Arbres & Méthodes d'agrégation

Septembre 2025

Cédric Dangeard
cedric.dangeard@orange.com

Objectifs

- Appréhender la data science en entreprise
- Installer et configurer son environnement de travail
- Visualiser et nettoyer les données
- Comprendre les algorithmes des arbres et les méthodes d'agrégation
- Utiliser Scikit-learn et les librairies R équivalentes
- Réaliser un projet de data science et le restituer

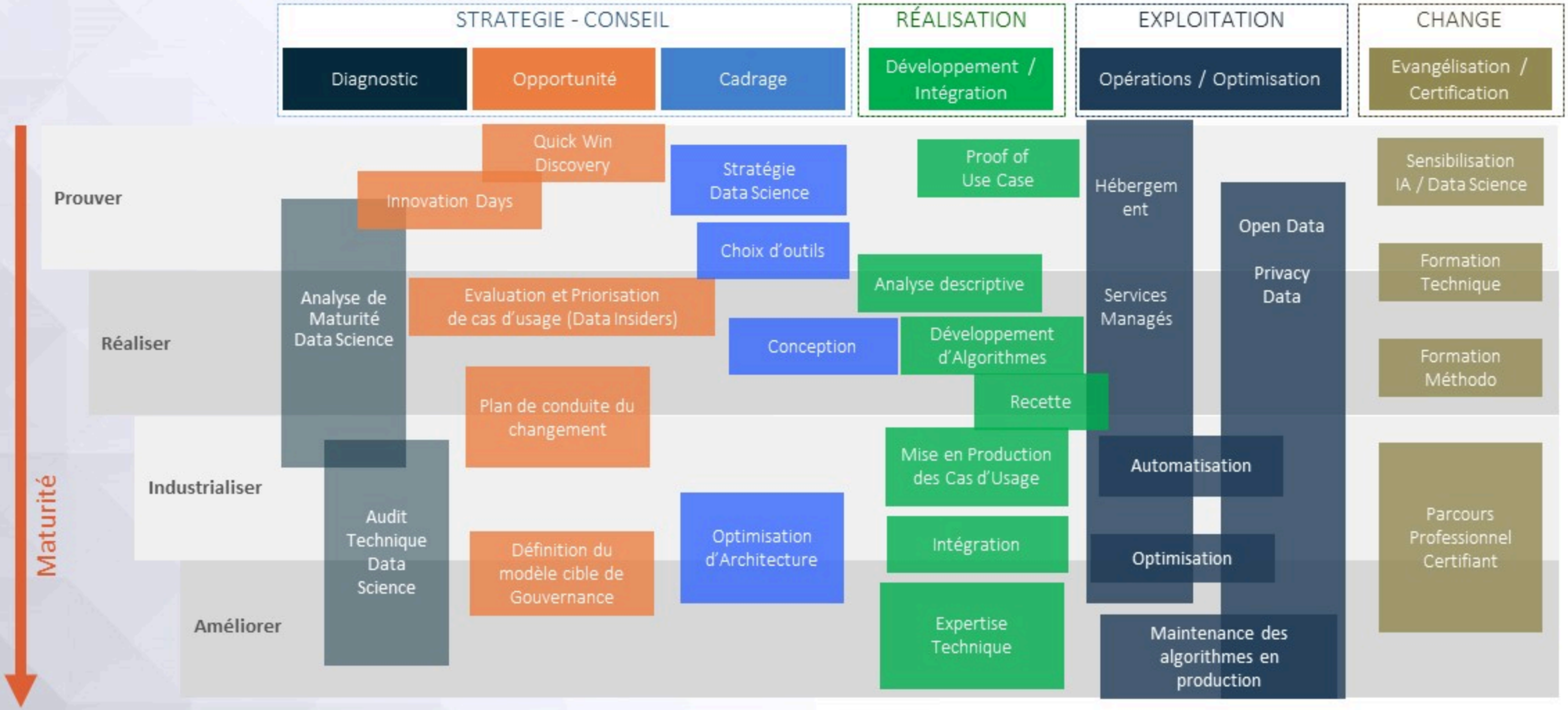
Fonctionnement du cours

- Cours théoriques
- Travaux pratiques (Python & R)
- QCM d'évaluation
- Projet de data science en groupe

Cours 1 : Au programme

- La data science en ESN
- Les étapes d'un projet de Machine Learning
- TP : Installation et setup
- La collecte de données
- Le nettoyage de données
- TP - Découverte et nettoyage du dataset

CATALOGUE DE SERVICES DATA SCIENCE & IA



QUELLES DIFFÉRENCES ENTRE LES PROFILS ?

Data Analyst

- Technique
- Métier
- Statistique
- Optimisation
- Analytique

Data Engineer

- Technique
- Métier
- Statistique
- Optimisation
- Analytique

Data Scientist

- Technique
- Métier
- Statistique
- Optimisation
- Analytique

ML Engineer

- Technique
- Métier
- Statistique
- Optimisation
- Analytique

QUELLES DIFFÉRENCES ENTRE LES PROFILS ?

