Intelligence Artificielle

Pr. Hiba Chougrad
Année-universitaire: 2023-2024

Plan

1. Introduction générale et Agents Intelligents
2. Logique du premier ordre
3. Machine Learning : Pré-traitement des données
4. Machine Learning : Supervised vs Unsupervised
5. Machine Learning : Construire un bon modèle
6. Machine Learning : Raisonnement probabiliste et réseaux bayésiens
7. Machine Learning: Algorithmes d’apprentissage automatique
8. Machine Learning: Apprentissage par renforcement, vision par ordinateur, NLP, Deep Learning

Plan

1. Introduction générale et Agents Intelligents
2. Logique du premier ordre
3. **Machine Learning : Pré-traitement des données**
4. Machine Learning : Supervised vs Unsupervised
5. Machine Learning : Construire un bon modèle
6. Machine Learning : Raisonnement probabiliste et réseaux bayésiens
7. Machine Learning: Algorithmes d'apprentissage automatique
8. Machine Learning: Apprentissage par renforcement, vision par ordinateur, NLP, Deep Learning

Machine Learning : Pré-traitement des données

Machine Learning

Sommaire

- Qu’est-ce que l’apprentissage automatique?
- Concepts.
- Algorithmes de bases.
  - Regression linéaire/multiple/polynomiale
  - Naïve Bayes
  - Decision Trees
  - K-Nearest Neighbors
  - K-Means
  - Neural Networks (Réseaux de Neurones)

Qu’est-ce que l’apprentissage?

- L’**apprentissage** est la capacité pour un agent intelligent de tirer profit de son **expérience passée** et de ses **observations** dans l’environnement pour **améliorer** **ses performances dans le futur.**

- *Rappel:* modèle PEAS (Performance measure, Environment, Actuators, Sensors)
  - On mesure *l’intelligence* d’un agent intelligent à l’aide d’une **mesure de performance**.

Pourquoi un agent intelligent devrait-il pouvoir apprendre?

1) **Impossible de prévoir toutes les situations possibles** (ex: un robot qui navigue dans des labyrinthes doit apprendre la configuration de chaque labyrinthe)

2) **Impossible d'anticiper les changements au fil du temps** (ex: systèmes de détection de fraudes, de pourriels, prédiction du cours boursier, etc.)

3) **Nous ne savons pas trop comment résoudre certains problèmes** (ex: reconnaissance de visage).

Introduction

- **Machine Learning** qui est une branche de **l'Intelligence Artificielle**, appelée aussi **Apprentissage Automatique** est un outil essentiel de la science des données (**Data Science**).

- Le ***Machine Learning*** a grandement fait parlé de lui ces dernières années surtout avec les applications qu'il a permit de construire.

- ***Machine Learning*** représente un ensemble de techniques puissantes permettant de créer des modèles **prédictifs** à partir de **données**, **sans avoir été explicitement programmées.**

- Nous allons voir les notions fondamentales du Machine Learning:
  1. Les caractéristiques du Machine Learning
  2. Les différentes familles d'algorithmes et leurs applications pratiques
  3. C'est quoi l'apprentissage? et comment l'implémenter concrètement?

Formulation du problème d’apprentissage

- Un agent **apprend** s'il améliore sa performance sur des tâches futures avec l'**expérience**.

- On va se concentrer sur un problème d'apprentissage simple (supervisé) mais ayant beaucoup d'applications:

> « Etant donnée une collection de paires (**entrées** , **sorties**) appelées **exemples d'apprentissage**, comment apprendre **une fonction** qui peut prédire correctement une sortie étant donnée une nouvelle entrée »

Formes d’apprentissage

- Il existe plusieurs sortes de problèmes d'apprentissage, qui se distinguent par la nature de la supervision offertes par nos **données**.

- **Apprentissage supervisé**: sortie désirée (*cible* ou *target*) est fournie explicitement par les données
  (ex: reconnaissance de caractères a l'aide d'un ensemble de paires (images, identité du caractère))

- **Apprentissage non-supervisé**: les données ne fournissent pas de signal explicite et le modèle doit extraire de l'information uniquement a partir de la structure des entrées
  (ex: identifier différents thèmes de livres en regroupant les livres similaires "*clustering*")

- **Apprentissage par renforcement**: le signal d'apprentissage correspond seulement a des récompenses et punitions
  (ex: est-ce que le modèle a gagné la partie d'échec (1) ou pas (-1))

Et plusieurs autres …

Représentation des entrées et sorties

- L'**entrée** doit généralement être **prétraitée** et mise dans une **représentation factorisé** (standardisé, normalisé).

- Par exemple, l'entrée peut être un vecteur ***X*** de ***n*** valeurs discrètes ou réelles.

- La sortie ***y*** (i.e. cible ou target), peut être :
  - une classe (problème de classification);
    - Ex: {ensoleillé, nuageux, pluie}.
    - Cas particulier : classification binaire.
  - une valeur réelle (problème de régression).

Représentation des données: Exemple

Exemple: **Prediction de la taille**

- Etant donnée la dataset suivante:

- Peut-on prédire la taille d'une personne âgée de 8 ans ?

- L'entrée x dans cet exemple est l'age $x_i \in \{5,7,11,18\}$ et $y_i \in \{47,52,61,71\}$
- Soit les pairs {(5,47),(7,52),(11,61),(18,71)}

- On veut trouver la valeur de y associée a une nouvelle entrée x=8
- Comment programmer la fonction f ?

