Analyse de données

Séance 3 - Préparer un dataset (2/2)

Master MIAS - M1
hadriensalem@gmail.com

Des questions par rapport à la séance précédente ?

Nous avons travaillé sur l’analyse des données

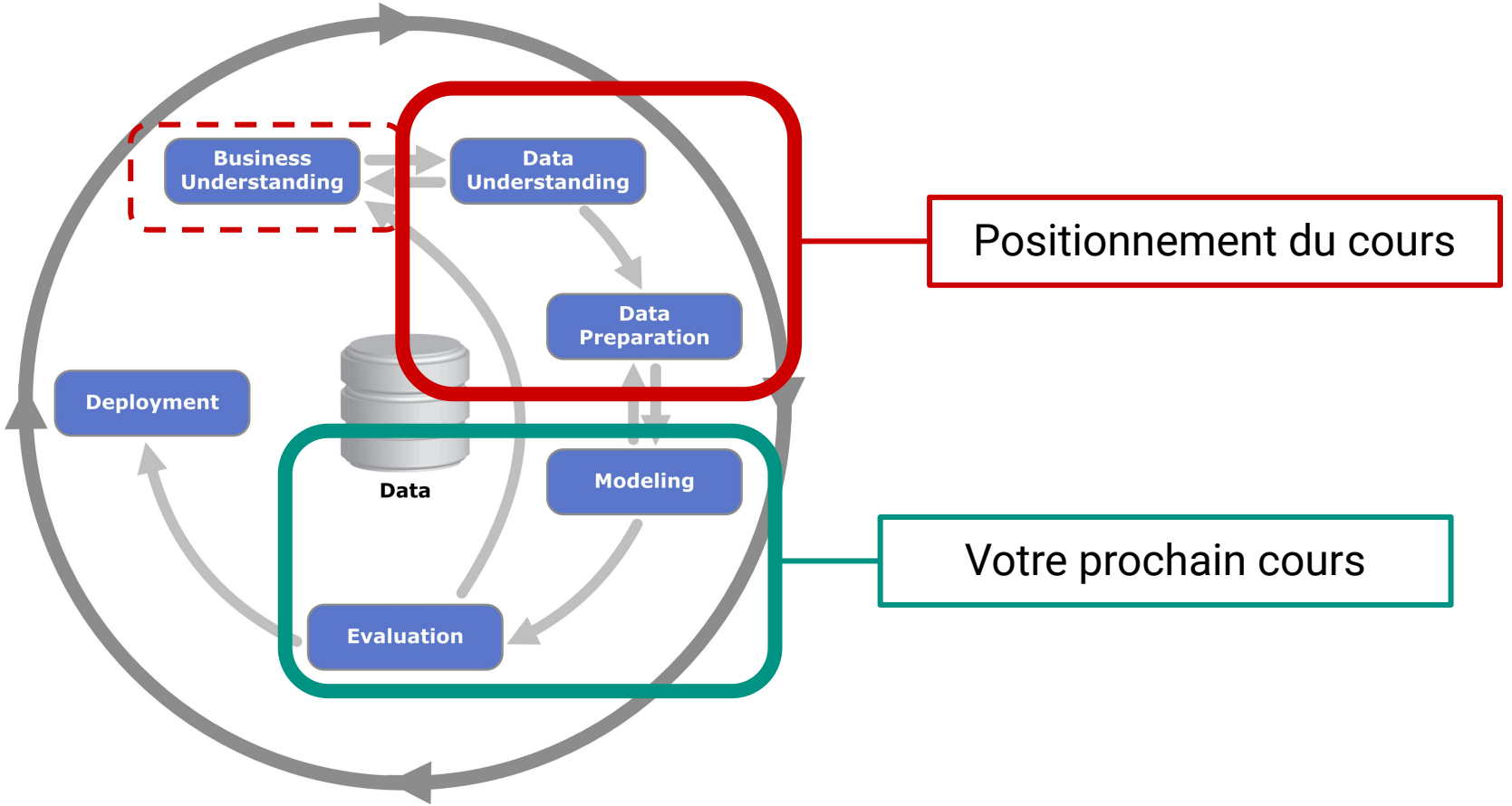
Introduction
Rappels - Positionnement

La méthode CRISP-DM

Cross-**I**ndustry **S**tandard **P**rocess for **D**ata **M**ining

➔ Publiée en 1999
➔ Méthode suivie dans l'industrie
➔ Toujours d'actualité

(Source de l'image: Wikipedia)



(Source de l'image: Wikipedia)

Séance d'aujourd'hui

La dernière fois, nous avons vu :

- Comment standardiser ou normaliser des données
- Différentes techniques pour gérer les données manquantes

(Source de l'image: Wikipedia)

Plan de cours

Séance 1 : Comprendre un dataset

- Étude exploratoire des données
- Visualisation des données

Séance 2 : Préparer un dataset (1/2)