Data Analyst

Technique	★★
Métier	★★★★★
Statistique	★★
Optimisation	★
Analytique	★★★★★

Data Engineer

Technique	★★★★★
Métier	★
Statistique	★★
Optimisation	★★★★★
Analytique	★★

Data Scientist

Technique	★
Métier	★★★★
Statistique	★★★★★
Optimisation	★★★
Analytique	★★

ML Engineer

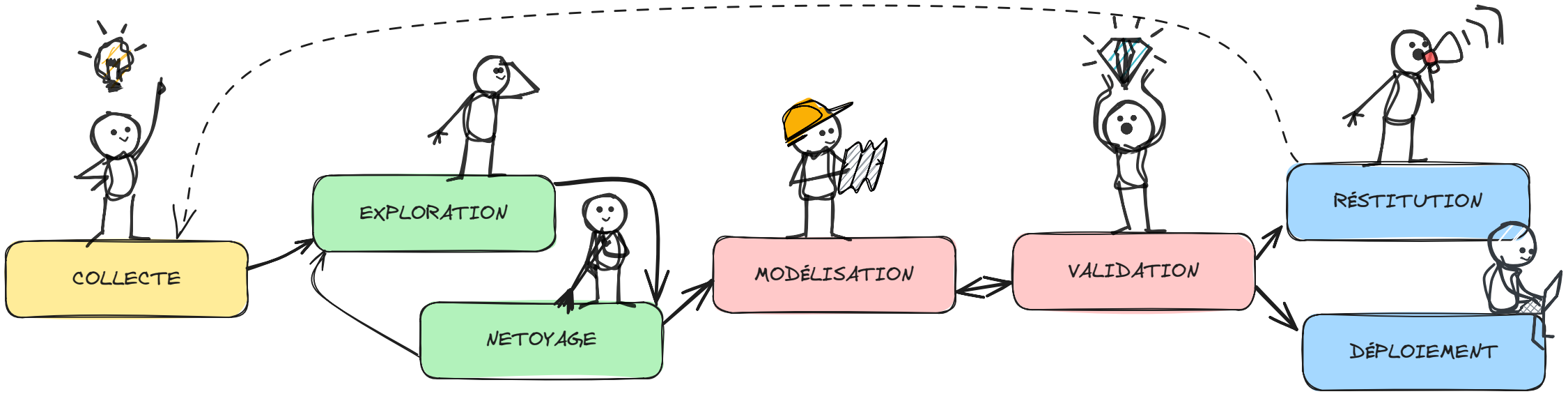
Technique	★★★★★
Métier	★
Statistique	★★★★
Optimisation	★★
Analytique	★

ÉTAPES D'UN PROJET ML

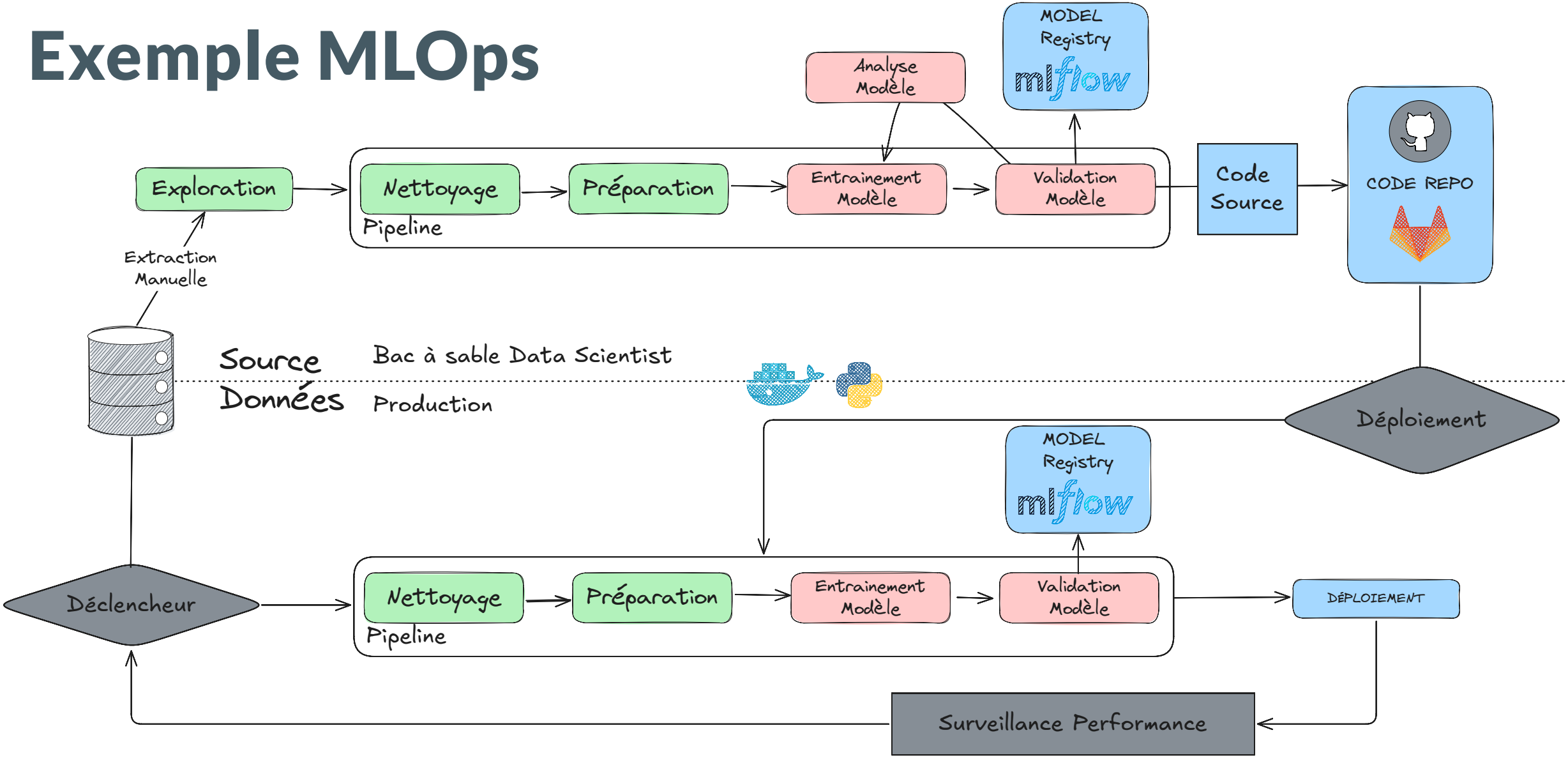
Quelles sont les étapes d'un projet de Machine Learning ?