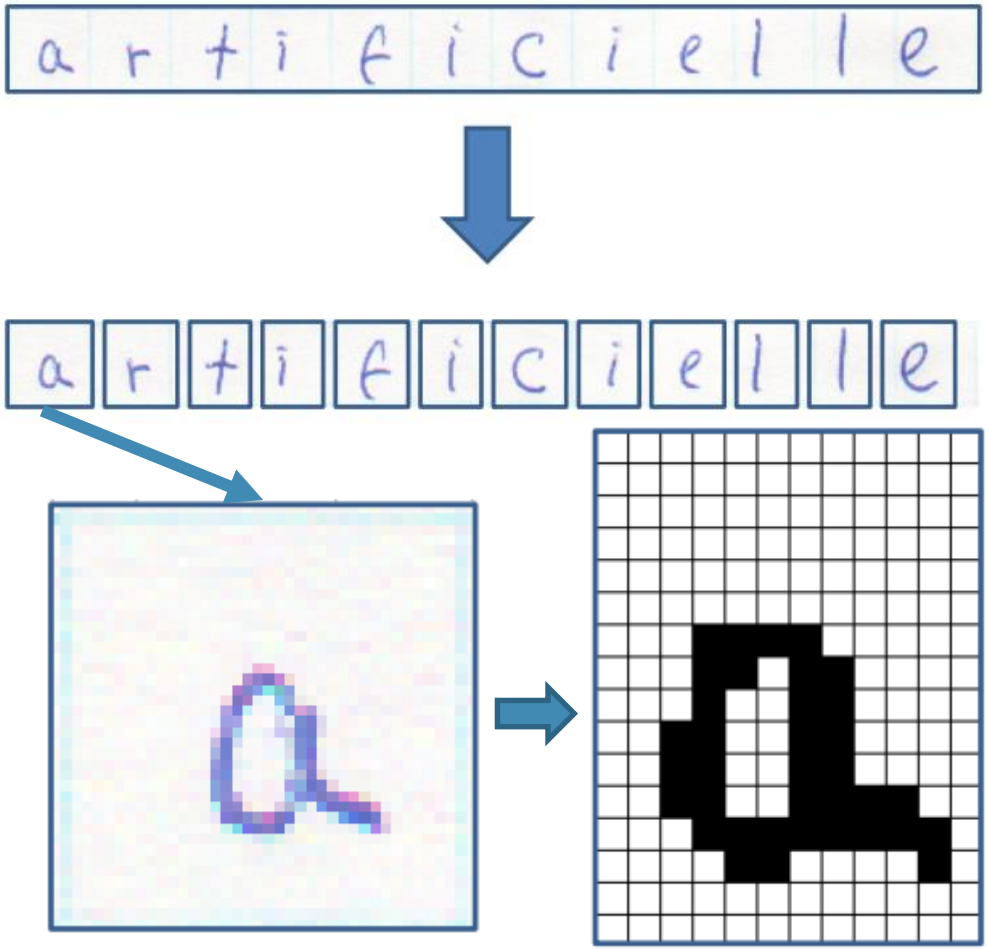
$$f: x \rightarrow y$$

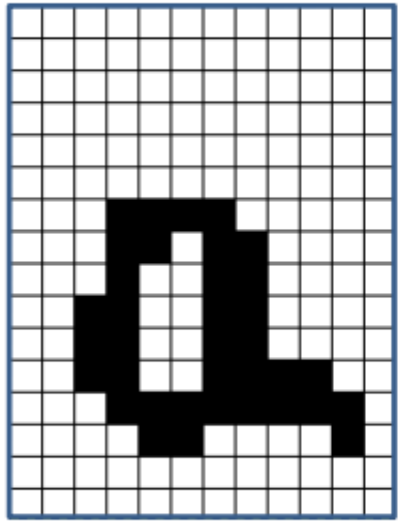
Dataset

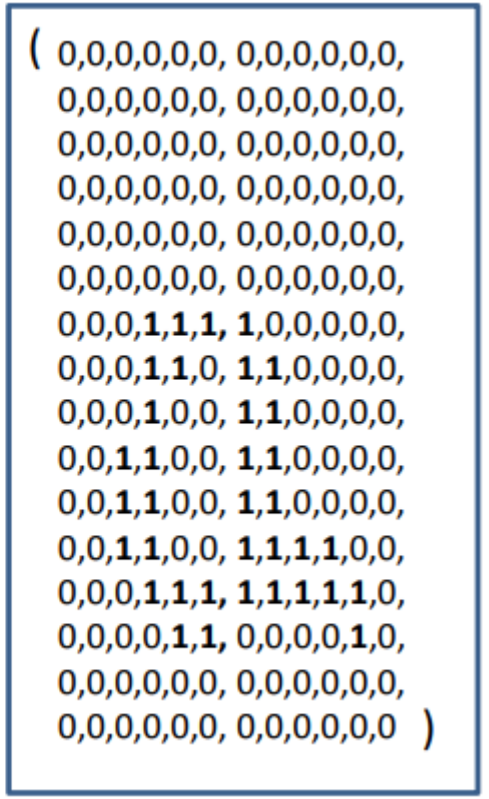
Age	Height (inches)
5	47
7	52
11	61
18	71
8	?

Représentation des données: Exemple

- Exemple: Reconnaissance de caractères OCR



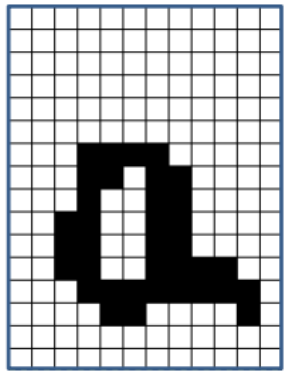




Représentation des données

Représentation des données: Exemple

- Entrée : un vecteur de 12 x 16 = 192 valeurs (dans {0,1} ou [0,1]), chacune représentant un pixel en entrée.



(0,0,0,0,0,0, 0,0,0,0,0,0,
0,0,0,0,0,0, 0,0,0,0,0,0,
0,0,0,0,0,0, 0,0,0,0,0,0,
0,0,0,0,0,0, 0,0,0,0,0,0,
0,0,0,0,0,0, 0,0,0,0,0,0,
0,0,0,0,0,0, 0,0,0,0,0,0,
0,0,0,**1,1,1,** 1,0,0,0,0,0,
0,0,0,**1,1**,0, **1,1**,0,0,0,0,
0,0,0,**1**,0,0, **1,1**,0,0,0,0,
0,0,**1,1**,0,0, **1,1**,0,0,0,0,
0,0,**1,1**,0,0, **1,1**,0,0,0,0,
0,0,**1,1**,0,0, **1,1,1,1**,0,0,
0,0,0,**1,1,1,** 1,1,1,1,1,0,
0,0,0,0,**1,1,** 0,0,0,0,**1**,0,
0,0,0,0,0,0, 0,0,0,0,0,0,
0,0,0,0,0,0, 0,0,0,0,0,0)

- Sortie: un caractère : {a, b, c, d, …, x, y, z, […], 0, 1, 2, … 8, 9}
- Comment programmer la fonction f ?

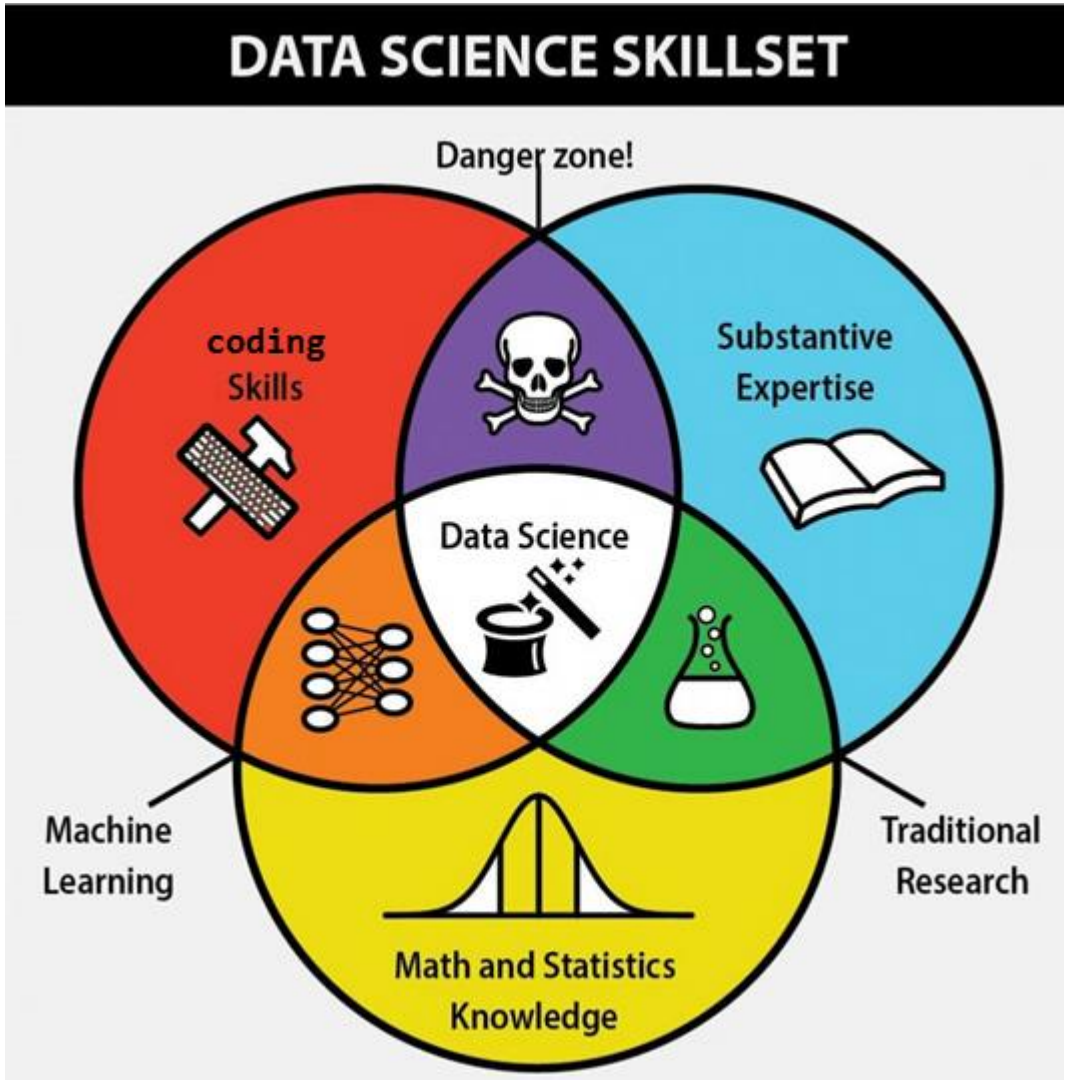
$$f : \{0,1\}^{12\times16} \rightarrow \{a,b,c,d,...,y,z,0,1,...,9\}$$

