Machine Learning

Training

Douala, Mai-Juin 2024

WHO AM I ?

TOUZA ISAAC

- PhD student in Computer Science
- Mobile, Web, and Desktop Developer
- Machine Learning Enthusiast
- Data Scientist
- Computer Science Teacher
- GDG Maroua Organizer
- Grand Nord Developers Community Lead
- Digital Entrepreneur

IncH.Class
Powered by SIMPLON.CO

@touzaisaac

Contacts :

Phone: 691805321 – 672277579

E-mail: isaac_touza@outlook.fr

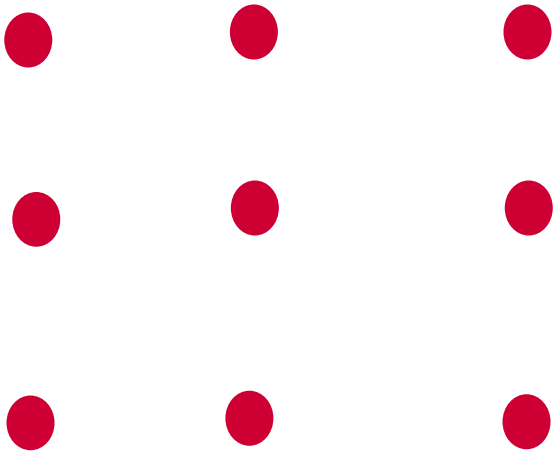
< Program of training />

1. Introduction et Concepts de Base du Machine Learning
2. Collecte et Préparation des Données
3. Traitement du Langage Naturel (NLP)
4. Règles d’association
5. Régression Linéaire et logistique
6. La Classification et Évaluation des Modèles
7. Méthodes de Clustering
8. Techniques d'Optimisation d’un modèle
9. Réseaux de Neurones et Deep Learning
10. Projet Final et Conclusion

IncH.Class
Powered by SIMPLON.CO

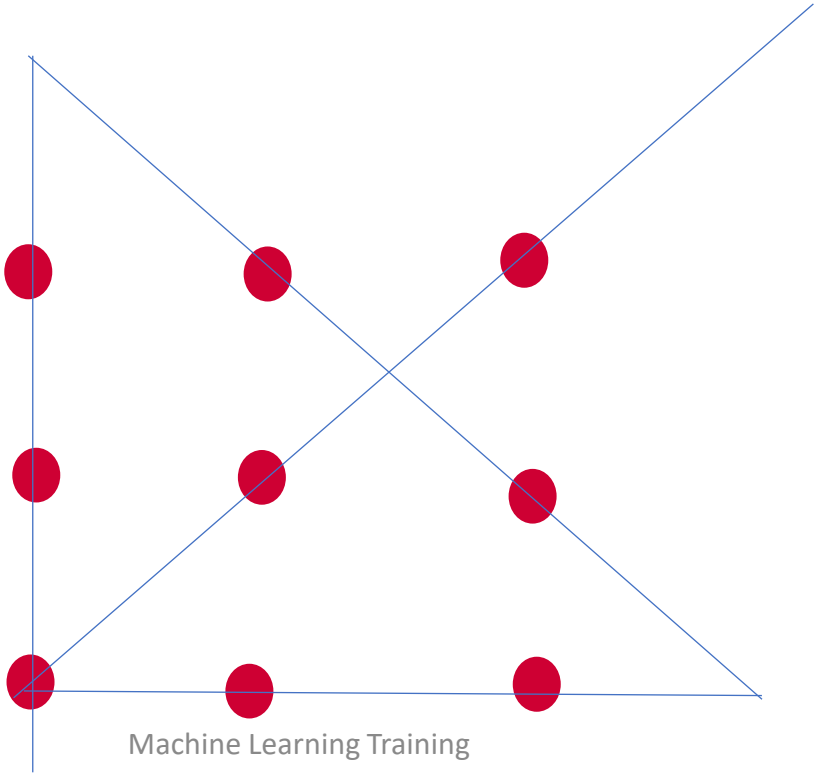
On réfléchi un peu avant de commencer.

Relier ces neufs points par quatre droites sans soulever le stylo.



On réfléchi un peu avant de commencer.

Relier ces neufs points par quatre droites sans soulever le stylo.



On réfléchi un peu avant de commencer.

Leçons :

- Sortir de sa zone de confort pour résoudre les problèmes
- Explorer au-delà de votre expérience personnelle
- Penser en dehors de la boîte
- Rompre les règles
- Il faut oser à aller plus loin
- …

1

Introduction et Concepts de Base du Machine Learning

1. Introduction au Machine Learning

Définition

Le **Machine Learning (ML)** est

- un sous-domaine de l'intelligence artificielle
- qui permet aux systèmes informatiques d'apprendre
- et d'améliorer automatiquement à partir de **l'expérience** sans être explicitement programmés.

C'est quoi :

- L'intelligence artificielle ?
- Systèmes informatiques ?
- La programmation ?

1. Introduction au Machine Learning

Historique

- Les débuts du Machine Learning sont souvent associés à ceux de l'IA avec la création du célèbre test de Turing en 1950.

- Mais c'est en 1959 que le terme “machine learning” apparait pour la première fois, utilisé par **Arthur Samuel** pour son programme créé en 1952 capable d'apprendre à jouer aux dames au fil des parties.

1. Introduction au Machine Learning

Historique

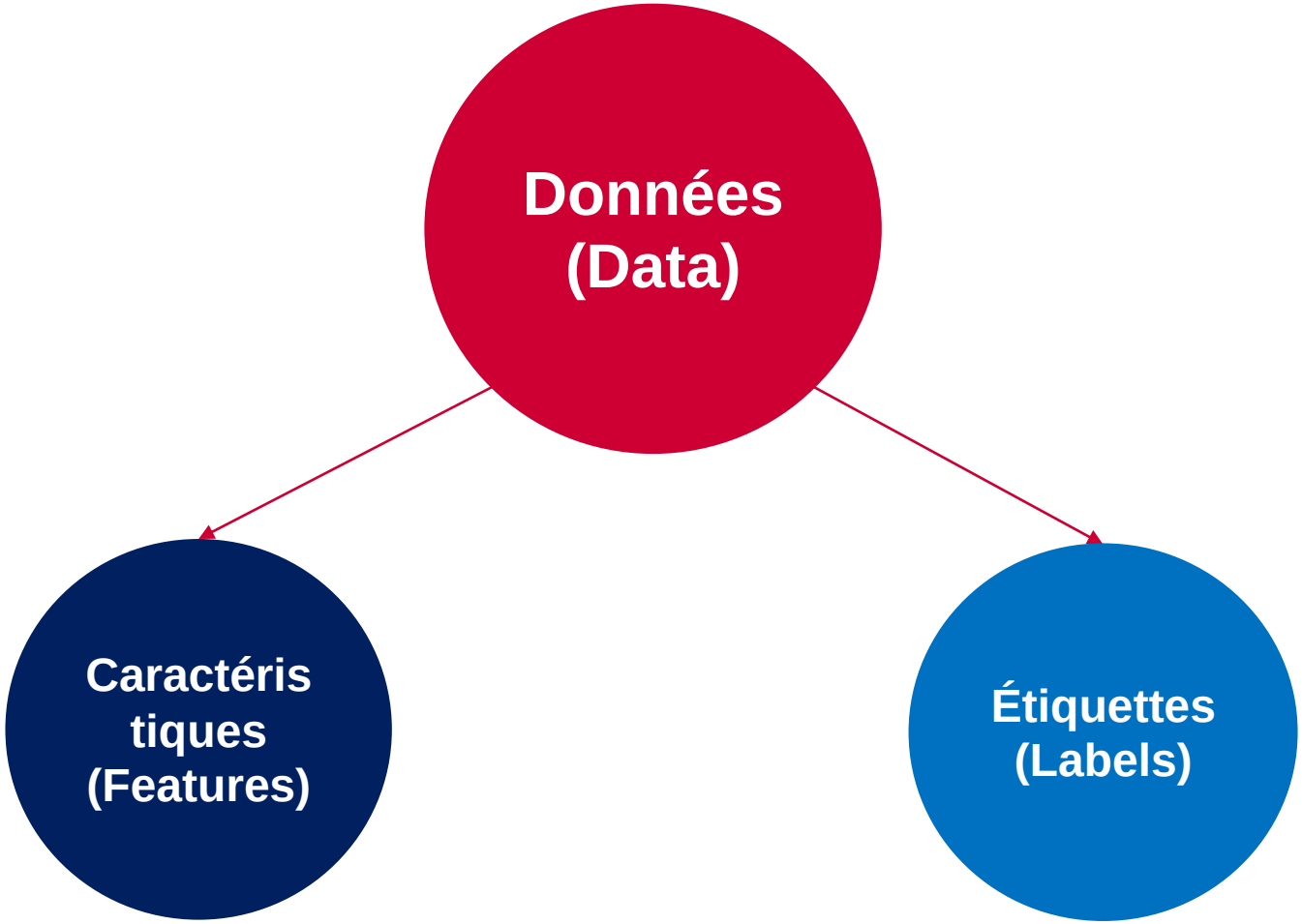
- Les progrès significatifs ont eu lieu avec :

    1. l'augmentation de la puissance de calcul
    2. l'amélioration des algorithmes et
    3. l'accès à de grandes quantités de données..