- Overview des types de pré-traitement des data
- Gestion des valeurs manquantes & absurdes

Séance 3 : Préparer un dataset (2/2)

- Introduction à la réduction de dimension
- Typologie des algorithmes de machine learning

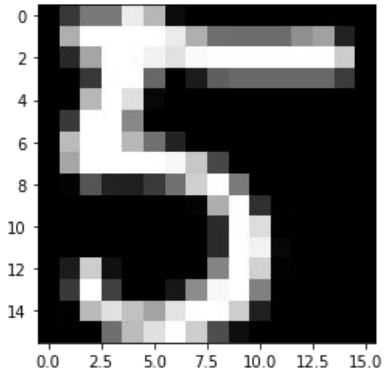
Qu’est-ce que la réduction de dimension ?

L’exemple du dataset MNIST

Exemple du dataset MNIST

Modified National Institute of Standards and Technology database

- Base de données de chiffres écrits à la main
- Chaque chiffre est représenté par le niveau de gris de ses pixels (valeur entre 0 et 255)
- Le dataset original a des images de taille 28x28, mais nous utiliserons des images 16x16 dans le TP

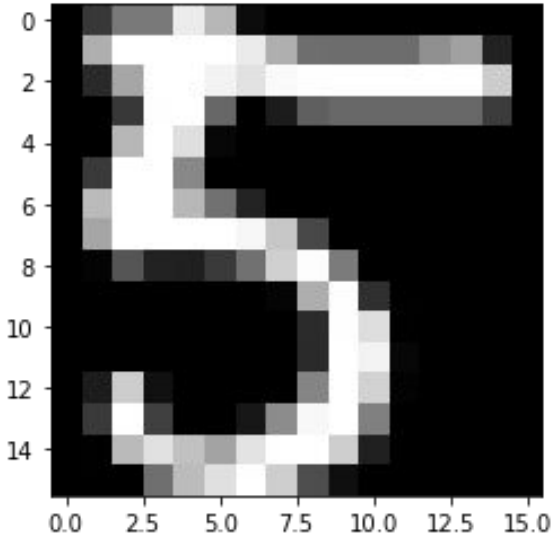


Exemple du dataset MNIST

On voit dans cet exemple que le nombre de features peut facilement devenir très important.

Certains datasets comportes des centaines, voire des milliers de colonnes!

Imaginez essayer de faire apprendre un algorithme sur des images en HD!



16 x 16 = 256 pixels

⇒ 256 dimensions
⇒ 256 colonnes dans le dataframe

Quels intérêts pour la réduction de dimension ?

Quels intérêts pour la réduction de dimension ?

Dans beaucoup de cas, avoir trop de features est un désavantage.

La réduction de dimension permet de :

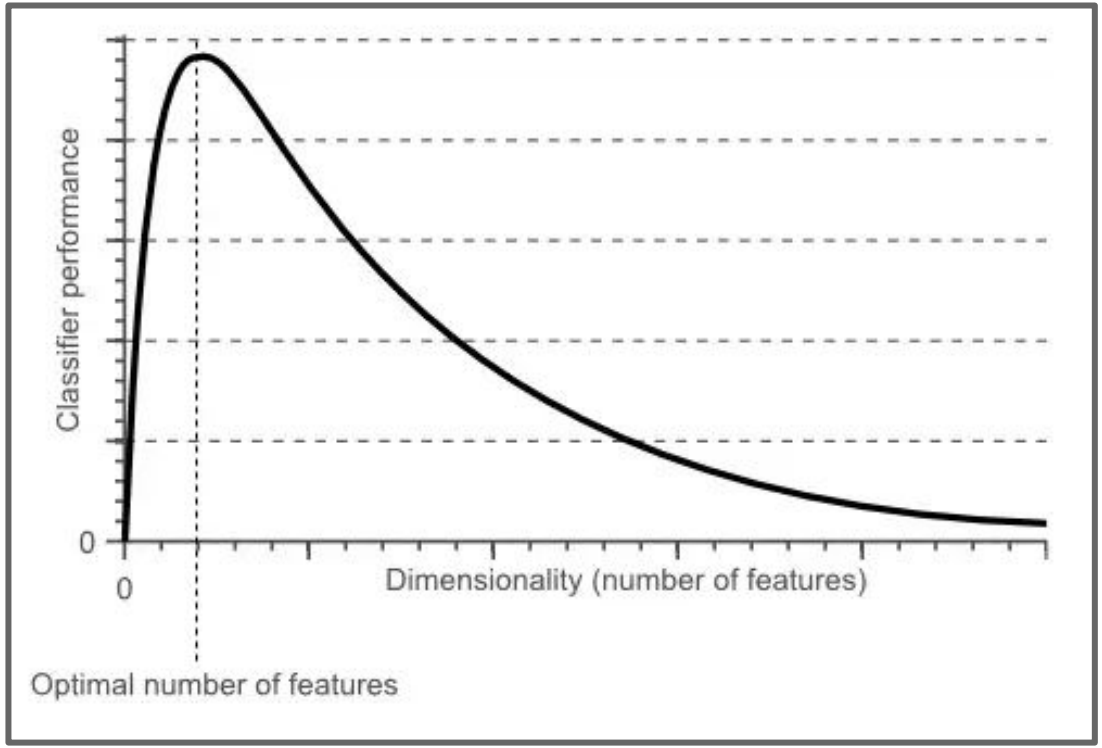
- Réduire le coût machine des calculs
  (et donc diminuer le temps d'entraînement)
- Réduire le bruit dans le dataset
- Visualiser les données (en 2D / 3D)
- Atténuer le "mal de la dimension"
  (*curse of dimensionality)*