- ?

ÉTAPES D'UN PROJET ML



Exemple MLOps



Mettre en place son environnement

De quoi a-t-on besoin ?

De quoi parle-t-on ?

- Python
- IPython
- ipykernel
- Jupyter Notebooks
- Spyder
- Google Colab
- Anaconda

Python

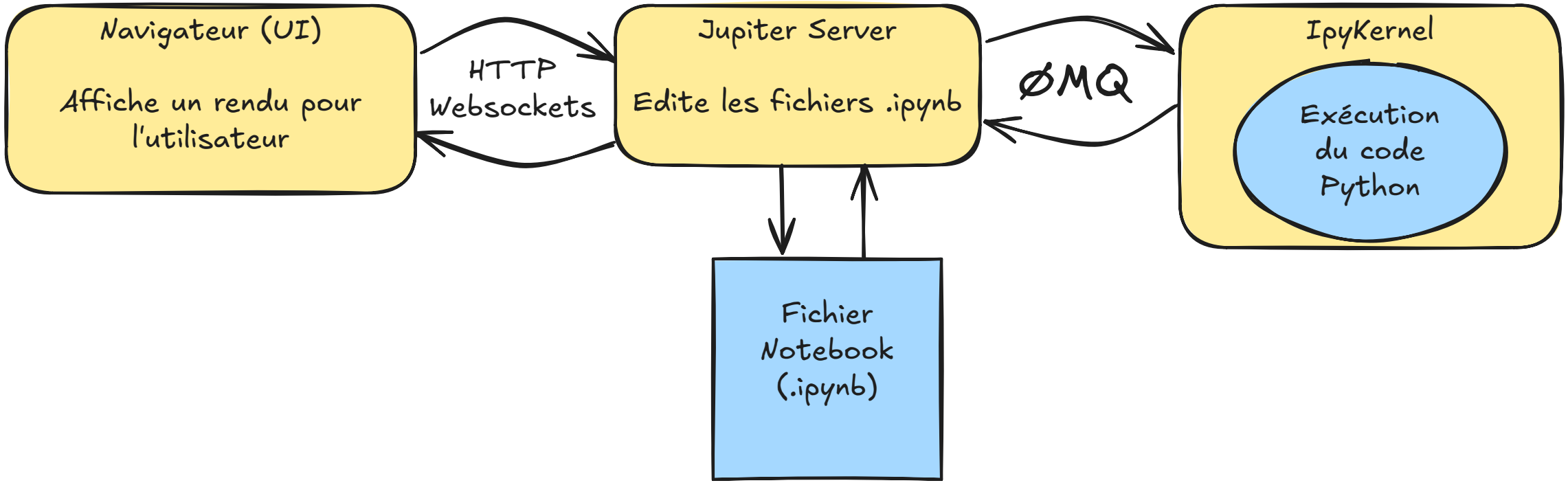
- Langage de programmation
- Créé en 1991 par Guido van Rossum
- Langage interprété
- Langage multi-paradigme

Que fait Jupyter Server ?

- Permet d'exécuter du code dans un navigateur web
- Permet d'éditer des notebooks
- Communique avec un kernel (ex : ipykernel pour Python)
- Communique avec le client (interface web)

Pour aller plus loin : article

Jupyter Server



IDE : Integrated Development Environment

- Environnement de développement intégré
- Permet d'éditer, exécuter et déboguer du code
- Exemples : PyCharm, VSCode, Spyder
- Permet d'installer des extensions

VS Code

- Gratuit et Open Source
- Multi-langages
- Projet Microsoft
- IDE le plus utilisé
- Nombreuses extensions

Pourquoi un environnement virtuel ?

Pourquoi un environnement virtuel ?

- Permet d'isoler les dépendances d'un projet
- Permet d'installer plusieurs versions d'une même bibliothèque
- Permet de partager les dépendances d'un projet
- C'est comme ça qu'on travaille en entreprise

Outils : virtualenv, venv, pipenv, conda

Créer son environnement

- venv est la librairie standard de Python pour créer des environnements virtuels, depuis Python 3.3.
- Il existe d'autres outils comme virtualenv ou pipenv.

```
# Créer un environnement (un dossier monProjetEnv qui contiendra l'environnement)
python -m venv monProjetEnv
# Activer l'environnement (sur Windows)
.\monProjetEnv\Scripts\activate
# Activer l'environnement (sur MacOS/Linux)
source monProjetEnv/bin/activate
# Désactiver l'environnement
deactivate
```

Installer les bibliothèques

```
pip install numpy pandas plotly seaborn scikit-learn
```

ou

```
pip install -r requirements.txt
```

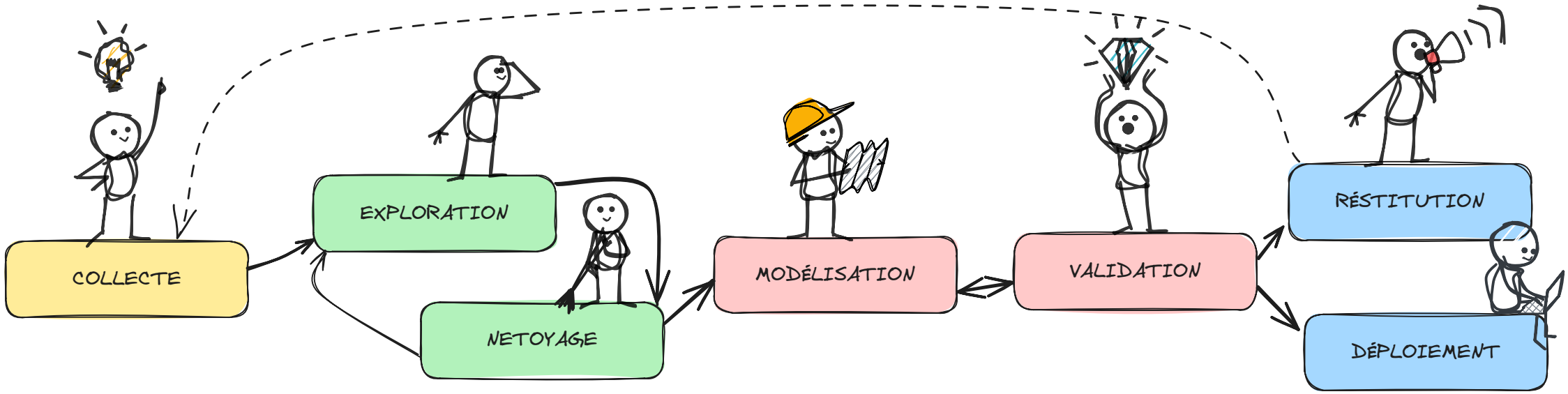
Installer Jupyter

```
# Installe JupyterLab et ipykernel
pip install jupyterlab ipykernel
# Crée un noyau à l'aide de votre environnement
python -m ipykernel install --user --name=monProjetEnv
# Lance le noyau
jupyter lab
```

TP

- Télécharger le notebook ICI

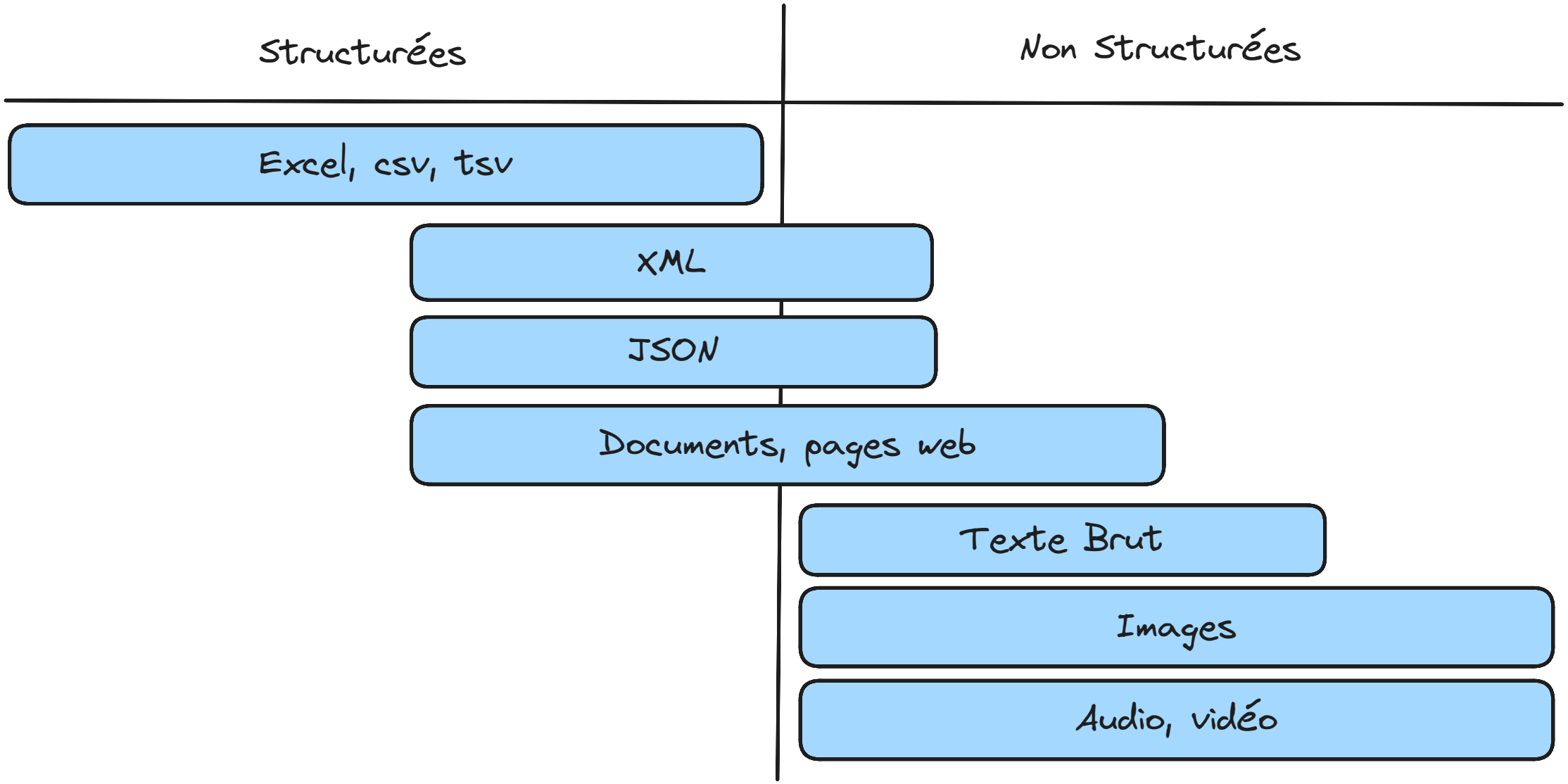
ÉTAPES D'UN PROJET ML



Collecte

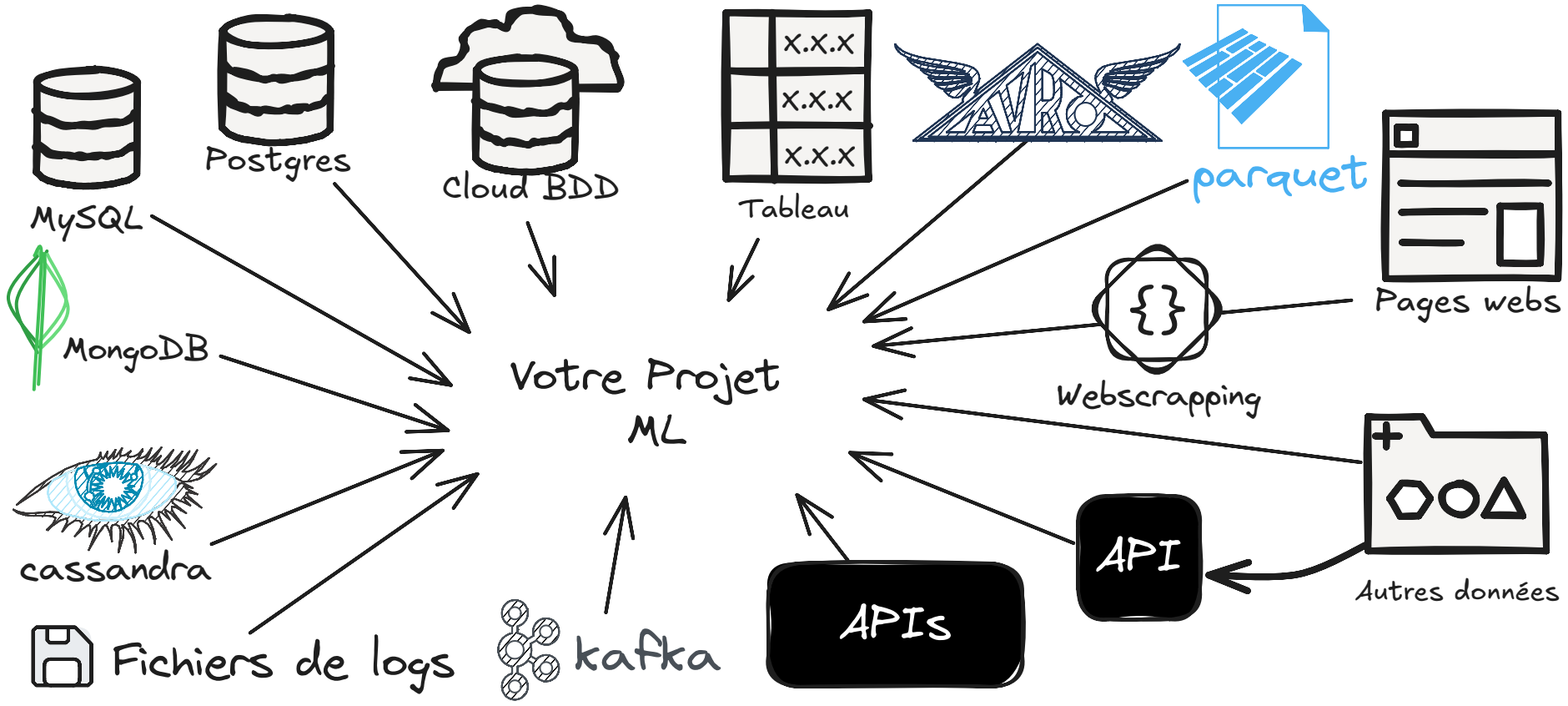
- S'assurer que le projet est réalisable
  - Données disponibles
  - Données suffisantes
  - Données de qualité
- S’assurer de disposer des ressources nécessaires
  - Matériel
  - Délais
- S’assurer de la conformité RGPD du projet

Avec quelles données ?


Structurées
Non Structurées
Excel, csv, tsv
XML
JSON
Documents, pages web
Texte Brut
Images
Audio, vidéo

Avec quelles sources de données ?

Sources de données



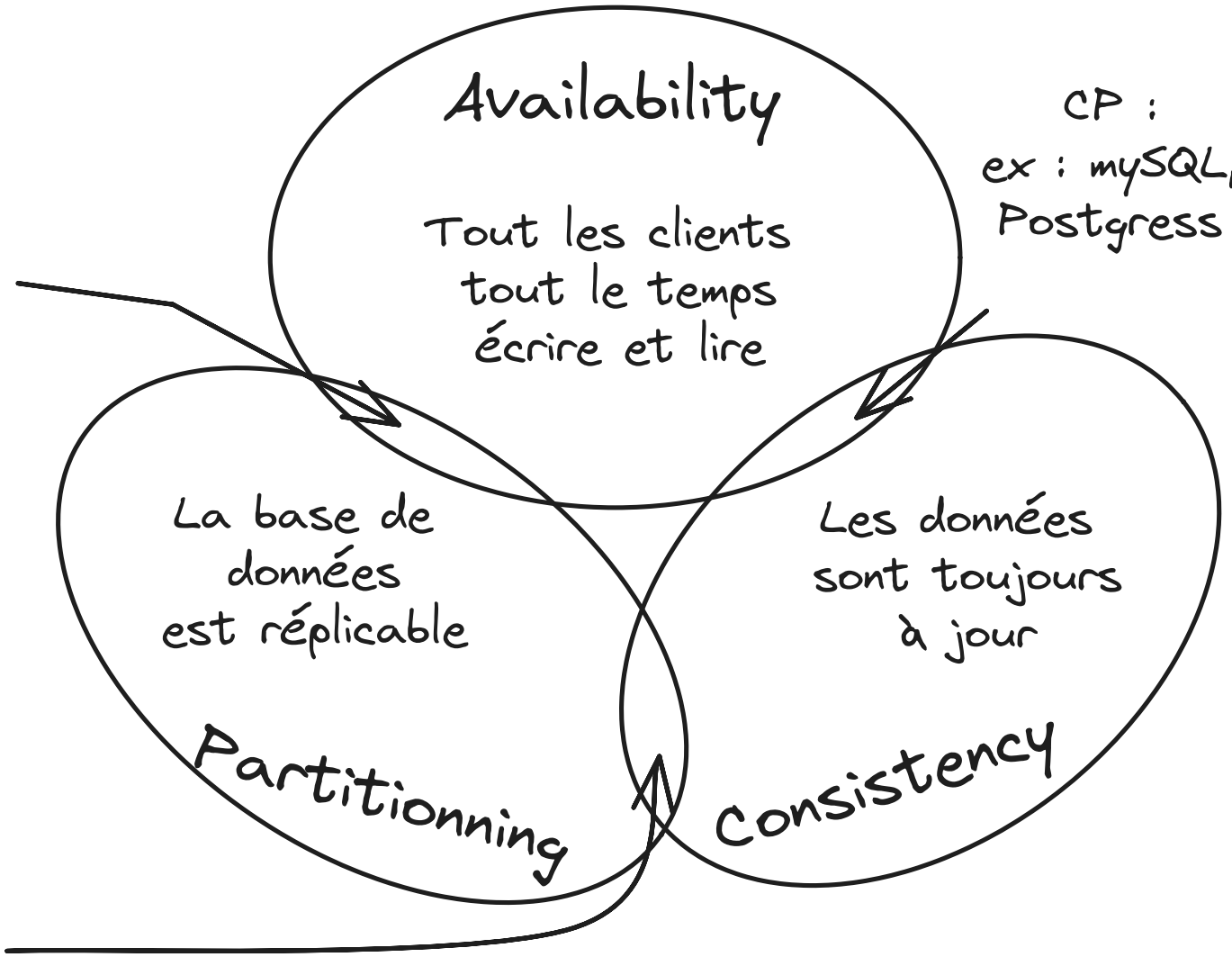
Big Data (5V)

- Volume : Quantité de données
- Vélocité : Vitesse de génération des données
- Variété : Diversité des types de données
- Véracité : Fiabilité des données
- Valeur : Utilité des données

CAP theorem

AP :
Rapidité,
toujours accessible,
mais données
pas forcement à jour
ex : Cassandra,
dynamoDB...

CP:
Données correctes,
mais avec latence
ex : MongoDB,
HBase, Redis...



CP :
ex : mySQL,
Postgress

NoSQL

- Bases de données non relationnelles
- Types
  - Clé-Valeur (Redis, DynamoDB)
  - Document (MongoDB, CouchDB)
  - Colonne (Cassandra, HBase)
  - Graphes (Neo4j)

Formats de fichiers orientés colonne

Dans les fichiers orientés lignes (comme CSV), toutes les données d'une même entrée sont stockées ensemble, et donc à la suite sur le disque.

Avantages :

- Facile à lire et écrire
- Bon pour les transactions

Inconvénients :

- Mauvais pour les analyses
- Mauvais pour la compression

Pourquoi les fichiers orientés colonnes ?

Dans les fichiers orientés colonnes, les données d'une même colonne sont stockées ensemble, et donc à la suite sur le disque.

Avantages :

- Performance des requêtes
- Meilleure efficacité de stockage (compression)

Inconvénients :

- Plus complexe à lire et écrire
- Mauvais pour les transactions

Exemple de fichier orienté colonne

1. Apache Parquet
   Le format de fichier colonne le plus utilisé aujourd'hui (Apache Spark, AWS Athena, Google BigQuery, ...)
2. Apache ORC
   Historiquement utilisé avec Apache Hive. Il a la particularité d'intégrer un
3. Apache Arrow
   Un format en mémoire pour le traitement analytique. Utilisé par Pandas, Spark, Dask, ...

RGPD

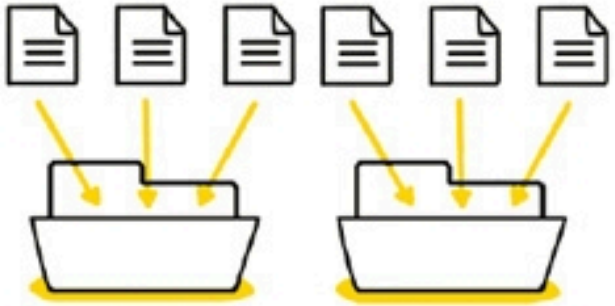
PASSER À L'ACTION

en 4 étapes

1

Constituez un registre de vos traitements de données

2



Faites le tri dans vos données

3

Respectez les droits des personnes

4

Sécurisez vos données

source : CNIL

Exploration : objectifs

1. Comprendre les données
2. Vérifier intégrité & cohérence des données
   - Valeurs manquantes
   - Doublons
   - Valeurs aberrantes
   - Biais
3. Création d'indicateurs
4. Visualiser

Nettoyage des données

Quels problèmes peut-on rencontrer ?

	Nom	Sexe	Ville	Code Postal	latitude	longitude	Age	Taille	Salaire	Client	Num fidélité	Code Concurrent
0	Paul	M	Paris	75000	48.8566	2.35222	18	1.8	1500	True	1235	
1	Pierre	M	Nante	44000	47.2184	-1.55362	25	1.75	20000	False		A
2	Jacques	M	Lyon	69000	45.764	4.83566	32	170	2500	True	1237	
3	Julie	F	Paris	75000	48.8566	2.35222	45	1.65	nan	False		
4	Anne	F	Nantes	44000	47.2184	-1.55362	18	1.8	1500	True	1238	B
5	Marie	F	Lyon	69000	45.764	4.83566	25	1.75	2030	False		
6	Andr%e	F	Paris	75000	48.8566	2.35222	322	1.7	2500	True		
7	Fassou	M	Nantes	44000	47.2184	-1.55362	45	1.65	3000	False		
8	James	M	Lyon	69000	45.764	4.83566	18	1.8	1500	True	1240	
9	Bob	M	Paris	75000	48.8566	2.35222	25	1.75	2000	False		

Quelles solutions ?

Nettoyage des données

- Variables corrélées
- Variable non pertinentes
- Valeurs manquantes
- Doublons
- Valeurs aberrantes
- Données déséquilibrées
- Biais
- Création de nouvelles variables

Données manquantes

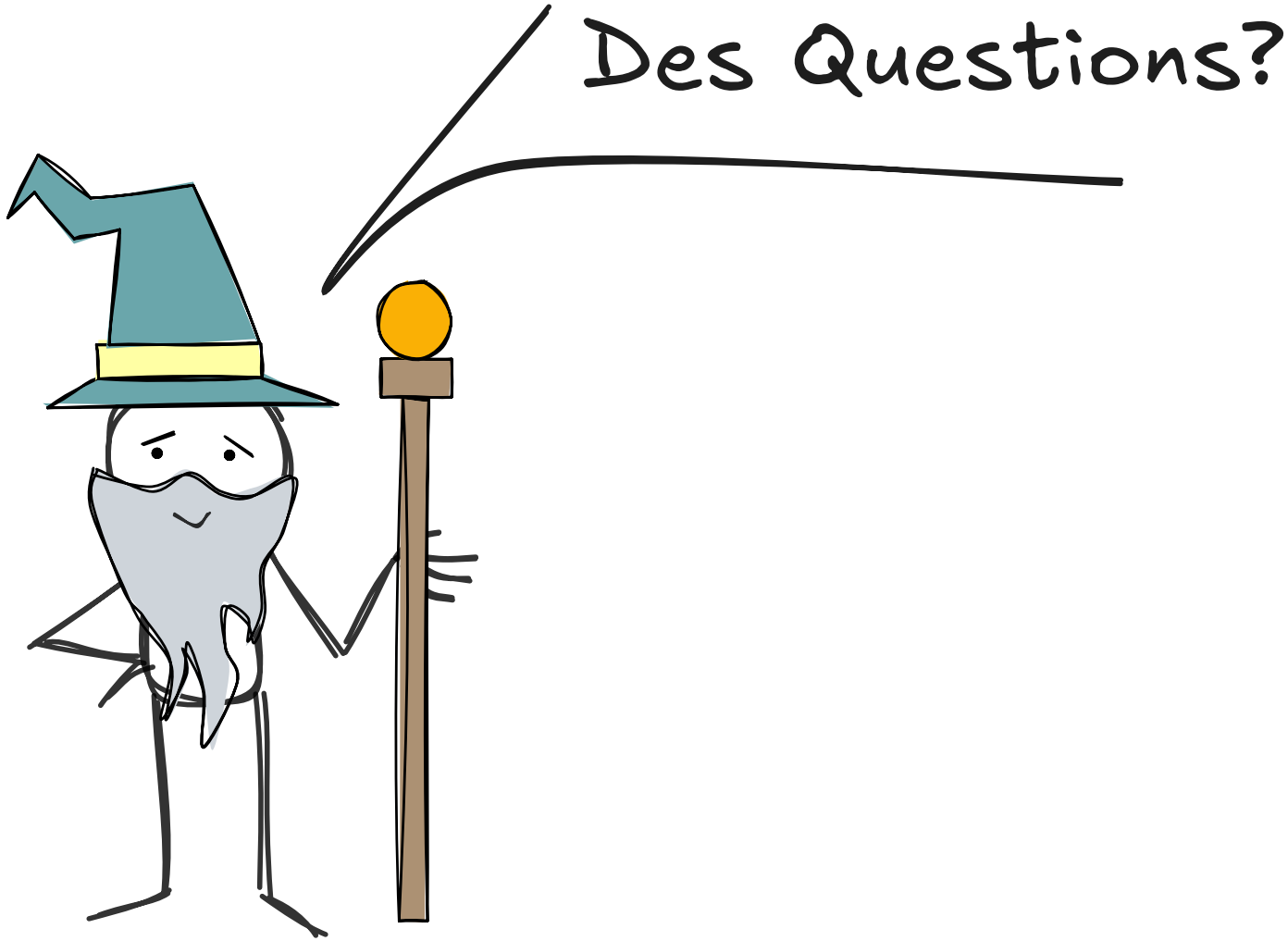
Que faire des données manquantes ?

Données manquantes

- Supprimer les lignes/colonnes
- Imputer avec la moyenne/médiane/modale
- Imputer avec un modèle prédictif (KNN, Régression, ...)
- Marqueur de données manquantes

Pour aller plus loin : cours F. Husson

RETOUR AU TP



Bibliographie

- ipykernel
- Jupyter
- venv
- virtualenv
- pipenv
- CAP theorem