Machine Learning et Data Science

- Le **Machine Learning** (ou *apprentissage automatique*) et la **data science** (ou *science des données* en) sont deux mots très en vogue lorsque l'on parle de la révolution **Big Data**, de prédiction des comportements ou tout simplement de la transformation numérique des entreprises.

- Le métier de **Data Scientist** est apparu pour trois raisons principales:
  1. l'explosion de la quantité de données produites et collectées par les humains
  2. l'amélioration et l'accessibilité plus grande des algorithmes de **Machine Learning**
  3. l'augmentation exponentielle des capacités de calcul des ordinateurs

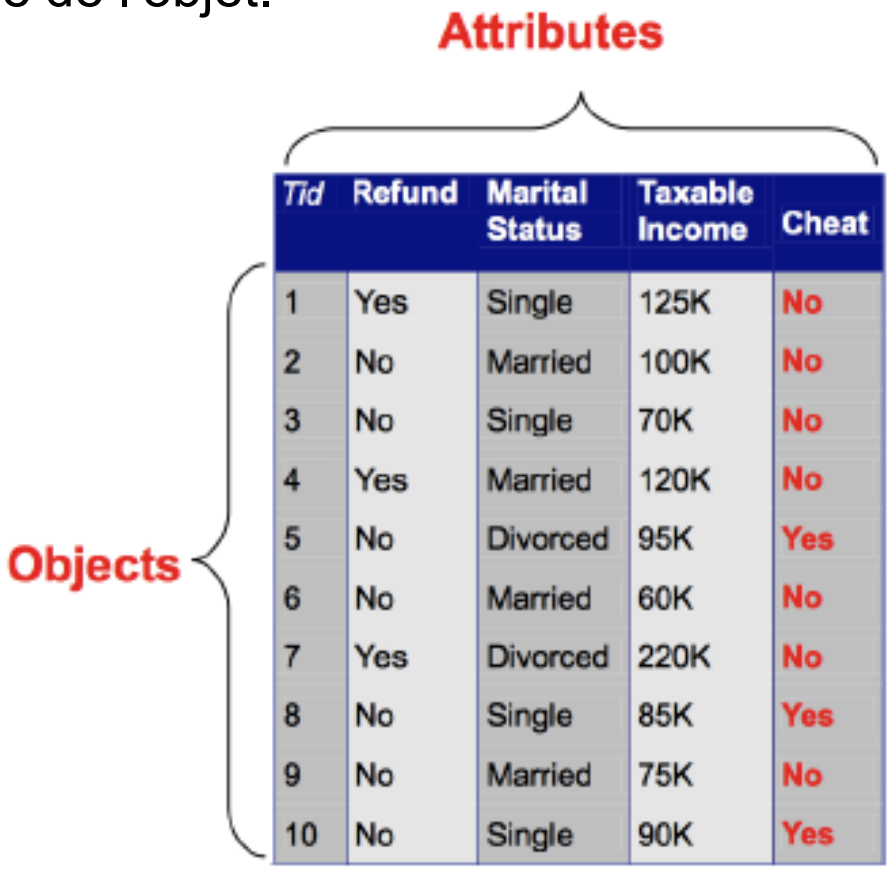
Machine Learning et Data Science



Les données?

Les **données** peuvent être vues comme une collection d'instances d'objets (enregistrements) et leurs attributs.

- Un attribut est une propriété et ou une caractéristique de l'objet.
- Un ensemble d'attributs décrit un objet



Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Les données?

Types

- **Quantitative** (numérique, exprime une quantité)
  - Discrète (ex : nombre d'étudiants dans un cours)
  - Continue (ex : longueur)
- **Qualitative**
  - Variable ordinale (classement à un concours, échelle de satisfaction client)
  - Variable catégorique (couleur de yeux, diplôme obtenu,…)

Modalités

- Les modalités d'un attribut sont l'ensemble des valeurs qu'il prend dans les données.
- Ex : les modalités de l'attribut note sont {0;1;2;…;20} et les modalités de l'attribut couleur sont {bleu,vert,noir,...}

Attributs

- L’attribut est une caractéristique de l’ensemble de données. Il a un type et des valeurs contraintes par son type.

- Le type d’un attribut peut être:
  - **Catégorique** (Categorical)
  - **Numérique** (Numerical)
  - **Ordinal**

Attributs à valeurs Catégoriques

- Données qualitatives sans signification mathématique (Région de résidence, Catégorie de produit, Parti politique, …)
- Appelée Catégorique ou Nominale. Les valeurs sont des symboles (des noms)
- Exemple:
  - Les valeurs de l'attribut *Temps* sont *{Ensoleillé, Pluvieux, Neigeux, Gris}*

- Aucune relation (ordre ou distance) entre les nominaux n'existe.
- Seuls des tests d'égalité peuvent être exécutés
- Exemple de règle:
  - If *Temps* = *Pluvieux* Then *Jouer_Match* = *No*

- On peut attribuer des nombres aux catégories afin de les représenter de manière plus compacte, mais **les nombres n'ont pas de signification mathématique**.
- Cas particulier: Les attributs de type booléen

Attributs à valeurs Numériques

- Représente une mesure quantitative (Taille des personnes, temps de chargement des pages, cours d'actions en bourse,etc)
- Exemple:
  - L'attribut *Longueur_d'une_pétale_de_fleur* est décrite par une valeur $\epsilon \mathbb{R}$
  - L'attribut *Température* est numérique exprimée en degrés Celsius ou Fahrenheit

- Données discrètes: Entières et présentent le compte d'un événement.
  - Combien d'achats un client a-t-il fait en un an?
  - Combien de fois j'ai eu «Pile»?
- Données continues: Possèdent un nombre infini de valeurs possibles.
  - Combien de temps a-t-il fallu pour passer à la caisse?
  - Combien de pluie est tombée un jour donné?

Attributs à valeurs Ordinales

- Un mélange de numérique et de catégorique
- Ce sont des données catégoriques ayant une signification mathématique.