2. Concepts Clés du Machine Learning

Données (Data)	Caractéristiques (Features)	Étiquettes (Labels)
Ensemble de valeurs recueillies pour l'analyse pouvant être structurées (bases de données relationnelles) ou non structurées (textes, images).	Variables d'entrée utilisées par le modèle pour effectuer des prédictions. Elles doivent être pertinentes et représentatives du problème.	Variables de sortie ou cible que le modèle essaie de prédire (utilisées dans l'apprentissage supervisé).

2. Concepts Clés du Machine Learning



2. Concepts Clés du Machine Learning

Exemple

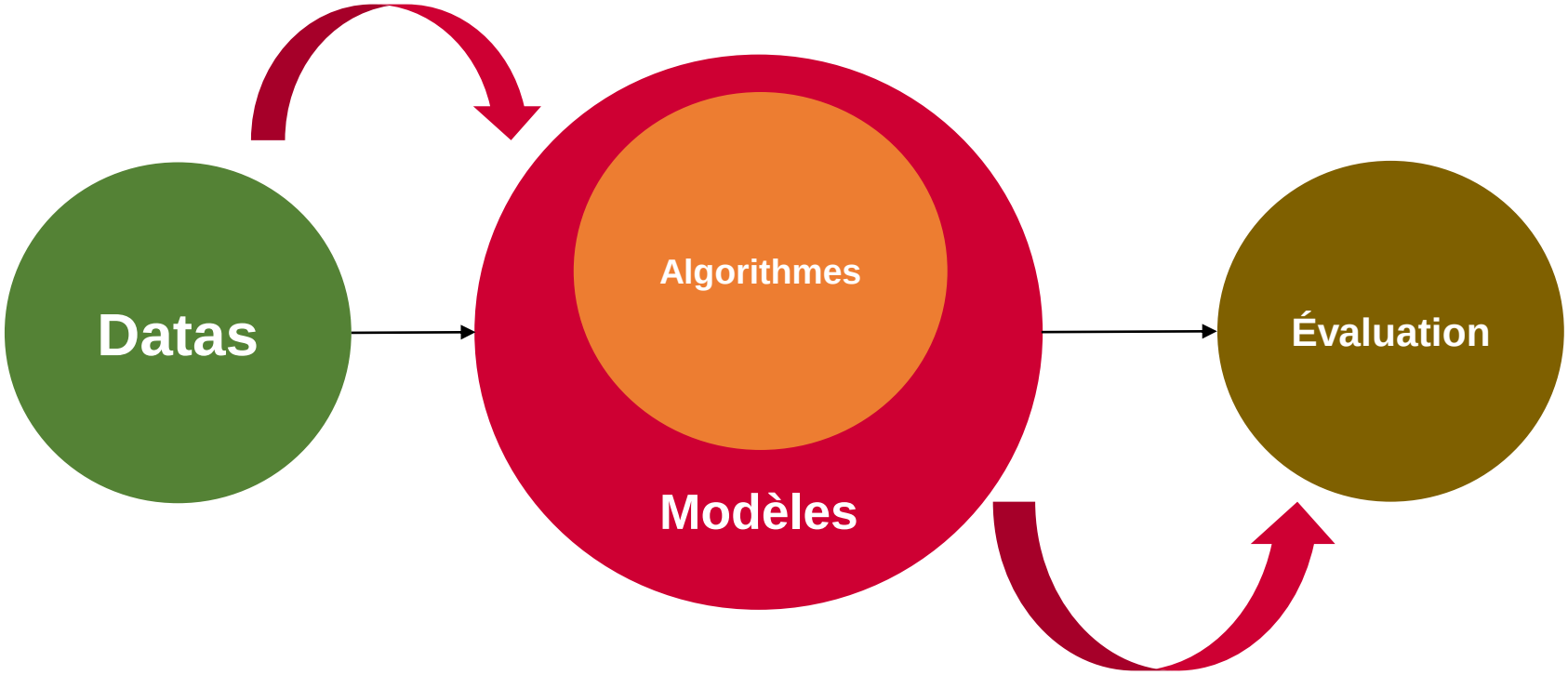
ID	Surface (m^2)	Nombre de Chambres	Nombre de Salles de Bain	Âge	Type de Toiture	Localisation	Besoin de Rénovation
1	150	3	2	50	Tuile	Centre	Oui
2	200	4	3	60	Zinc	Banlieue	Non
3	100	2	1	40	Ardoise	Campagne	Oui
4	180	5	4	70	Tuile	Centre	Non

Question : Identifier les features et le label.

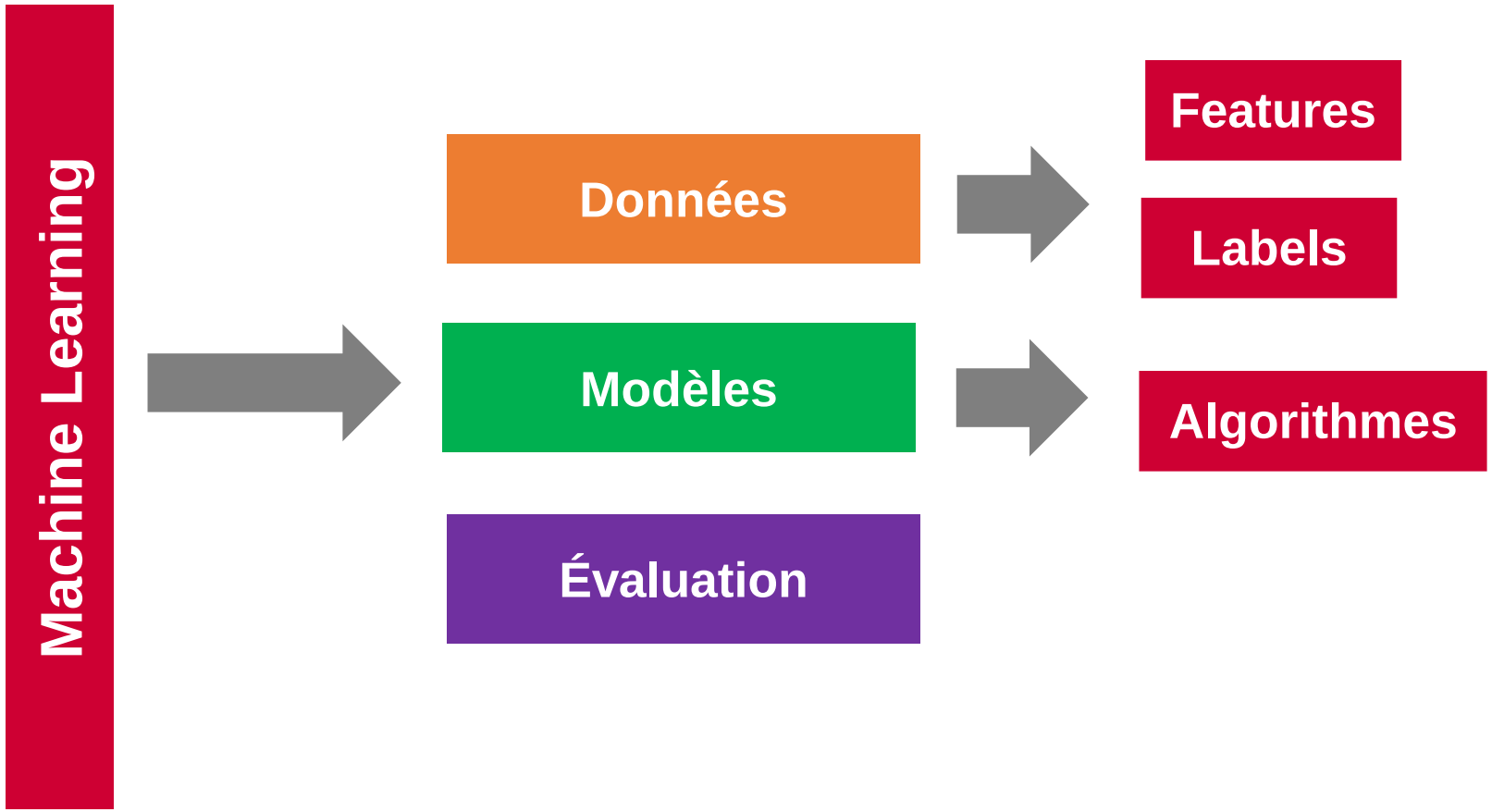
2. Concepts Clés du Machine Learning

Modèles	Algorithmes	Évaluation
Structures mathématiques ou statistiques qui représentent les relations entre les caractéristiques et les étiquettes. Ils sont entraînés sur des données pour apprendre ces relations.	Ensemble de règles ou instructions utilisées pour résoudre un problème spécifique en ML. Les algorithmes sont utilisés pour créer et ajuster les modèles.	Processus de mesure de la performance d'un modèle.

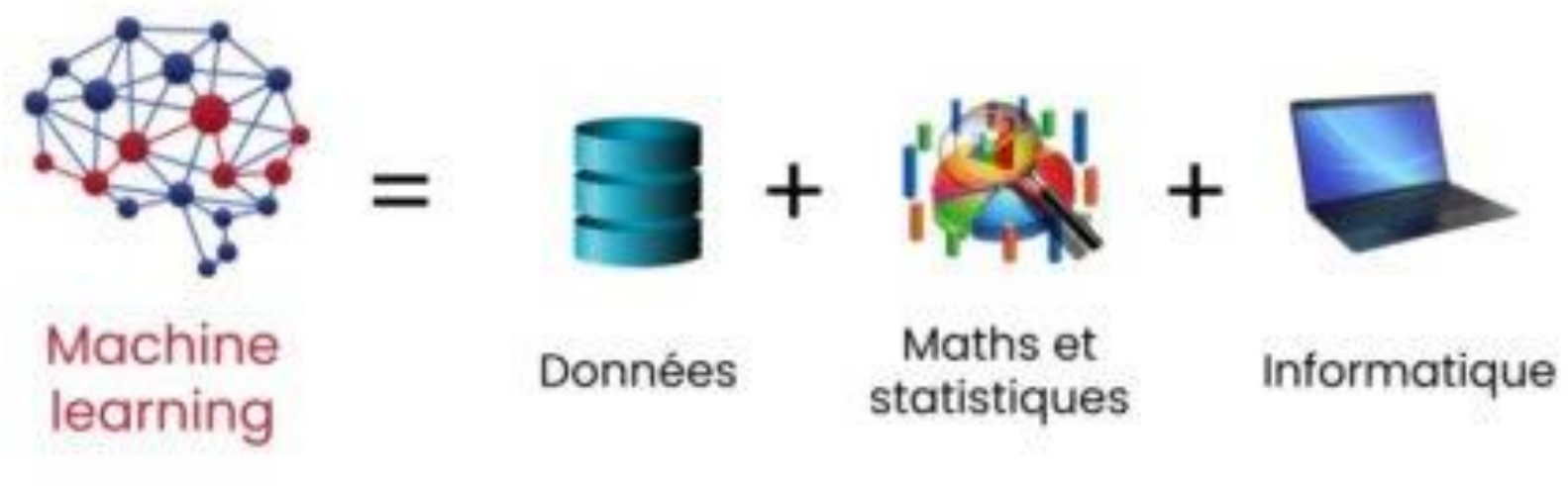
2. Concepts Clés du Machine Learning



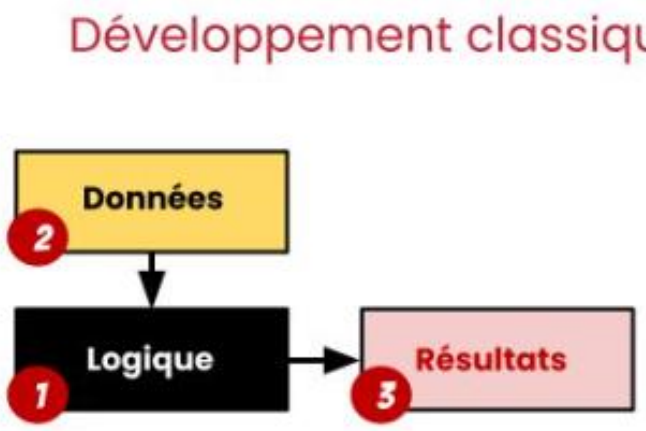
2. Concepts Clés du Machine Learning



2. Concepts Clés du Machine Learning



3. ML Vs Programme classique

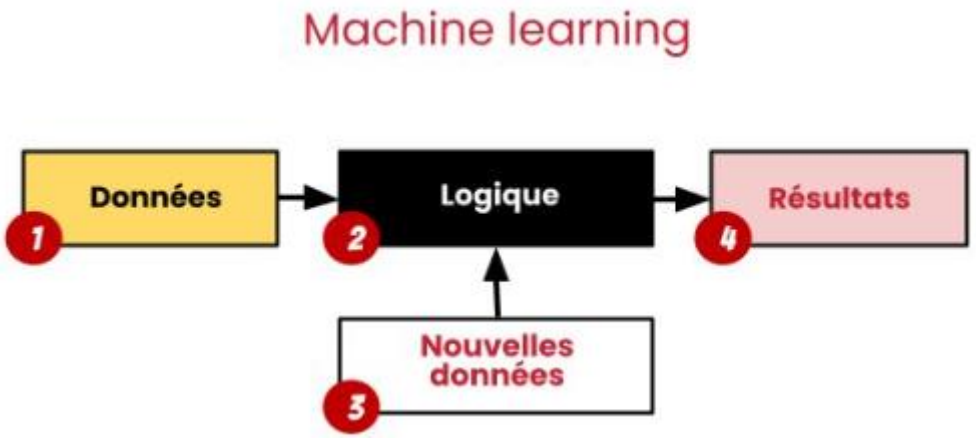


1. Une **logique** est écrite dans un langage de développement (python, C++, java, php, etc)

2. Des **données** interagissent avec cette logique (input de l'utilisateur, tableaux, etc)

3. Des **résultats** sont obtenus (calculs, affichage à l'écran, exécution d'une commande, etc)

3. ML Vs Programme classique



1. La **donnée** est collectée (ex :différentes photos de pommes sont recueillies)
2. La **logique** est calculée par la machine (ex : La machine trouve toute seule qu'est-ce que fait qu'une photo soit ou pas une pomme). À ce moment-ci, on dit que le modèle de machine learning a été "entraîné"
3. Des **nouvelles données** interagissent avec la logique précédemment trouvée (ex : on utilise des nouvelles photos de fruits jamais utilisées auparavant)
4. Des **résultats** sont obtenus (la logique trouvée par l'ordinateur jugera si la nouvelle photo ressemble ou pas à une pomme, en fonction des exemples utilisés lors de la phase d'apprentissage)

4. Tâches du Machine Learning

Classification :	Régression :	Clustering :
Affectation de catégories aux données. Exemples : reconnaissance des visages, détection de spam.	Prédiction de valeurs continues. Exemples : prédiction des prix immobiliers, prévisions météorologiques.	Groupement de données non étiquetées. Exemples : segmentation de clients, regroupement de documents similaires.

4. Tâches du Machine Learning

Règles d'Association :	Estimation :	Réduction de Dimensionnalité :
Identifier des relations significatives entre différents éléments dans un ensemble de données.	Produire une valeur approximative d'une variable cible basée sur des données d'entrée	Réduction du nombre de caractéristiques tout en préservant les informations importantes.

5. Domaines d'Application du ML

- **Santé :** Diagnostic médical, analyse des images médicales, prédiction des maladies.
- **Finance:** Détection de fraude, évaluation du risque de crédit, trading algorithmique.
- **Marketing et Commerce :** Recommandation de produits, segmentation des clients, personnalisation de l'expérience utilisateur.
- **Transport :** Conduite autonome, optimisation des itinéraires, maintenance prédictive.
- **Technologie et Industrie :** Maintenance prédictive, automatisation des processus, optimisation de la production.

Agriculture : Prédiction des rendements, détection des maladies des plantes, gestion de l'irrigation.

6. Méthodes du Machine Learning

Algorithmes de base :

- Régression linéaire et logistique
- K-Nearest Neighbors (KNN)
- Support Vector Machines (SVM)
- Naive Bayes
- Arbres de décision

Algorithmes avancés :

- Random Forest
- Gradient Boosting Machines (GBM)
- XGBoost
- Réseaux de neurones profonds (Deep Learning)
- Transformers (pour le NLP)

7. Outils et Bibliothèques du ML

Langages de programmation :

- **Python :** Langage le plus utilisé en ML pour sa simplicité et ses bibliothèques riches.

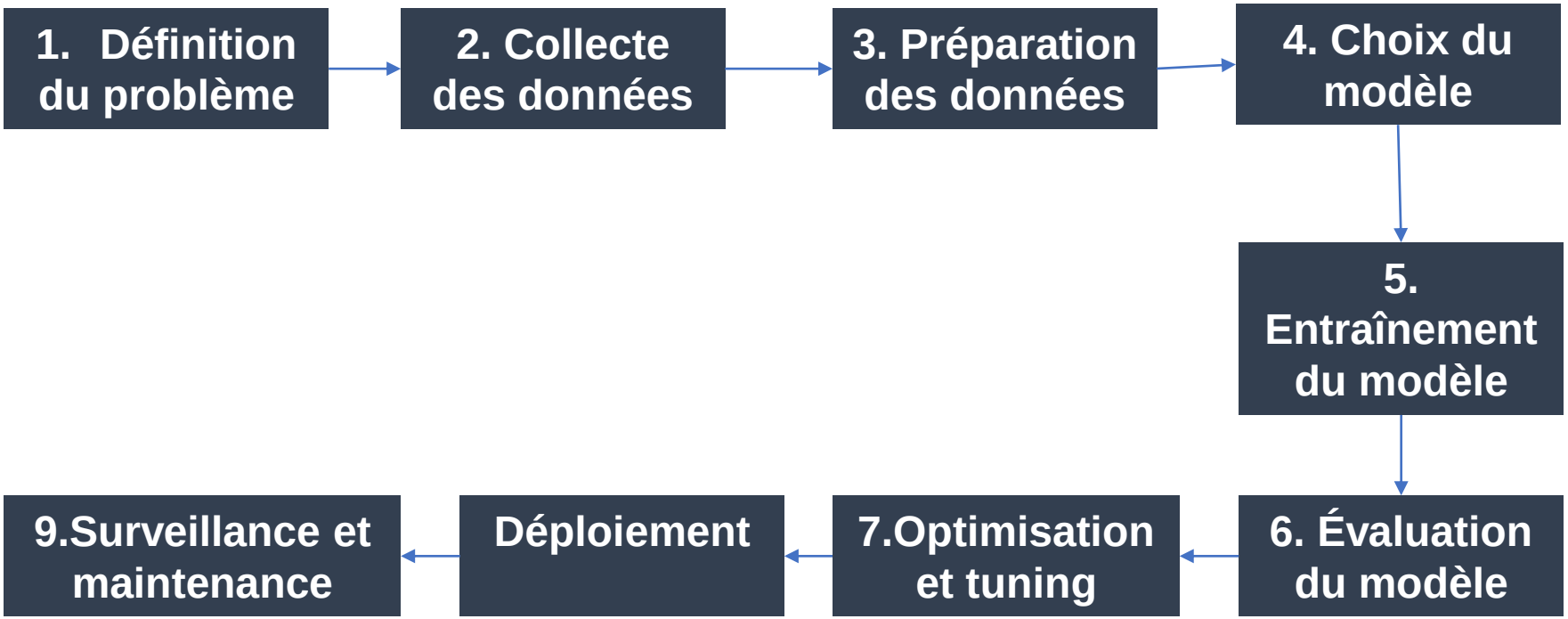
- **R :** Utilisé pour les statistiques et l'analyse des données.

7. Outils et Bibliothèques du ML

Bibliothèques et frameworks :

- **Scikit-learn :** Bibliothèque pour les algorithmes de ML de base.
- **TensorFlow et Keras :** Frameworks pour le Deep Learning.
- **PyTorch :** Framework flexible et efficace pour le Deep Learning.
- **Pandas :** Bibliothèque pour la manipulation et l'analyse des données.
- **Numpy :** Bibliothèque pour les opérations mathématiques et scientifiques.
- **Matplotlib et Seaborn :** Bibliothèques pour la visualisation des données.

8. Méthodologie d'un Projet de ML



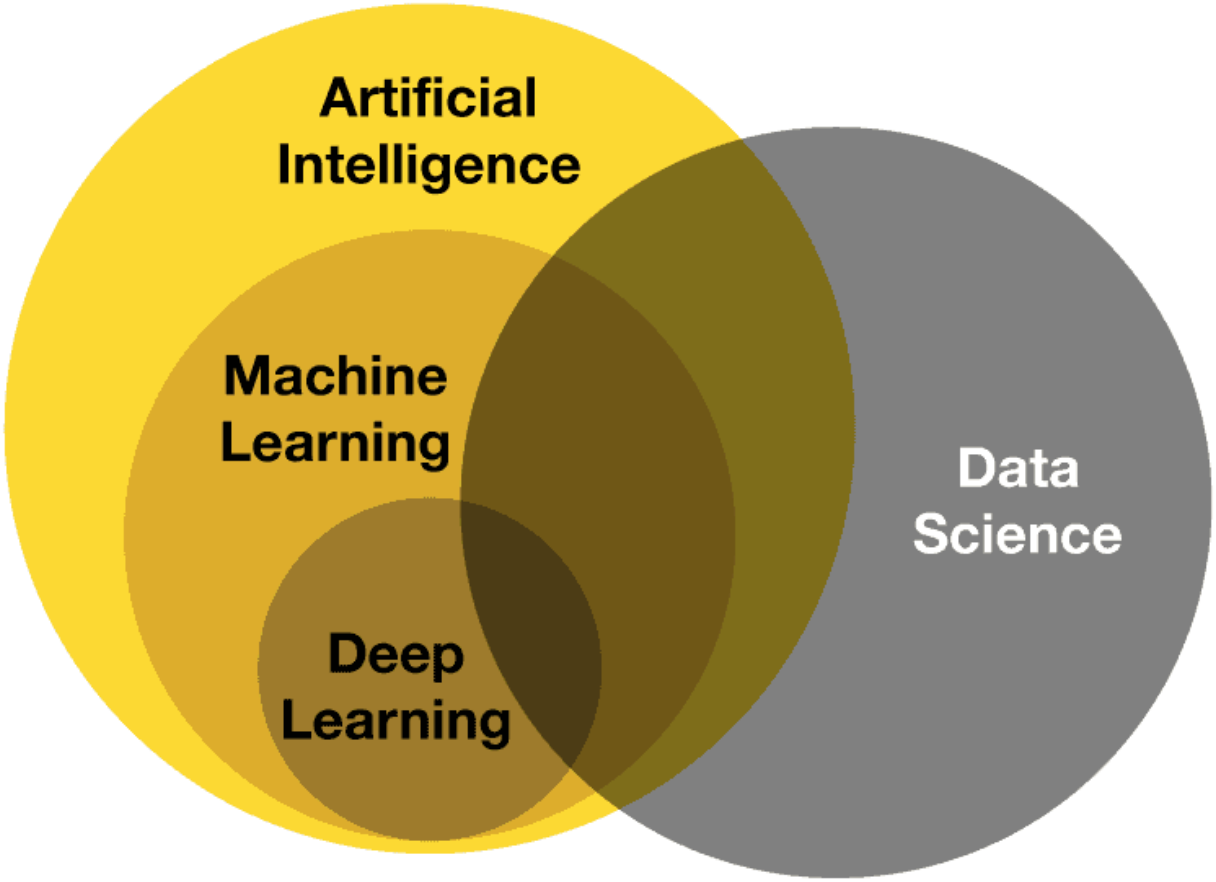
8. Méthodologie d'un Projet de ML

1. **Définition du problème :** Identifier et formuler clairement le problème à résoudre.
2. **Collecte des données :** Rassembler les données nécessaires pour l'analyse.
3. **Préparation des données :** Nettoyer et transformer les données pour les rendre exploitables (gestion des valeurs manquantes, normalisation, etc.).
4. **Choix du modèle :** Sélectionner l'algorithme ou les algorithmes à utiliser pour créer le modèle.
5. **Entraînement du modèle :** Utiliser les données d'entraînement pour ajuster les paramètres du modèle.
6. **Évaluation du modèle :** Tester le modèle sur des données de test pour mesurer ses performances.
7. **Optimisation et tuning :** Ajuster les hyperparamètres et améliorer le modèle.
8. **Déploiement :** Mettre en production le modèle pour des prédictions en conditions réelles.
9. **Surveillance et maintenance :** Suivre les performances du modèle et le mettre à jour en cas de besoin.

8. ML vs DS vs DM vs AI

Aspect	Data Science	Data Mining	Machine Learning	Artificial Intelligence (AI)
Définition	Science interdisciplinaire des données	Processus de découverte de motifs dans les données	Apprentissage automatique des ordinateurs	Systèmes intelligents capables de réaliser des tâches humaines
Méthodes	Statistiques, Exploration de données, ML	Clustering, Classification, Association	Supervised, Unsupervised, Reinforcement Learning	ML, Logique floue, Réseaux de neurones
Domaines	Analyse prédictive, BI, Visualisation	Marketing, Détection de fraudes, Analyse de marché	Reconnaissance d'image, Recommandation	Robots, Assistants virtuels, NLP
Outils	Python, R, SQL, Pandas, NumPy	Weka, RapidMiner, Python, SQL	Python, R, Scikit-learn, TensorFlow	Python, Java, TensorFlow, PyTorch
Exemples	Visualisation des tendances de ventes	Segmentation de clients pour marketing	Classification d'e-mails en spam	Assistant virtuel, Voiture autonome

8. ML vs DS vs DM vs AI



9. Rappel

Générer 1000 enregistrements sur les données spécifiées :

- Âge,
- Sexe (M,F),
- Profession (Etudiant, Fonctionnaire, Pas de Fonction, Entrepreneur),
- Gain (entre 10000 et 500000)
- Eligible_au_pret? (Oui, Non)

9. Rappel

Charger les données :

1. Utiliser **pandas** pour charger un fichier CSV dans un DataFrame.
2. Afficher les premières lignes et les informations générales sur les données.

Nettoyer les données :

1. Identifier les valeurs manquantes
2. Remplacer ou supprimer les valeurs manquantes

Opérations statistiques de base :

1. Calculer la moyenne, la médiane et l'écart-type de colonnes numériques

Histogrammes et boîtes à moustaches avec Matplotlib :

1. Créer un histogramme pour visualiser la distribution d'une colonne.
2. Créer une boîte à moustaches pour visualiser la distribution d'une colonne.
3. Créer un graphique en barres pour comparer les moyennes de groupes.
4. Créer un pairplot pour visualiser les relations entre plusieurs paires de colonnes.

2

Collecte et Préparation des Données

1. Introduction à la Collecte des Données

Définition : Collecte des Données

La **collecte des données** est le processus systématique d'acquisition et de mesure d'informations provenant de diverses sources, dans le but de répondre à des questions de recherche, de tester des hypothèses, d'évaluer des résultats ou de prendre des décisions informées.

2. Objectifs de la Collecte des Données

1. **Obtenir des Informations Précises et Fiables** : Assurer que les données recueillies sont exactes, complètes et représentatives de la réalité étudiée.

2. **Supporter les Analyses et Prises de Décisions** : Fournir une base solide de données pour mener des analyses, formuler des conclusions et prendre des décisions éclairées.

3. **Faciliter la Recherche et le Développement** : Contribuer à l'avancement des connaissances scientifiques et au développement de nouvelles technologies ou solutions.

3. Méthodes de Collecte des Données

1. Enquêtes et Questionnaires :

- Utilisés pour collecter des données directement auprès des individus ou des groupes.
- Peuvent être administrés en ligne, par téléphone ou en face à face.

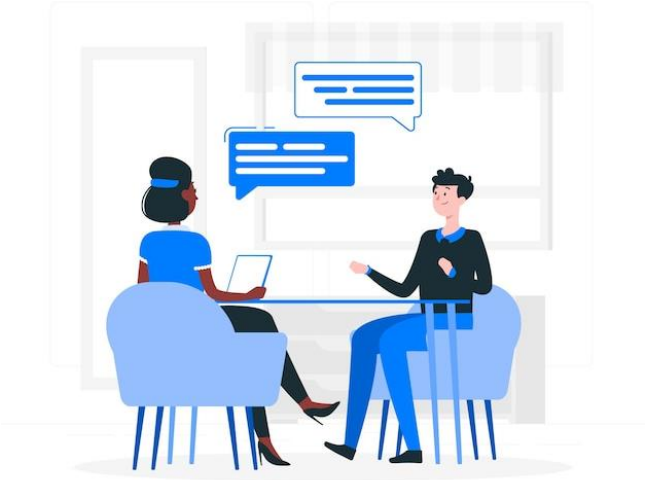
3. Méthodes de Collecte des Données

2. Observations Directes :

- Implique l'observation systématique et l'enregistrement des comportements ou des événements tels qu'ils se produisent.
- Utilisé dans des contextes comme les études de marché, la recherche en psychologie, etc.

3. Méthodes de Collecte des Données

3. Interviews et Entretiens :



- Entrevues structurées, semi-structurées ou non structurées pour recueillir des informations détaillées.
- Permet d'explorer des sujets en profondeur.

3. Méthodes de Collecte des Données

4. Extraction de Données (Data Mining)

- Utilisation d'outils et de techniques pour extraire des données pertinentes de grandes bases de données.
- Utilisé couramment en marketing, finance, et autres domaines nécessitant l'analyse de grandes quantités de données

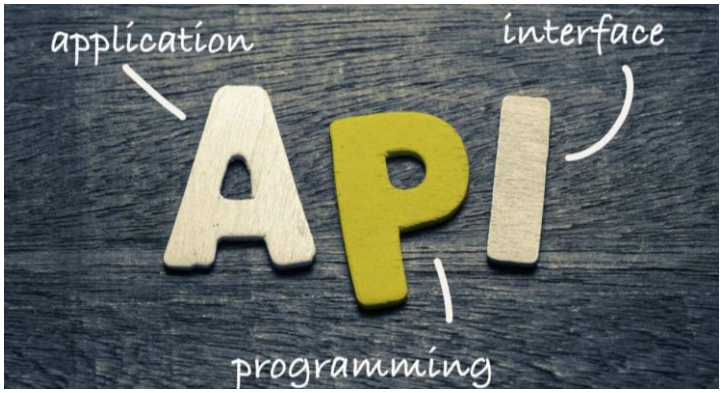
3. Méthodes de Collecte des Données

5. Web Scraping :

- Technique automatisée pour extraire des données de sites web.
- Implique l'utilisation de scripts pour récupérer des informations spécifiques.

3. Méthodes de Collecte des Données

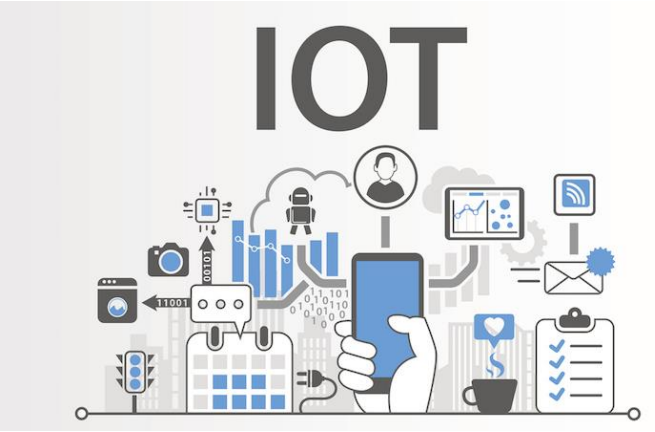
6. Utilisation d'APIs (Application Programming Interfaces) :



- Accès programmatique aux données via des points de terminaison API fournis par des services web.
- Permet de collecter des données en temps réel et de manière structurée.

3. Méthodes de Collecte des Données

7. Capteurs et Appareils IoT (Internet of Things) :



- Utilisation de capteurs et d'appareils connectés pour collecter des données en temps réel.
- Utilisé dans des domaines comme la domotique, les villes intelligentes, la santé, etc.

4. Sources de Données

- Bases de données relationnelles et non relationnelles
- APIs (Application Programming Interfaces)
- Web scraping
- Fichiers CSV, Excel, JSON, etc.
- Données ouvertes (Open Data)
- Kaggle
- etc.

Application 1: Collecte des données

Accédez au site suivant
https://www.acfas.ca/publications/magazine/articles/
puis scraper à l'aide du navigateur les informations suivantes sur tous les articles des différentes pages :

- le titre,
- l'image descriptive,
- le texte de l'article,
- date de publication ou de mise à jour (si elle existe),
- la nature de l'article (entretien, décryptages, récit, reportage, histoire, carte blanche, tribune, etc..)
- et le nom de l'auteur.

Et exportez les données obtenues sous format CSV.

Application 2: Collecte des données

Les classements de l'équipe du Cameroun aux championnats passés se trouvant à l'adresse https://fr.wikipedia.org/wiki/%C3%89quipe_du_Cameroun_de_football

1. Démarrer le logiciel JupyterLab puis importer les modules suivants :
    - urllib
    - bs4
    - pandas
    - request
    - os
    - requests
2. Accédez au site mentionné plus haut puis scraper les informations sur le classement FIFA de l'équipe du Cameroun puis les stocker dans une base des données et l'exporter sous format Excel.

5. Nettoyage des Données

Définition :

Le **nettoyage de données** (ou **data cleaning** en anglais) est le processus d'identification et de correction des erreurs et des incohérences dans les données pour améliorer leur qualité. Cela inclut :

- la gestion des valeurs manquantes,
- la correction des erreurs de saisie,
- l'élimination des doublons,
- et l'ajustement des valeurs aberrantes.

L'objectif principal du nettoyage des données est de rendre les données plus précises, complètes, et utilisables pour des analyses fiables.

5. Nettoyage des Données

Objectifs du Nettoyage de Données :

- **Améliorer la Qualité des Données** : Assurer que les données sont précises, cohérentes et complètes.
- **Préparer les Données pour l'Analyse** : Rendre les données utilisables et prêtes pour des analyses et des modèles statistiques.
- **Minimiser les Biais et les Erreurs** : Réduire les biais introduits par des données incorrectes ou incomplètes.

5. Nettoyage des Données

Objectifs du Nettoyage de Données :

- **Améliorer la Qualité des Données** : Assurer que les données sont précises, cohérentes et complètes.
- **Préparer les Données pour l'Analyse** : Rendre les données utilisables et prêtes pour des analyses et des modèles statistiques.
- **Minimiser les Biais et les Erreurs** : Réduire les biais introduits par des données incorrectes ou incomplètes.

5. Nettoyage des données

Suppression des lignes ou des colonnes avec des valeurs manquantes

```
import pandas as pd

# Exemple de DataFrame avec des valeurs manquantes
data = {'A': [1, 2, None], 'B': [4, None, 6], 'C': [7, 8, 9]}
df = pd.DataFrame(data)

# Suppression des lignes avec des valeurs manquantes
df_dropped_rows = df.dropna()

# Suppression des colonnes avec des valeurs manquantes
df_dropped_cols = df.dropna(axis=1)
```

5. Nettoyage des données

Imputation des valeurs manquantes

```
from sklearn.impute import SimpleImputer


imputer = SimpleImputer(strategy='mean')
df_imputed = pd.DataFrame(imputer.fit_transform(df), columns=df.columns)


print(df_imputed)
```

5. Nettoyage des données

Gestion des Duplicatas

```
df = pd.DataFrame({'A': [1, 2, 2, 4], 'B': [4, 5, 5, 6]})
df_deduplicated = df.drop_duplicates()


print(df_deduplicated)
```

5. Nettoyage des données

Normalisation des formats de date

```
df = pd.DataFrame({'date': ['2023-01-01', '01/02/2023', 'March 3, 2023']})
df['date'] = pd.to_datetime(df['date'])
```

5. Nettoyage des données

Nettoyage des Chaînes de Caractères

```
df = pd.DataFrame({'text': ['  Hello  ', 'World', '  Foo Bar ']})
df['text'] = df['text'].str.strip()
```

5. Nettoyage des données

Nettoyage des Chaînes de Caractères

```
df = pd.DataFrame({'text': ['  Hello  ', 'World', '  Foo Bar ']})
df['text'] = df['text'].str.strip()
```

6. Transformation des Données

Définition :

La **transformation des données** est le processus de modification, de conversion et de structuration des données brutes en un format approprié pour l'analyse.
Cela inclut des opérations telles que:

- le regroupement,
- la normalisation,
- l'encodage,
- l'agrégation et
- la création de nouvelles fonctionnalités.

L'objectif de la transformation des données est d'améliorer la qualité et l'utilité des données afin qu'elles soient prêtes pour des modèles d'apprentissage automatique, des analyses statistiques ou des visualisations.

6. Transformation des Données

Objectifs :

- **Rendre les Données Compatibles :** Adapter les données à un format ou une structure compatible avec les exigences des algorithmes ou des systèmes d'analyse.

- **Améliorer la Qualité des Données :** Corriger les biais, les incohérences et les erreurs pour rendre les données plus fiables et précises.

- **Faciliter l'Analyse et la Modélisation :** Simplifier les données et les rendre plus pertinentes pour les analyses et les modèles prédictifs.

6. Transformation des Données

Avantages :

- **Amélioration de la Précision des Modèles :** Les données transformées sont mieux adaptées aux algorithmes de machine learning, améliorant ainsi la performance des modèles.
- **Facilitation de l'Exploration des Données :** Des données bien transformées rendent l'analyse exploratoire plus facile et plus efficace.
- **Réduction de la Complexité :** Simplifier les données complexes pour une meilleure compréhension et une manipulation plus aisée.
- **Traitement des Données Hétérogènes :** Uniformiser les données provenant de sources variées pour une analyse cohérente.

6. Transformation des Données

Techniques Courantes de Transformation des Données

Normalisation : Mise à l'échelle des données dans une plage spécifique, souvent entre 0 et 1.

$$X_{\text{scaled}} = \frac{X - X_{\min}}{X_{\max} - X_{\min}} \times (\text{feature\_range}_{\max} - \text{feature\_range}_{\min}) + \text{feature\_range}_{\min}$$

Où :

- X est la valeur originale.
- $X_{\min}$ et $X_{\max}$ sont les valeurs minimale et maximale de la caractéristique originale.
- $\text{feature\_range}_{\min}$ et $\text{feature\_range}_{\max}$ sont les limites inférieure et supérieure de la plage souhaitée, par défaut 0 et 1.

6. Transformation des Données

Techniques Courantes de Transformation des Données

```
from sklearn.preprocessing import MinMaxScaler
import pandas as pd


df = pd.DataFrame({'A': [1, 2, 3], 'B': [4, 5, 6]})
scaler = MinMaxScaler()
df_normalized = pd.DataFrame(scaler.fit_transform(df), columns=df.columns)
print(df_normalized)
```

6. Transformation des Données

Techniques Courantes de Transformation des Données

Standardisation : Transformation des données pour qu'elles aient une moyenne de 0 et un écart-type de 1.

```
from sklearn.preprocessing import StandardScaler


scaler = StandardScaler()
df_standardized = pd.DataFrame(scaler.fit_transform(df), columns=df.columns)
print(df_standardized)
```

6. Transformation des Données

Techniques Courantes de Transformation des Données

Encodage one-hot : Conversion des variables catégorielles en variables indicatrices binaires.de 0 et un écart-type de 1.

```
df = pd.DataFrame({'color': ['red', 'blue', 'green']})
df_encoded = pd.get_dummies(df, columns=['color'])
print(df_encoded)
```

6. Transformation des Données

Techniques Courantes de Transformation des Données

Encodage ordinal : Conversion des catégories en valeurs ordinales.

```
from sklearn.preprocessing import OrdinalEncoder


df = pd.DataFrame({'size': ['small', 'medium', 'large']})
encoder = OrdinalEncoder(categories=[['small', 'medium', 'large']])
df['size_encoded'] = encoder.fit_transform(df[['size']])
print(df)
```

6. Transformation des Données

Techniques Courantes de Transformation des Données

Combinaison de fonctionnalités : Création de nouvelles colonnes en combinant les données existantes.

```
df['A+B'] = df['A'] + df['B']
print(df)
```

6. Transformation des Données

Techniques Courantes de Transformation des Données

Transformation mathématique : Application de transformations mathématiques comme les logarithmes, les puissances, etc.

```
df['A_squared'] = df['A'] ** 2
df['log_B'] = df['B'].apply(lambda x: np.log(x) if x > 0 else 0)
print(df)
```

6. Transformation des Données

Techniques de Manipulation des Données Textuelles

Nettoyage et Préparation des Textes : Suppression des caractères spéciaux, conversion en minuscules, etc.

```
df = pd.DataFrame({'text': ['Hello World!', 'Data Science is fun.']})
df['cleaned_text'] = df['text'].str.lower().str.replace('[^\w\s]', '')
print(df)
```

6. Transformation des Données

Techniques de Manipulation des Données Textuelles

Nettoyage et Préparation des Textes : Suppression des caractères spéciaux, conversion en minuscules, etc.

```
df = pd.DataFrame({'text': ['Hello World!', 'Data Science is fun.']})
df['cleaned_text'] = df['text'].str.lower().str.replace('[^\w\s]', '')
print(df)
```

Application 3: Transformation des données

1. Importation des Bibliothèques et Chargement des Données
- Importez les bibliothèques nécessaires.
- Generer une donnée de 5000 enregistrements avec les champs suivants: id, name, age, gender et salary join_date puis exportez sous format excel.
- Dans un autre programme, chargez les données à partir d'un fichier precedent.

2. Inspection Initiale des Données
- Affichez les premières lignes du DataFrame.
- Affichez des informations résumées sur les données (types de données, valeurs manquantes, etc.).
- Affichez un résumé statistique des données.

Application 3: Transformation des données

3. Nettoyage des Données

- Affichez le nombre de valeurs manquantes pour chaque colonne.
- Imputez les valeurs manquantes de la colonne 'age' avec la moyenne.
- Imputez les valeurs manquantes de la colonne 'salary' avec la médiane.
- Supprimez les lignes avec des valeurs manquantes dans la colonne 'gender'.
- Uniformisez les formats des noms (par exemple, mettez tous les noms en minuscules).
- Convertissez la colonne 'join_date' en format date.
- Identifiez et supprimez les doublons dans le DataFrame.
- Détectez et ajustez les valeurs aberrantes dans la colonne 'salary' (par exemple, remplacez les salaires supérieurs à un certain seuil par la médiane)

Application 3: Transformation des données

4. Transformation des Données

- Encodez la colonne 'gender' en utilisant le one-hot encoding.
- Normalisez les colonnes 'age' et 'salary' pour qu'elles soient comprises entre 0 et 1.
- Standardisez les colonnes 'age' et 'salary' pour qu'elles aient une moyenne de 0 et un écart-type de 1.
- Créez une nouvelle colonne 'years_since_join' qui représente le nombre d'années depuis la date d'adhésion.

3

Traitement du Langage Naturel (NLP)

1. Définition et application

Définition de NLP

Le **Traitement du Langage Naturel** (NLP) est une branche de l'intelligence artificielle qui permet aux ordinateurs de comprendre, interpréter et générer le langage humain. Il combine l'informatique, la linguistique et l'apprentissage automatique pour analyser les textes.

1. Définition et application

Applications du NLP :

- **Chatbots :** Améliorer le service client avec des assistants virtuels.
- **Analyse de sentiments :** Comprendre les opinions des utilisateurs sur les réseaux sociaux, les avis produits, etc.
- **Traduction automatique :** Traduire des textes d'une langue à une autre.
- **Résumé automatique :** Résumer de longs textes pour une lecture rapide.

2. Bibliothèques Python pour le NLP

- **NLTK (Natural Language Toolkit):** NLTK est une bibliothèque complète pour le traitement de texte.
- **spaCy** est une bibliothèque NLP rapide et précise. Elle est utilisée pour le tokenization, le POS tagging, l'analyse de dépendance, la reconnaissance d'entités nommées (NER), etc.
- **Gensim** est utilisé pour le traitement thématique et la modélisation de sujets. Il est également connu pour son implémentation de Word2Vec.
- **Transformers** est utilisé pour les modèles de langue de pointe comme BERT, GPT, et leurs variantes. Il est excellent pour les tâches de classification, de génération de texte, de traduction, etc
- **TextBlob** est utilisé pour les tâches NLP de base comme le POS tagging, l'analyse de sentiment, et la traduction.
- **Scikit-learn** est utilisé pour le prétraitement du texte et la classification, notamment avec des outils comme CountVectorizer et TfidfVectorizer.

Prétraitement et Représentation de Textes

Travaux en équipes:

- Présentez chacune des opérations de prétraitement ou de représentation de textes ci-dessous puis écrire un exemple de code en python montrant son utilisation

	Tâche	Groupe
Prétraitement des Textes	Tokenization	1
	Stop words	2
	Stemming et Lemmatization	3
Représentation des Textes	Bag of Words (BoW)	4
	Term Frequency-Inverse Document Frequency (TF-IDF)	5
	Word Embeddings (Word2Vec)	6

Prétraitement et Représentation de Textes

Restitution des travaux par chaque Équipe.

3. Prétraitement des Textes

Tokenization : Découper le texte en unités plus petites (mots, phrases).

```
import nltk
nltk.download('punkt')
from nltk.tokenize import word_tokenize

text = "Natural Language Processing is an exciting field of study."
tokens = word_tokenize(text)
print("Tokens:", tokens)
```

3. Prétraitement des Textes

Stop Words : Éliminer les mots fréquents mais peu significatifs.

```
from nltk.corpus import stopwords
nltk.download('stopwords')

stop_words = set(stopwords.words('english'))
filtered_tokens = [word for word in tokens if word.lower() not in stop_words]
print("Filtered Tokens:", filtered_tokens)
```

3. Prétraitement des Textes

Stemming et Lemmatization : Réduire les mots à leur forme de base.

```
from nltk.stem import PorterStemmer, WordNetLemmatizer
nltk.download('wordnet')

stemmer = PorterStemmer()
lemmatizer = WordNetLemmatizer()

stemmed_tokens = [stemmer.stem(word) for word in filtered_tokens]
print("Stemmed Tokens:", stemmed_tokens)

lemmatized_tokens = [lemmatizer.lemmatize(word) for word in filtered_tokens]
print("Lemmatized Tokens:", lemmatized_tokens)
```

3. Prétraitement des Textes

Part-of-Speech (POS) Tagging : Associe à chaque mot dans un texte une catégorie grammaticale (nom, verbe, adjectif, etc.).

```
import nltk
from nltk.tokenize import word_tokenize

# Téléchargement des ressources nécessaires
nltk.download('punkt')
nltk.download('averaged_perceptron_tagger’)

# Étiquetage des parties du discours (POS tagging)
pos_tags = nltk.pos_tag(tokens)
print("POS Tags:", pos_tags)
```

3. Représentation des Textes

Bag of Words (BoW) : Convertir les textes en vecteurs de mots.

```
from sklearn.feature_extraction.text import CountVectorizer

corpus = [
    "Natural Language Processing is fun.",
    "I love learning about NLP.",
    "Text mining and data analysis are interesting."
]

vectorizer = CountVectorizer()
X_bow = vectorizer.fit_transform(corpus)
print("Bag of Words Representation:\n", X_bow.toarray())
```

4. Représentation des Textes

TF-IDF : Prendre en compte la fréquence des mots et leur importance relative.

```
from sklearn.feature_extraction.text import TfidfVectorizer

tfidf_vectorizer = TfidfVectorizer()
X_tfidf = tfidf_vectorizer.fit_transform(corpus)
print("TF-IDF Representation:\n", X_tfidf.toarray())
```

4. Représentation des Textes

Word Embeddings : Utiliser des représentations vectorielles avancées comme Word2Vec, GloVe.

```
from gensim.models import Word2Vec

sentences = [text.split() for text in corpus]
model = Word2Vec(sentences, vector_size=100, window=5, min_count=1, workers=4)
word_vectors = model.wv
print("Word Embedding for 'Natural':", word_vectors['Natural'])
```

5. Application

Tâches à réaliser :

1. Prétraitement

- Normalisez le texte en le convertissant en minuscules.
- Tokenisez le texte en mots.
- Supprimez les stop words et la ponctuation.
- Appliquez la lemmatisation aux tokens.

2. Extraction des Caractéristiques

- Créer une matrice Bag of Words.
- Créer une matrice TF-IDF.

3. Effectuez le POS tagging

Résultats Attendus:

- Une liste de tokens prétraités.
- Une matrice Bag of Words et une matrice TF-IDF, avec les noms des caractéristiques.
- Une liste des tokens avec leurs étiquettes POS et dépendances, ainsi que les entités nommées détectées dans le texte.

IncH.Class
Powered by SIMPLON.CO

5. Application

Appliquer toutes les techniques de prétraitement et de représentation de texte étudiées dans le cours sur le texte suivant:

Le traitement du langage naturel (NLP) est un domaine de l'intelligence artificielle (IA) qui vise à permettre aux ordinateurs de comprendre, interpréter et générer le langage humain. Le NLP combine la linguistique informatique, l'intelligence artificielle et l'apprentissage automatique pour traiter des données textuelles. Parmi les applications du NLP, on trouve la traduction automatique, l'analyse des sentiments, la reconnaissance vocale et les chatbots. Grâce aux avancées technologiques récentes, le NLP a fait des progrès significatifs, permettant des interactions homme-machine plus naturelles et intuitives. Cependant, des défis persistent, notamment la gestion des ambiguïtés du langage et la compréhension du contexte culturel.

4

Règles d'association

1.Introduction aux Règles d'Association

Définition:

Les **règles d'association** sont des techniques utilisées pour identifier les relations entre les éléments dans de grandes bases de données.
Elles sont principalement utilisées dans les systèmes de recommandation, l'analyse de panier d'achat, et le marketing de détail.

1.Introduction aux Règles d'Association

Définition:

Une **règle d'association** est une application de la forme X→ Y

- où X et Y sont des ensembles d'items disjoints.

Dans notre cas , une règle peut s'écrire :

SI Produit1 **ALORS** Produit2

1.Introduction aux Règles d'Association

Applications

- **Analyse de panier d'achat** : Identifier les produits fréquemment achetés ensemble.
- **Systèmes de recommandation** : Recommander des produits ou des services basés sur les comportements d'achat passés.
- **Détection de fraudes** : Identifier les transactions suspectes en détectant des modèles inhabituels.

1.Introduction aux Règles d'Association

Concepts clés :

- **Itemset** : Un ensemble d'éléments.
- **item** : un élément d'un ensemble (un produit)
- **sup(itemset)** : nombre de transactions d'apparition simultanée des produits
- **card(itemset)** : nombre de produits dans l'ensemble.

1.Introduction aux Règles d'Association

Concepts clés :

Une **transaction** représente un ensemble d'articles ou objets dans une base des données.

Nous pouvons représenter les transactions comme :

1. Liste
2. Représentation verticale
3. Représentation horizontale

1.Introduction aux Règles d'Association

Concepts clés :

Une Liste

- Chaque ligne représente une transaction
- Chaque ligne liste les items achetés par le consommateur
- Les lignes peuvent avoir un numéro différent de colonnes

Représentation verticale : seulement deux colonnes

- une colonne pour les numéros de la transaction (id)
- Une colonne indiquant un item présent

Représentation horizontale : Les transactions se représentent avec une matrice binaire

- Chaque ligne de la matrice représente une transaction
- Chaque colonne représente un article ou item
- Si un item est présent dans une transaction sera représenté avec
- un 1
- Si un item est absent sera représenté avec un 0

1.Introduction aux Règles d'Association

Concepts clés :

Soit la représentation de l'ensemble de transactions suivantes :

Transaction 1: [pain, lait, fromage]
Transaction 2: [lait, beurre]
Transaction 3: [beurre, fromage]
Transaction 4: [pain, lait, beurre]
Transaction 5: [pain, fromage]

Question : De quelle de représentation s'agit-il ? Donner deux autres représentations possible.

1.Introduction aux Règles d'Association

Concepts clés :

Représentation verticale

Id	Transaction
1	**{ pain, lait, fromage }**
2	**{ lait, beurre }**
3	**{ beurre, fromage }**
4	**{ pain, lait, beurre }**
5	**{ pain, fromage }**

1.Introduction aux Règles d'Association

Concepts clés :

Représentation horizontale

Transaction	pain	lait	fromage	beurre
1	1	1	1	0
2	0	1	0	1
3	0	0	1	1
4	1	1	0	1
5	1	0	1	0

1.Introduction aux Règles d'Association

Concepts clés :

Support : fait référence au nombre de transactions (observées) qui le contient. C'est un indicateur de fiabilité

Calcul du support d'une règle d'association :

$$\sigma(X \rightarrow Y) = \frac{\sigma(X \bigcup Y)}{N}$$

N est le nombre de transactions.

Confiance : Elle est définie comme le nombre de fois qu'une combinaison d'articles a été vendue ensemble, divisé par le nombre de fois que le premier article apparaît dans l'ensemble de données. C'est un indicateur de précision

Calcul de la confiance d'une règle d'association :

$$c(X \rightarrow Y) = \frac{\sigma(X \bigcup Y)}{\sigma(X)}$$

1.Introduction aux Règles d'Association

Concepts clés :

Lift : Mesure de l'intérêt d'une règle d'association. Un lift supérieur à 1 indique une corrélation positive entre les items.

$$Lift(X \rightarrow Y) = \frac{\sigma(X \rightarrow Y)}{\sigma(X) * \sigma(Y)}$$

C'est un indicateur de pertinence

2. Méthodes de Génération des Règles

Algorithme Apriori : Algorithme de base pour la génération de règles d'association. Il utilise une approche itérative pour trouver les itemsets fréquents.

Algorithme Eclat : Basé sur une approche de recherche en profondeur.

FP-Growth : Utilise une structure de données appelée arbre FP (Frequent Pattern) pour compresser la base de données et générer les itemsets fréquents.

2. Méthodes de Génération des Règles

Algorithme Apriori : Algorithme de base pour la génération de règles d'association. Il utilise une approche itérative pour trouver les itemsets fréquents.

Algorithme Eclat : Basé sur une approche de recherche en profondeur.

FP-Growth : Utilise une structure de données appelée arbre FP (Frequent Pattern) pour compresser la base de données et générer les itemsets fréquents.

2. Méthodes de Génération des Règles

Algorithme Apriori

L'algorithme Apriori s'execute en deux étapes :

1. Génération de tous les itemsets fréquents c'est-à-dire :

$$\text{IF} = \{ X_i \subseteq T \mid \text{supp}(X_i) = X_i.count \geqslant \text{minsupp}, \quad i=1,2..,n \}$$

2. Génération de toutes les règles d'associations de confiance à partir des itemsets fréquents, c'est-à-dire

$$\{ X_i, Y_j \subseteq IF \mid X_i \cap Y_j = \varnothing \wedge \text{Conf}(X_i \rightarrow Y_j) \geqslant \text{minconf} \quad i=1,2,\ldots,p \; j=1,2,\ldots,q \}$$

minsupp est l'indice de support minimum donné, et **minconf** l'indice de confiance donné.

3. Exemple d'application

Vous disposez d'un ensemble de données représentant les transactions des clients dans un magasin de vêtements. Chaque transaction se compose d'articles achetés par un client.

- Transaction 1 : {Chemise, Pantalon, Cravate}
- Transaction 2 : {Chemise, Veste, Chaussures}
- Transaction 3 : {Pantalon, Chaussures, Chapeau}
- Transaction 4 : {Chemise, Pantalon, Chaussures}
- Transaction 5 : {Veste, Chapeau}

1. Donner une représentation binaire de ces données
2. Utilisez l'algorithme Apriori pour trouver tous les ensembles fréquents avec un support minimum de 2 transactions.
3. En utilisant les ensembles fréquents trouvés à l'étape précédente, générez toutes les règles d'association possibles avec une confiance minimale de 50%. Puis calculer le support et la confiance pour chaque règle d'association générée.
4. Identifiez les règles d'association intéressantes. Et Interpréter ces règles.

5

Régression linéaire ET logistique

1. Introduction à la Régression Linéaire

Définition:

La régression linéaire est une méthode statistique utilisée pour modéliser la relation entre une variable dépendante (ou réponse) et une ou plusieurs variables indépendantes (ou prédictions) X.
Elle est largement utilisée dans divers domaines tels que la finance, la santé, l'économie, etc., pour prédire des résultats futurs basés sur des données historiques.
L'objectif est de trouver la meilleure ligne droite (modèle linéaire) qui décrit la relation entre les variables. La variable Y est appelée **variable dépendante**, ou **variable à expliquer** et les variables Xj (j=1,...,q) sont appelées **variables indépendantes**, ou **variables explicatives.**

2. Formule de la Régression Linéaire

Pour une variable indépendante X et une variable dépendante Y , la relation est modélisée par l'équation : $Y = aX + b$

- a: est le coefficient de pente (la inclinaison de la ligne).
- b : intercept (est l'ordonnée à l'origine : le point où la ligne coupe l'axe des ordonnées)

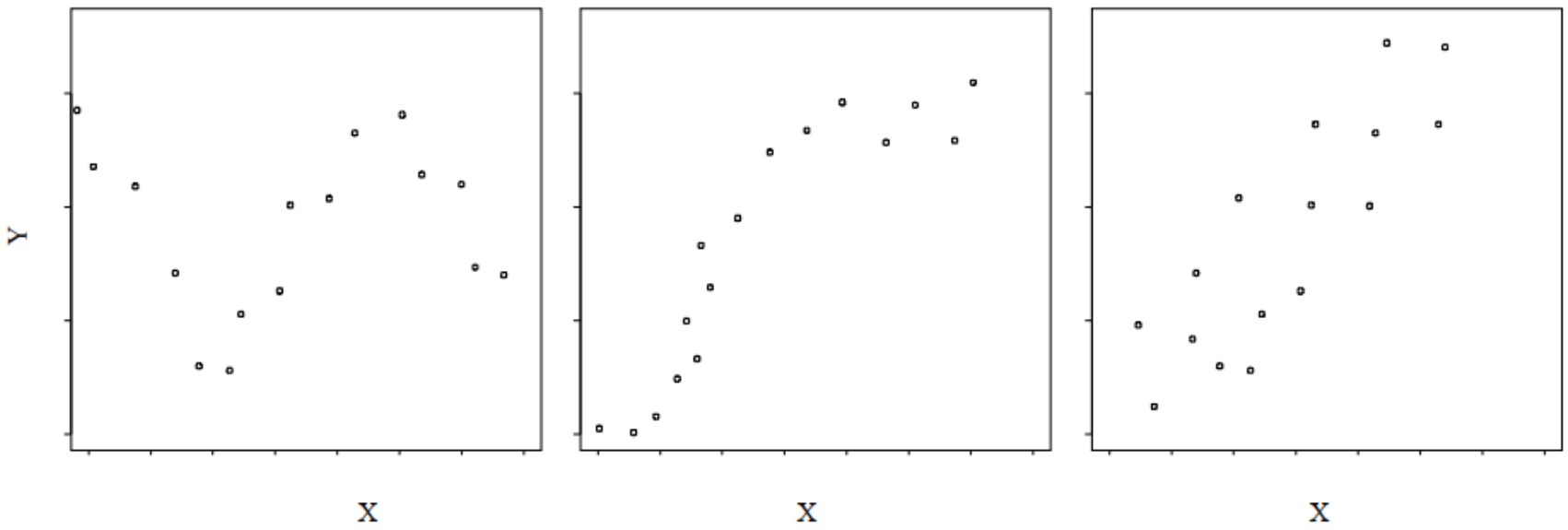
3. Représentation graphique

Avant toute analyse, il est intéressant de représenter les données. Le but de la régression simple est de chercher une fonction f telle que

$$y_i \approx f(x_i).$$

Ainsi une étude de régression simple débute toujours par un tracé des observations (xi, yi), i = 1, ..., n. Cette première représentation permet de savoir si le modèle linéaire est pertinent. Le graphique suivant représente trois nuages de points déférents.

3. Représentation graphique



Au vue du graphique, il semble inadéquat de proposer une régression linéaire pour les 2 premiers graphiques, le tracé présentant une forme sinusoïdale ou sigmoïdale. Par contre, la modélisation par une droite de la relation entre *X* et *Y* pour le dernier graphique semble correspondre à une bonne approximation de la liaison.

3. Méthode des Moindres Carrés

La méthode des moindres carrés est une approche statistique utilisée pour estimer les paramètres a et b en minimisant la somme des carrés des écarts entre les valeurs observées et les valeurs prédites par le modèle.

Principe de la Méthode des Moindres Carrés

La méthode des moindres carrés consiste à ajuster une ligne droite (ou une hyperplane dans le cas de plusieurs variables explicatives) à un ensemble de données de telle sorte que la somme des carrés des différences verticales entre les points de données réels et les points de données prédits par la ligne soit minimale.

3. Méthode des Moindres Carrés

L'équation de la ligne de régression est donnée par :

où :

$$y = \beta_0 + \beta_1 x$$

- y est la variable dépendante,
- x est la variable indépendante,
- β_0 est l'ordonnée à l'origine (intercept),
- β_1 est la pente de la ligne (coefficient de régression).

3. Méthode des Moindres Carrés

Formulation Mathématique

Pour un ensemble de données (x_i, y_i) où $i = 1,2, \dots, n$ les coefficients β_0 et β_1 sont déterminés en minimisant la fonction de coût suivante :

$$\text{SSE} = \sum_{i=1}^{n} (y_i - (\beta_0 + \beta_1 x_i))^2$$

La minimisation de cette somme des carrés des erreurs (SSE) conduit aux équations normales suivantes :

$$\beta_1 = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n} (x_i - \bar{x})^2}$$

$$\beta_0 = \bar{y} - \beta_1 \bar{x}$$

3. Méthode des Moindres Carrés

Formulation Mathématique

où :

- $\bar{x}$ est la moyenne des x_i,
- $\bar{y}$ est la moyenne des y_i.

3. Étapes de la Régression Linéaire

1. Importation des Bibliothèques Nécessaires
2. Chargement et Préparation des Données
3. Séparation des Données en Ensembles d'Entraînement et de Test
4. Création du Modèle de Régression Linéaire
5. