Cours d'introduction à la Data Science

Chargé du cours : Jules DEGILA, PhD, MCF
Assistante : Aurélie KPOZE, PhD

jules.degila@imsp-uac.org, satou.kpoze@imsp-uac.org

18 Novembre 2025

Présentations (1/2)

- **Formation :**
  - Licence Professionnelle — IUT Lokossa
  - Master de Recherche — IMSP
  - Doctorat — (IMSP, UL)

- **Domaines de recherche :** Sécurité informatique, intelligence artificielle, continuum IoT–Edge–Cloud, systèmes de contrôle industriels.

- **Poste actuel :** Chargée de recherche à **INRIA**

Plan du Cours

- **Chapitre 1 : Premiers pas en Data Science**

Plan du Cours

- **Chapitre 1 : Premiers pas en Data Science**
- **Chapitre 2 : De la collecte au stockage des données**

Plan du Cours

- **Chapitre 1 : Premiers pas en Data Science**
- **Chapitre 2 : De la collecte au stockage des données**
- **Chapitre 3 : Préparation, exploration et visualisation des données**

Plan du Cours

- **Chapitre 1 : Premiers pas en Data Science**
- **Chapitre 2 : De la collecte au stockage des données**
- **Chapitre 3 : Préparation, exploration et visualisation des données**
- **Chapitre 4 : Introduction au Machine Learning**

Plan du Cours

- **Chapitre 1 : Premiers pas en Data Science**
- **Chapitre 2 : De la collecte au stockage des données**
- **Chapitre 3 : Préparation, exploration et visualisation des données**
- **Chapitre 4 : Introduction au Machine Learning**
- **Chapitre 5 : Sécurité, éthique et gouvernance des données**

Plan du Cours

- **Chapitre 1 : Premiers pas en Data Science**
- **Chapitre 2 : De la collecte au stockage des données**
- **Chapitre 3 : Préparation, exploration et visualisation des données**
- **Chapitre 4 : Introduction au Machine Learning**
- **Chapitre 5 : Sécurité, éthique et gouvernance des données**
- **Chapitre 6 : Projet final de Data Science**

Objectifs du cours

À la fin de ce cours, vous devrez être capables de :

- Comprendre les fondements de la Data Science

Objectifs du cours

À la fin de ce cours, vous devrez être capables de :

- Comprendre les fondements de la Data Science
- Maîtriser les étapes d'un projet de data science, de la collecte des données à la présentation des résultats

Objectifs du cours

À la fin de ce cours, vous devrez être capables de :

- Comprendre les fondements de la Data Science
- Maîtriser les étapes d'un projet de data science, de la collecte des données à la présentation des résultats
- Utiliser les outils essentiels du data scientist

Objectifs du cours

À la fin de ce cours, vous devrez être capables de :

- Comprendre les fondements de la Data Science
- Maîtriser les étapes d'un projet de data science, de la collecte des données à la présentation des résultats
- Utiliser les outils essentiels du data scientist
- Effectuer des analyses statistiques descriptives et produire des visualisations claires et interprétables.

Objectifs du cours

À la fin de ce cours, vous devrez être capables de :

- Comprendre les fondements de la Data Science
- Maîtriser les étapes d'un projet de data science, de la collecte des données à la présentation des résultats
- Utiliser les outils essentiels du data scientist
- Effectuer des analyses statistiques descriptives et produire des visualisations claires et interprétables.
- Concevoir et entraîner des modèles simples de Machine Learning pour la régression ou la classification.

Objectifs du cours

À la fin de ce cours, vous devrez être capables de :

- Comprendre les fondements de la Data Science
- Maîtriser les étapes d'un projet de data science, de la collecte des données à la présentation des résultats
- Utiliser les outils essentiels du data scientist
- Effectuer des analyses statistiques descriptives et produire des visualisations claires et interprétables.
- Concevoir et entraîner des modèles simples de Machine Learning pour la régression ou la classification.
- Appliquer les bonnes pratiques de sécurité, d'éthique et de gouvernance des données.

Mode d'évaluation

Élément d'évaluation	**Barème**	**Description**
Travaux pratiques (TP)	25 pts	5 mini TP notés à la fin de chaque chapitre.
Projet final de Data Science	25 pts	Réalisation complète d'un projet : collecte, préparation, analyse, modélisation et présentation.
Épreuve finale	45 pts	Examen individuel couvrant l'ensemble des notions du cours.
Participation et engagement	5 pts	Présence active, participation aux discussions et travaux de groupe.
Total	**100 pts**	

Note finale = 25 (TP) + 25 (Projet) + 45 (Épreuve) + 5 (Participation) = 100

Pré-requis du cours *"Introduction à la Data Science"*

1. **Bases en Python** : notions de variables, structures de données (`list`, `dict`), boucles et fonctions simples.
2. **Notions élémentaires en mathématiques et statistiques** : moyenne, pourcentage, représentation graphique.
3. **Connaissances informatiques de base** : manipulation de fichiers, utilisation d'un navigateur et d'un environnement numérique.
4. **Aucune connaissance préalable en Data Science n'est requise.**
5. **Matériel nécessaire** : ordinateur portable, connexion Internet et accès à Google Colab ou Jupyter Notebook.

Pré-requis du cours *"Introduction à la Data Science"*

Ressources complémentaires à ce cours :

- **Documentation officielle :**
  - Python — `https://docs.python.org`
  - Pandas — `https://pandas.pydata.org/docs`
  - Scikit-learn — `https://scikit-learn.org/stable`

Pré-requis du cours *"Introduction à la Data Science"*

Ressources complémentaires à ce cours :

- **Documentation officielle :**
  - Python — `https://docs.python.org`
  - Pandas — `https://pandas.pydata.org/docs`
  - Scikit-learn — `https://scikit-learn.org/stable`
- **Cours et plateformes d'apprentissage :**
  - Coursera — *IBM Data Science Specialization*
  - Kaggle Learn — *Python, Pandas, Machine Learning*
  - DataCamp — *Introduction to Data Science with Python*

Pré-requis du cours *"Introduction à la Data Science"*

Ressources complémentaires à ce cours :

- **Documentation officielle :**
  - Python — `https://docs.python.org`
  - Pandas — `https://pandas.pydata.org/docs`
  - Scikit-learn — `https://scikit-learn.org/stable`
- **Cours et plateformes d'apprentissage :**
  - Coursera — *IBM Data Science Specialization*
  - Kaggle Learn — *Python, Pandas, Machine Learning*
  - DataCamp — *Introduction to Data Science with Python*
- **Livres recommandés :**
  - Wes McKinney — *Python for Data Analysis*
  - Aurélien Géron — *Hands-On Machine Learning with Scikit-Learn & TensorFlow*

Présentations (2/2)

- Parcours, attentes, préoccupations particulières..

Quelques conseils pour bien débuter en Data Science

Gardez à l'esprit :

- Restez curieux

Quelques conseils pour bien débuter en Data Science

Gardez à l'esprit :

- Restez curieux
- Gardez un esprit critique

Quelques conseils pour bien débuter en Data Science

Gardez à l'esprit :

- Restez curieux
- Gardez un esprit critique
- Apprenez en pratiquant

Quelques conseils pour bien débuter en Data Science

Gardez à l'esprit :

- Restez curieux
- Gardez un esprit critique
- Apprenez en pratiquant
- Documentez vos travaux

Quelques conseils pour bien débuter en Data Science

Gardez à l'esprit :

- Restez curieux
- Gardez un esprit critique
- Apprenez en pratiquant
- Documentez vos travaux
- Collaborez ! la Data Science est un travail d'équipe : partagez, discutez, demandez de l'aide.

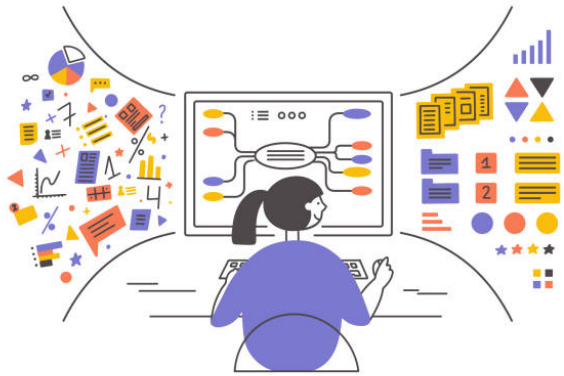
Quelques conseils pour bien débuter en Data Science

Gardez à l'esprit :

- Restez curieux
- Gardez un esprit critique
- Apprenez en pratiquant
- Documentez vos travaux
- Collaborez ! la Data Science est un travail d'équipe : partagez, discutez, demandez de l'aide.
- Restez éthique : respectez la confidentialité des données et évitez les biais dans vos analyses.

