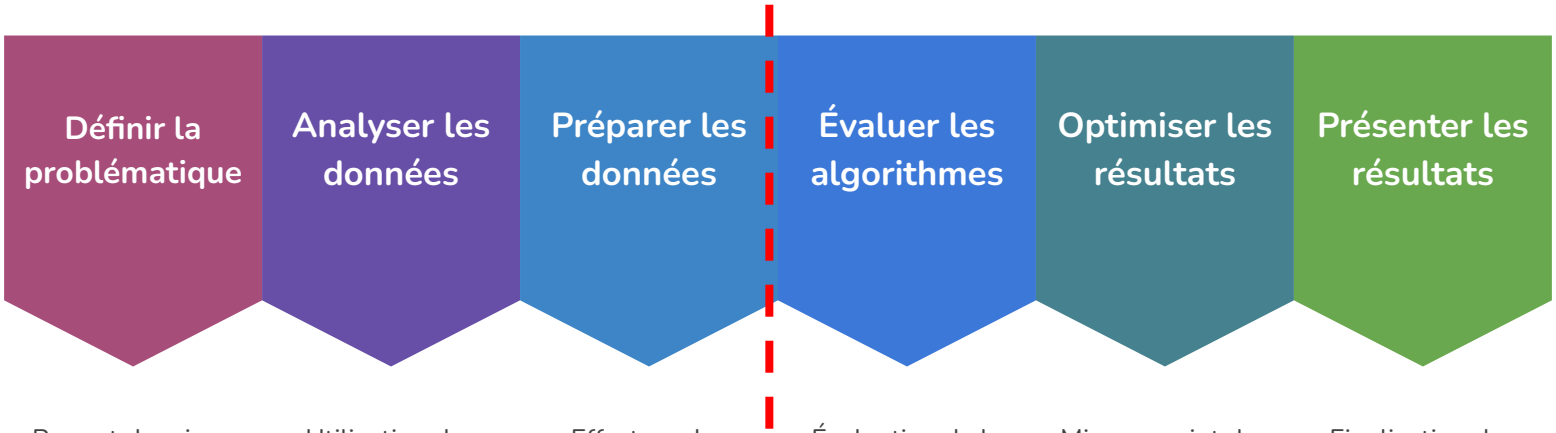
# Préparation des données

Atelier #6

# Principales tâches d’un projet d’apprentissage automatique




Permet de mieux comprendre les objectifs du projet.

Utilisation des statistiques descriptives et des visualisations pour explorer les données sous la main.