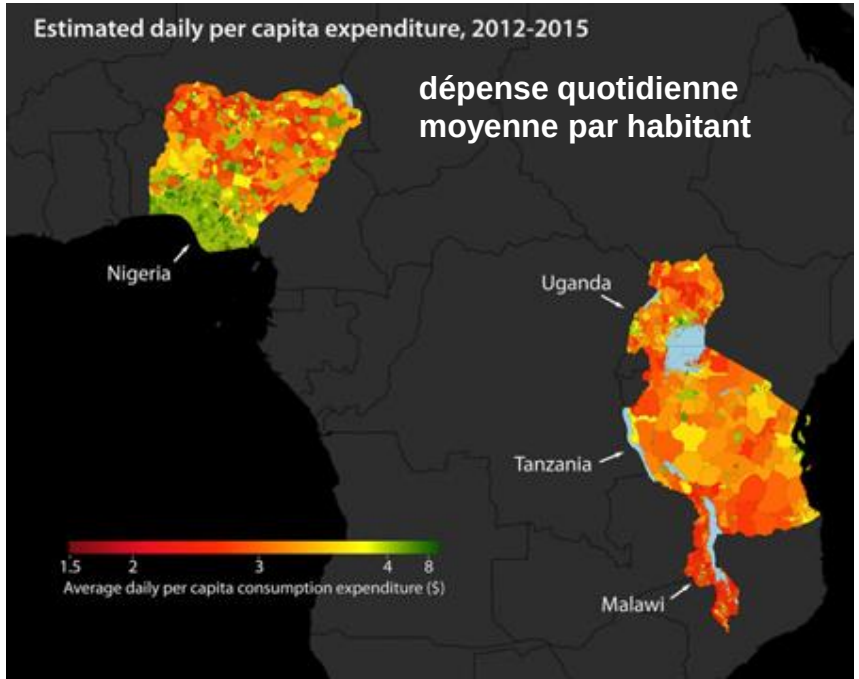
- Exemple: Evaluations de films sur une échelle de 1 à 5.
  - Les notes doivent être 1, 2, 3, 4 ou 5
  - Mais ces valeurs ont une signification mathématique; 1 signifie que c'est un film pire qu'un qui a une note de 2.

1ère étape : trouvez les données

- Pour s’attaquer à un problème, la première chose à faire est d'explorer toutes les pistes possibles pour **récupérer les données**.
- En effet, **les données** constituent **l'expérience**, ce sont les exemples que vous allez fournir à votre algorithme de **Machine Learning** afin qu'il puisse apprendre et devenir plus performant.
- Les données destinées à alimenter un algorithme de Machine Learning, sont appelées **Dataset** (jeu de données)

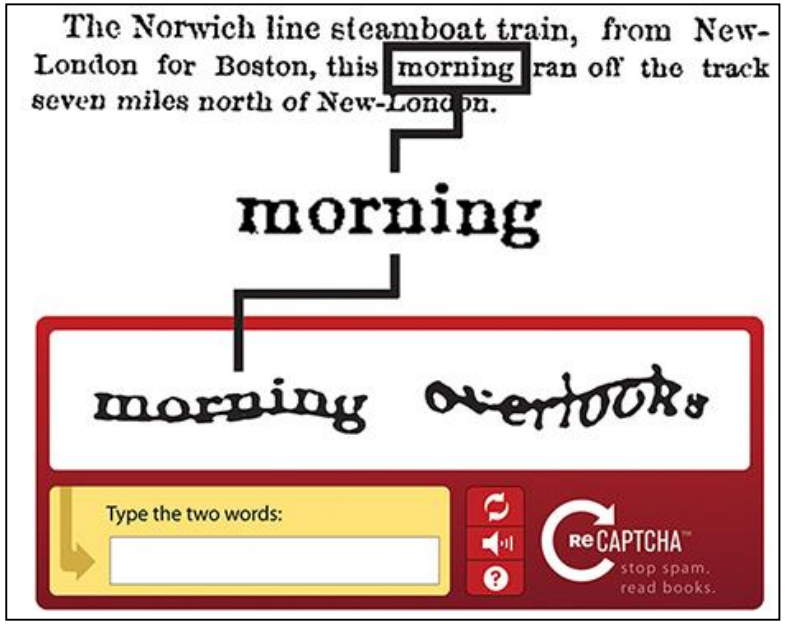
Exemples d'utilisation originale des donnés

- **Les images satellites pour évaluer le niveau de pauvreté:** Des chercheurs ont utilisé le Machine Learning pour pouvoir cartographier les zones de pauvreté de manière automatique, simplement à partir d'images satellites!
- Un des chercheurs a expliqué **« Notre machine a appris à remarquer des indices, , comme l'état des routes, les zones urbaines, les voies navigables ainsi que les hectares d'agriculture. »** À partir de ces données, l'algorithme est capable de déterminer le niveau de richesses.



Exemples d’utilisation originale des donnés

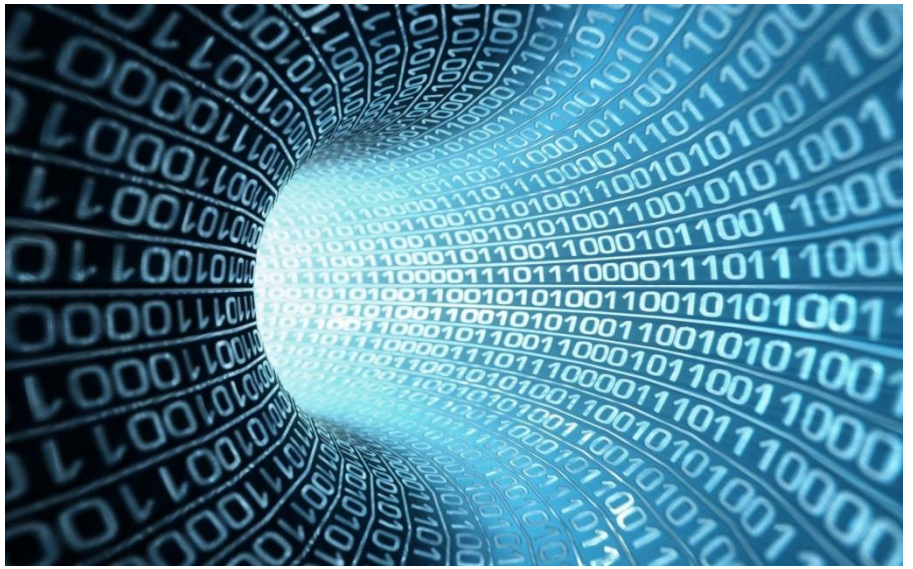
- **Les CAPTCHAs pour la digitalisation automatique de livres:** Luis von Ahn, entrepreneur et chercheur, a créé un célèbre système de reCAPTCHA qui permettait à la fois aux sites web de valider que les formulaires étaient bien remplis par des humains, et qui alimentait en même temps la base de données d'un algorithme de digitalisation de livres.
- Grâce aux nombreux exemples renseignés directement par des humains, l'algorithme a fini par avoir suffisamment de **données d'exemples** pour réussir ensuite seul à retranscrire en texte des images scannées de livres, avec un taux d'erreur très faible.



Les données?

L'idée est que **plus vous aurez une bonne compréhension de vos données, plus vous serez en mesure de bien les utiliser lors de la phase d'entraînement de votre modèle d'apprentissage**.

Ci-après vous aurez une vue d'ensemble du type de données habituelles rencontrées en Machine Learning.



Les bases de données

Les bases de données constituent la source principale de récupération de données par les Data Scientists. Il existe plusieurs technologies (de Hadoop à SQL) pour assurer la récupération et le stockage de ces données.