Chapitre 1

Premiers pas en Data Science



Introduction aux concepts, rôles et outils fondamentaux de la Data Science.

Plan du Chapitre 1

Objectifs du Chapitre 1 — Premiers pas en Data Science

À la fin de ce chapitre, vous serez capables de :

- Définir la **Data Science** et comprendre son importance dans le monde actuel.
- Identifier les **étapes principales d'un projet data** (collecte → nettoyage → analyse → modélisation → communication).
- Distinguer les concepts de *Data Science*, *Machine Learning*, *Intelligence Artificielle* et *Big Data*.
- Connaître les **rôles clés** : data scientist, data analyst, data engineer.
- Découvrir les **outils essentiels** : Python, Pandas, Matplotlib, Scikit-learn.

Introduction

- C'est quoi une donnée ?

1

[1] source image: https://blog.techlearnindia.com/data-science-terms

Introduction

- C'est quoi une donnée ?
- Qu'est ce que la data science ?

1

[1] source image: https://blog.techlearnindia.com/data-science-terms

Introduction

Définition

Une donnée est une information qui peut être enregistrée, mesurée, stockée ou analysée par un système informatique. C'est la plus petite unité d'information utilisée en Data Science.
Exemple: Un nom, un âge, un son, etc...
Ensemble de données = Jeu de données (Dataset)

Définition

Discipline qui combine **statistiques**, **programmation** et **connaissance métier** pour extraire de la valeur à partir des données.

- Objectif : *passer des données brutes aux décisions*.

Introduction

Définition

Une donnée est une information qui peut être enregistrée, mesurée, stockée ou analysée par un système informatique. C'est la plus petite unité d'information utilisée en Data Science.
Exemple: Un nom, un âge, un son, etc...
Ensemble de données = Jeu de données (Dataset)

Définition

Discipline qui combine **statistiques**, **programmation** et **connaissance métier** pour extraire de la valeur à partir des données.

- Objectif : *passer des données brutes aux décisions*.
- Applications : santé, finance, transport, industrie, marketing, énergie...

Introduction

Définition

Une donnée est une information qui peut être enregistrée, mesurée, stockée ou analysée par un système informatique. C'est la plus petite unité d'information utilisée en Data Science.
Exemple: Un nom, un âge, un son, etc...
Ensemble de données = Jeu de données (Dataset)

Définition

Discipline qui combine **statistiques**, **programmation** et **connaissance métier** pour extraire de la valeur à partir des données.

- Objectif : *passer des données brutes aux décisions*.
- Applications : santé, finance, transport, industrie, marketing, énergie...
- Différences (haut niveau) : *Data Science* vs *ML* vs *IA* vs *Big Data*.

Introduction

Comment tirer de la valueur des données ? Grâce aux modèles.
Un **modèle** est une représentation mathématique qui apprend à partir des données afin de **prédire**, **classer**, **détecter** ou **estimer** quelque chose.

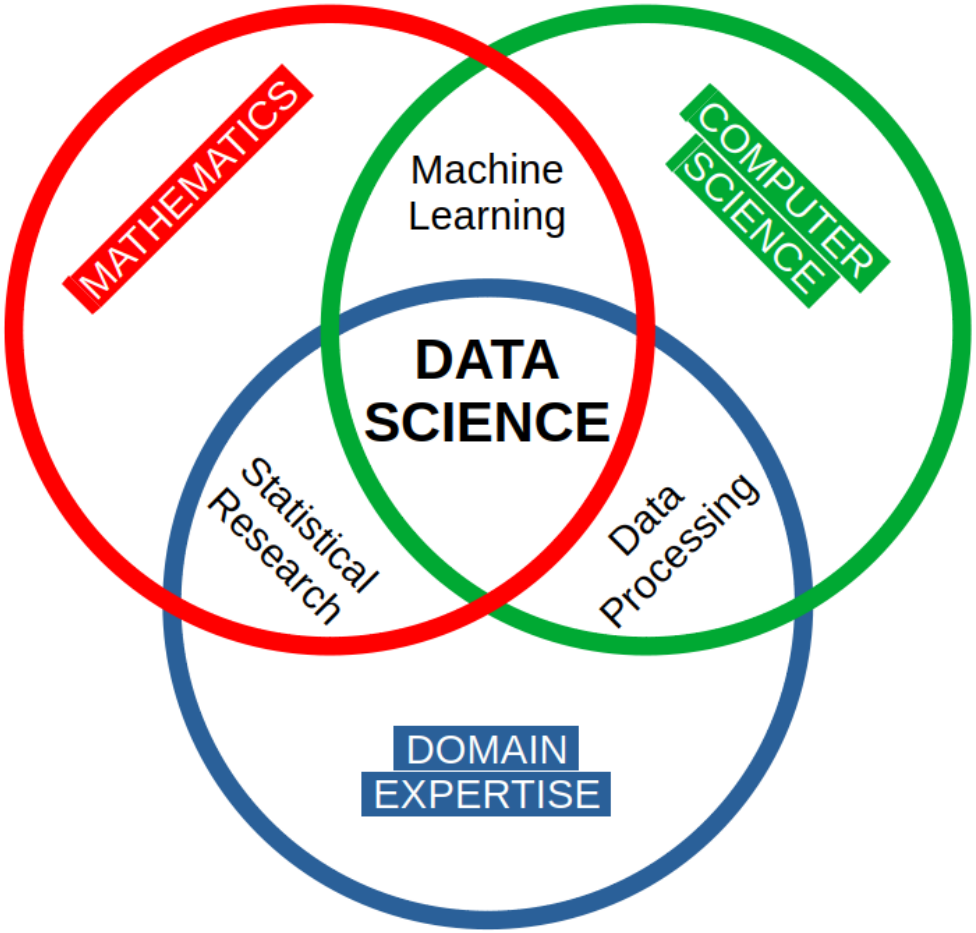
Un modèle apprend grâce à :

- **Les données d'entrée (features)** : âge, salaire, ville, historique...
- **La cible (label)** : ce que l'on veut prédire (ex. quitter l'abonnement : oui/non)

Exemple : Un modèle apprend qu'un client jeune avec un faible salaire quitte souvent un service. Il peut ensuite **prédire** si un nouveau client risque de partir.

À retenir : Un modèle = une fonction intelligente qui **apprend** à partir des données et **généralise** à de nouveaux cas.
Différents types de modèles selon la tâche à réaliser.

Introduction



2

[2] source image: Wikipedia

Quelques faits marquants sur la Data Science :

- Chaque jour, plus de **300 millions de téraoctets de données** sont générés dans le monde (réseaux sociaux, capteurs, transactions, etc.).

Quelques faits marquants sur la Data Science :

- Chaque jour, plus de **300 millions de téraoctets de données** sont générés dans le monde (réseaux sociaux, capteurs, transactions, etc.).
- En 2025, on estime que la quantité de données produites dépassera **175 zettaoctets** (IDC, 2023).

Quelques faits marquants sur la Data Science :

- Chaque jour, plus de **300 millions de téraoctets de données** sont générés dans le monde (réseaux sociaux, capteurs, transactions, etc.).
- En 2025, on estime que la quantité de données produites dépassera **175 zettaoctets** (IDC, 2023).
- Les entreprises qui utilisent la Data Science pour orienter leurs décisions sont en moyenne **23% plus rentables** (McKinsey, 2022).

Quelques faits marquants sur la Data Science :

- Chaque jour, plus de **300 millions de téraoctets de données** sont générés dans le monde (réseaux sociaux, capteurs, transactions, etc.).
- En 2025, on estime que la quantité de données produites dépassera **175 zettaoctets** (IDC, 2023).
- Les entreprises qui utilisent la Data Science pour orienter leurs décisions sont en moyenne **23% plus rentables** (McKinsey, 2022).
- La Data Science est utilisée dans tous les secteurs : **santé, transport, énergie, finance, éducation, sécurité, IA, IoT**.

Quelques faits marquants sur la Data Science :

- Chaque jour, plus de **300 millions de téraoctets de données** sont générés dans le monde (réseaux sociaux, capteurs, transactions, etc.).
- En 2025, on estime que la quantité de données produites dépassera **175 zettaoctets** (IDC, 2023).
- Les entreprises qui utilisent la Data Science pour orienter leurs décisions sont en moyenne **23% plus rentables** (McKinsey, 2022).
- La Data Science est utilisée dans tous les secteurs : **santé, transport, énergie, finance, éducation, sécurité, IA, IoT**.
- Les métiers liés à la Data (Data Scientist, Data Engineer, Analyste) figurent parmi les **professions les plus demandées au monde**.

Introduction

Quelques faits marquants sur la Data Science :

- Chaque jour, plus de **300 millions de téraoctets de données** sont générés dans le monde (réseaux sociaux, capteurs, transactions, etc.).
- En 2025, on estime que la quantité de données produites dépassera **175 zettaoctets** (IDC, 2023).
- Les entreprises qui utilisent la Data Science pour orienter leurs décisions sont en moyenne **23% plus rentables** (McKinsey, 2022).
- La Data Science est utilisée dans tous les secteurs : **santé, transport, énergie, finance, éducation, sécurité, IA, IoT**.
- Les métiers liés à la Data (Data Scientist, Data Engineer, Analyste) figurent parmi les **professions les plus demandées au monde**.
- Le **Big Data** désigne le volume considérable d'informations structurées et non structurées que les humains et les machines génèrent.

Importance de la Data Science

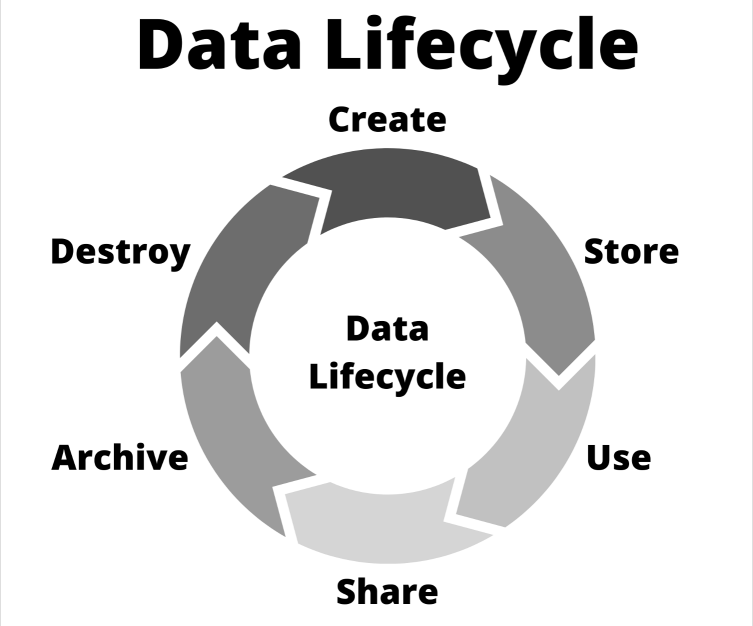
Quelques exemples concrets d'application

- **Netflix, Amazon, Spotify** : recommandation personnalisée de films, produits ou musiques selon le comportement des utilisateurs.
- **Hôpitaux et laboratoires** : aide au diagnostic, prédiction d'épidémies, optimisation des traitements grâce à l'analyse de données médicales.
- **Agriculture intelligente** : utilisation de capteurs et d'images satellites pour optimiser l'irrigation et anticiper les rendements.
- **ChatGPT et IA générative** : analyse et apprentissage sur des milliards de textes pour produire des réponses cohérentes et naturelles.

Concepts fondamentaux

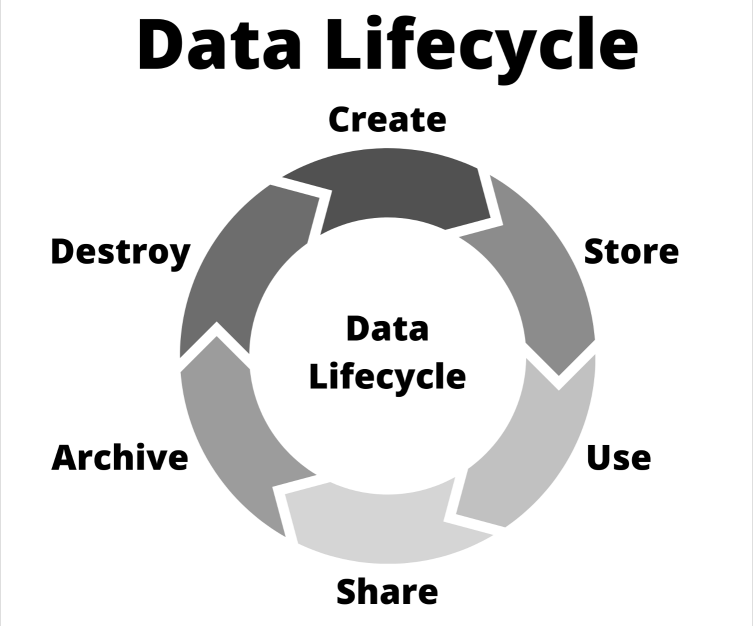
Cycle de vie des données

1. Création/Acquisition



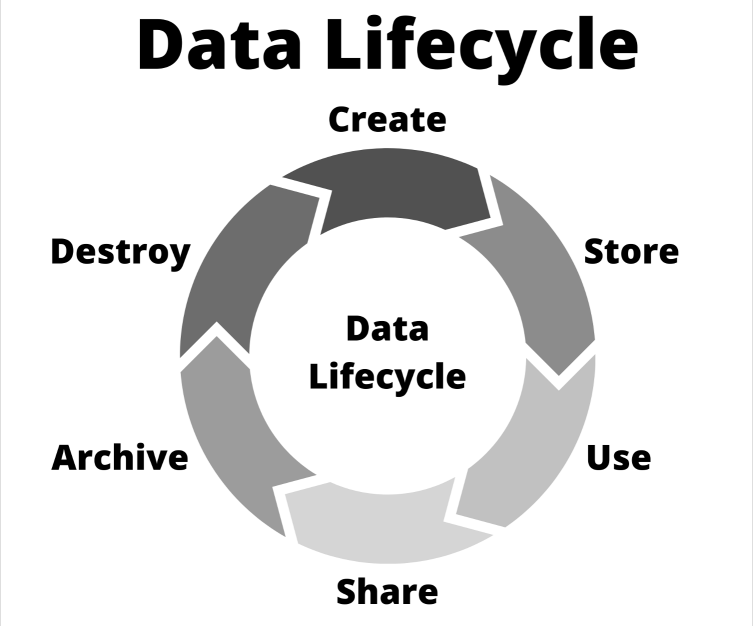
Cycle de vie des données

1. Création/Acquisition
2. Stockage



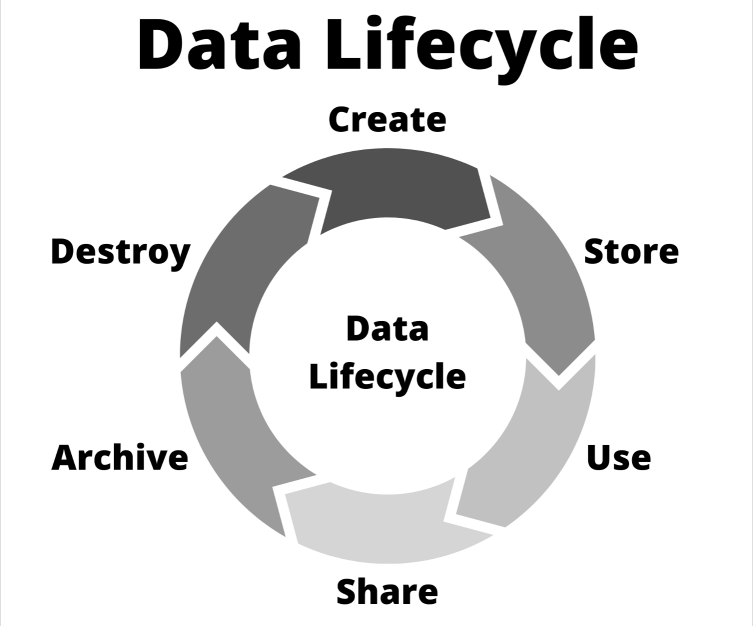
Cycle de vie des données

1. Création/Acquisition
2. Stockage
3. Traitement



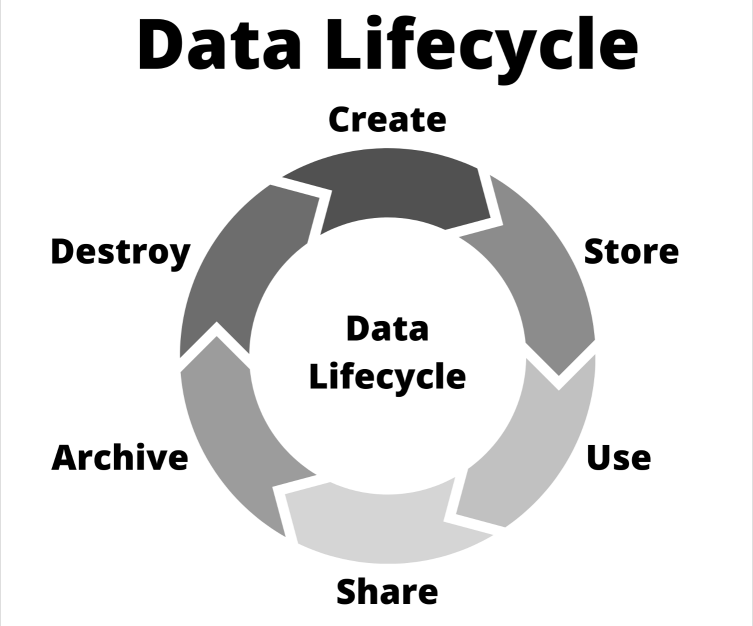
Cycle de vie des données

1. Création/Acquisition
2. Stockage
3. Traitement
4. Analyse/Consommation



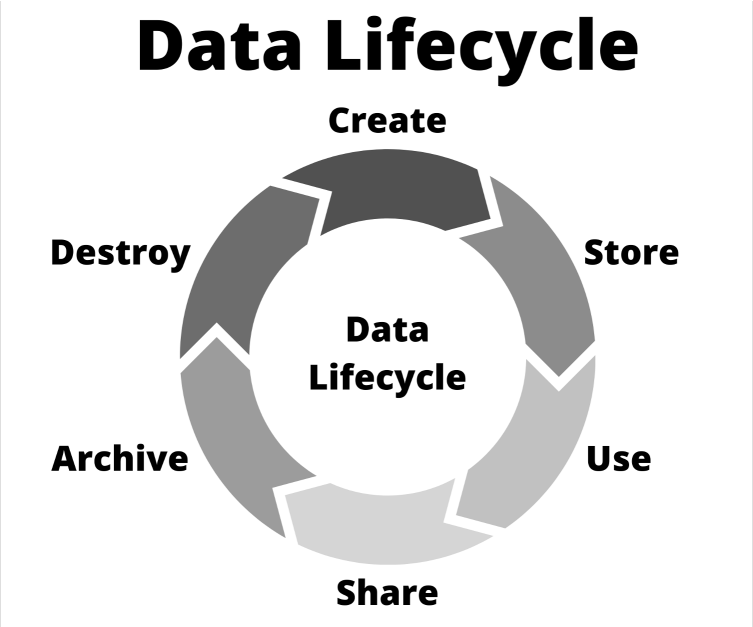
Cycle de vie des données

1. Création/Acquisition
2. Stockage
3. Traitement
4. Analyse/Consommation
5. Partage/Diffusion



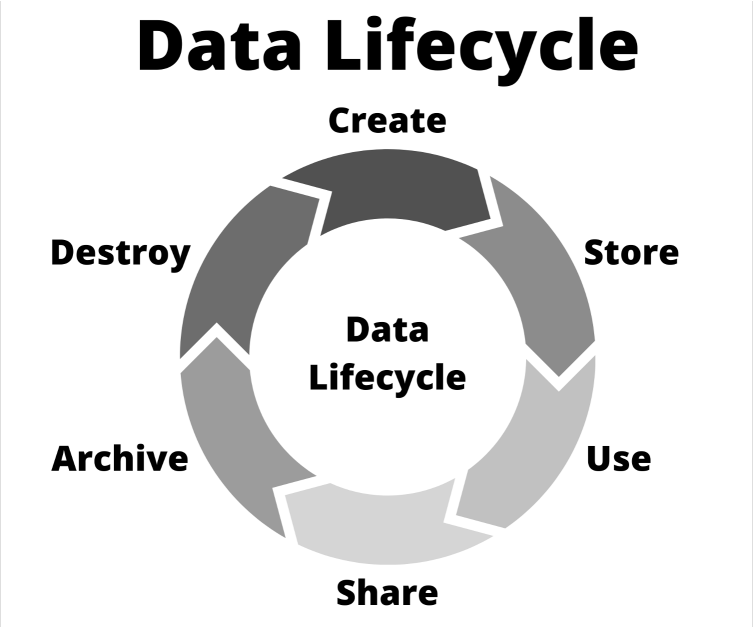
Cycle de vie des données

1. Création/Acquisition
2. Stockage
3. Traitement
4. Analyse/Consommation
5. Partage/Diffusion
6. Archivage



Cycle de vie des données

1. Création/Acquisition
2. Stockage
3. Traitement
4. Analyse/Consommation
5. Partage/Diffusion
6. Archivage
7. Destruction



Cas d'étude : Cycle de vie des données

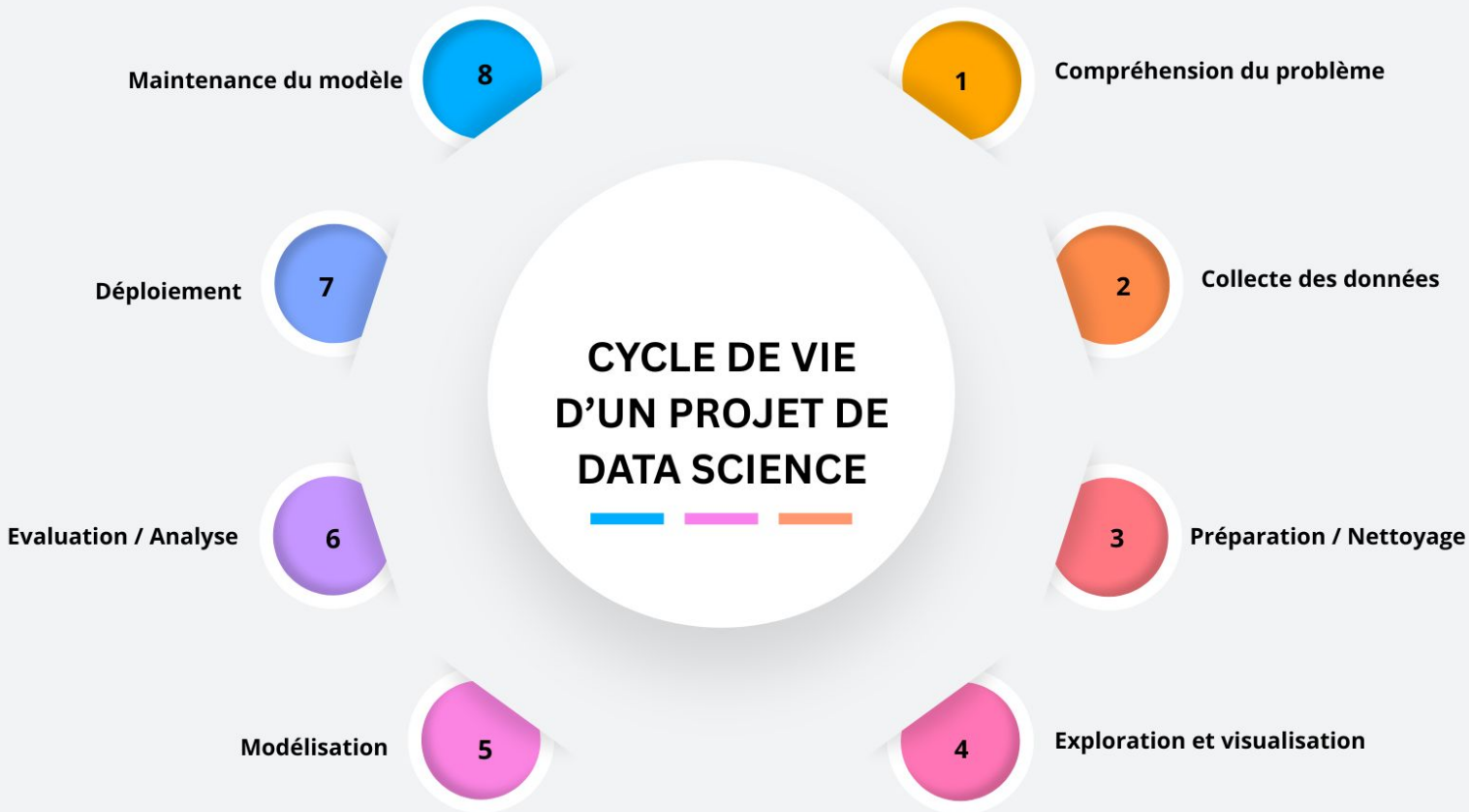
Exemple : Service de livraison (type Uber Eats): Acquisition, stockage, traitement, analyse, partage, archivage, destruction

Cas d'étude : Cycle de vie des données

Exemple : Service de livraison (type Uber Eats): Acquisition, stockage, traitement, analyse, partage, archivage, destruction

1. **Acquisition** Le client passe une commande $\rightarrow$ l'app enregistre adresse, panier, heure.
2. **Stockage** Les données sont sauvegardées dans une base SQL.
3. **Traitement** Nettoyage, calcul du temps estimé de livraison, assignation d'un livreur.
4. **Analyse** Statistiques internes : temps moyen, zones à forte demande.
5. **Partage** L'utilisateur voit le suivi en temps réel dans l'application.
6. **Archivage** Anciennes commandes conservées pour l'historique.
7. **Destruction** Suppression des données si l'utilisateur supprime son compte.

Cycle de vie d'un projet de Data Science



Cas d'étude 1 : Prédire une perte de clients

Objectif : Prédire quels clients risquent de ne plus utiliser Uber Eats et de passer à un service concurrent.

Cycle du projet :

Cas d'étude 1 : Prédire une perte de clients

Objectif : Prédire quels clients risquent de ne plus utiliser Uber Eats et de passer à un service concurrent.

Cycle du projet :

- **Compréhension du besoin** L'entreprise souhaite anticiper la perte de clients afin de mieux cibler les actions de fidélisation.

Cas d'étude 1 : Prédire une perte de clients

Objectif : Prédire quels clients risquent de ne plus utiliser Uber Eats et de passer à un service concurrent.

Cycle du projet :

- **Compréhension du besoin** L'entreprise souhaite anticiper la perte de clients afin de mieux cibler les actions de fidélisation.
- **Collecte des données** Historique des commandes, fréquence d'utilisation, délais de livraison, réclamations, satisfaction, promotions utilisées.

Cas d'étude 1 : Prédire une perte de clients

Objectif : Prédire quels clients risquent de ne plus utiliser Uber Eats et de passer à un service concurrent.

Cycle du projet :

- **Compréhension du besoin** L'entreprise souhaite anticiper la perte de clients afin de mieux cibler les actions de fidélisation.
- **Collecte des données** Historique des commandes, fréquence d'utilisation, délais de livraison, réclamations, satisfaction, promotions utilisées.
- **Préparation des données** Nettoyage, gestion des valeurs manquantes, encodage des catégories, création de nouvelles variables (*baisse d'activité, temps moyen de livraison, score d'insatisfaction*).

Cas d'étude 1 : Prédire une perte de clients

Objectif : Prédire quels clients risquent de ne plus utiliser Uber Eats et de passer à un service concurrent.

Cycle du projet :

- **Compréhension du besoin** L'entreprise souhaite anticiper la perte de clients afin de mieux cibler les actions de fidélisation.
- **Collecte des données** Historique des commandes, fréquence d'utilisation, délais de livraison, réclamations, satisfaction, promotions utilisées.
- **Préparation des données** Nettoyage, gestion des valeurs manquantes, encodage des catégories, création de nouvelles variables (*baisse d'activité, temps moyen de livraison, score d'insatisfaction*).
- **Exploration** Analyse des comportements associés au départ : diminution de commandes, mauvaises expériences, délais trop longs.

Cas d'étude 1 : Prédire une perte de clients

- **Modélisation** Construction d'un modèle de classification (Logistic Regression, Random Forest, XGBoost) pour prédire "reste" vs "part".

Cas d'étude 1 : Prédire une perte de clients

- **Modélisation** Construction d'un modèle de classification (Logistic Regression, Random Forest, XGBoost) pour prédire "reste" vs "part".
- **Évaluation** Utilisation de métriques adaptées : précision, rappel, F1-score, matrice de confusion.

Cas d'étude 1 : Prédire une perte de clients

- **Modélisation** Construction d'un modèle de classification (Logistic Regression, Random Forest, XGBoost) pour prédire "reste" vs "part".
- **Évaluation** Utilisation de métriques adaptées : précision, rappel, F1-score, matrice de confusion.
- **Déploiement** Intégration du modèle dans le système interne pour alerter automatiquement les équipes marketing.

Cas d'étude 1 : Prédire une perte de clients

- **Modélisation** Construction d'un modèle de classification (Logistic Regression, Random Forest, XGBoost) pour prédire "reste" vs "part".
- **Évaluation** Utilisation de métriques adaptées : précision, rappel, F1-score, matrice de confusion.
- **Déploiement** Intégration du modèle dans le système interne pour alerter automatiquement les équipes marketing.
- **Suivi et amélioration** Surveillance de la performance du modèle, réentraînement périodique en fonction des nouveaux comportements des clients.

Cas d'étude 2 : prédire les préférences d'achat d'un client

Contexte : Une entreprise souhaite recommander les bons produits ou afficher les bonnes publicités à ses clients afin d'augmenter les ventes et améliorer l'expérience utilisateur.

Objectif : Prédire quel type de produit un client est le plus susceptible d'acheter prochainement.

- Compréhension du besoin ?
- Collecte des données ?
- Préparation des données ?
- Exploration ?
- Modélisation ?
- Evaluation ?
- Déploiement ?
- Suivi et amélioration ?

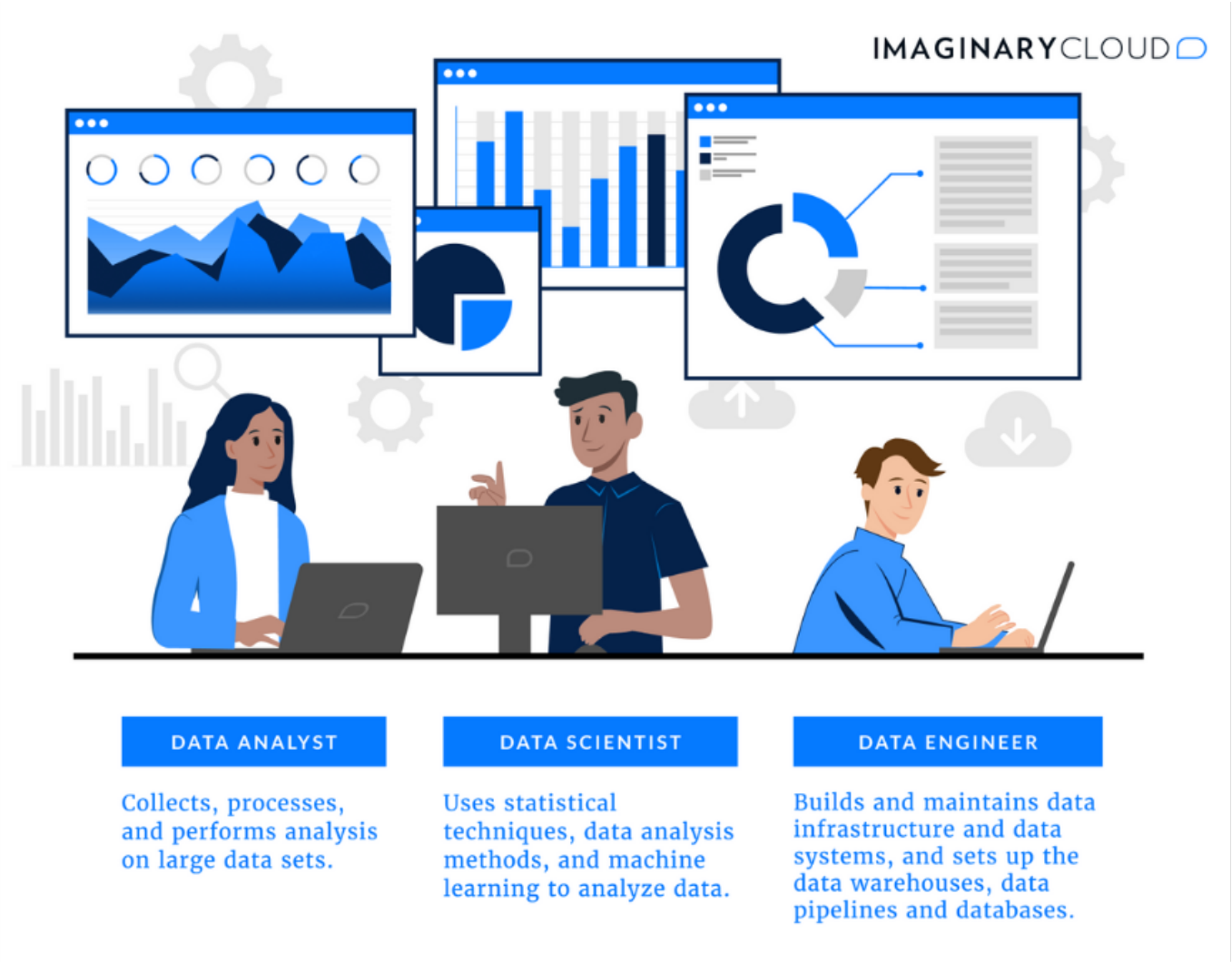
Les principaux rôles en Data Science

- **Data Scientist** : conçoit et évalue les modèles d'analyse.
- **Data Analyst** : interprète les données et produit des rapports.
- **Data Engineer** : construit les infrastructures et pipelines de données.

Collaboration

Ces métiers travaillent en synergie pour transformer les données en valeur.

Les principaux rôles en Data Science



Cas d'étude: Prédiction du départ des clients (Customer Churn)

Contexte : Une entreprise de télécommunication souhaite identifier les clients susceptibles de résilier leur abonnement, afin d'anticiper les départs et d'améliorer la fidélisation.

Objectif du projet : Construire un modèle de Machine Learning capable de prédire si un client risque de partir dans les 3 prochains mois.

Cas d'étude: Prédiction du départ des clients (Customer Churn)

Rôles et contributions :

- **Data Engineer :** Collecte des données clients (CRM, appels au service client, paiements), nettoyage, et stockage dans une base SQL.
- **Data Analyst :** Exploration et visualisation des données (âge, ancienneté, satisfaction, incidents). Identification des corrélations avec le départ des clients.
- **Data Scientist :** Entraînement d'un modèle de classification (Random Forest) pour prédire le churn, évaluation et interprétation des résultats.
- **ML Engineer :** Intégration du modèle dans le système d'information pour une utilisation en temps réel par le service marketing.

Données: sources, types et caractéristiques

Sources de données

Où trouver les données ?

- **Internes** : bases métiers, CRM/ERP, logs d'applications, entrepôts de données.
- **Externes** : open data (data.gouv.fr, UCI, Kaggle), partenaires, fournisseurs de données.
- **APIs** : REST/GraphQL (ex. météo, finance, réseaux sociaux).
- **Fichiers** : CSV, Excel, JSON, Parquet ; exports ponctuels.
- **Capteurs/IoT** : séries temporelles, télémétrie, messages MQTT.
- **Web scraping** : pages web, documents ; attention aux aspects légaux/éthiques.

Critères de choix : qualité, coût, volumétrie, droit d'usage (licence), sécurité & confidentialité.

Données : structurées, semi-structurées, non structurées

Catégorie	Définition	Exemples & formats
Structurées	Données tabulaires avec schéma fixe (colonnes typées).	Tables SQL, feuilles Excel, CSV (transactions, ventes, clients).
Semi-structurées	Schéma flexible + balises/attributs.	JSON, XML, YAML, logs & événements, documents NoSQL (MongoDB).
Non structurées	Sans schéma explicite, contenu libre.	Texte libre (PDF, emails), images, audio, vidéo, posts réseaux sociaux.

Impacts pratiques : outils/stockage (SQL vs NoSQL), parsing/ETL, coût de préparation, performances d'analyse.

Types de données (statistiques)

Échelles de mesure :

- **Nominal** : catégories sans ordre (*sexe*, *pays*, *couleur*).
- **Ordinal** : catégories ordonnées (*niveau d'étude*, *satisfaction 1–5*).
- **Quantitatif discret** : entiers comptables (*nombre d'achats*, *enfants*).
- **Quantitatif continu** : mesures réelles (*taille*, *revenu*, *température*).

Pourquoi c'est crucial ?

- Choix des *statistiques* : moyenne/médiane, écart-type, corrélations.
- Choix des *graphes* : barplot (nominal/ordinal), histogramme/boxplot (quantitatif).
- Choix des *modèles* : encodage des variables, métriques d'évaluation adaptées.

Pratique : mapping types $\leftrightarrow$ outils

- **Pandas** : `object` (texte/catégories), `category` (catégoriel encodé), `int64`, `float64`, `bool`, `datetime64`.
- **SQL** : `VARCHAR/TEXT`, `INT/BIGINT`, `DECIMAL/FLOAT`, `BOOLEAN`, `DATE/TIMESTAMP`.
- **NoSQL** : documents flexibles (JSON), schéma à la lecture, indices par champ.

Bonnes pratiques

- Définir les types dès l'ingestion (*schema-on-write* si possible).
- Normaliser les catégories (valeurs cohérentes, casse/typos).
- Gérer les *missing values* et dates/temps (timezone !).

Choisir sa source et son format

Questions clés

- *Quel besoin métier ?* KPI, exploration, prédiction, temps réel ?
- *Contraintes* : volumétrie, latence, budget, sécurité/RGPD, licence.
- *Durée de vie* : ponctuel (CSV) vs récurrent (DB/API, pipeline).

Règles simples

- **Tabulaire & stable** $\Rightarrow$ SQL / Parquet.
- **Événements/logs** $\Rightarrow$ JSON + data lake / streaming.
- **Non structuré** $\Rightarrow$ stockage objet (S3/Blob) + indexation.

Qu'est-ce qu'un dataset ?

Définition : Un **dataset** (jeu de données) est un ensemble structuré d'observations. Il se présente généralement sous forme de **tableau**, où :

- les **lignes** représentent des exemples / individus ;
- les **colonnes** représentent des variables (features).

Exemples de datasets

- Titanic : informations sur les passagers.
- Housing : caractéristiques de maisons.
- MNIST : images de chiffres manuscrits.

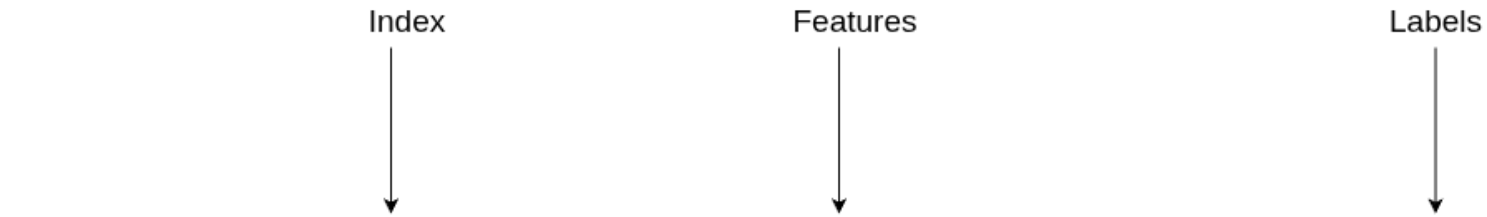
Features, Label et Variables

Feature : une colonne du dataset utilisée comme information d'entrée (ex : âge, prix, classe, sexe).
Label / Target : la variable que l'on cherche à prédire (ex : Survived dans Titanic, Price dans Housing).

Types de variables

- **Numériques** : Age, Fare, Salary
- **Catégorielles** : Sex, Embarked, Ville
- **Binaires** : 0/1 (ex : survie)
- **Textuelles ou images** (données non structurées)

Features, Label et Variables



Exemples / instances →

#	Age	Has_Job	Own_House	Credit_Rating	Class
1	Young	false	false	fair	No
2	Young	false	false	good	No
3	Young	true	false	good	Yes
4	Young	true	true	fair	Yes
5	Young	false	false	fair	No
6	middle	false	false	fair	No
7	middle	false	false	good	No
8	middle	true	true	good	Yes
9	middle	false	true	excellent	Yes
10	middle	false	true	excellent	Yes
11	old	false	true	excellent	Yes
12	old	false	true	good	Yes
13	old	true	false	good	Yes
14	old	true	false	excellent	Yes
15	old	false	false	fair	No

Train set, Test set et Validation

Pourquoi séparer les données ? Pour évaluer la capacité du modèle à généraliser sur de nouvelles données.

- **Train set (Ensemble d'entraînement)** Utilisé pour apprendre les paramètres du modèle. Représente en général **70–80%** du dataset.
- **Test set (Ensemble de test)** Utilisé uniquement à la fin pour mesurer les performances. Représente **20–30%** du dataset.
- **Validation set (facultatif)** Sert à optimiser les hyperparamètres. Dans *train_test_split*, il peut être remplacé par la validation croisée.

Règle essentielle : Le modèle **ne doit jamais voir le test set pendant l'entraînement**.

Qu'est-ce qu'un modèle en Data Science ?

Un **modèle** est un programme mathématique ou statistique qui apprend à faire une tâche à partir de données.

- Il prend des **features** en entrée (âge, prix, taille...).
- Il produit une **prédiction** en sortie (classe, valeur numérique...).
- Il apprend en analysant des exemples et en ajustant ses paramètres.

Exemples de modèles

- Régression linéaire $\rightarrow$ prédire un prix.
- Arbre de décision $\rightarrow$ prédire une catégorie.
- KNN $\rightarrow$ classer un individu par similarité.

Comment un modèle apprend ?

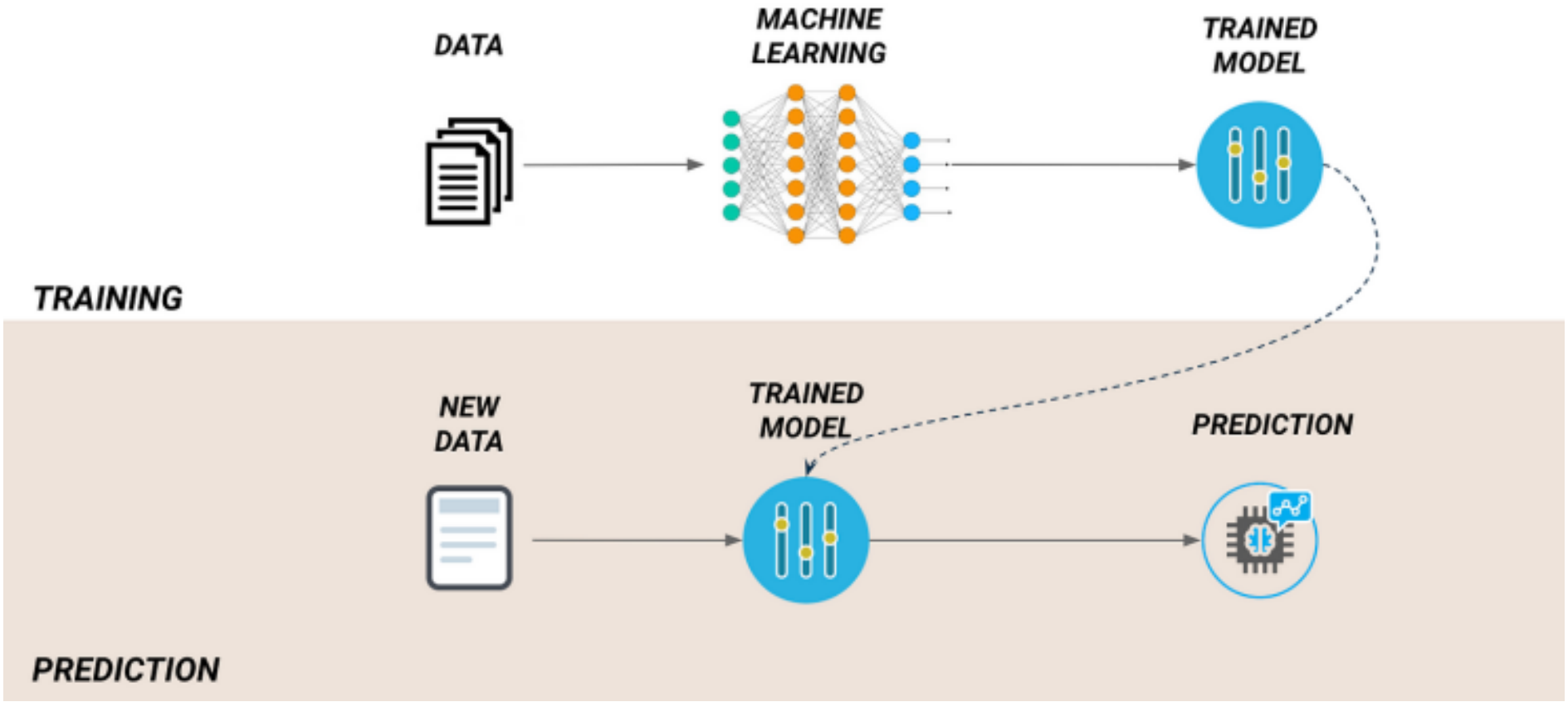
Principe général :

1. On fournit des exemples (train set)
2. Le modèle fait une prédiction
3. On mesure l'erreur (fonction de perte)
4. Le modèle ajuste ses paramètres pour réduire l'erreur
5. On répète jusqu'à ce que l'erreur soit minimale

Important

Un modèle ne mémorise pas les données : il **apprend des relations générales**.

L'essentiel à retenir



Les outils du Data Scientist

Les outils du Data Scientist

Langages :

- Python
- R
- SQL

Bibliothèques Python :

- NumPy – calcul scientifique
- Pandas – manipulation de données
- Matplotlib / Seaborn – visualisation
- Scikit-learn – Machine Learning

Librairies Python essentielles en Data Science

Python est le langage le plus utilisé en Data Science. Voici les bibliothèques indispensables :

- **NumPy** : calcul numérique, tableaux multidimensionnels.
- **Pandas** : manipulation de données, DataFrames.
- **Matplotlib** : graphiques simples.
- **Seaborn** : visualisations statistiques avancées.
- **Scikit-learn** : Machine Learning (classification, régression...).
- **Jupyter** : environnement interactif pour exécuter Python.

Ces librairies seront utilisées tout au long du cours.

NumPy et Pandas : les fondations

NumPy

- Base du calcul scientifique en Python.
- Manipulation rapide de tableaux (`ndarray`).

Pandas

- Offre la structure **DataFrame**, similaire à un tableau Excel.
- Import, nettoyage et analyse de données.
- Outils pour visualisation et statistiques rapides.

Matplotlib et Seaborn : visualisation des données

Matplotlib

- Bibliothèque de base pour tracer des graphiques.
- Très flexible, mais parfois verbeuse.

Seaborn

- Basée sur Matplotlib mais avec un style plus moderne.
- Idéale pour les analyses statistiques et EDA.

Scikit-learn : créer vos premiers modèles

Scikit-learn est la bibliothèque principale de Machine Learning en Python.
Fonctionnalités :

- Classification (logistic regression, KNN, arbres...)
- Régression (linéaire, forêt aléatoire...)
- Clustering (K-means)
- Prétraitement : normalisation, encodage...
- Séparation train/test, validation croisée

Idéal pour débuter dans le Machine Learning.

Rappels Python

Quelques bases indispensables :

- **Types de données** : `int`, `float`, `str`, `bool`
- **Structures principales** : `list`, `tuple`, `dict`, `set`

Boucles et conditions :

```
for i in range(5):
    if i % 2 == 0:
        print(i, "est pair")
```

Fonctions :

```
def carre(x):
    return x**2
```

Bibliothèques essentielles : `NumPy`, `Pandas`, `Matplotlib`, `Seaborn` (voir Notebook 1)

Les DataFrames en Python

Définition : Un **DataFrame** est une structure de données bidimensionnelle (tableau) fournie par la bibliothèque `pandas`. C'est l'équivalent d'une feuille Excel : chaque colonne a un nom et un type de données.