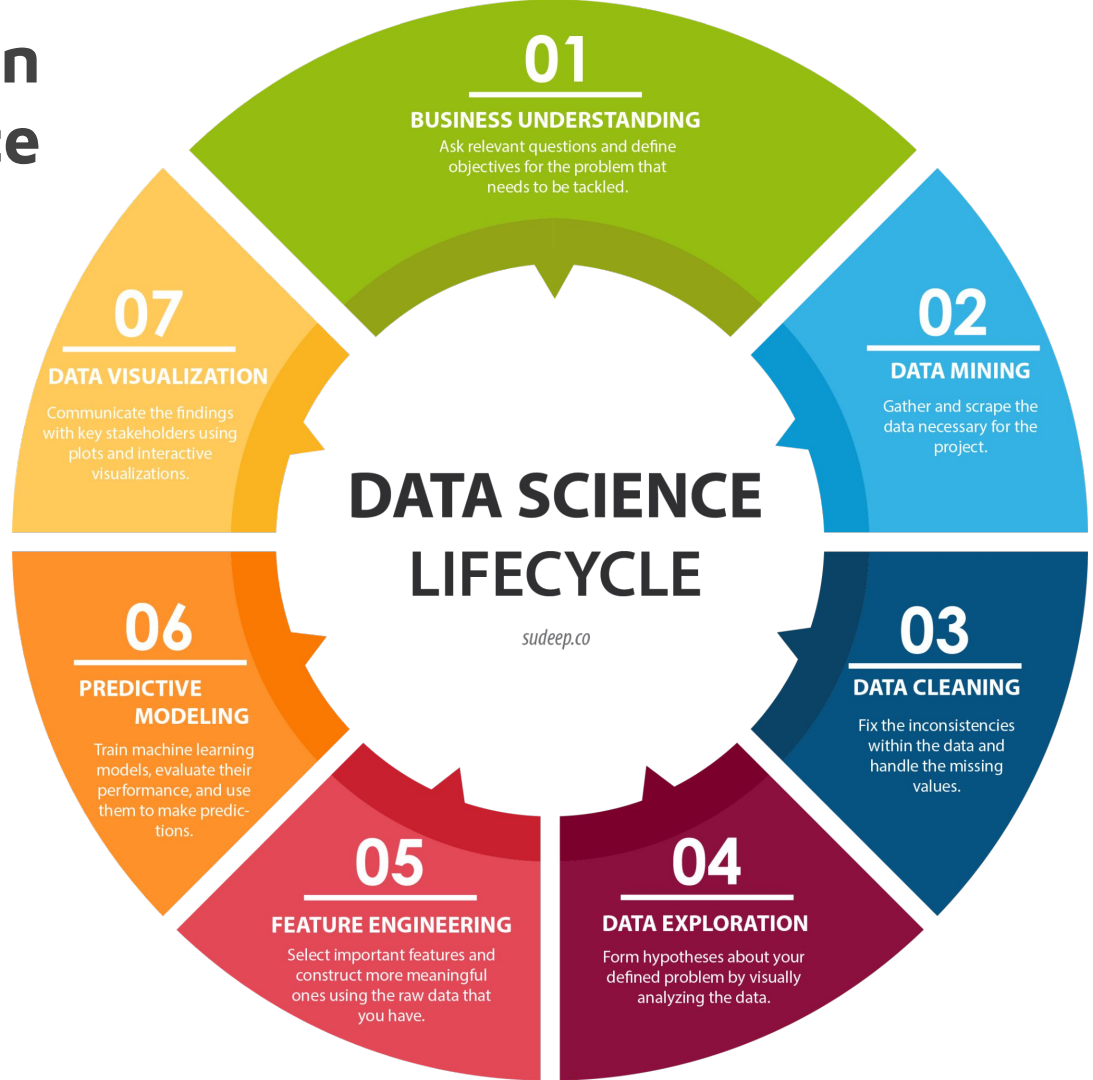
Effectuer des transformations sur les données pour les préparer à l’application des algorithmes.

Évaluation de la performance et fiabilité des algorithmes sur les données pour sélectionner les meilleurs solutions.

Mise au point des algorithmes pour maximiser la performance prédictive.

Finalisation des modèles et présentation des résultats.

# Cycle de vie d'un projet en science des données



# Pourquoi préparer les données?

La préparation des données est une étape essentielle avant de pouvoir appliquer des algorithmes.