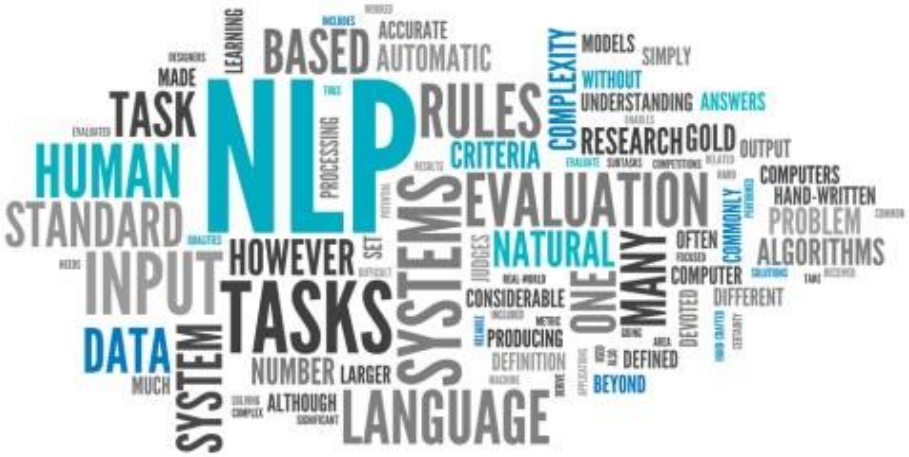
Ces bases de données peuvent comprendre différents types d'information, une bonne partie généralement spécifiques à l'activité de l'entreprise.

À titre d'exemple et de manière non-exhaustive :

- les logs d'un serveur web
- le catalogue produit d'un site de e-commerce
- les transactions bancaires
- les comportements des utilisateurs d'un site
- ...
-

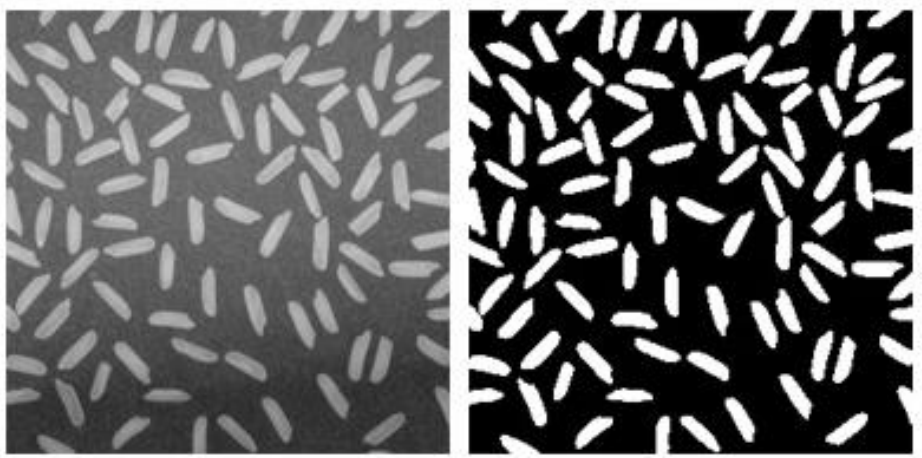
Les données brutes: le texte

- Le texte, rédigé en langage naturel (humain), est ainsi une autre source de données principale pour le travail de Data Scientist. Cela comporte tous les types de texte auxquels on peut penser naturellement (articles, livres, messages,… etc), mais aussi d'autres types de textes tels que du code HTML ou encore des séquences d'ADN.
- Le traitement du texte (appelé NLP **N**atural **L**anguage **P**rocessing) constitue un domaine de recherche à part entière.



Les données brutes: Les images (et vidéos)

- Les images sont aussi une des sources de captation de l'environnement souvent utile pour la résolution de nombreuses problématiques. Beaucoup d'entreprises, hôpitaux et centres de recherches ont des banques d'images à traiter pour les classer par type ou autre.
- Le traitement des images et vidéos constitue un domaine de recherche à part entière (machine vision).



Un exemple de pré-traitement d'image appelé seuillage, qui permet de simplifier ensuite l'apprentissage d'un modèle statistique (crédits : The MathWorks, Inc.)

Les données brutes: IoT (Internet des objet)

- Les objets connectés sont une autre source de données brutes, qui récupèrent un grand nombre de données grâce à leurs capteurs.
- Un bon exemple est l'entreprise ***Nest*** qui a utilisé la Data Science pour créer un **thermostat intelligent** qui optimise la consommation d'électricité en surveillant à la fois la température, la présence des habitants, etc.

2ème étape: prétraitement des données

- Le **traitement** et la **compréhension** des données est probablement la partie la plus importante de l'apprentissage automatique.

- Il ne sera pas efficace de choisir au hasard un algorithme et d'y jeter vos données. Chaque algorithme est différent en termes de type de données et de paramètre de problème pour lequel il fonctionne le mieux.

- Il est nécessaire de **comprendre** ce qui se passe dans la **Dataset** avant de commencer à créer un modèle.

2ème étape: prétraitement des données

- Il est important d'avoir **une représentation des données d'entrée** qu'un ordinateur peut comprendre.

- Il est souvent utile de considérer les données comme **un tableau**.

  C.-à-d., chaque point de données sur lequel vous souhaitez raisonner (chaque e-mail, chaque client, chaque transaction) est **une ligne,** et chaque attribut qui décrit ce point de données (par exemple, l'âge d'un client ou le montant ou l'emplacement d'une transaction) est **un colonne**.

- En Machine Learning, chaque entité ou ligne est un échantillon (i.e. **data point**), tandis que les colonnes contiennent les attributs qui décrivent ces entités (i.e. **features**).

2ème étape: prétraitement des données

- Algorithmes d'apprentissage sont sensibles aux valeurs d'entrées.
- On a rarement accès à des données bien formatées et complètes, prêtes à être utilisées telles quelles.

- L'étape de **prétraitement** consiste à:
  - Nettoyer les données (corrections des doublons, des erreurs de saisie, …)
  - Contrôle sur l'intégrité des domaines de valeurs : détection des valeurs aberrantes(Outliers).
  - Détection des informations manquantes(Missing Values)
  - Echelles des variables doivent être comparables : Normalisation / Standardisation
  - Enrichissement des données

Ajustement d’echelle (Normalisation)

- La **Normalisation** permet d'ajuster une série de valeurs (représentant typiquement un ensemble de mesures, *par exemple, une variable stockant les tailles de personnes, en cm*) suivant une fonction de transformation (i.e. une mise à l’échelle) pour les rendre comparables avec quelques points de référence spécifiques.
- La normalisation est la mise à l’échelle pour avoir un petit intervalle spécifié.
  - **Normalisation** (Min-Max)
  - **Standardisation** (Z-score)

Rappel: (Min, Max, Mean, StdDev)

- **Minimum (Min):** la plus petite valeur des valeurs enregistrées à un moment donné.
- **Maximum (Max):** la plus grande valeur des valeurs enregistré que l'on retrouve.
- **Moyenne (Mean):** la somme des valeurs numériques divisée par le *nombre* de ces valeurs numériques.

$$\overline{X} = \frac{1}{n} \sum_{i=1}^{n} x_i \quad avec \quad x_i \epsilon X \quad et \quad i = 1, \dots, n$$

- **Variance:** La variance (σ^2) mesure la dispersion d'un ensemble de valeurs autour de leur moyenne.

$$V = \sigma^2 = \frac{1}{n} \sum_{i=1}^{n} (x_i - \overline{X})^2$$

- **Ecart-type (Standard deviation):** L'écart-type σ est que la racine carrée de la variance.

$$\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (x_i - \overline{X})^2}$$

Mean

Mean : La somme des valeurs numériques divisée par le *nombre* de ces valeurs numériques.

$$\overline{X} = \frac{1}{n} \sum_{i=1}^{n} x_i \qquad avec \quad x_i \epsilon X \quad et \quad i = 1, \dots, n$$

Exemple:

- Nombre d'enfants dans chaque maison de mon quartier:

  {0, 2, 3, 2, 1, 0, 0, 2, 0}

- Le MEAN (moyenne) est (0+2+3+2+1+0+0+2+0) / 9 = **1.11**

Variance

- La **variance** ($\boldsymbol{\sigma^2}$) est simplement la moyenne des différences (valeur et moyenne) au carré.

$$\boldsymbol{V = \sigma^2 = \frac{1}{n} \sum_{i=1}^{n} (x_i - \bar{X})^2}$$

- **Exemple:** C'est quoi la variance de cette Dataset {1, 4, 5, 4, 8}?