⇒ Augmenter la performance des algorithmes de machine learning

Augmenter le nombre de *features* peut augmenter la performance des algorithmes jusquà à certain point.

Cependant, s'il y a "trop" de colonnes et pas assez d'exemples pour distinguer des motifs, l'apprentissage devient plus difficile.

Les effets de "grande dimension" peuvent apparaître dès la dimension 5!



Le mal de la dimension, ou phénomène de Hughes

Source de l'image

Comment réduire la dimension d’un dataset?
Méthode 1 : Feature Selection

Comment sélectionner les features les plus importantes ?

Comment sélectionner les features les plus importantes ?

Il existe plusieurs méthodes de feature selection

- Sélection d'après des connaissances métier (peut être contre-productif, e.g. médecine)
- Suppression des variables à faible variance
- *Feature Importance* à partir d'un premier modèle de machine learning (e.g. Random Forests)
- Choisir de façon itérative à partir d'un modèle de machine learning
  - Méthode *forward* : ajouter des variables
  - Méthode *backward* : retirer des variables
  - Méthode *mixt* : mélange des deux
  - Choix aléatoire
- D'autres méthodes sont proposées par Scikit-learn

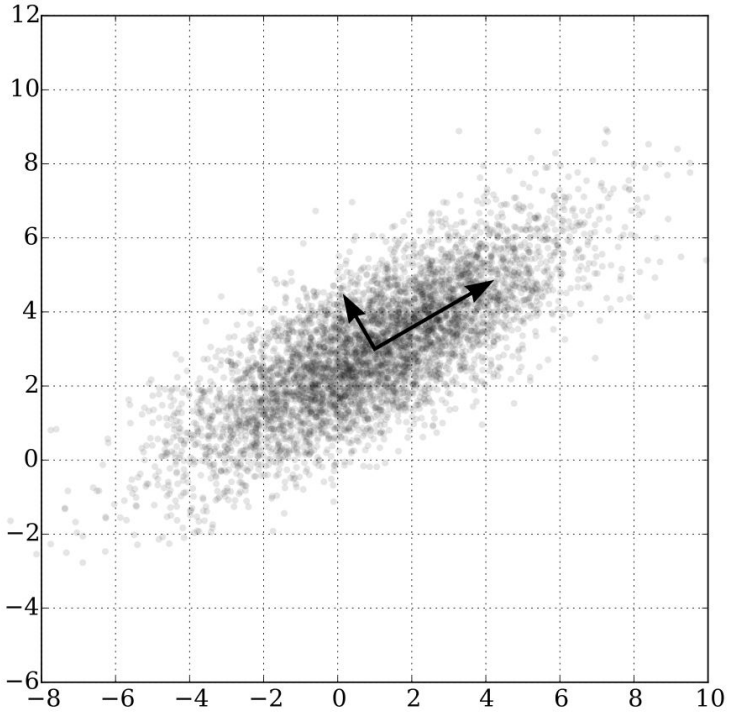
Comment réduire la dimension d’un dataset?
Méthode 2 : Feature extraction

Feature extraction

Créer des colonnes en utilisant des colonnes existantes

Les colonnes ainsi créées doivent être plus significatives
que les colonnes existantes initialement dans le dataset

Il peut s’agir de simples combinaisons linéaires, ou de méthodes plus avancées



Principe intuitif

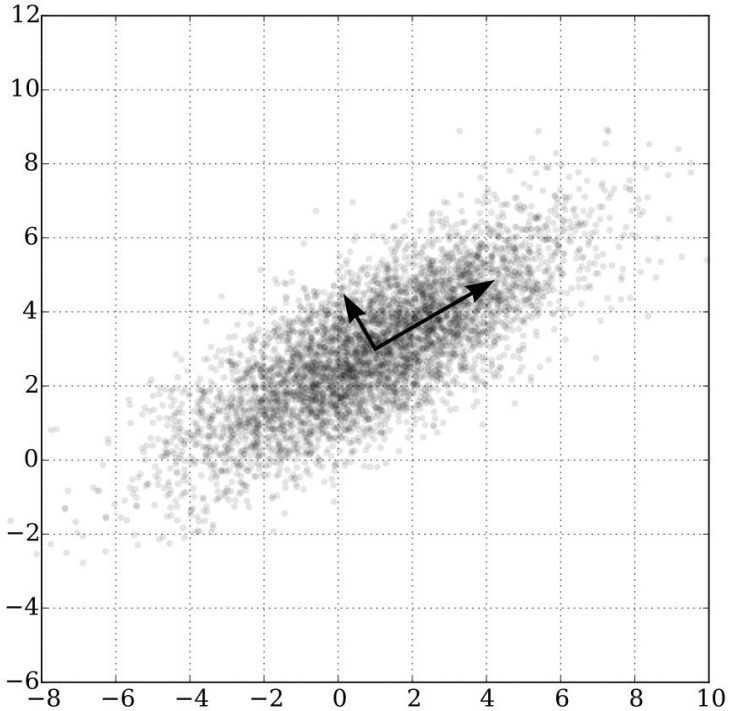
Trouver les directions de plus grande variance

Principe

On cherche à trouver une **nouvelle base** telle que la **variance de chaque composante projetée est maximisée**.

Autrement dit, la PCA n'est **pas une méthode de réduction de dimension en soi**. Toutefois, dans la construction de la nouvelle base, on assure que les premiers vecteurs sont les directions de plus grande variance.

Pour utiliser la PCA comme méthode de réduction de dimension, il suffit **de sélectionner les N premiers vecteurs de la nouvelle base**.



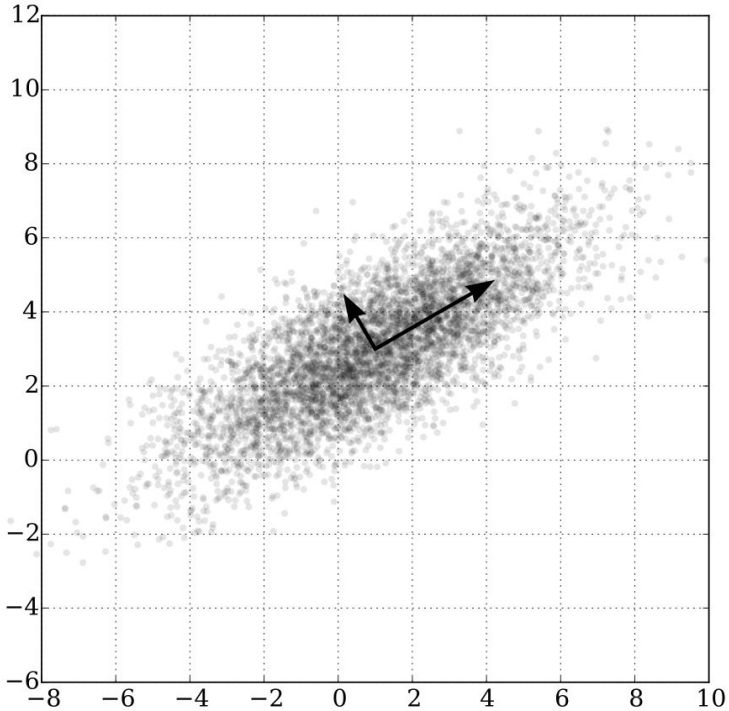
Formellement

On cherche une matrice de changement de base qui minimise l'erreur d'approximation

$$U = \underset{U^T U = 1}{\arg\min} \sum_{n=1}^{N} ||x_n - \underbrace{U U^T x_n}_{x_{\text{approx}}}||^2$$

On peut montrer que c'est équivalent à chercher les **vecteurs propres** de la matrice de covariance.

$$\text{Cov}(X, Y) \equiv \text{E}[(X - \text{E}[X])\,(Y - \text{E}[Y])]$$



Points d’attention pour la PCA

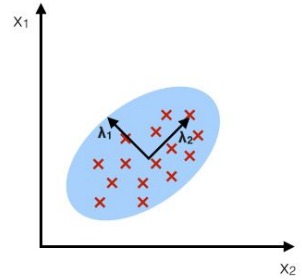
→ Il est indispensable de **standardiser les données** avant de l’appliquer

→ Même si on conserve les dimensions les plus “importantes”, on perd de l’information

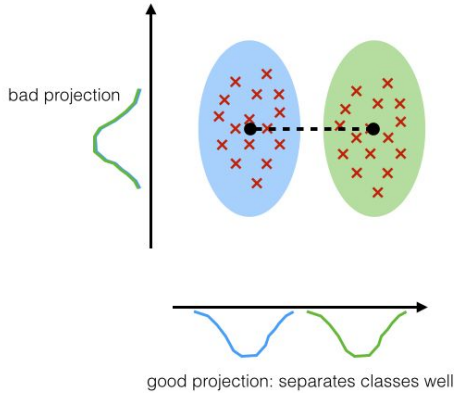
→ Les features ne sont plus interprétables

→ Dans un problème de classification, la PCA ne tient pas compte de la séparation entre classes.
Une autre méthode cherchant à maximiser à la fois les variance intra et inter-classes est l’analyse discriminante linéaire de Fisher.

PCA:
component axes that maximize the variance



LDA:
maximizing the component axes for class-separation



[Source de l’image]

À vos notebooks !

Le TP se trouve au même endroit que pour la dernière séance :

https://github.com/SnowHawkeye/mias-data

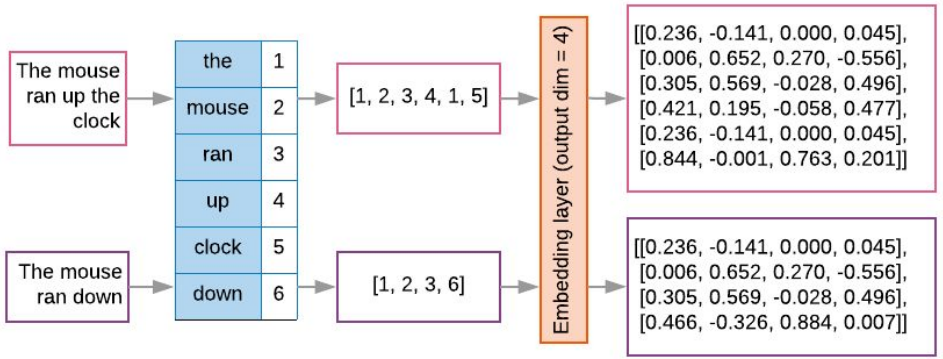
D’autres types de prétraitement

Tout type de traitement pouvant aider les algorithmes à apprendre est pertinent.

Comme nous l’avons vu, le pré-processing des données peut être aussi influent que le choix d’algorithme lui-même.

Le pré-processing peut également inclure

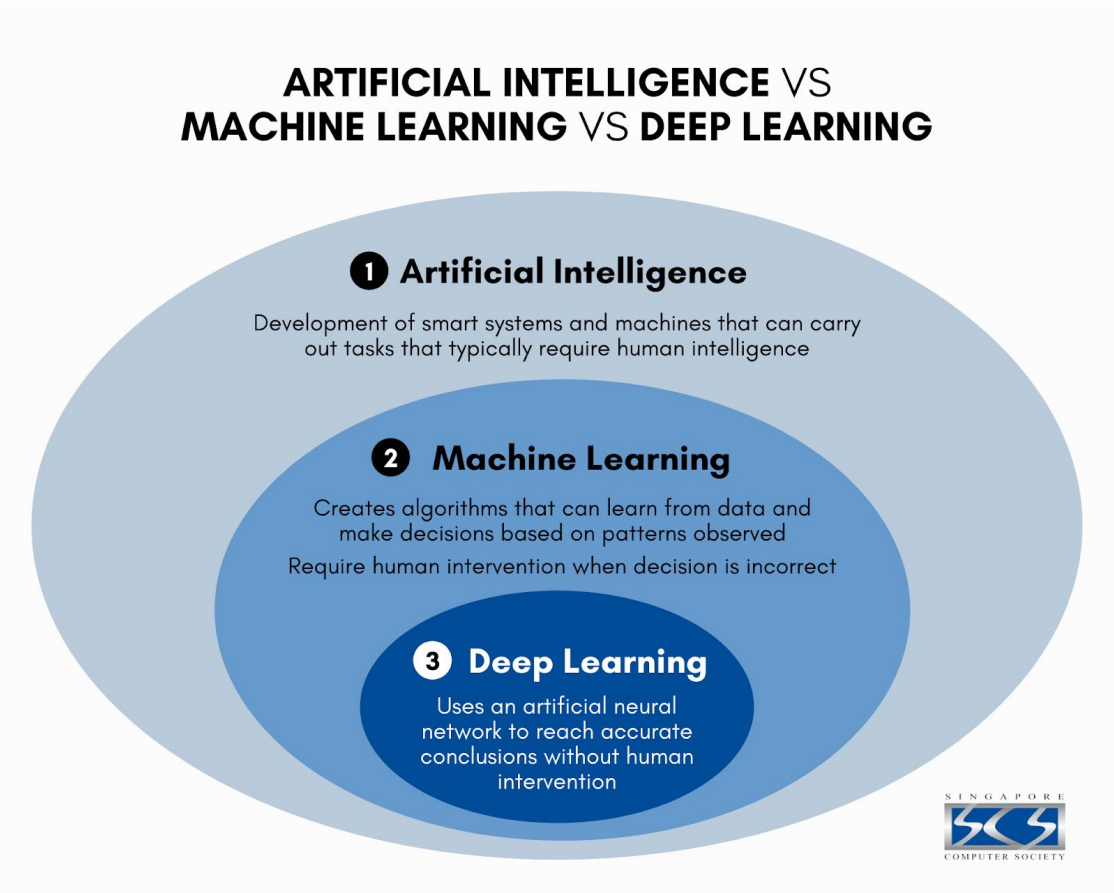
- La vectorisation de variables littérales



