} \quad DSC = \frac{2TP}{2TP + FP + FN}$$

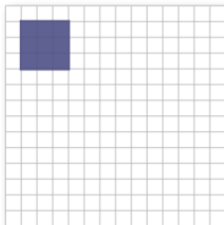
$$IoU = \frac{|A \cap B|}{|A \cup B|} \quad \text{or} \quad IoU = \frac{TP}{TP + FP + FN}$$

- Relation entre les 2 métriques

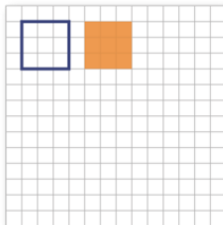
Source : Reinke et al, 2021

# Overlap : exemple

**Reference**



**Prediction 1**

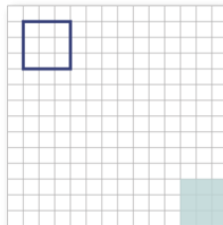


$DSC = 0.0$

$Volume = 9.0$

$HD = 5.0$

**Prediction 2**



$DSC = 0.0$

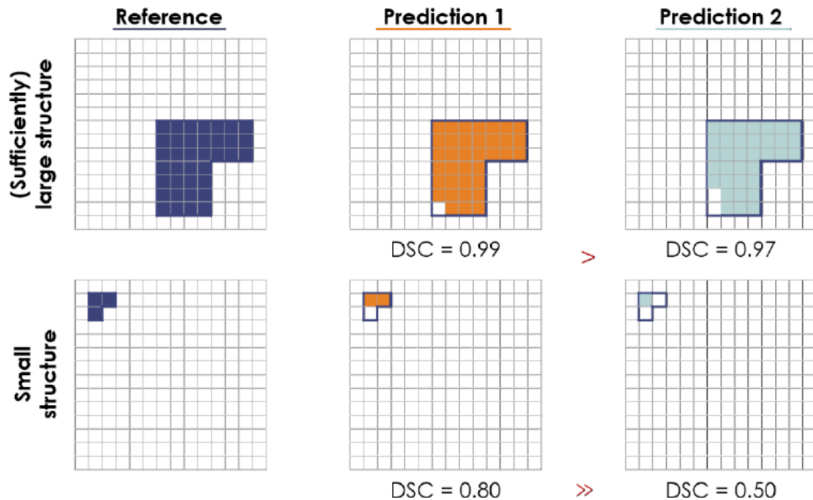
$Volume = 9.0$

$HD = 14.14$

=  
=  
<<

Source : Reinke et al, 2021

# Overlap : exemple

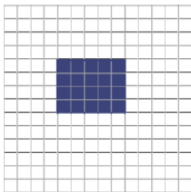


Source : Reinke et al, 2021

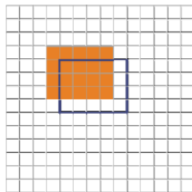
# Overlap : exemple

- Problème : ne prend pas en compte la forme

Reference

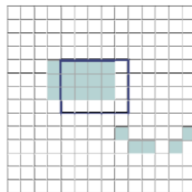


Prediction 1



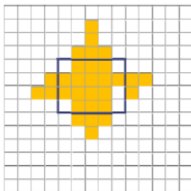
DSC = 0.60

Prediction 2



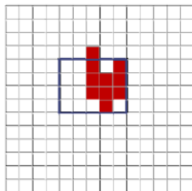
DSC = 0.60

Prediction 3



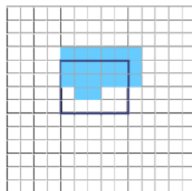
DSC = 0.60

Prediction 4



DSC = 0.60

Prediction 5



DSC = 0.60

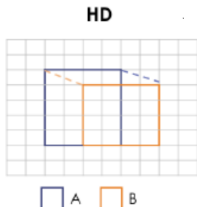
# Qualité de la frontière

- Distance de Hausdorff

$$d_H(X, Y) = \max \left\{ \sup_{x \in X} \inf_{y \in Y} d(x, y), \sup_{y \in Y} \inf_{x \in X} d(x, y) \right\}$$

- Average symmetric surface distance (ASSD) : moins sensible aux outliers que HD

$$ASSD = \frac{\sum_{x \in X} d(x, Y) + \sum_{y \in Y} d(y, X)}{|X| + |Y|}$$



$$HD(A, B) = \max \left\{ \sup_{a \in A} d(a, B), \sup_{b \in B} d(b, A) \right\}$$

$$d(x, Y) = \inf_{y \in Y} d(x, y)$$

Source : Reinke et al, 2021

# Mesures pour la régression

- mesures de performance typiques pour les problèmes de régression
  - RMSE (*Root Mean Square Error*)

$$RMSE(X, h) = \sqrt{\frac{1}{m} \sum_{i=1}^m (h(x^{(i)}) - y^{(i)})^2}$$

- MAE (*Mean Absolute Error*)

$$MAE(X, h) = \frac{1}{m} \sum_{i=1}^m |h(x^{(i)}) - y^{(i)}|$$