- Premièrement on calcule le Mean:

  (1+4+5+4+8)/5 = 4.4

- Deuxièmes on calcule différences (valeur et moyenne) :

  (-3.4, -0.4, 0.6, -0.4, 3.6)

- On calcule le carré des différences:

  (11.56, 0.16, 0.36, 0.16, 12.96)

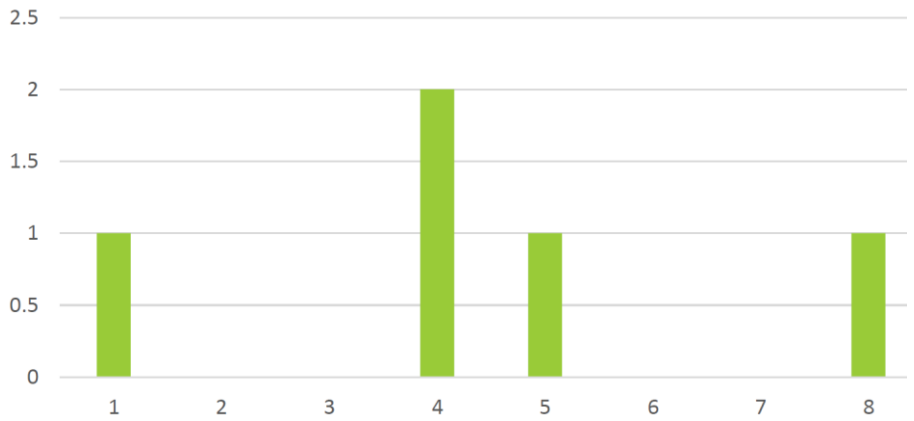
- On calcule la moyenne des différences au carré:

  $\boldsymbol{\sigma^2}$= (11.56 + 0.16 + 0.36 + 0.16 + 12.96) / 5 = 5.04

Standard Deviation σ

- **Standard Deviation** (L'écart type) σ est la racine carrée de la variance.

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{X})^2}$$



- **Exemple:**

  C'est quoi la variance de cette Dataset {1, 4, 5, 4, 8}?

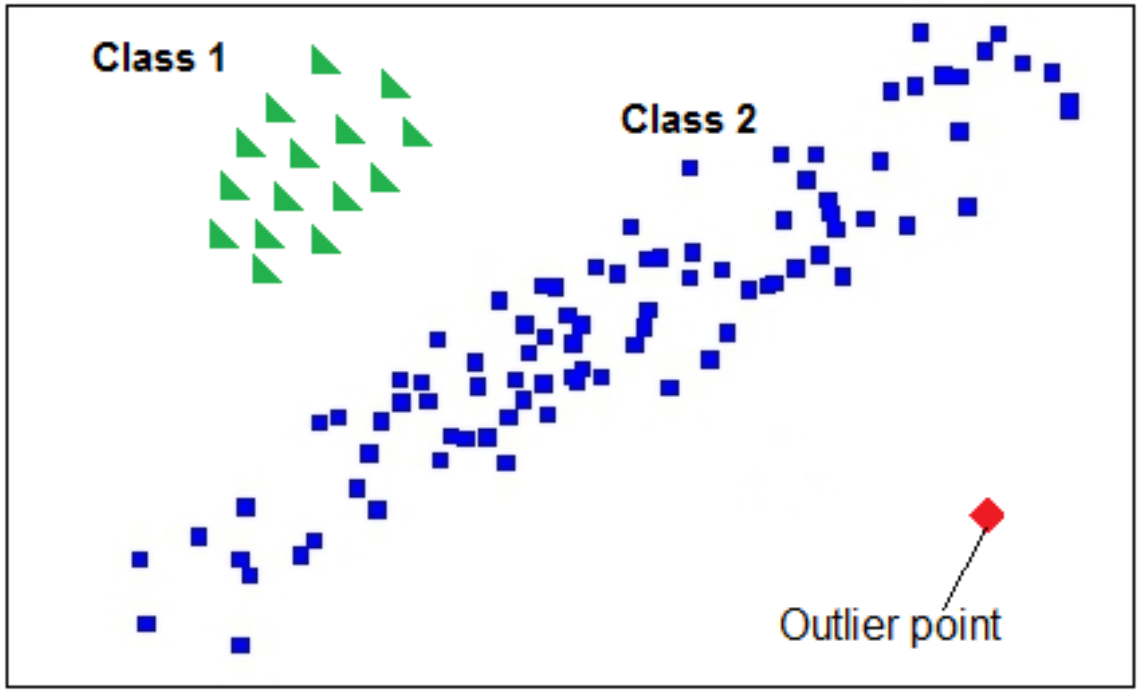
  $V = \sigma^2 = 5.04$ et donc $\sigma = \sqrt{5.04} = 2.24$

- Standard deviation est 2.24 pour cette Dataset

- La **StdDev** est généralement utilisé comme moyen pour identifier les valeurs aberrantes (Outliers). Les Data points qui se situent **à plus d'un σ du Mean** peuvent être considérés comme inhabituels.
- On peut décrire un data point comme extrême, en parlant de «**combien de sigmas**» **est-il loin du Mean.**

Outliers (Données aberrantes)

- En statistique, **une donnée aberrante** est une valeur ou une observation qui est "distante" des autres observations effectuées sur le même phénomène, c'est-à-dire qu'elle contraste grandement avec les valeurs "normalement" mesurées.
- Une donnée aberrante peut être due à la variabilité inhérente au phénomène observé ou bien elle peut aussi indiquer une erreur expérimentale. Normalement, les données aberrantes doivent être exclues de la série de données.



Exemple

On considère 5 instances qui ont l'attribut A avec les valeurs suivantes: {-5, 6, 9, 2, 4}

- Calculez le Min, Max, Mean, Variance et StdDev?