Source de l’image

- L’over/undersampling pour traiter des données déséquilibrées (*unbalanced data*)

Introduction au Machine Learning



Attention à différencier les concepts

Source de l'image

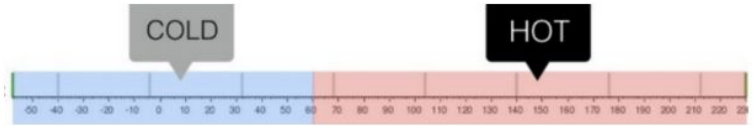
Les grands types de problème

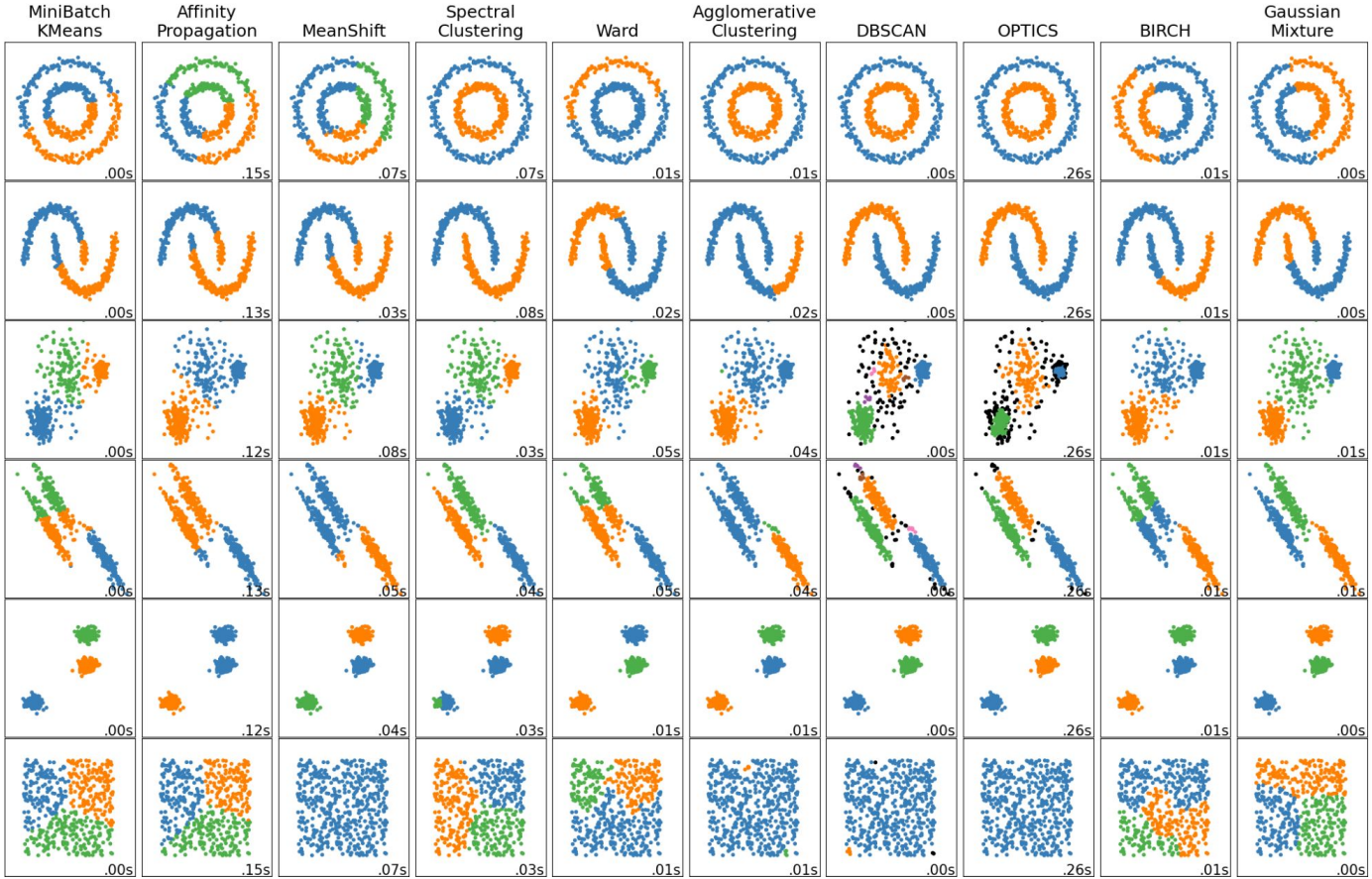
Regression

What will be the temperature tomorrow?

Classification

Will it be hot or cold tomorrow?