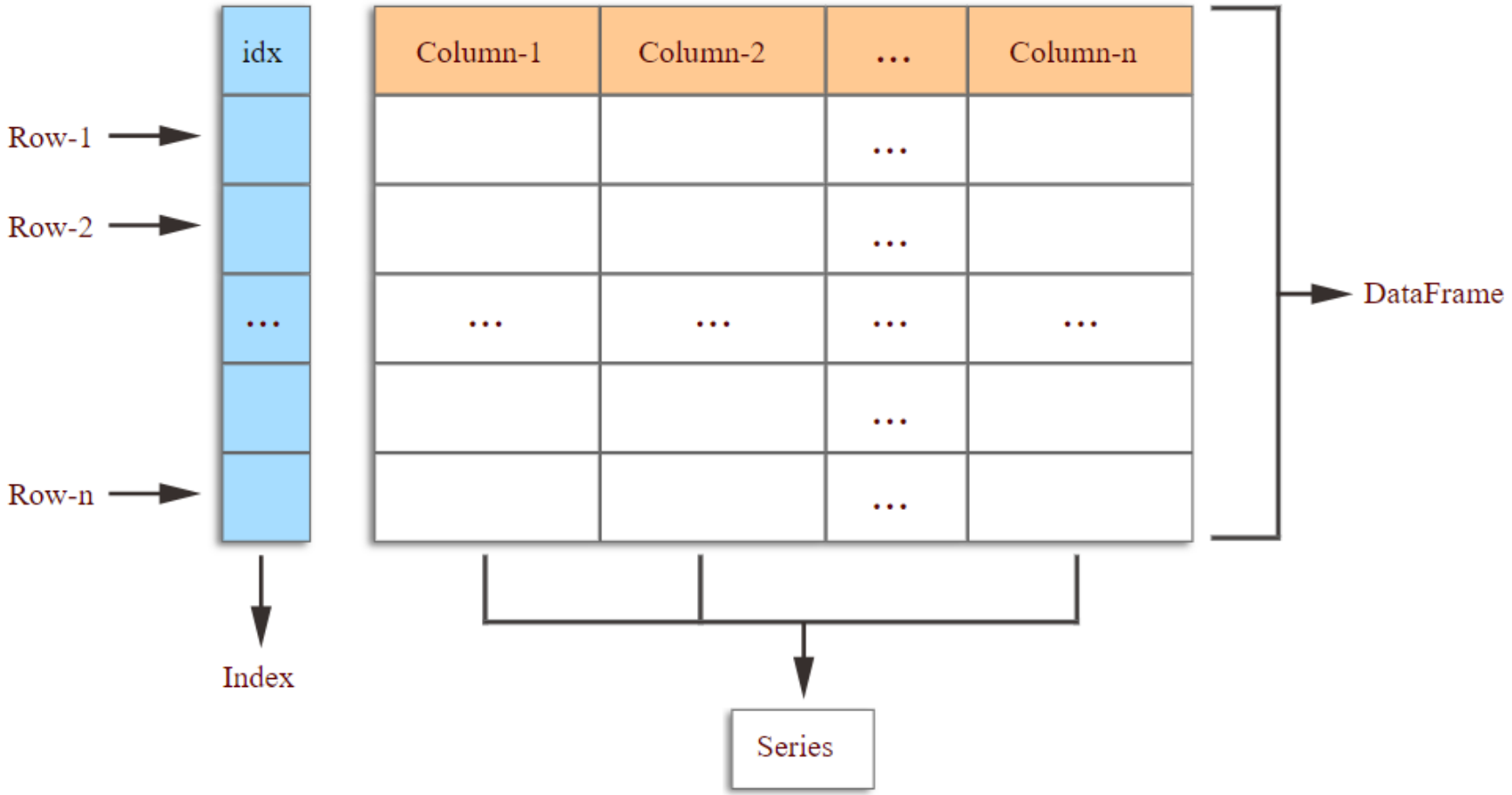
Caractéristiques principales :

- Contient des lignes (observations) et des colonnes (variables)
- Index automatique ou personnalisé
- Manipulation et analyse rapides
- Compatible avec CSV, Excel, SQL, JSON, etc.

Importer pandas :

```
import pandas as pd
```

Les DataFrames en Python



Créer et explorer un DataFrame

Créer un DataFrame à partir d'un dictionnaire :

```
import pandas as pd

data = {
    "Name": ["Alice", "Bob", "Charlie"],
    "Age": [25, 30, 35],
    "City": ["Paris", "Lyon", "Marseille"]
}

df = pd.DataFrame(data)
print(df)
```

Créer et explorer un DataFrame

Créer un DataFrame à partir d'un dictionnaire :

```
import pandas as pd

data = {
    "Name": ["Alice", "Bob", "Charlie"],
    "Age": [25, 30, 35],
    "City": ["Paris", "Lyon", "Marseille"]
}

df = pd.DataFrame(data)
print(df)
```

Explorer les données :

```
df.head()       # premi res lignes
df.info()       # r sum du tableau
df.describe()   # statistiques descriptives
df.columns      # noms des colonnes
df.shape        # dimensions du DataFrame
```

Manipulations simples avec Pandas

Exemples de base :

```
# Sélection d'une colonne
df["City"]

# Filtrer les lignes
df[df["Age"] > 28]

# Ajouter une colonne
df["Salary"] = [3000, 3500, 4000]

# Trier les données
df.sort_values(by="Age", ascending=False)

# Moyenne d'une colonne
df["Age"].mean()
```

Astuce : Explorez toujours vos données avant de les transformer !

Rappels statistiques

Notions essentielles de statistiques descriptives :

- **Moyenne (Mean)** — mesure de la tendance centrale : $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$
- **Médiane (Median)** — valeur centrale d'un ensemble de données trié. Elle sépare les données en deux parties égales.
- **Écart-type (Standard deviation)** — mesure de la dispersion autour de la moyenne : $s = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2}$
- **Variance** — carré de l'écart-type, indique la variabilité des données.

Rappels statistiques

- **Corrélation (r)** — mesure la relation linéaire entre deux variables : $-1 \leq r \leq 1$ *(r proche de 1 $\rightarrow$ forte corrélation positive, r proche de -1 $\rightarrow$ négative, r 0 $\rightarrow$ aucune corrélation)*
- **Distribution** — représentation graphique (histogramme, boxplot) pour observer la forme des données. *En Python :* `df.describe()`, `df.corr()`, `plt.hist()`, `sns.boxplot()`

Travaux Pratiques

Qu'est-ce que Jupyter Notebook ?

Jupyter Notebook est un environnement interactif pour écrire et exécuter du code Python.

- Interface web simple et intuitive.
- Mélange de code, graphiques et texte explicatif.
- Très utilisé en Data Science, Machine Learning et enseignement.

Format du fichier : `.ipynb`

Comment fonctionne Jupyter Notebook ?

Principe :

- Vous créez un **notebook** composé de cellules.
- Les cellules peuvent contenir :
  - du code Python,
  - du texte formaté (Markdown),
  - des graphiques,
  - des tableaux.
- Vous exécutez chaque cellule indépendamment.

Avantages :

- Idéal pour l'expérimentation et l'analyse.
- Permet de documenter et visualiser votre travail au même endroit.

Comment lancer Jupyter Notebook ?

1. **Activer l'environnement virtuel (recommandé)**
   - Linux/macOS : `source venv/bin/activate`
   - Windows : `venv\Scripts\activate`
2. **Lancer Jupyter Notebook**
   - `jupyter notebook`
   - Le navigateur s'ouvrira automatiquement.
3. **Lancer JupyterLab (plus moderne)**
   - `jupyter lab`

Travail pratique 1 — Explorer un dataset

Objectif : Découvrir la structure d'un dataset à l'aide de Pandas. **Tâches :**

1. Importer un fichier CSV (ex : `Titanic.csv`).
2. Afficher les 5 premières lignes (`df.head()`).
3. Observer les colonnes, types et valeurs manquantes (`df.info()`, `df.isna().sum()`).

Travail pratique 2 — Décrire les données

Objectif : Produire des statistiques descriptives simples. **Tâches :**

- Calculer la moyenne, médiane et écart-type (`df.describe()`).
- Analyser la distribution d'une variable (ex : âge, prix).
- Identifier les variables numériques et catégorielles.

Outil : Python (Pandas) dans Google Colab ou Jupyter.

Résumé du chapitre

- Compréhension du rôle et de l'importance de la Data Science.
- Introduction au cycle de vie d'un projet de données.
- Découverte des rôles, outils et langages utilisés.
- Manipulation d'un premier dataset avec Python.

Compétences acquises

Savoir explorer, décrire et comprendre un jeu de données simple.

Évaluation formative

Mini-exercice — Analyse exploratoire

1. TP noté : Pour aller plus loin avec les DataFrames

Livrable

Un notebook Python commenté avec 5 à 10 lignes d'analyse.

Auto-évaluation

- Puis-je expliquer ce qu'est la Data Science et ses domaines d'application ?
- Ai-je compris les étapes d'un projet de données ?
- Suis-je capable d'explorer un dataset avec Pandas ?
- Ai-je compris les bases de Python et des statistiques simples ?