Exemple

Dataset **{-5, 6, 9, 2, 4}**

- Le minimum ***Min = -5***
- Le maximum ***Max= 9***
- La moyenne:

Mean = (-5+6+9+2+4) / 5 = 3.2

- On soustrait la moyenne depuis chaque valeur et on met la valeur au carré:

(-5-3.2)^2 = 67.24
(6-3.2)^2 = 7.84
(9-3.2)^2 = 33.64
(2-3.2)^2 = 1.44
(4-3.2)^2 = 0.64

Variance= (67.24 + 7.84 + 33.64 + 1.44 + 0.64) / 5 =22.16

- Et on retrouve l'écart-type:

StdDev = sqrt ((67.24 + 7.84 + 33.64 + 1.44 + 0.64) / 5) = 4.71

Normalisation et standardisation

Normalisation

- min-max

$$v' = \frac{v - min_A}{max_A - min_A}(new\_max_A - new\_min_A) + new\_min_A$$

Standardisation

- z-score

$$v' = \frac{v - mean_A}{Stand\_dev_A}$$

Exemple

On considère 5 instances qui ont l'attribut A avec les valeurs suivantes:
{-5, 6, 9, 2, 4}

1. Donnez l’intervalle de définition de l’attribut A?
2. Normalisez ses valeurs tel que le nouvel intervalle est [0,1]?

Exemple

On considère 5 instances qui ont l'attribut A avec les valeurs suivantes:
{-5, 6, 9, 2, 4}

1. L’intervalle de définition de l’attribut A est **[-5,9]**
2. Normalisation dans [0,1]:

   {-5, 6, 9, 2, 4} ⟹ **{0, 0.78, 1, 0.5 , 0.64}**

Standardisation

- **Standardisation** *(centrer-réduire)*: c'est la conversion des valeurs vers un standard commun (les rendre conforme à un standard). On veut ramener la distribution de chaque variable a une loi normale centrée-réduite $\boldsymbol{x'_i} \sim \boldsymbol{N}(0,1)$

  - Centrer la moyenne a zéro et ajuster pour un écart-type unitaire

$$\boldsymbol{x'_i} = \frac{\boldsymbol{x_i - \mu}}{\boldsymbol{\sigma}} \quad \boldsymbol{avec} \quad \boldsymbol{x_i \epsilon X} \quad \boldsymbol{et} \quad \boldsymbol{i = 1, \dots, n}$$

  - La standardisation transforme les données en soustrayant à chaque valeur le **Mean** et en la divisant par la **StdDev**. Ceci rendra toutes les valeurs (indifféremment de leurs distributions et unités de mesures originales) en unités compatibles avec **une Moyenne de 0** et **d'écart-type 1**.
  - Moins sensible aux valeurs aberrantes que la normalisation Min-Max.

Exercice

On considère 5 instances qui ont l'attribut A avec les valeurs suivantes:
{-5, 6, 9, 2, 4}

1. Calculez le Min, Max, Mean, Variance et StdDev?
2. Normalisez les valeurs en utilisant « la standardisation». Quelles sont les nouvelles valeurs?
3. Quel le nouvel intervalle?
4. Quelle est le nouveau « Mean » (Moyenne) et «StdDev » (écart-type)?

Median

- la **Médiane** d'un ensemble de valeurs est une valeur x qui permet de couper l'ensemble des valeurs en deux parties égales
- La médian met d'un côté une moitié des valeurs, qui sont toutes inférieures ou égales à x et de l'autre côté l'autre moitié des valeurs, qui sont toutes supérieures ou égales à x
- **Exemple:**

  On commence par mettre les valeurs de la Dataset en ordre:

  0, 2, 3, 2, 1, 0, 0, 2, 0 → 0, 0, 0, 0, **1**, 2, 2, 2, 3

  La Mediane est donc 1

Median

- La Médiane est moins sensible aux Outliers que le Mean.

- **Exemple:**
- La moyenne (Mean) des revenus aux États-Unis est de 72 641 $, mais la médiane n'est que de 51 939 $ parce que la moyenne est biaisée (skewed) par une poignée de milliardaires.
- La médiane représente mieux l'américain «typique» dans cet exemple.

InterQuartile Range

- En statistiques, **l’écart interquartile** ou étendue interquartile (IQR) est une mesure de dispersion qui s'obtient en faisant la différence entre le troisième et le premier quartile : IQR = Q3 - Q1.
- L‘IQR est un estimateur statistique robuste.

InterQuartile Range (exercice)

- Trouvez l'IQR de la dataset suivant: {3, 5, 7, 8, 9, 11, 15, 16, 20, 21}.

  Etape 1: mettre les nombres en ordre.
  3, 5, 7, 8, 9, 11, 15, 16, 20, 21.
  Etape 2: Marquer le centre du dataset:
  3, 5, 7, 8, 9, | 11, 15, 16, 20, 21.
  Etape 3: mettre des parenthèses pour entourer les nombres se trouvant avant et après la marque faites à l'étape 2 ainsi on peut facilement distinguer Q1 et Q3.
  (3, 5, 7, 8, 9), | (11, 15, 16, 20, 21).
  Etape 4: Trouver Q1 et Q3
  Q1 est la médiane (le milieu) de la première moitié des données, et Q3 est la médiane (le milieu) de l'autre partie des données.
  *(3, 5, **7**, 8, 9), | (11, 15, **16**, 20, 21).*
  Q1 = 7 et Q3 = 16.
  Etape 5: Soustraire Q1 de Q3.
  IQR= Q3-Q1= 16 – 7 = 9.

Exemple 2:
Dans la série **10; 25; 30; 40; 41; 42; 50; 55; 70; 101; 110; 111, l'IQR est 40.** En effet, **Q3 valant 70** et **Q1 valant 30**, il suffit de calculer **70-30.**

Mode

- **Mode** : la valeur la plus fréquente dans un ensemble de données. Il n’est pas utilisé dans le cas des valeurs numériques continues.
- **Exemple**:
  - Nombre d'enfants dans chaque maison de mon quartier:

{0, 2, 3, 2, 1, 0, 0, 2, 0}

On compte la fréquence de chaque valeur?

0: 4, 1: 1, 2: 3, 3: 1

Le **Mode** est donc **0**

Missing values

Information manquante

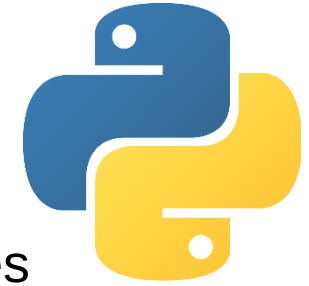
- Cas où les champs ne contiennent aucune donnée.
- Parfois, il est intéressant de conserver ces enregistrements car l'absence d'information peut être informative (e.g. fraude).

Que faire si des valeurs de variables sont manquantes ?

- Retirer les données avec valeurs manquantes
  - Perte de données pour l'apprentissage
- Marquer les variables manquantes pour l'algorithme d'apprentissage
  - Certains algorithmes d'apprentissage peuvent gérer les variables manquantes
- **Imputation:** Assigner une valeur aux variables manquantes
  - On remplace les valeurs manquantes des attributs categoriques et numériques du dataset respectivement avec le mode et la moyenne/ou la médiane.

Outils et Environnement

Python



- **Python** est devenu **le langage** de la science des données. Il combine la puissance des langages de programmation à usage général avec la facilité d'utilisation des langages de script spécifiques au domaine comme MATLAB ou R.

- Python intègre des **bibliothèques** pour le chargement, la visualisation, les statistiques, le traitement du langage naturel, le traitement des images, etc. Il fournit aux scientifiques un large éventail de fonctionnalités générales et spéciales.

- L'un des principaux avantages de l'utilisation de Python est la possibilité d'interagir directement avec le code, en utilisant un terminal ou d'autres outils comme le **bloc-notes Jupyter**.

- L'apprentissage automatique et l'analyse des données sont des processus fondamentalement itératifs, dans lesquels les données conduisent l'analyse. Il est essentiel que ces processus disposent d'outils qui permettent **une itération rapide** et **une interaction facile**.

Scikit-learn

- **Scikit-learn** est open-source, il a une communauté d'utilisateurs et développeurs très active, il s'améliore constamment.

- Scikit-learn intègre des algorithmes d'apprentissage automatique de pointe avec documentation. C'est la librairie python la plus utilise pour le Machine Learning.

- Il est largement utilisé dans l'industrie et le monde universitaire, et il existe une multitude de tutoriels et extraits de code en ligne pour les intéressés.

- Scikit-learn dépend de deux autres packages Python, NumPy et SciPy. Pour le traçage (plot) et le développement interactif, vous devez également installer matplotlib et le bloc-notes Jupyter.

Anaconda

- On va utiliser le gestionnaire de paquets Anaconda.
- **Anaconda**: une distribution Python conçue pour le traitement de données à grande échelle, l'analyse prédictive et le calcul scientifique.
- Anaconda intègre NumPy, SciPy, matplotlib, pandas, IPython, Jupyter Notebook et Scikit-learn.
- Disponible sur Mac OS, Windows et Linux, c'est une solution pratique pour les débutants qui n'ont pas une installation existante des packages scientifiques Python.

- Remarques:
  - Téléchargez la dernière version compatible avec votre OS.
  - Double click pour l’installation sous Windows
  - Et suivez ces étapes pour Linux https://docs.anaconda.com/anaconda/install/linux/

Jupyter

- **Jupyter Notebook** est un environnement de programmation interactif basé sur le navigateur.
- Jupyter permet d'exécuter du code dans le navigateur. C'est un excellent outil pour l'analyse des données et est largement utilisé par les Data scientists.
- il existe deux modes dans Jupyter Notebook: le **mode commande** et le **mode édition**.
- Shortcuts:

  - `Shift + Enter` run the current cell, select below
  - `Ctrl + Enter` run selected cells
  - `Alt + Enter` run the current cell, insert below
  - `Ctrl + S` save and checkpoint

Jupyter

Command mode

(Tapez **ESC** pour l'activer)

- `Enter` take you into edit mode
- `H` show all shortcuts
- `Up` select cell above
- `Down` select cell below
- `Shift + Up` extend selected cells above
- `Shift + Down` extend selected cells below
- `A` insert cell above
- `B` insert cell below
- `X` cut selected cells
- `C` copy selected cells
- `V` paste cells below
- `Shift + V` paste cells above
- `D, D (press the key twice)` delete selected cells
- `Z` undo cell deletion
- `Z` undo cell deletion
- `S` Save and Checkpoint
- `Y` change the cell type to *Code*
- `M` change the cell type to *Markdown*
- `Shift + Space` scroll notebook up
- `Space` scroll notebook down

Jupyter

Edit mode

(Tapez **ENTER** pour l'activer)

- `Esc` take you into command mode
- `Tab` code completion or indent
- `Shift + Tab` tooltip
- `Ctrl + ]` indent
- `Ctrl + [` dedent
- `Ctrl + A` select all
- `Ctrl + Z` undo
- `Ctrl + Shift + Z` or `Ctrl + Y` redo
- `Ctrl + Home` go to cell start
- `Ctrl + End` go to cell end
- `Ctrl + Left` go one word left
- `Ctrl + Right` go one word right
- `Down` move cursor down
- `Up` move cursor up

Autres packages

NumPy

- **NumPy** est l'un des packages fondamentaux pour le calcul scientifique en Python. Il contient des fonctionnalités pour les tableaux multidimensionnels, des fonctions mathématiques de haut niveau telles que les opérations d'algèbre linéaire, les générateurs de nombres pseudo-aléatoires…
- En scikit-learn, le tableau NumPy (NumPy array) est la structure de données fondamentale. scikit-learn prend les données sous forme de NumPy arrays. Toutes les données que vous utilisez devront alors être converties en un tableau NumPy.
- Exemple:

```
import numpy as np

x = np.array([[1, 2, 3], [4, 5, 6]])
print("x:\n{}".format(x))
```

Out[1]:

```
x:
[[1 2 3]
 [4 5 6]]
```

Autres packages

- **SciPy** est une librairie visant à unifier et fédérer un ensemble de bibliothèques Python à usage scientifique. Scipy utilise les tableaux et matrices du module NumPy.

- Cette distribution de modules est destinée à être utilisée avec le langage interprété Python afin de créer un environnement de travail scientifique très similaire à celui offert par Scilab, GNU Octave, Matlab voire R.

- Il contient par exemple des modules pour l'optimisation, l'algèbre linéaire, les statistiques, le traitement du signal ou encore le traitement d'images.

- Afin d'obtenir d'excellentes performances d'exécution (point faible des langages interprétés), la plupart des algorithmes de SciPy et NumPy sont codés en C.

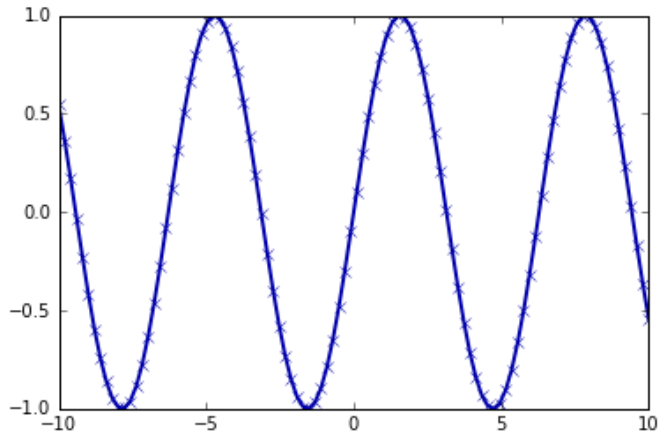
Autres packages

- **Matplotlib** est la principale librairie de traçage (Plotting) scientifique en Python. Elle fournit des fonctions pour effectuer des visualisations de qualité telles que des graphiques linéaires, des histogrammes, des diagrammes de dispersion, etc.

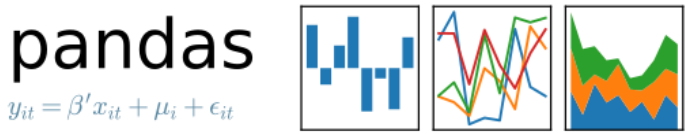
- La visualisation de vos données et des différents aspects de votre analyse peut vous fournir des informations importantes. Sur Jupyter, vous pouvez afficher des figures directement dans le navigateur à l'aide des commandes **%matplotlib notebook** et **%matplotlib inline**

```
%matplotlib inline
import matplotlib.pyplot as plt

# Generate a sequence of numbers from -10 to 10 with 100 steps in between
x = np.linspace(-10, 10, 100)
# Create a second array using sine
y = np.sin(x)
# The plot function makes a line chart of one array against another
plt.plot(x, y, marker="x")
```



Autres packages



- **Pandas** est une librairie Python pour l'analyse des données. Elle est construite autour d'une structure de données appelée DataFrame. Un pandas DataFrame est un tableau, semblable à une feuille de calcul Excel.
- Pandas fournit une large gamme de méthodes pour modifier et opérer sur cette table. Contrairement à NumPy, qui exige que toutes les entrées d'un tableau soient du même type, pandas permet à chaque colonne d'avoir un type distinct (par exemple, des nombres entiers, des dates, et des chaînes).
- Pandas supporte plusieurs formats de fichiers et de bases de données, comme SQL, les fichiers Excel et les fichiers de valeurs séparées par des virgules (CSV).

```
import pandas as pd

# create a simple dataset of people
data = {'Name': ["John", "Anna", "Peter", "Linda"],
        'Location' : ["New York", "Paris", "Berlin", "London"],
        'Age' : [24, 13, 53, 33]
       }

data_pandas = pd.DataFrame(data)
# IPython.display allows "pretty printing" of dataframes
# in the Jupyter notebook
display(data_pandas)
```