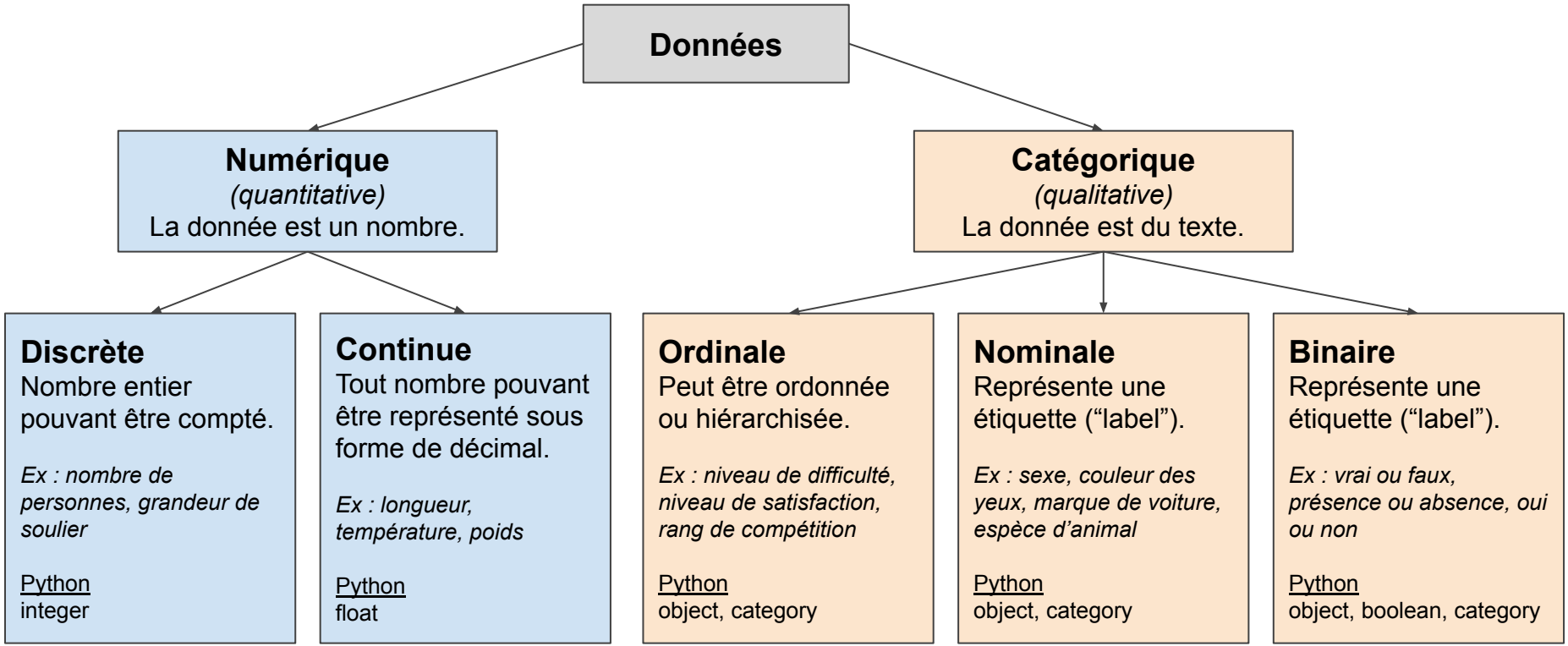
Comme on a vu avec les statistiques sur la distribution normale, les données ne sont pas généralement pas parfaitement distribuer. C'est pourquoi on doit les préparer!

**Approche**

Chaque algorithme va nécessiter une différente approche dans la préparation et l'application de transformations sur les variables.

Il est donc recommandé de tester plusieurs différentes transformations et de sélectionner celle qui performe le mieux. On va y revenir dans une autre leçon.

# Types de données



# Types de données

Tableau des types de données utilisés par pandas et numpy

| Pandas dtype | Python type | NumPy type | Usage |
|---|---|---|---|
| object | str or mixed | string_, unicode_, mixed types | Text or mixed numeric and non-numeric values |
| int64 | int | int_, int8, int16, int32, int64, uint8, uint16, uint32, uint64 | Integer numbers |
| float64 | float | float_, float16, float32, float64 | Floating point numbers |
| bool | bool | bool_ | True/False values |
| datetime64 | datetime | datetime64[ns] | Date and time values |
| timedelta[ns] | NA | NA | Differences between two datetimes |
| category | NA | NA | Finite list of text values |

# Librairie Scikit-Learn (sklearn)

C'est **LA LIBRAIRIE** essentielle pour l'apprentissage automatique!

**Description** : Répertoire d'outils pour l'analyse de données qui sont construits à partir de numpy, scipy et matplotlib.

Lien : https://scikit-learn.org/stable/index.html

# Encoder vs Transformer les données

**Encoder** : On va parler d'encodage lorsqu'on transforme des données catégoriques en format numérique.

*Exemple : ['un', 'deux', 'trois'] encoder = [0, 1, 2]*

**Transformer** : On va parler de transformation lorsqu'on applique une méthode d'ajustement sur des données numériques.

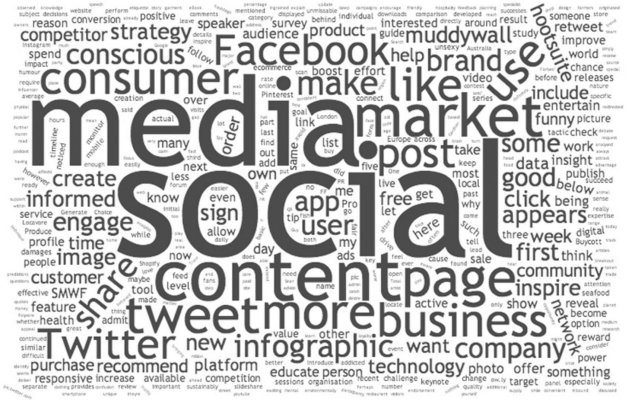
*Exemple : [3, 6, 8, 1] transformer en binaire avec un seuil de 5 = [0, 1, 1, 0]*

# Préparation de données catégoriques

Les algorithmes d'apprentissage automatique ne fonctionnent qu'avec des données numériques.

**Il faut donc transformer les données catégoriques contenant du texte en format numérique.**

Généralement pas nécessaire, ni recommandé, de faire des transformations sur le format numérique des données catégoriques.



# Préparation de données catégoriques

**Méthodes pour ENCODER les données catégoriques :**

1. Catégorique binaire : Remplacer directement les deux valeurs par 1 et 0.
    a. *Ex : [True, False, False] = [1, 0, 0]*
2. Catégorique ordinale : Définir un gradient numérique correspondant aux étiquettes.
    a. *Ex : [facile, difficile, moyen] = [0, 2, 1]*
3. Catégorique nominale : Utiliser un encodage “One-Hot”.
    a. *Ex : [‘chat’, ‘rat’, ‘chien’, ‘rat’] = [[1, 0, 0], [0, 1, 0], [0, 0, 1], [0, 1, 0]]*

# Préparation de données numériques

**Méthodes pour TRANSFORMER les données numériques :**

1. Redimensionner les données (*rescale*)
2. Standardiser les données (*standardize*)
3. Normaliser les données (*normalize*)
4. Binariser les données (*binarize*)

**Information** : https://scikit-learn.org/stable/modules/preprocessing.html#preprocessing-data

# Étapes de préparation

**Chaque méthode suit les mêmes étapes :**

1. Télécharger le jeu de données
2. Séparer le jeu de données en variables d'entrée et de sortie
3. Appliquer les méthodes de transformation
4. Imprimer les valeurs et observer la transformation

**Il est aussi possible d'appliquer une transformation à une seule variable.*

# 1 - Redimensionner (*"rescale"*)

**Description**

- Cette méthode vise à redistribuer les données dans un intervalle prédéfini.
- Permet d'avoir toutes les variables d'un jeu de données sur la même échelle.
- Améliore l'interprétation des données.
- Aide aussi à optimiser les algorithmes d'apprentissage automatique.
- Peut aussi permettre de centrer les données.

**Lien** :
https://scikit-learn.org/stable/modules/preprocessing.html#scaling-features-to-a-range

# 2 - Standardiser (*"standardize"*)

**Description**

- Méthode très pratique qui vise à standardiser les variables qui ont déjà une distribution normale autour d'une moyenne de 0 et un écart-type de 1.
- Fonctionne particulièrement bien en combinaison avec les modèles de régression linéaire, régression logistique et d'analyse discriminante linéaire.
- C'est une des méthodes les plus fréquemment utilisées et applicables à une grande variété de source de données.

**Lien** :
https://scikit-learn.org/stable/modules/preprocessing.html#standardization-or-mean-removal-and-variance-scaling

# 3 - Normaliser (*"normalize"*)

**Description**

- Transformer les données d'une rangée pour que la longueur du vecteur soit égale à 1 en algèbre linéaire.
- Méthode utilisée pour certains algorithmes précis.
- Utile lorsqu'un jeu de données a beaucoup de zéros ou s'il est très étendu.

**Lien** : https://scikit-learn.org/stable/modules/preprocessing.html#normalization

# 4 - Binariser (“*binarize*”)

**Description**

- Transformer les données selon un seuil défini.
- Toutes les valeurs au-dessus du seuil deviendront 1 et celles plus petites ou égales deviendront 0.
- Le seuil est arbitraire, choisi soi-même selon le contexte.
- Utile lorsqu’on travaille avec des probabilités.

**Lien** : https://scikit-learn.org/stable/modules/preprocessing.html#feature-binarization

# Fin de de la phase d’analyse exploratoire des données

***(Exploratory data analysis EDA)***