

Big Data et Data Mining

- Dispensé par **MWAMBA KASONGO Dahouda**
- Docteur en génie logiciel et systèmes d'information
- Machine and Deep Learning Engineer
- Assisté par Master Yani KALOMBA
- E-mail : dahouda37@gmail.com
dahouda37@hanyang.ac.kr
dahouda37@upl-univ-ac
- Tel.: +243 99 66 55 265



Fiche matière

ISTA / Kolwezi

Faculté de Sciences Informatiques

Enseignant responsable de la matière : **Dr. Dahouda MWAMBA**

Assisté par MSc. Yani KALOMBA

Contact: dahouda37@gmail.com / dahouda37@hanyang.ac.kr / dahouda37@upl-univ.ac

Matière : **Big Data et Data Mining**

Domaine/ Filiere/Specialite : **GENIE LOGICIEL**

Crédit: **3 [45H]**

Volume horaire d'enseignement hebdomadaire: **Cours (Nombre d'heures par semaine): H**

Big Data et Data Mining

CHAPITRE 1 INTRODUCTION AU BIG DATA

Le **Big Data** désigne un ensemble de technologies et de méthodes permettant de traiter et d'analyser des volumes massifs de données, souvent en temps réel.

Il est utilisé pour extraire des informations précieuses qui peuvent aider à la prise de décision, l'optimisation des processus et l'innovation dans divers domaines



CHAPITRE 1 INTRODUCTION AU BIG DATA

1.1 Définition et importance du Big Data

❖ Définition du Big Data

Le Big Data se réfère à des ensembles de données **extrêmement volumineux, variés et générés à grande vitesse**, qui ne peuvent pas être traités efficacement par les bases de données traditionnelles.

Ces données proviennent de diverses sources, telles que :

- ✓ Les **réseaux sociaux** (publications, likes, partages, commentaires).
- ✓ Les **capteurs IoT** (données en temps réel des objets connectés).
- ✓ Les **transactions financières** (achats, paiements en ligne).
- ✓ Les **données médicales** (dossiers patients, imagerie médicale).



CHAPITRE 1 INTRODUCTION AU BIG DATA

1.1 Définition et importance du Big Data

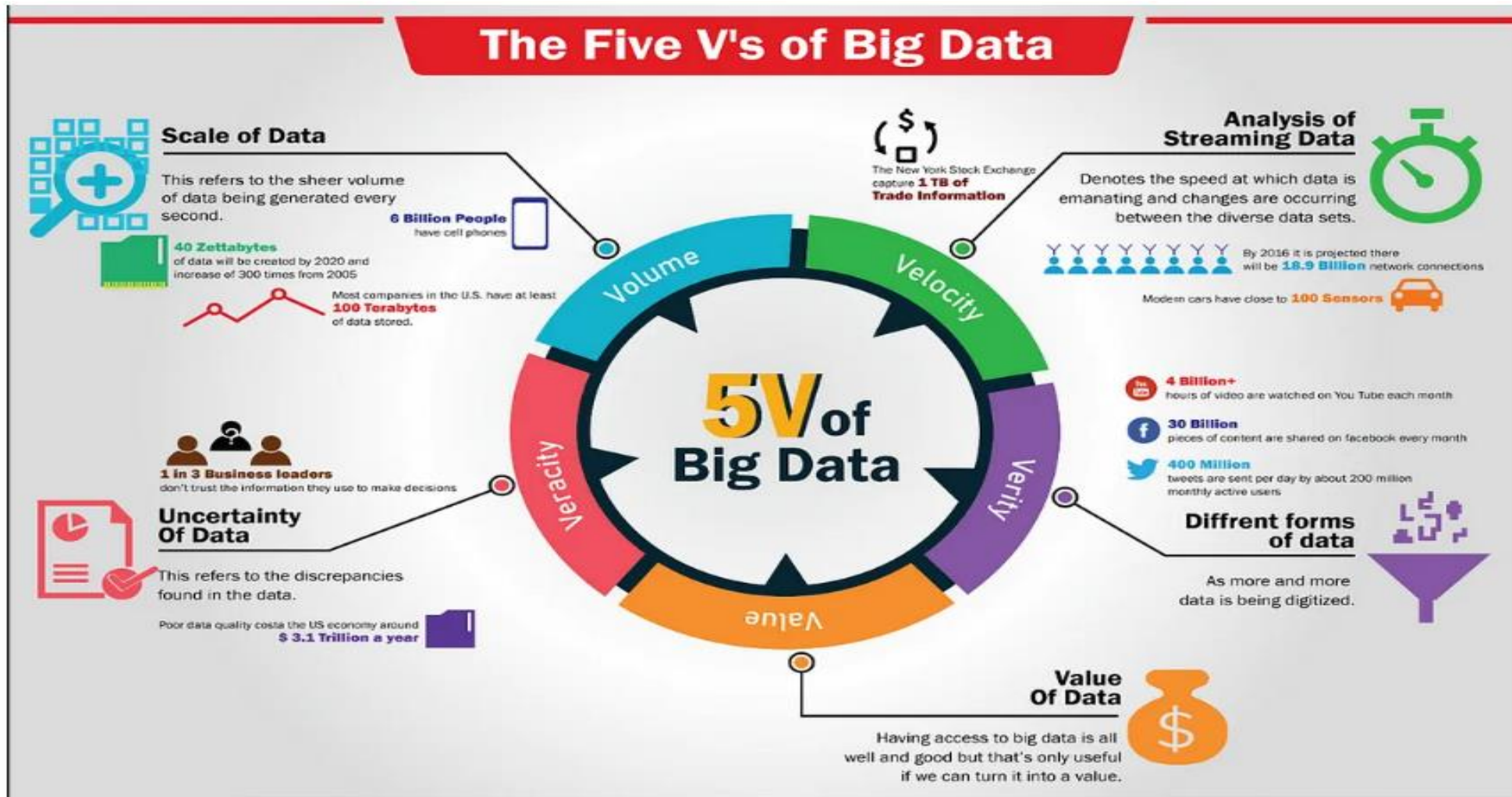
❖ Importance du Big Data

Le Big Data est crucial pour plusieurs raisons :

- ✓ **Meilleure prise de décision** : Grâce à l'analyse de grandes quantités de données, les entreprises et organisations peuvent identifier des tendances et optimiser leurs stratégies.
- ✓ **Amélioration de l'expérience utilisateur** : Par exemple, Netflix et Amazon utilisent le Big Data pour recommander du contenu personnalisé.
- ✓ **Innovation technologique** : L'intelligence artificielle et le Machine Learning utilisent le Big Data pour entraîner des modèles précis.
- ✓ **Efficacité opérationnelle** : Les entreprises optimisent leurs coûts et leurs processus en analysant des flux de données en temps réel.
- ✓ **Avancées dans la recherche scientifique** : En médecine, l'analyse de grandes bases de données permet de découvrir de nouveaux traitements et d'identifier des corrélations entre maladies.

CHAPITRE 1 INTRODUCTION AU BIG DATA

1.2 Les 5V du Big Data



CHAPITRE 1 INTRODUCTION AU BIG DATA

1.2 Les 5V du Big Data

❑ Les 5V sont les principales caractéristiques qui définissent le Big Data :

✓ **Volume**: Quantité massive de données générées chaque seconde (ex. : Google traite 40 000 recherches par seconde).

Exemples : Réseaux sociaux, transactions bancaires, vidéos en streaming.

✓ **Vitesse** : Rapidité avec laquelle les données sont générées, stockées et analysées.

Exemples : Analyse des tendances Twitter en temps réel, mises à jour boursières instantanées.

✓ **Variété** : Multiplicité des formats de données (structurées, semi-structurées et non structurées).

Exemples : Texte, images, vidéos, audio, données issues des capteurs.

✓ **Véracité** : Fiabilité et qualité des données (élimination des erreurs et incohérences).

Exemples : Fake news sur les réseaux sociaux, erreurs dans les bases de données médicales.

✓ **Valeur** : Capacité à extraire de la valeur exploitable à partir des données collectées.

Exemples : Personnalisation des recommandations Netflix, prédictions en maintenance industrielle.

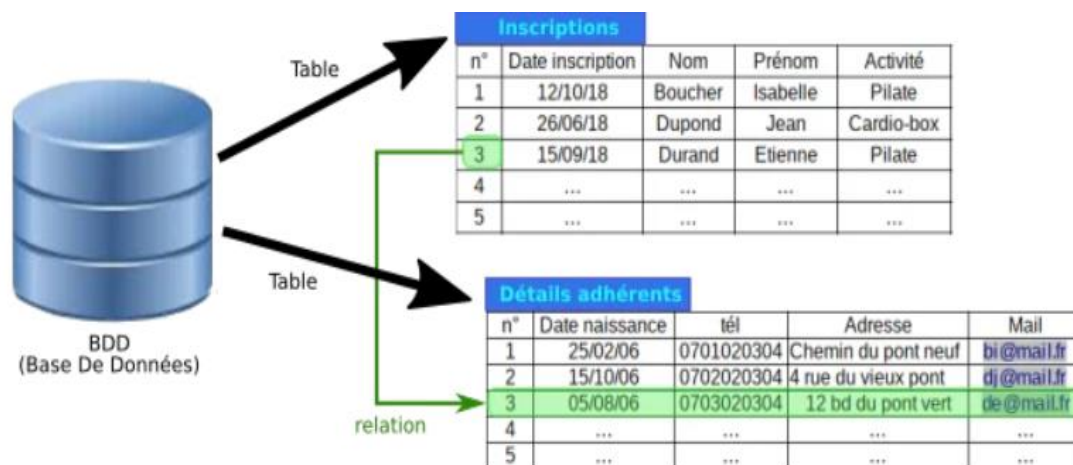
CHAPITRE 1 INTRODUCTION AU BIG DATA

1.3 Différences entre bases de données traditionnelles et Big Data

1.3.1 Données traditionnelles

Les données traditionnelles sont des données structurées et stockées dans des bases de données pouvant être gérées à partir d'un seul ordinateur ; il se présente sous la forme d'un tableau contenant des valeurs numériques ou textuelles.

Le terme « données traditionnelles » ne fait pas partie du vernaculaire officiel. C'est quelque chose que nous introduisons pour plus de clarté. Nous pensons que cela aide à souligner la distinction entre les méga données et les non méga données.



Une base de données (BDD) est une technique pour stocker des données de manière structurée.

Généralement une BDD, contient plusieurs tables qui peuvent être reliées entre elles ou pas.

Lorsque les tables sont reliées entre elles, on parle de Base de Données Relationnelles.

CHAPITRE 1 INTRODUCTION AU BIG DATA

1.3 Différences entre bases de données traditionnelles et Big Data

1.3.2 Big Data

Le Big Data est plus volumineux que les données traditionnelles, mais pas au sens trivial. Il s'agit de données extrêmement volumineuses, réparties sur un réseau d'ordinateurs, mais elles ne se caractérisent pas seulement par leur volume.

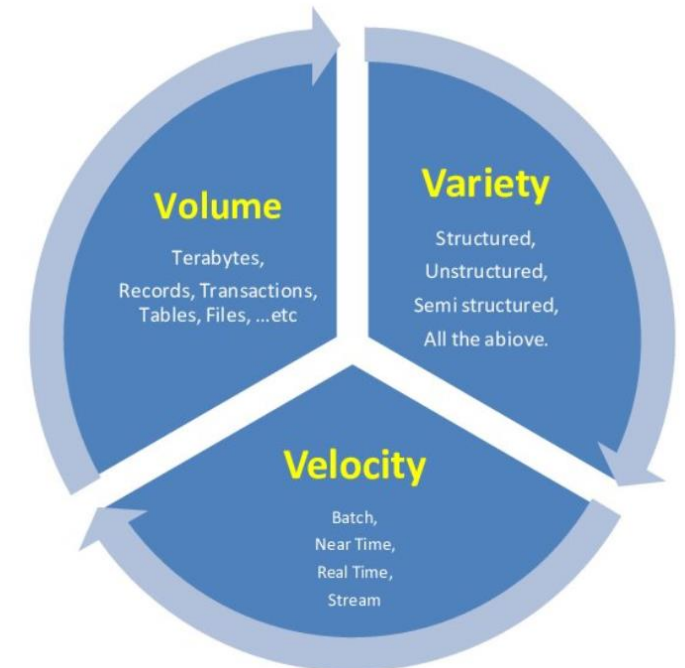
Ces données peuvent être sous différents formats ; il peut être structuré, semi-structuré ou non structuré; et vous verrez souvent des méga données caractérisées par la lettre « V ». Cela découle des « 3V du big data » :

✓ **Variété** - chiffres, texte, mais aussi images, audio, données mobiles, etc.

Les méga données peuvent être dans différents formats.

✓ **Vélocité** - elle est récupérée et calculée en temps réel

✓ **Volume** - les données volumineuses sont mesurées en peta-exaoctets



CHAPITRE 1 INTRODUCTION AU BIG DATA

1.3 Différences entre bases de données traditionnelles et Big Data

Critères	Bases de données traditionnelles (SQL)	Big Data (NoSQL, Hadoop, Spark)
Volume	Gère des téraoctets (To)	Gère des pétaoctets et exaoctets
Structure	Structurée (tables, relations)	Structurée, semi-structurée et non structurée
Stockage	Serveurs centralisés	Systèmes distribués (HDFS, S3, BigQuery)
Traitement des données	Transactions classiques	Traitement parallèle et massivement distribué
Vitesse	Temps de réponse limité	Temps de réponse rapide en temps réel
Exemples	MySQL, PostgreSQL, Oracle	Hadoop, MongoDB, Apache Spark

- Les bases de données traditionnelles (SQL) sont optimisées pour des **données structurées et des transactions complexes**, mais deviennent inefficaces face aux volumes massifs de données.
- Les solutions **Big Data** utilisent des architectures distribuées (comme **Hadoop et Spark**) permettant de traiter rapidement d'énormes quantités de données hétérogènes.

CHAPITRE 1 INTRODUCTION AU BIG DATA

1.4 Cas d'utilisation du Big Data

☐ Santé

- **Analyse prédictive** : Identification des patients à risque grâce à l'IA.
 - **Médecine personnalisée** : Adaptation des traitements en fonction des données génétiques.
 - **Gestion hospitalière** : Optimisation des flux de patients et des ressources médicales.
- ✓ Exemple : IBM Watson analyse des milliers de publications médicales pour aider au diagnostic du cancer.

☐ Finance et assurance

- **Détection de fraudes** : Analyse des comportements suspects en temps réel.
- **Trading algorithmique** : Utilisation du Machine Learning pour anticiper les fluctuations du marché.
- **Personnalisation des offres bancaires** : Analyse du profil des clients pour proposer des services adaptés.

Exemple : Mastercard utilise le Big Data pour détecter les transactions frauduleuses en temps réel.

CHAPITRE 1 INTRODUCTION AU BIG DATA

1.4 Cas d'utilisation du Big Data

❑ Marketing et commerce

- **Publicité ciblée** : Personnalisation des campagnes marketing en fonction des comportements des clients.
- **Optimisation de la relation client** : Analyse des avis clients pour améliorer les services.
- **Prédiction des tendances de consommation** : Analyse des comportements d'achat.
- ✓ **Exemple** : Amazon utilise le Big Data pour recommander des produits en fonction de l'historique des achats.

❑ Internet des Objets (IoT)

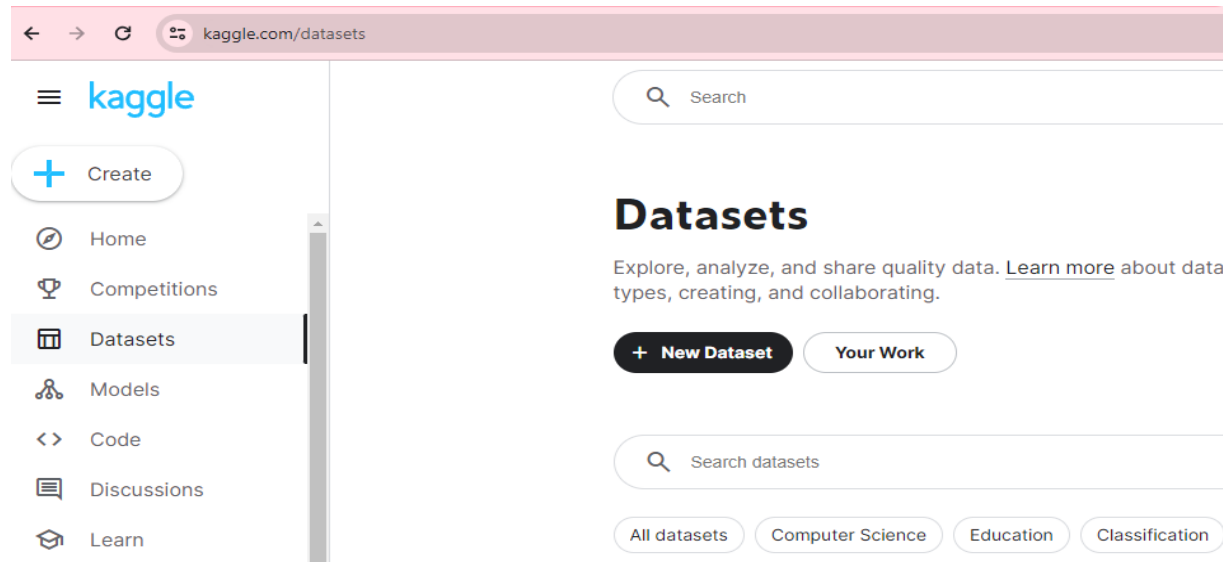
- **Villes intelligentes** : Gestion optimisée du trafic et de l'énergie dans les smart cities.
- **Maintenance prédictive** : Anticipation des pannes dans les machines industrielles.
- **Domotique** : Optimisation de la consommation énergétique des maisons connectées.
- ✓ **Exemple** : Tesla collecte et analyse les données de ses véhicules en temps réel pour améliorer l'expérience utilisateur et la sécurité.

CHAPITRE 1 INTRODUCTION AU BIG DATA

1.5 Sources de données publiques

Il existe de nombreuses sources où vous pouvez obtenir des données pour l'analyse de la science des données, en fonction de vos intérêts spécifiques et du type d'analyse que vous souhaitez effectuer. Voici quelques options populaires :

1. Kaggle : Kaggle est une plate-forme de concours de science des données, mais elle héberge également un grand nombre d'ensembles de données disponibles gratuitement pour l'exploration et l'analyse. Lien du Kaggle : <https://www.kaggle.com/datasets>



CHAPITRE 1 INTRODUCTION AU BIG DATA

1.5 Sources de données publiques

2. Référentiel UCI Machine Learning : le référentiel UCI Machine Learning est un ensemble de bases de données, de théories de domaine et de générateurs de données largement utilisés par la communauté de Machine Learning. Lien de UCI : <https://archive.ics.uci.edu/>

[Datasets](#) [Contribute Dataset](#) [About Us](#)

Welcome to the UC Irvine Machine Learning Repository

We currently maintain 664 datasets as a service to the machine learning community. Here, you can donate and find datasets used by millions of people all around the world!

[VIEW DATASETS](#) [CONTRIBUTE A DATASET](#)

Popular Datasets



Iris

A small classic dataset from Fisher, 1936. One of the earliest known datasets used for ev...

🔍 Classification 📊 150 Instances 📋 4 Features



Dry Bean

Images of 13,611 grains of 7 different registered dry beans were taken with a high-resol...

🔍 Classification 📊 13.61K Instances 📋 16 Features

New Datasets



PhiUSIIL Phishing URL (Website)

PhiUSIIL Phishing URL Dataset is a substantial dataset comprising 134,850 legitimate an...

🔍 Classification 📊 235.8K Instances 📋 54 Features



RT-IoT2022

The RT-IoT2022, a proprietary dataset derived from a real-time IoT infrastructure, is intro...

🔍 Classification, Regressi... 📊 123.12K Instances 📋 84 Features

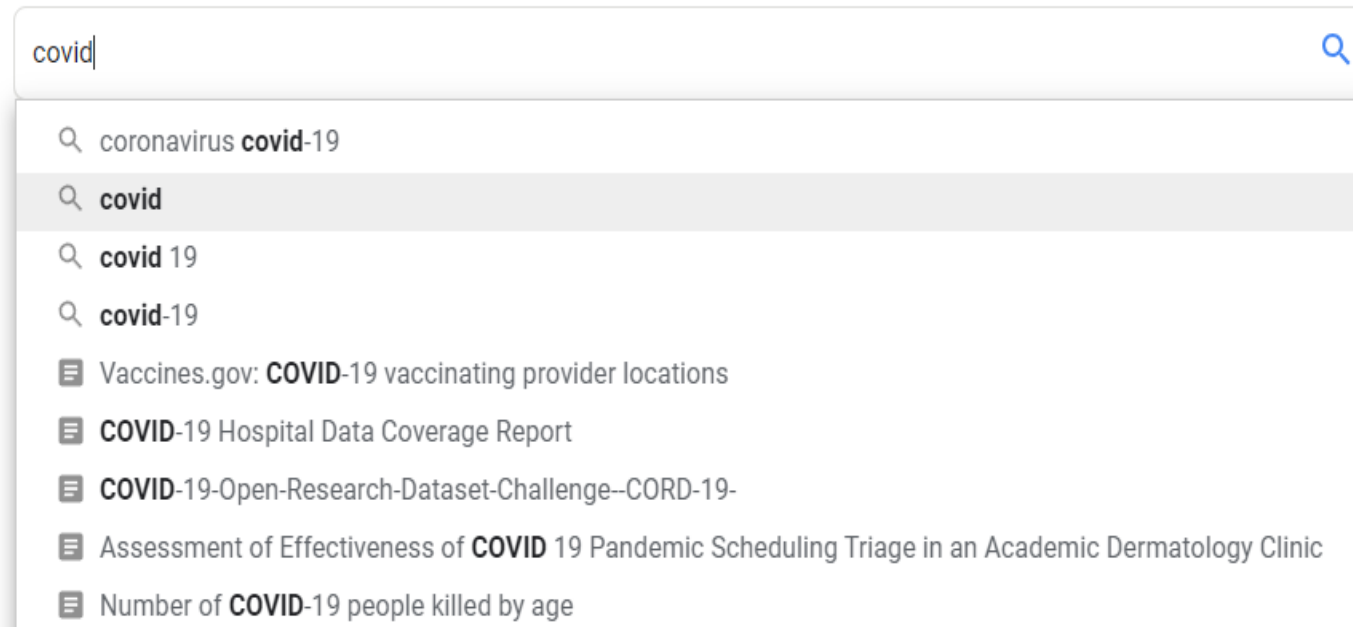
CHAPITRE 1 INTRODUCTION AU BIG DATA

1.5 Sources de données publiques

3. Recherche d'ensembles de données Google : Google Dataset Search vous aide à trouver des ensembles de données stockés sur le Web.

C'est un outil utile pour découvrir des ensembles de données provenant de diverses sources : <https://datasetsearch.research.google.com/>

Dataset Search



CHAPITRE 1 INTRODUCTION AU BIG DATA

1.5 Sources de données publiques

4. **Portails de données ouvertes gouvernementaux** : de nombreux gouvernements proposent des portails de données ouvertes où vous pouvez trouver des ensembles de données liés à la démographie, à l'économie, à la santé, etc. Lien : <https://data.gov/>

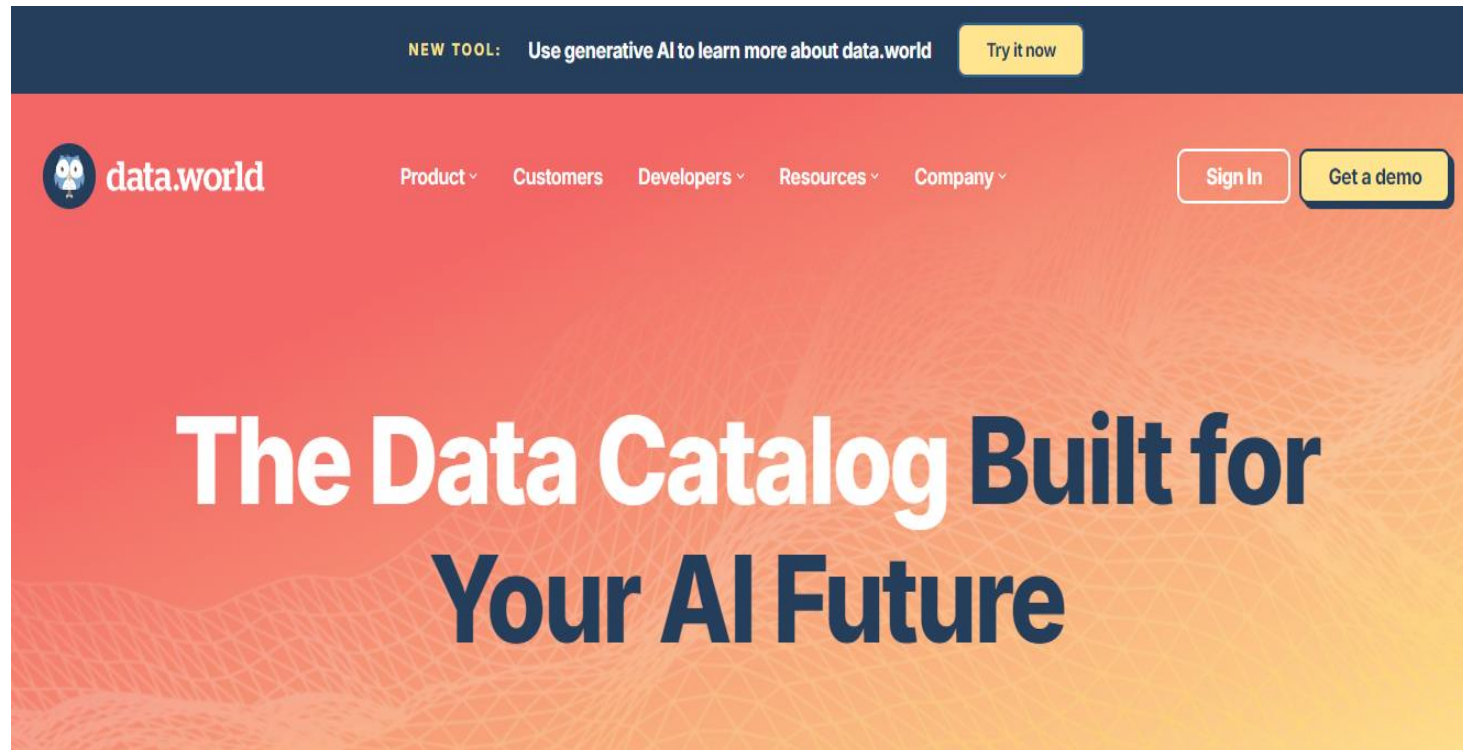


CHAPITRE 1 INTRODUCTION AU BIG DATA

1.5 Sources de données publiques

5. **Data.world** : Data.world est une plateforme où vous pouvez trouver et partager des ensembles de données.

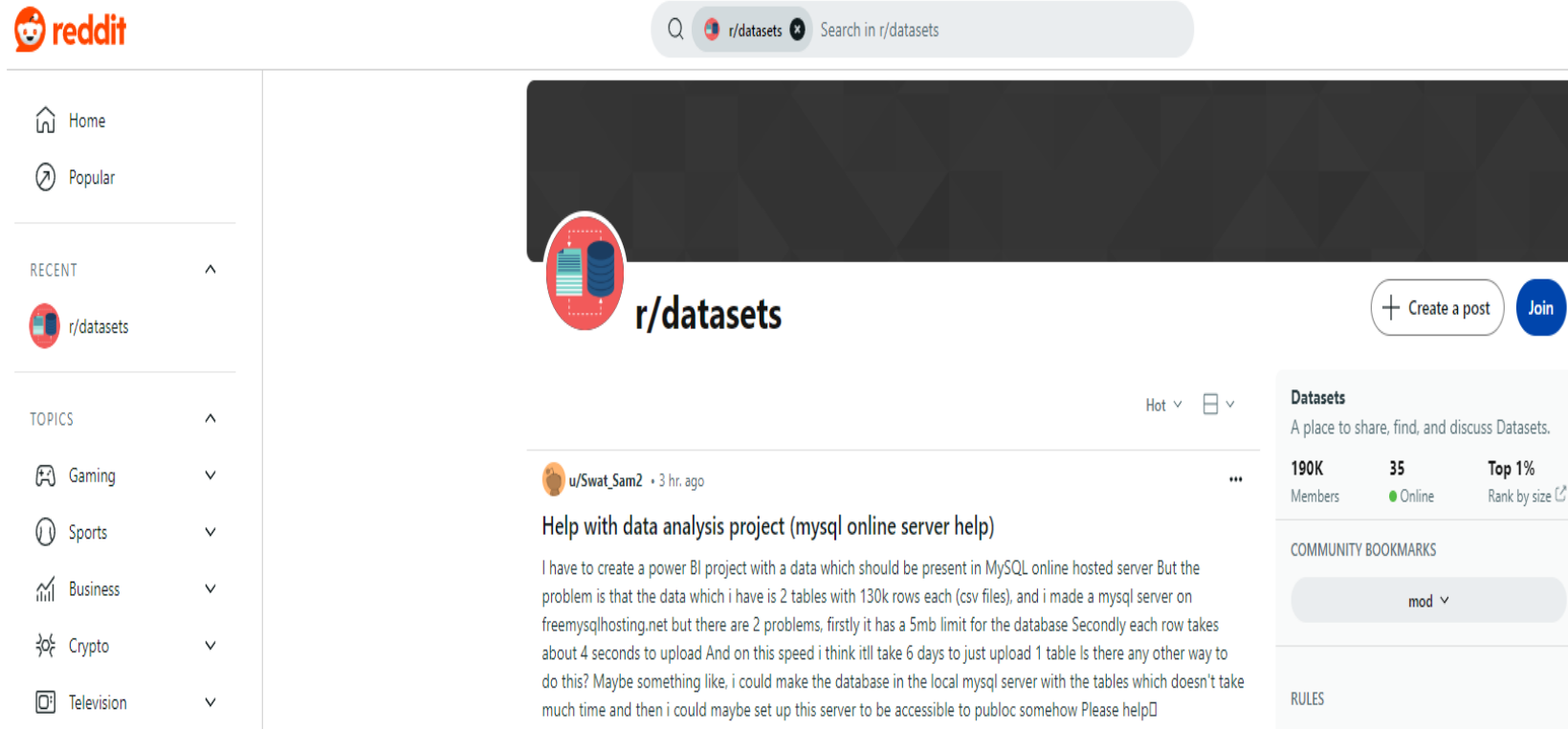
Il héberge une gamme diversifiée d'ensembles de données fournis par la communauté. Lien : <https://data.world/>



CHAPITRE 1 INTRODUCTION AU BIG DATA

1.5 Sources de données publiques

6. Ensembles de données Reddit : le sous-reddit r/datasets est une communauté où les gens partagent et demandent des ensembles de données. Vous pourriez trouver des ensembles de données intéressants ici <https://www.reddit.com/r/datasets/>



The screenshot shows the Reddit interface for the r/datasets subreddit. On the left is a sidebar with the Reddit logo, navigation links (Home, Popular), a 'RECENT' section listing 'r/datasets', and a 'TOPICS' section with categories like Gaming, Sports, Business, Crypto, and Television. The main content area features the subreddit header with a search bar, a banner image, the 'r/datasets' name, and buttons for 'Create a post' and 'Join'. Below the header is a post by user 'u/Swat_Sam2' titled 'Help with data analysis project (mysql online server help)'. The post text describes a user's difficulty with uploading CSV files to a MySQL server on freemysqlhosting.net. On the right side of the post, there is a summary box for the subreddit: 'Datasets' with a description 'A place to share, find, and discuss Datasets.', '190K Members', '35 Online', and 'Top 1% Rank by size'. Below this are sections for 'COMMUNITY BOOKMARKS' (showing a 'mod' button) and 'RULES'.

CHAPITRE 2 ÉCOSYSTEME DU BIG DATA



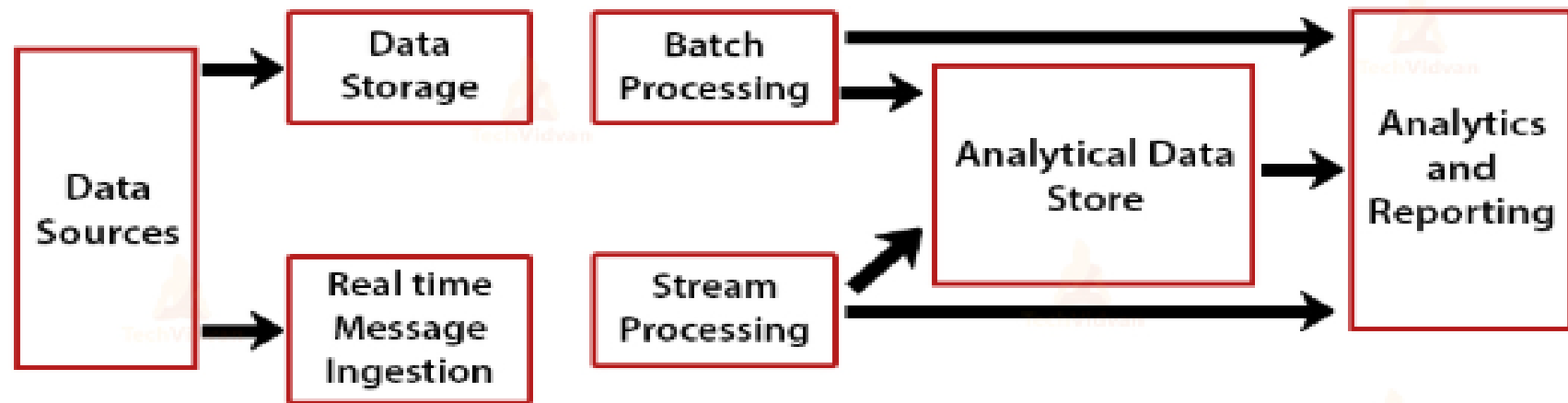
CHAPITRE 2 ÉCOSYSTEME DU BIG DATA

Ce chapitre explore les **technologies et infrastructures** essentielles à la gestion et au traitement des données massives.