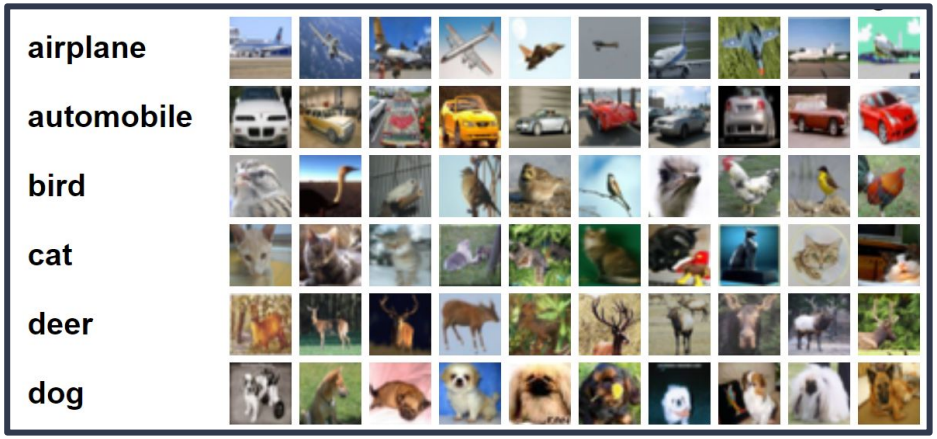
Clustering

Source de l'image

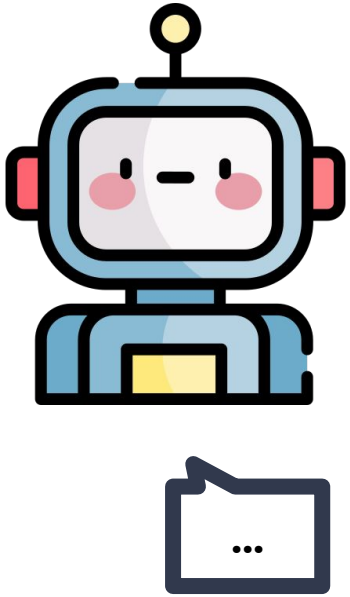
Typologie des algorithmes

On distingue trois grands types d’algorithmes

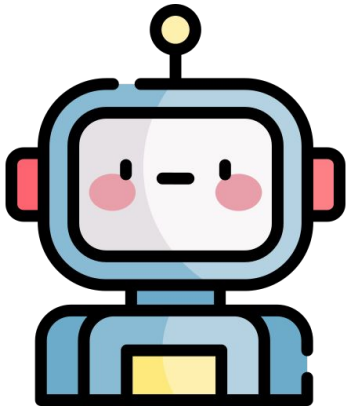
Apprentissage **supervisé**, apprentissage **non supervisé** et apprentissage **par renforcement**



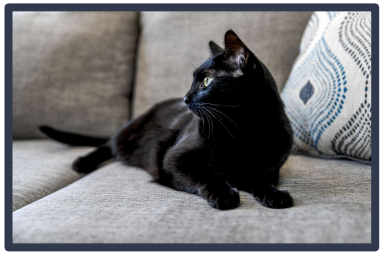
Apprentissage

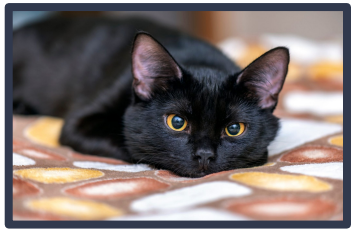


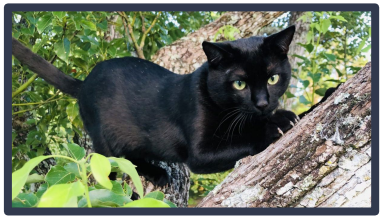
Exemple inconnu



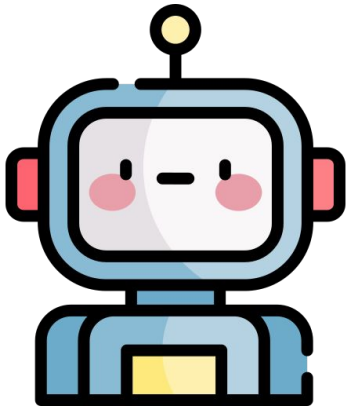
Dog





Apprentissage

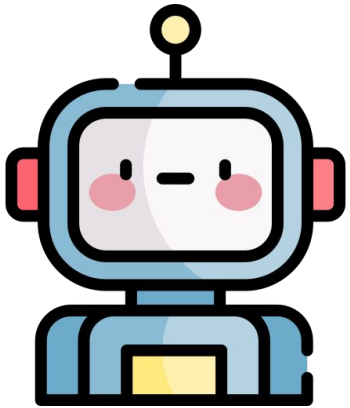


Apprentissage non supervisé



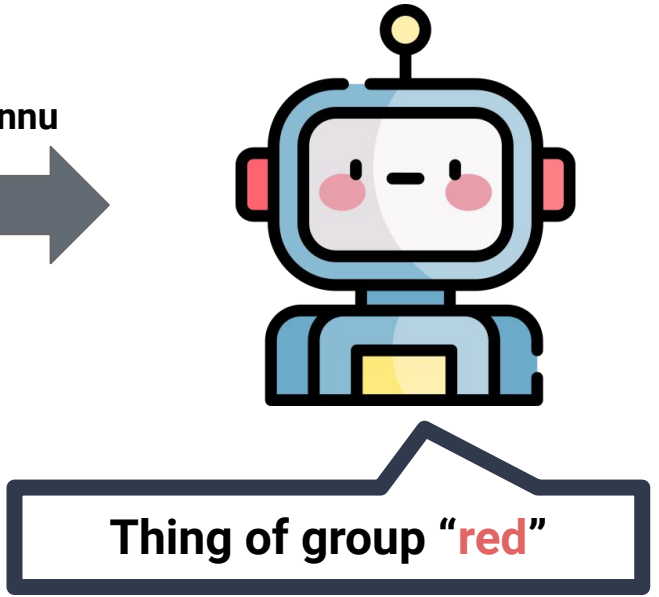


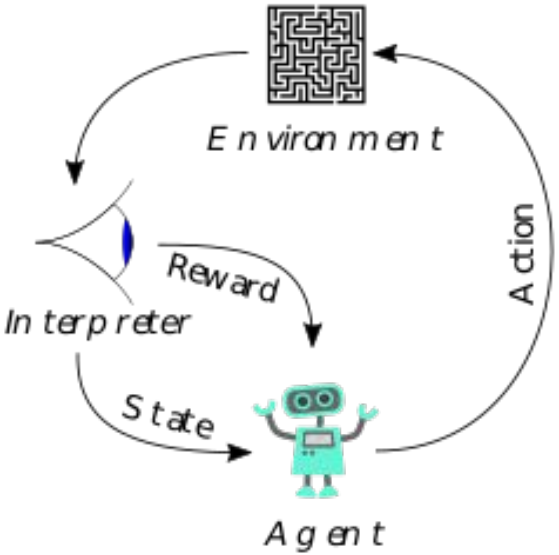

Apprentissage

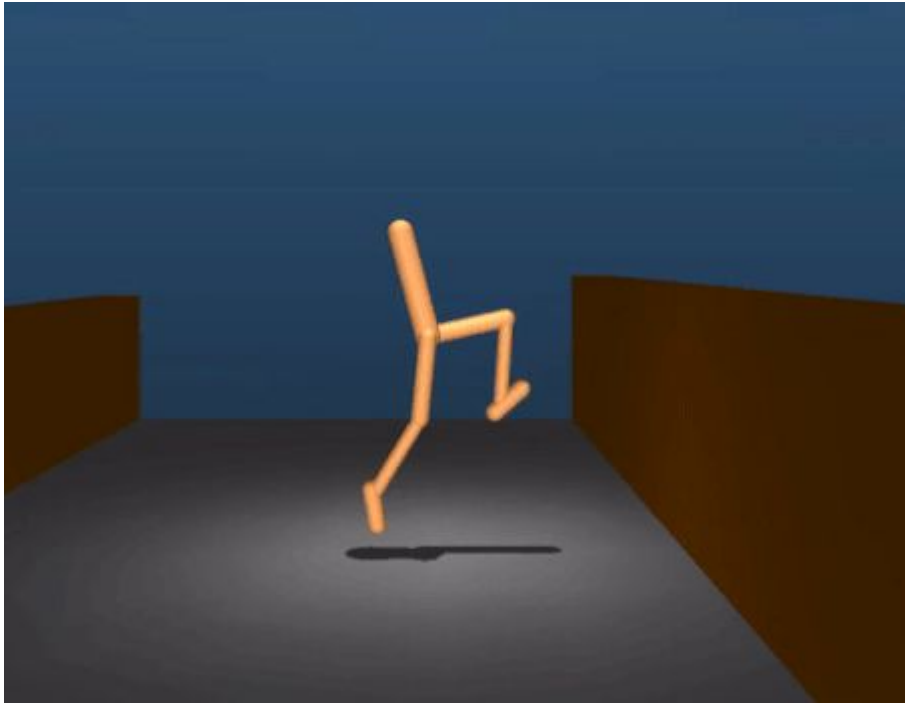


Apprentissage non supervisé

Exemple inconnu




Environment
Reward
Interpreter
Action
State
Agent



Conclusion et perspectives

Conclusion et perspectives

La compréhension et la préparation des données ne sont que le début du travail.

Par ailleurs, l'exploitation des données est un processus cyclique, où on est amené à itérer sur chaque étape.

L'utilisation d'algorithmes s'inscrit dans un processus complexe, et ne représente finalement qu'une partie de la chaîne.