2.1 Architecture du Big Data

L'architecture du Big Data repose sur un ensemble de technologies permettant de **stocker**, **traiter** et **analyser** d'énormes volumes de données.

Big Data Architecture



L'architecture du Big Data est la base de l'analyse du Big Data.

Il s'agit du système global utilisé pour gérer de grandes quantités de données afin qu'elles puissent être analysées à des fins commerciales, orienter l'analyse des données et fournir un environnement dans lequel les outils d'analyse du Big Data peuvent extraire des informations commerciales vitales à partir de données autrement ambiguës.

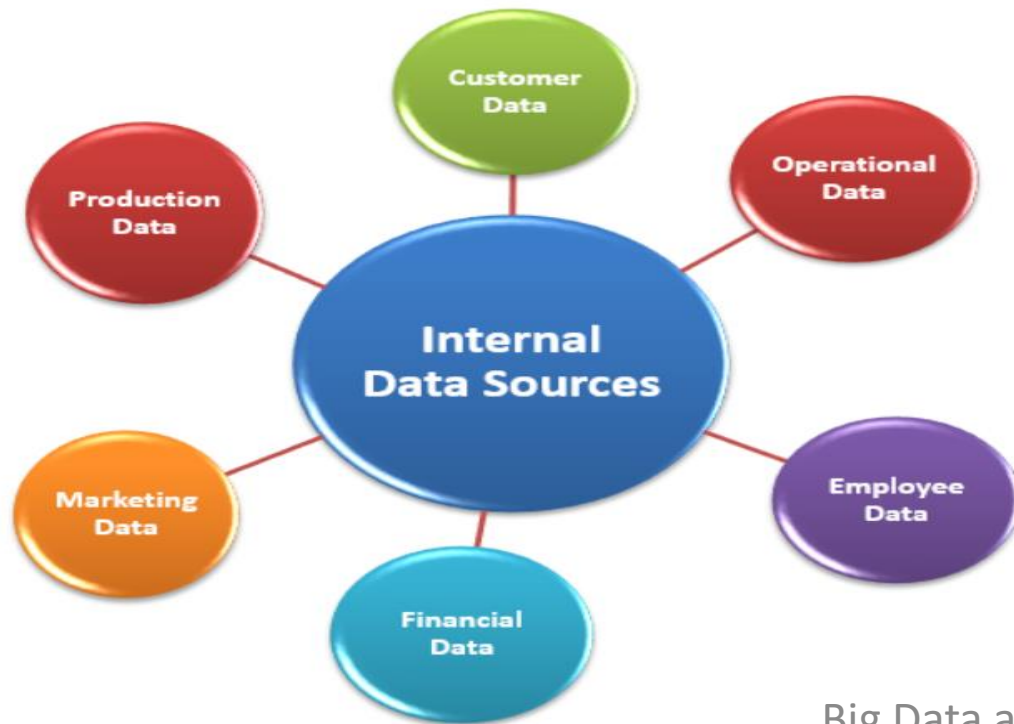
CHAPITRE 2 ÉCOSYSTEME DU BIG DATA

2.1 Architecture du Big Data

L'architecture du Big Data comprend généralement :

a. Ingestion des données

Les données proviennent de diverses sources (capteurs IoT, réseaux sociaux, logs, bases de données, etc.). Elles peuvent être **structurées** (SQL), **semi-structurées** (JSON, XML) ou **non structurées** (images, vidéos, textes).



**SOURCE OF
DATA**

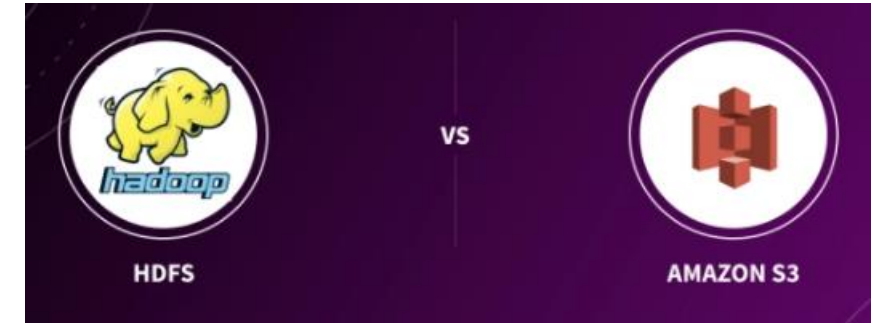
CHAPITRE 2 ÉCOSYSTEME DU BIG DATA

2.1 Architecture du Big Data

b. Stockage des données

Les systèmes de stockage doivent être capables de gérer un **grand volume** de données de manière évolutive.

Outils : Hadoop Distributed File System (**HDFS**), Amazon **S3**, Google BigQuery



c. Traitement des données

Le traitement des données peut être batch (traitement par lots) ou temps réel.

- ✓ **Batch Processing** : Apache Hadoop (**MapReduce**), **Apache Spark**
- ✓ **Streaming Processing** : **Apache Flink**, **Apache Storm**, **Kafka Streams**

d. Analyse et visualisation des données

Les données sont exploitées via des algorithmes de Data Mining et de Machine Learning, puis visualisées.

Outils : **Power BI**, Tableau, etc

CHAPITRE 2 ÉCOSYSTEME DU BIG DATA

2.2 Les bases de données NoSQL (MongoDB, Cassandra, HBase)

Les bases de données NoSQL sont conçues pour **gérer des volumes massifs de données** de manière flexible et performante, contrairement aux bases SQL classiques.

a. MongoDB (Base orientée documents)

- Stocke les données sous forme de **documents JSON/BSON**.
- ✓ Flexible et bien adaptée aux applications web et mobiles.
- ✓ Indexation avancée et forte scalabilité.

Exemple d'utilisation :

Stockage de profils utilisateur pour les applications web.



```
1  {  
2    _id: "5cf0029caff5056591b0ce7d",  
3    firstname: 'Jane',  
4    lastname: 'Wu',  
5    address: {  
6      street: '1 Circle Rd',  
7      city: 'Los Angeles',  
8      state: 'CA',  
9      zip: '90404'  
10   }  
11 }
```

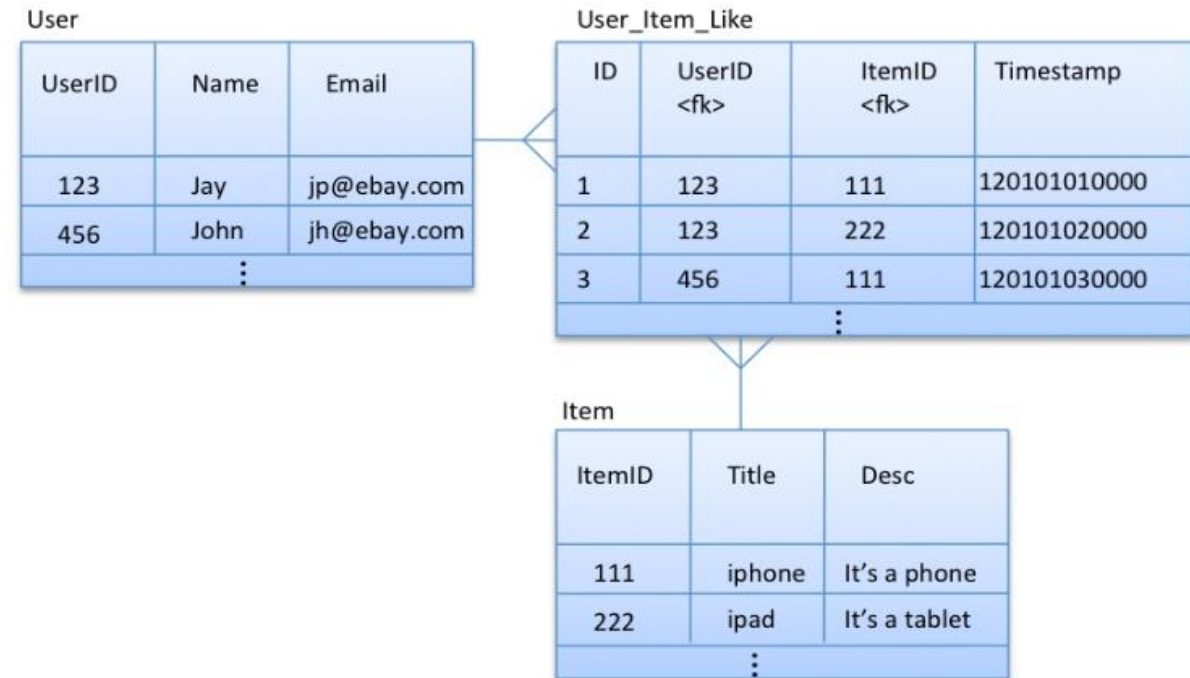

CHAPITRE 2 ÉCOSYSTEME DU BIG DATA

2.2 Les bases de données NoSQL (MongoDB, Cassandra, HBase)

b. Cassandra (Base orientée colonnes)

- Conçue pour une haute disponibilité et une scalabilité horizontale.
- ✓ Réplication automatique sur plusieurs serveurs.
- ✓ Idéale pour les applications distribuées (ex : Netflix).

- ✓ Exemple d'utilisation : Gestion des logs et des données IoT



CHAPITRE 2 ÉCOSYSTEME DU BIG DATA

2.2 Les bases de données NoSQL (MongoDB, Cassandra, HBase)

c. HBase (Base orientée colonnes, intégrée à Hadoop)

- Fonctionne sur HDFS, optimisée pour les grands volumes de données.
- ✓ Accès rapide en lecture/écriture grâce à la structure en colonnes.
- ✓ Utilisée pour des applications nécessitant un accès rapide à des données massives.

Il permet un accès aléatoire, strictement cohérent et en temps réel à des pétaoctets de données.

HBase s'intègre parfaitement à Apache Hadoop et à l'écosystème Hadoop et s'exécute sur le système de fichiers distribué Hadoop (HDFS) ou Amazon S3 à l'aide du système de fichiers Amazon Elastic MapReduce (EMR).

Exemple d'utilisation : Analyse des logs serveurs, moteur de recherche.



CHAPITRE 2 ÉCOSYSTEME DU BIG DATA

2.3 Comparaison SQL vs NoSQL

Critère	Bases SQL (Relationnelles)	Bases NoSQL
Structure	Tables avec schéma rigide	Flexible (JSON, colonnes, clé-valeur, graphes)
Scalabilité	Scalabilité verticale (ajout de CPU, RAM)	Scalabilité horizontale (ajout de serveurs)
Modèle de données	Relationnel (relations entre tables)	Non relationnel (données indépendantes)
Exemples	MySQL, PostgreSQL, Oracle	MongoDB, Cassandra, HBase
Cas d'usage	Transactions bancaires, ERP, CRM	Big Data, réseaux sociaux, IoT

CHAPITRE 2 ÉCOSYSTEME DU BIG DATA

2.4 Technologies de stockage distribué

Le **stockage distribué** permet de gérer de **très grands volumes de données** sur plusieurs serveurs en assurant la **redondance** et la **haute disponibilité**.

a. **HDFS** (Hadoop Distributed File System)

- Système de fichiers distribué utilisé par Apache Hadoop.
 - Stocke les fichiers en blocs sur plusieurs nœuds et assure la redondance.
- HDFS est une **technique de stockage** de base de données qui héberge la conception du système de fichiers distribué.
- Il fonctionne sur du matériel standard, est hautement **tolérant aux pannes** et est conçu à l'aide de matériel à faible coût.
- HDFS stocke de **grandes quantités de données** sur plusieurs machines afin de simplifier l'accès pour ses utilisateurs.
- Tous les fichiers qui utilisent HDFS sont stockés de **manière redondante** pour réduire les pertes de données et améliorer leurs capacités de traitement parallèle.



Avantages : tolérance aux pannes, scalable.

Limitation : conçu pour le batch processing, pas pour l'accès en temps réel.

CHAPITRE 2 ÉCOSYSTEME DU BIG DATA

2.4 Technologies de stockage distribué

b. Amazon S3 (Simple Storage Service)



- Stockage cloud scalable proposé par AWS.
- Utilisé pour le **stockage de fichiers volumineux** et l'intégration avec d'autres services AWS.
- Amazon S3 est un service Web de stockage de données évolutif, peu coûteux et à haut débit fourni par Amazon.
- Amazon S3 est conçu pour la sauvegarde et l'archivage en ligne des données et des programmes d'application.
- Amazon S3 stocke les données sous forme d'objets.
- Chaque objet se compose d'un fichier avec un ID et des métadonnées associés.
- Ces fichiers fonctionnent comme des enregistrements et des répertoires pour stocker des données dans votre région AWS.
- Amazon S3 vous permet de charger, de stocker et de télécharger tout type de fichier jusqu'à 5 To.
- Tous ses abonnés peuvent également accéder aux mêmes capacités de stockage qu'Amazon utilise sur son site Web.
- Amazon S3 est conçu pour donner à l'abonné un contrôle total sur l'accessibilité des données

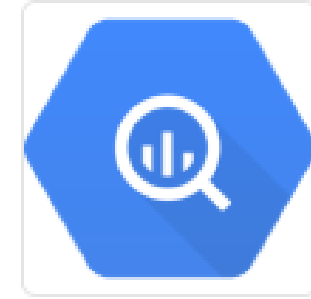
Avantages : sécurité, intégration cloud, pay-as-you-go.

Limitation : accès plus lent que HDFS pour les traitements Big Data intensifs.

CHAPITRE 2 ÉCOSYSTEME DU BIG DATA

2.4 Technologies de stockage distribué

c. Google BigQuery



Google BigQuery

- Service d'entrepôt de données entièrement managé par Google Cloud.
 - Optimisé **pour l'analyse en temps réel** et le traitement SQL sur Big Data.
- BigQuery est l'entrepôt de données d'analyse entièrement géré, à l'échelle du pétaoctet et économique de Google Cloud, qui vous permet d'exécuter des analyses sur de vastes volumes de données en temps quasi réel.
- Avec BigQuery, vous n'avez aucune infrastructure à configurer ou à gérer, ce qui vous permet de vous concentrer sur la recherche d'informations pertinentes à l'aide de GoogleSQL et de profiter de modèles de tarification flexibles avec des options à la demande et à tarif forfaitaire.

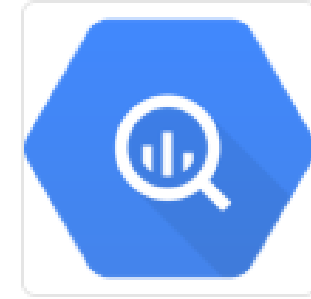
Avantages : rapide, sans gestion de serveur.

Limitation : plus cher pour des traitements fréquents et intensifs.

CHAPITRE 2 ÉCOSYSTEME DU BIG DATA

2.4 Technologies de stockage distribué

c. Google BigQuery



Google BigQuery

- Service d'entrepôt de données entièrement managé par Google Cloud.
 - Optimisé **pour l'analyse en temps réel** et le traitement SQL sur Big Data.
- BigQuery est l'entrepôt de données d'analyse entièrement géré, à l'échelle du pétaoctet et économique de Google Cloud, qui vous permet d'exécuter des analyses sur de vastes volumes de données en temps quasi réel.
- Avec BigQuery, vous n'avez aucune infrastructure à configurer ou à gérer, ce qui vous permet de vous concentrer sur la recherche d'informations pertinentes à l'aide de GoogleSQL et de profiter de modèles de tarification flexibles avec des options à la demande et à tarif forfaitaire.

Avantages : rapide, sans gestion de serveur.

Limitation : plus cher pour des traitements fréquents et intensifs.

CHAPITRE 3 Traitement des Données Massives (Big Data)



CHAPITRE 3 Traitement des Données Massives

Dans ce chapitre, nous allons explorer les différentes approches et technologies utilisées pour traiter efficacement de grandes quantités de données dans un environnement distribué.

3.1 Introduction aux Systèmes Distribués

■ Définition

Un système distribué est un ensemble de plusieurs machines interconnectées qui travaillent ensemble pour exécuter des tâches de calcul ou de stockage.

Ces systèmes permettent de traiter de grandes quantités de données en parallèle et d'optimiser les performances.

■ Pourquoi utiliser des systèmes distribués ?

- ✓ **Scalabilité** : Capacité d'ajouter de nouvelles machines pour gérer des charges de travail croissantes.
- ✓ **Fiabilité et tolérance aux pannes** : Si un nœud tombe en panne, les autres peuvent continuer le traitement.
- ✓ **Traitement parallèle** : Accélération du traitement des données grâce à la répartition des tâches sur plusieurs machines.
- ✓ **Optimisation des ressources** : Répartition de la charge de travail pour une meilleure utilisation des ressources matérielles.

■ Exemples de systèmes distribués

- Hadoop Distributed File System (HDFS) : Système de stockage de Hadoop.
- Amazon S3, Google Cloud Storage : Solutions de stockage distribuées dans le cloud.

CHAPITRE 3 Traitement des Données Massives

3.2 MapReduce : Principes et Fonctionnement

■ Définition

MapReduce est un modèle de programmation développé par Google pour traiter de grandes quantités de données en parallèle sur un cluster de machines.

Il est utilisé dans Hadoop pour exécuter des tâches de traitement massivement parallèles.

Les deux phases principales:

Map (Mappage) :

Divise les données d'entrée en petits morceaux appelés **blocs**.

Applique une fonction de transformation sur chaque bloc pour générer des paires clé-valeur.

Reduce (Réduction) :

Agrège les résultats de la phase Map en regroupant les valeurs ayant la même clé.

Produit un résultat final optimisé.

CHAPITRE 3 Traitement des Données Massives

3.3 Apache Spark : Traitement en Mémoire

■ Présentation de Spark

Apache Spark est un framework de traitement Big Data qui améliore MapReduce en exécutant les calculs en mémoire RAM au lieu d'écrire sur disque entre chaque étape. Il permet un traitement plus rapide et interactif des données.

■ Avantages de Spark par rapport à MapReduce

Traitement en mémoire : Réduit le temps d'accès aux données et améliore les performances.

Plusieurs API disponibles : Supporte Python (PySpark), Java, Scala et R.

Support du streaming : Peut traiter les données en temps réel avec Spark Streaming.

Bibliothèques intégrées :

MLlib : Machine Learning.

GraphX : Traitement des graphes.

Spark SQL : Manipulation de données avec SQL.

CHAPITRE 3 Traitement des Données Massives

3.3 Apache Spark : Traitement en Mémoire

- **Fonctionnement de Spark**

RDD (Resilient Distributed Dataset) : Structure de données distribuée qui stocke les données en mémoire et permet des calculs en parallèle.

Lazy Evaluation : Les transformations ne sont exécutées que lorsque c'est nécessaire, optimisant les performances.

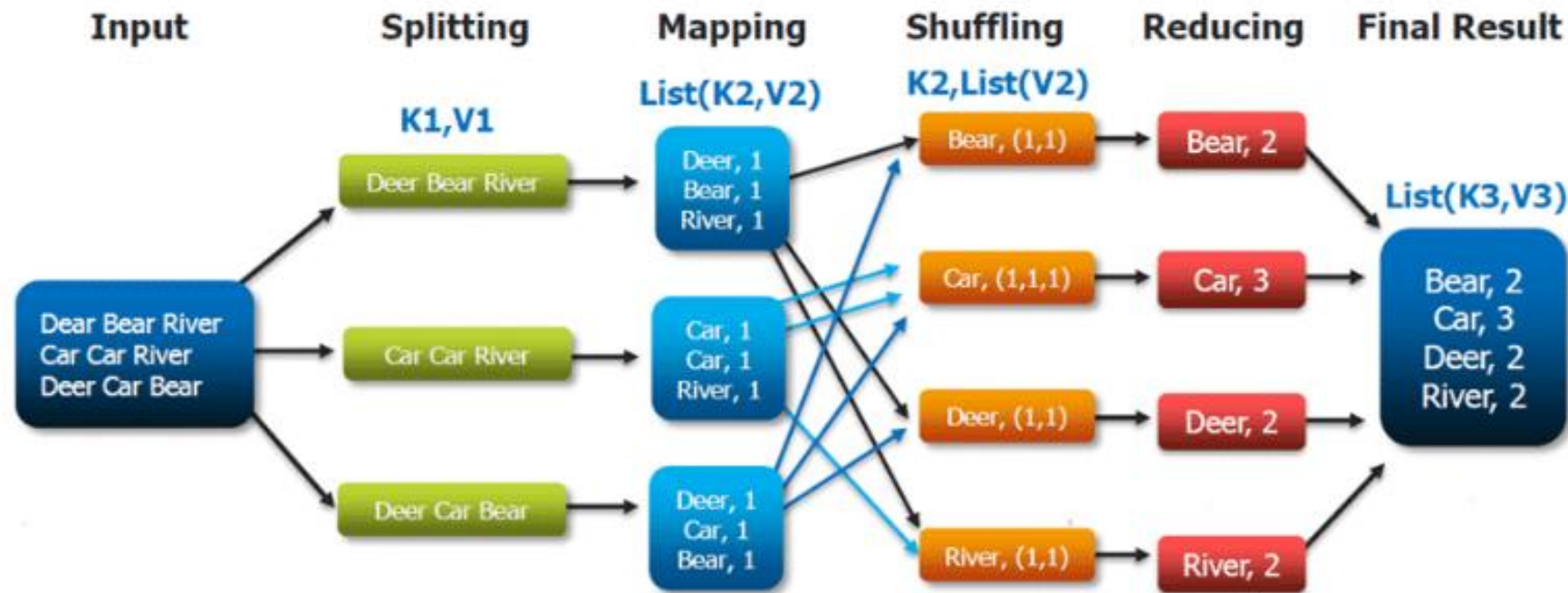
Exécution distribuée : Utilise un cluster de machines pour répartir les tâches.

CHAPITRE 3 Traitement des Données Massives

Dans ce chapitre, nous allons explorer les différentes approches et technologies utilisées pour traiter efficacement de grandes quantités de données dans un environnement distribué.

3.2 MapReduce : Principes et Fonctionnement

The Overall MapReduce Word Count Process



CHAPITRE 3 Traitement des Données Massives

Dans ce chapitre, nous allons explorer les différentes approches et technologies utilisées pour traiter efficacement de grandes quantités de données dans un environnement distribué.

3.2 MapReduce : Principes et Fonctionnement

■ Avantages de MapReduce

Traitement massivement parallèle : Exploite la puissance des clusters pour accélérer les calculs.

Tolérance aux pannes : Réexécute les tâches en cas d'échec d'un nœud.

Simplicité : Facilite le traitement des données massives sans gérer les détails de la distribution des tâches.

■ Limites de MapReduce

Lenteur : Temps d'attente élevé à cause de l'écriture des données sur le disque après chaque phase.

Non adapté au temps réel : Pas optimisé pour le traitement en streaming.

CHAPITRE 4 Introduction au data mining

Fin CHAPITRE 3



CHAPITRE 4 Introduction au Data Mining

4.1 Qu'est-ce que le data mining (DM) ?

Le data mining est le processus de découverte de modèles, de corrélations et d'informations utiles à partir de grands ensembles de données. Il est largement utilisé dans divers domaines tels que les affaires, la santé et la finance.



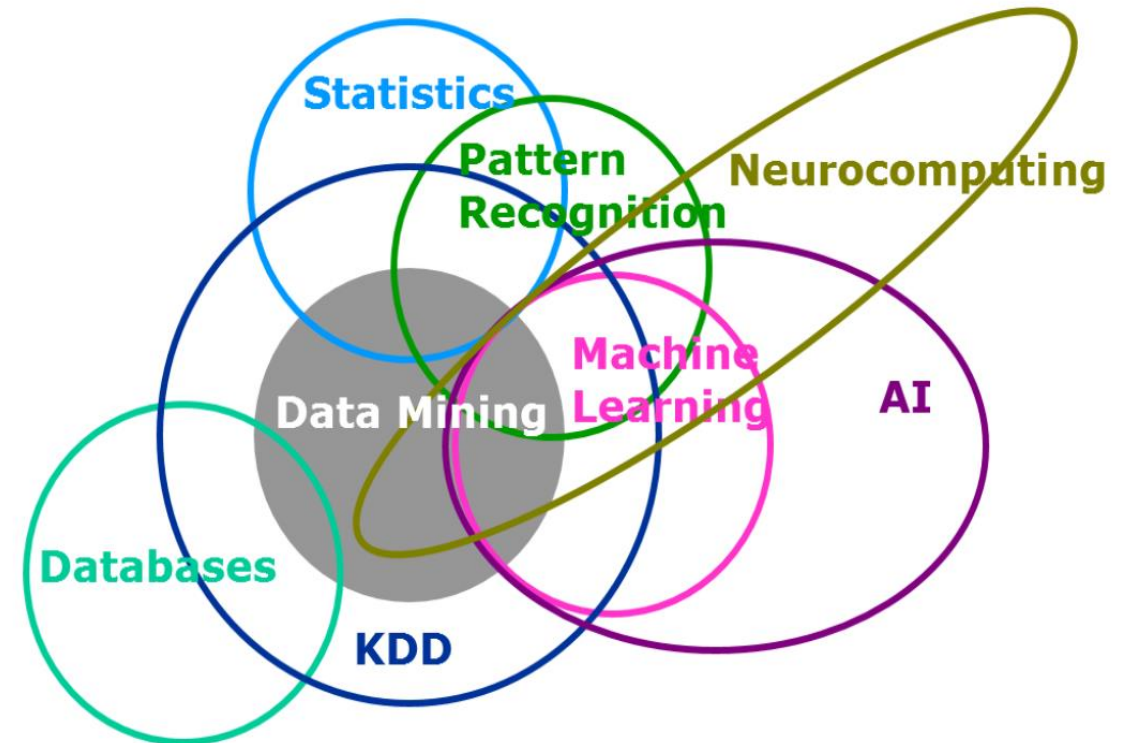
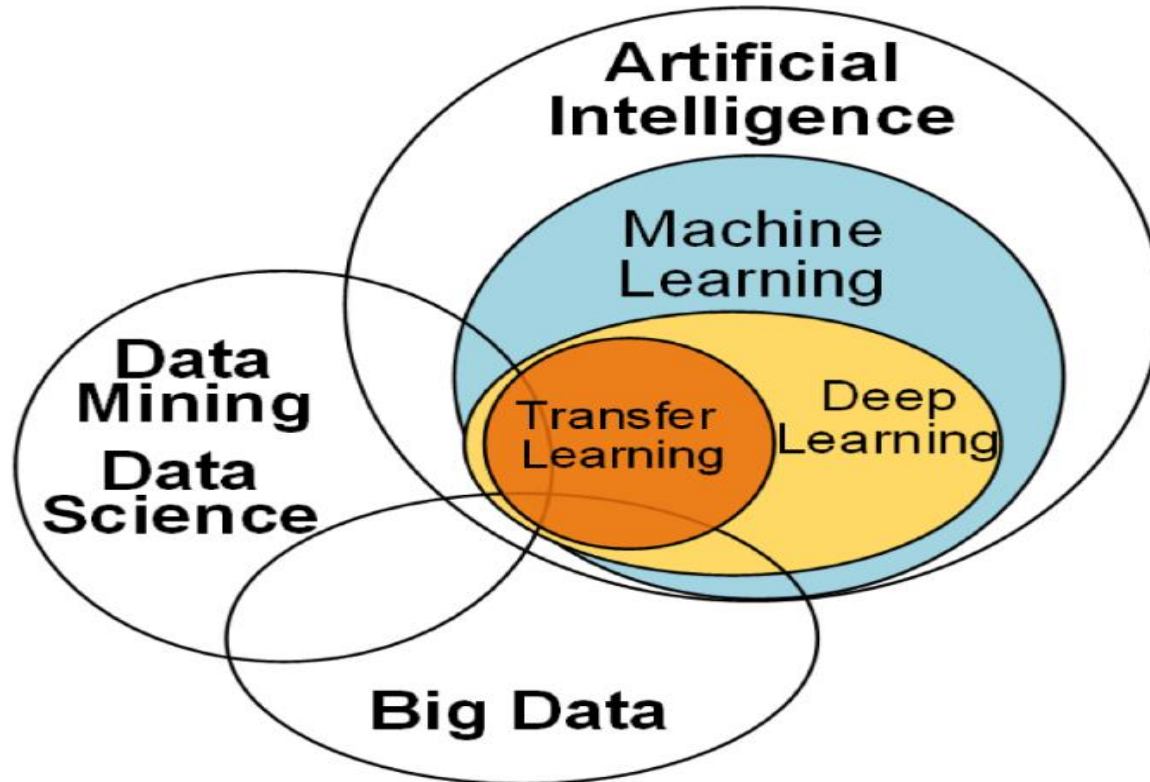
CHAPITRE 4 Introduction au Data Mining

4.2 Différence entre data mining, machine learning et big data

Data mining : Processus d'extraction de connaissances à partir de données.

Machine learning : Sous-ensemble de l'IA permettant aux machines d'apprendre des modèles à partir de données.

Big data : Désigne des ensembles de données extrêmement volumineux dont le traitement nécessite des outils spécialisés.



CHAPITRE 4 Introduction au Data Mining

4.3 Processus du Data Mining (CRISP-DM, KDD, DATLAS, SEMMA)

Le **CRISP-DM** (Cross-Industry Standard Process for Data Mining) et le **KDD** (Knowledge Discovery in Databases) sont deux méthodologies utilisées pour structurer les projets de Data Mining.

- ❖ **SEMMA** (Sample, Explore, Modify, Model, Assess)
- ❖ **DATLAS** (Data, Transformation, Learning, Assessment, and Sharing)

KDD	CRISP-DM
---	Business Understanding
Selection	Data Understanding
Pre-processing	
Transformation	Data Preparation
Data Mining	Modeling
Interpretation/Evaluation	Evaluation
---	Deployment

CHAPITRE 4 Introduction au Data Mining

4.3 Processus du Data Mining (KDD, CRISP-DM, DATLAS, SEMMA)

Il est largement adopté en raison de sa flexibilité et de son approche pragmatique.

■ Phases du CRISP-DM

1. Compréhension du problème

- ✓ Définir les objectifs métier.
- ✓ Traduire les objectifs en problèmes analytiques

2. Compréhension des données

- ✓ Collecter les données.
- ✓ Examiner leur qualité et leur pertinence.

3. Préparation des données

- ✓ Nettoyage des données (valeurs manquantes, outliers).
- ✓ Transformation et sélection des variables.

4. Modélisation

- ✓ Choisir les algorithmes (classification, clustering, régression).
- ✓ Ajuster les hyperparamètres.

5. Évaluation

- ✓ Vérifier la performance du modèle avec des métriques (précision, recall, precision, F1-Score, MSE).
- ✓ Comparer les résultats avec les objectifs métier.

6. Déploiement

Mettre le modèle en production.
Surveiller et améliorer les performances.

CHAPITRE 4 Introduction au Data Mining

4.3 Processus du Data Mining (KDD, CRISP-DM, DATLAS, SEMMA)

■ Phases du KDD

1. Sélection des données

- ✓ Choisir les sources de données pertinentes.

2. Prétraitement

- ✓ Nettoyage des données et gestion des valeurs aberrantes.

3. Transformation

- ✓ Extraction de caractéristiques, réduction de dimensionnalité.

4. Data Mining

- ✓ Application d'algorithmes pour détecter des motifs

5. Évaluation et interprétation

- ✓ Vérification de la pertinence des résultats.

The Knowledge Discovery Process

- **Data Mining v. Knowledge Discovery in Databases (KDD)**

- ▶ DM and KDD are often used interchangeably
- ▶ actually, DM is only part of the KDD process



CHAPITRE 4 Introduction au Data Mining

4.4 Comparaison entre KDD et CRISP-DM

Bien qu'elles poursuivent le même objectif : extraire des connaissances utiles à partir des données; elles diffèrent dans leur approche et leur structure.

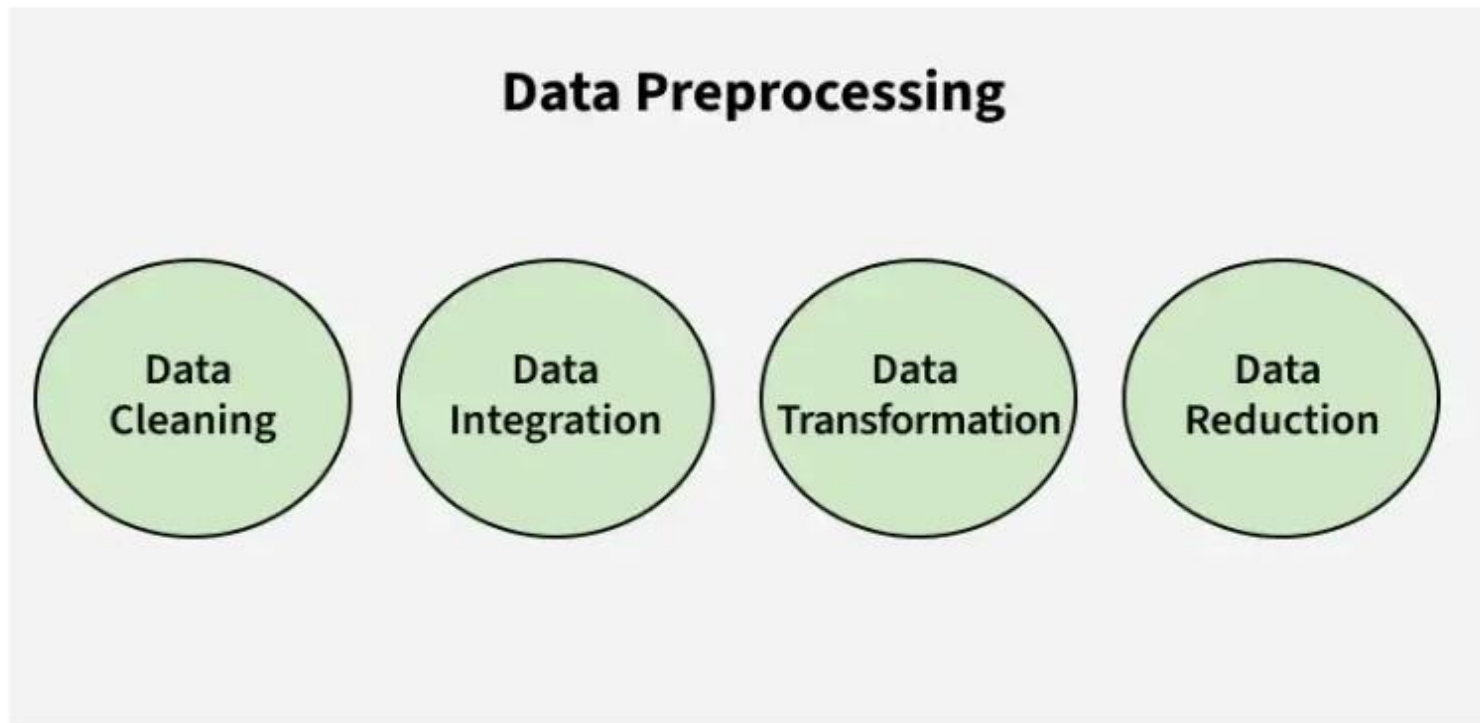
Critères	KDD	CRISP-DM
Objectif	Découverte de nouvelles connaissances	Déploiement en entreprise
Focus	Recherche et exploration des données	Processus métier et production
Structure	Plus flexible et exploratoire	Approche bien définie
Phases clés	Prétraitement, transformation, découverte	Modélisation, évaluation, déploiement
Utilisation	Universités, laboratoires de recherche	Entreprises, industrie

CHAPITRE 5 Prétraitement des données

5.1 Definition

Le prétraitement (Data Precessing) des données est une étape essentielle en **Data Mining** et **Machine Learning**, car **la qualité des données** impacte directement **la performance des modèles**.

Il comprend plusieurs sous-étapes comme le nettoyage, la transformation et la réduction de dimension.



CHAPITRE 5 Prétraitement des données

5.2 Étapes du Prétraitement des Données

1. Nettoyage des données

- ✓ Gestion des valeurs manquantes
- ✓ Détection et suppression des outliers
- ✓ Correction des erreurs et incohérences

Méthodes courantes :

Suppression : Supprimer les lignes ou colonnes contenant trop de valeurs manquantes.

Imputation : Remplacer par la moyenne, la médiane ou une valeur prédictive.

2. Transformation des données

- ✓ Normalisation et standardisation
- ✓ Encodage des variables catégorielles
- ✓ Feature engineering (création de nouvelles caractéristiques)

Normalisation (Min-Max Scaling) : Ramène les valeurs entre [0,1].

Standardisation (Z-score) : Centre les données autour de la moyenne ($\mu = 0$, $\sigma = 1$).

3. Réduction de dimension

- ✓ Sélection des caractéristiques (Feature Selection)
- ✓ Techniques de réduction de dimension

CHAPITRE 6 Algorithmes de classification

6.1 Les algorithmes de Machine Learning

Les algorithmes d'apprentissage automatique sont des techniques utilisés pour créer des systèmes capables d'apprendre à partir de données et de faire des prédictions ou de prendre des décisions sans être explicitement programmés.

6.1.1 Algorithmes d'apprentissage supervisé

Dans l'apprentissage supervisé, l'algorithme est formé sur des données étiquetées (où l'entrée et la sortie correspondante sont connues). L'objectif est d'apprendre une correspondance entre les entrées et les sorties.

1. Régression linéaire [Linear Regression]

- **Objectif** : Prédire des valeurs continues (par exemple, les prix des maisons).
- **Description** : Modélise la relation entre les caractéristiques d'entrée (variables indépendantes) et une sortie continue (variable dépendante) en ajustant une ligne droite aux données.

2. Régression logistique [Logistic Regression]

- **Objectif** : Problèmes de classification binaire (par exemple, détection de spam)
- **Description** : Similaire à la régression linéaire, mais utilisée pour prédire les résultats catégoriels.
Elle génère des probabilités à l'aide d'une fonction logistique.

CHAPITRE 6 Algorithmes de classification

6.1 Les algorithmes de Machine Learning

6.1.1 Algorithmes d'apprentissage supervisé

3. Arbres de décision [Decision Trees]

- Objectif : Classification et régression.

4. Random Forest

- Objectif : Classification et régression.

5. Support Vector Machines (SVM)

- Objectif : Classification

6. K-Nearest Neighbors (k-NN)

- Objectif : Classification et régression logistique.

7. Naive Bayes

- Objectif : Classification (par exemple, classification de texte, filtrage du spam)

CHAPITRE 6 Algorithmes de classification

6.1 Les algorithmes de Machine Learning

6.1.2. Algorithmes d'apprentissage non supervisé

Dans l'apprentissage non supervisé, l'algorithme travaille avec des données non étiquetées et essaie de trouver des modèles ou des structures cachés en leur sein.

1. Clustering k-Means [K-Means Clustering]

- **Objectif** : Clustering (regroupement de points de données similaires).
- **Description** : Partitionne les données en « k » clusters où chaque point de données appartient au cluster avec le centroïde le plus proche (moyenne).

2. Clustering hiérarchique [Hierarchical Clustering]

- **Objectif** : Clustering.

CHAPITRE 6 Algorithmes de classification

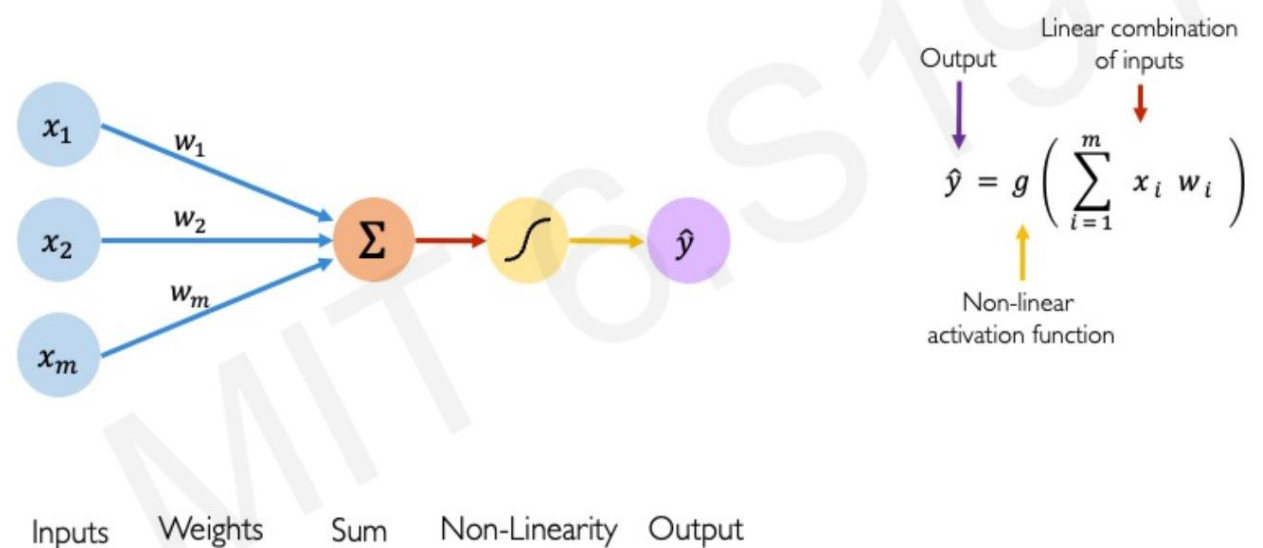
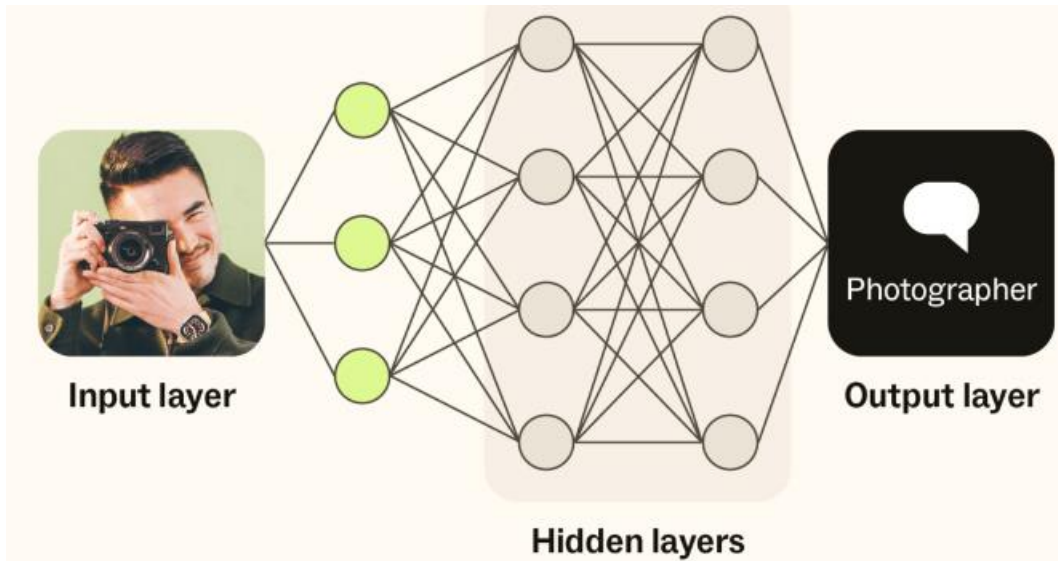
6.2 Les algorithmes de Deep Learning

❖ Qu'est-ce que le Deep Learning ?

Le Deep Learning est une branche de l'apprentissage automatique basée sur l'architecture d'un réseau neuronal artificiel.

Il s'agit d'un sous-ensemble de l'apprentissage automatique, les programmes résolvent des tâches sans être explicitement programmés.

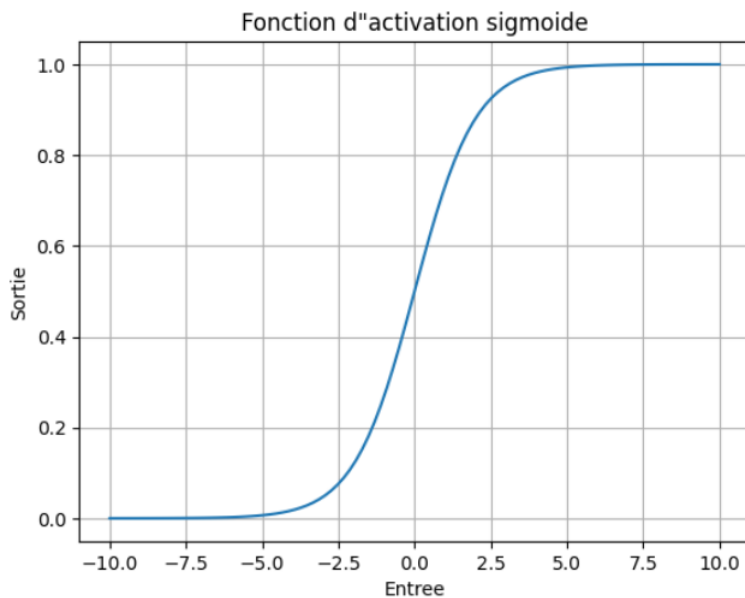
Il est plus spécifique car ils entraînent des **réseaux neuronaux artificiels**.



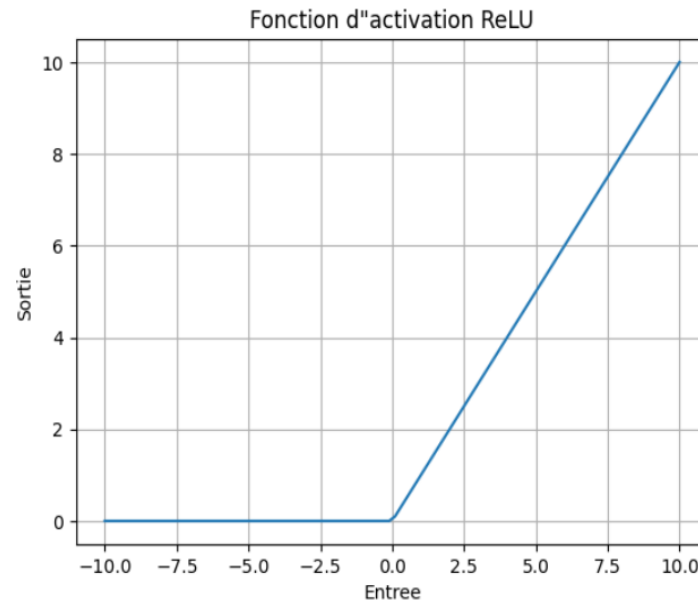
CHAPITRE 6 Algorithmes de classification

6.2 Les algorithmes de Deep Learning

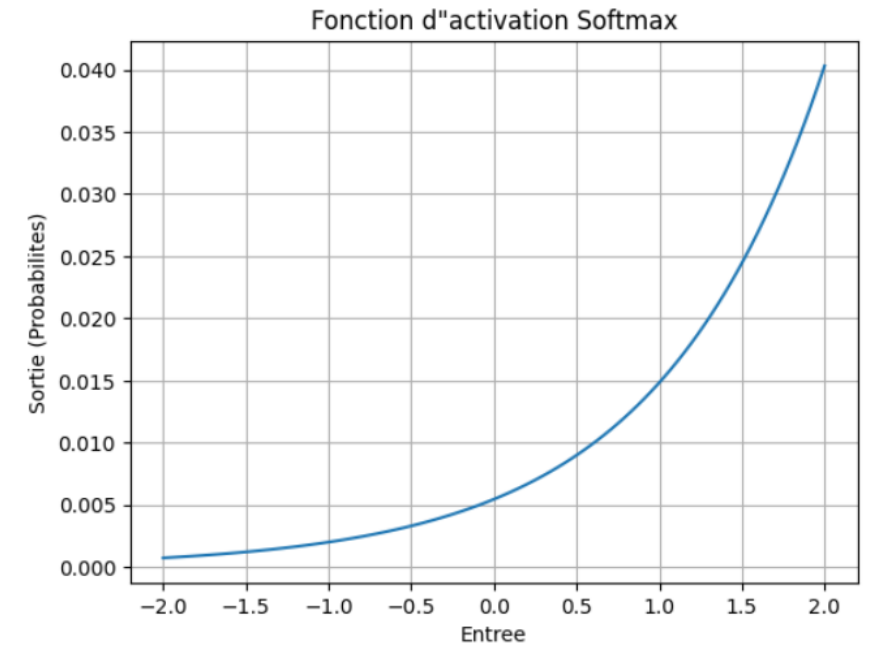
Fonction d'activation



$$\sigma(x) = \frac{1}{1 + e^{-x}}$$



$$f(x) = \max(0, x)$$



$$\text{softmax}(x_i) = \frac{e^{x_i}}{\sum_j e^{x_j}}$$

CHAPITRE 6 Algorithmes de classification

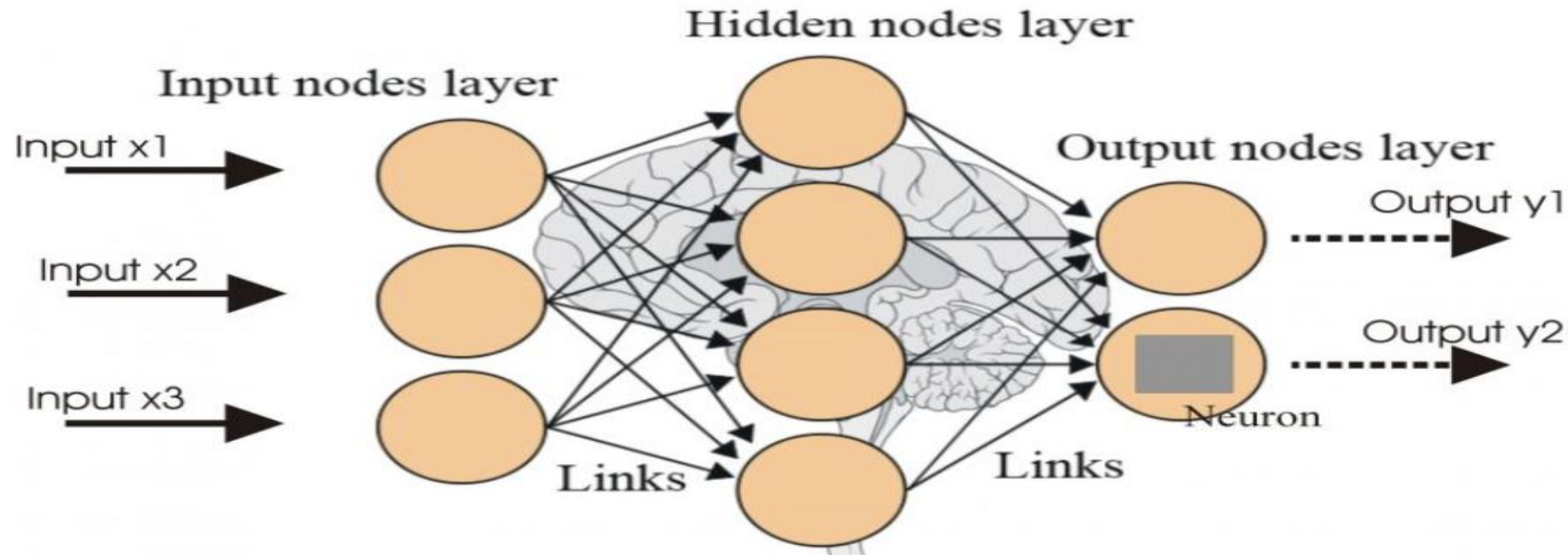
6.2 Les algorithmes de Deep Learning

6.2.1 Shallow Neural network

6.2.2 Réseaux de neurones convolutifs [Convolutional Neural Network (CNN)]

6.2.3 Réseaux de neurones récurrents [Recurrent Neural Network (RNN)]

6.2.4 Variant de RNN : Long Term Short Memory (LSTM)



CHAPITRE 6 Algorithmes de classification



6.2 Les algorithmes de Deep Learning

6.1.1. Shallow Neural Network

Pour calculer la sortie d'un réseau neuronal superficiel, passons en revue les étapes à l'aide d'un exemple.
Un réseau neuronal superficiel comporte généralement une couche cachée.

❖ Nous supposons ce qui suit

1. Supposons les couches suivantes:

[Couche d'entrée] *Input Layer*: x_1 et x_2

[Couche cachée] *Hidden Layer*: 2 neurones

[Couche de sortie] *output Layer*: 1 neurone

2. Paramètres du réseau neuronal superficiel

Nous utiliserons les éléments suivants :

w_{11}, w_{12} pour le neurone 1 de la couche cachée

w_{21}, w_{22} pour le neurone 2 de la couche cachée

3. Biais pour la couche cachée

b_1 pour le neurone 1

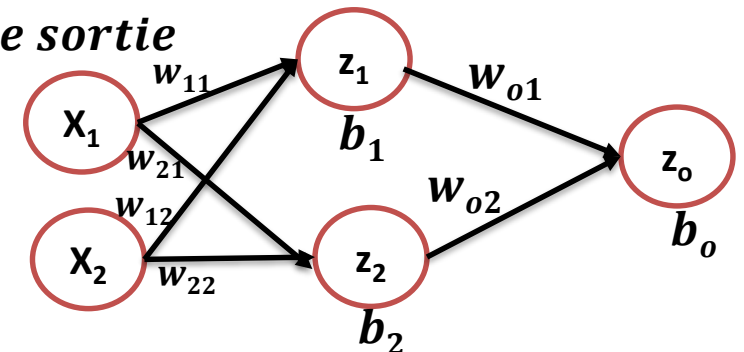
b_2 pour le neurone 2

4. Weights pour la couche de sortie

w_{o1}, w_{o2} sont les weights reliant les neurones de la couche cachée au neurone de sortie.

5. Biais pour la couche de sortie

b_0 pour la couche de sortie



CHAPITRE 6 Algorithmes de classification



6.2 Les algorithmes de Deep Learning

6.1.1 Shallow Neural Network

❖ Étape 1 : Calculer les sorties de la couche cachée

✓ Nous calculons d'abord l'entrée nette de chaque neurone de la couche cachée.

▪ Entrée nette du neurone caché 1 :

$$Z_1 = w_{11}x_1 + w_{12}x_2 + b_1$$

▪ Entrée nette du neurone caché 2 :

$$Z_2 = w_{21}x_1 + w_{22}x_2 + b_2$$

✓ Nous appliquons maintenant une fonction d'activation à ces entrées nettes.

✓ Supposons que nous utilisons la fonction d'activation ReLU :

$$a_1 = \text{ReLU}(Z_1) = \max(0, Z_1)$$

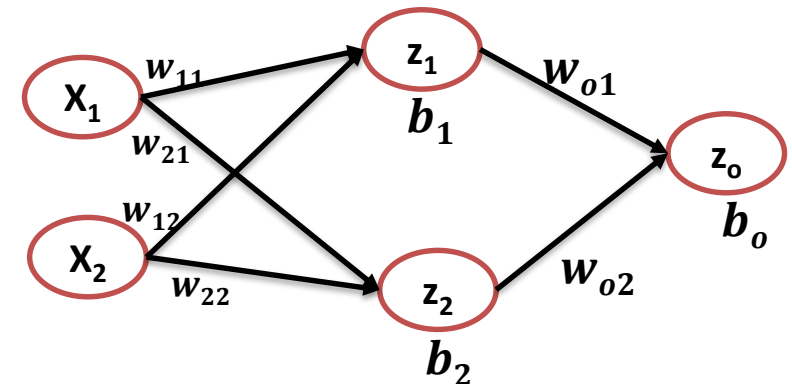
$$a_2 = \text{ReLU}(Z_2) = \max(0, Z_2)$$

❖ Étape 2 : Calculer la couche de sortie

✓ Maintenant, calculez l'entrée nette de la couche de sortie :

$$Z_o = w_{o1}a_1 + w_{o2}a_2 + b_o$$

Le résultat final est simplement la valeur de Z_o , en supposant que nous utilisons une activation linéaire au niveau de la couche de sortie.



Shallow Neural Network

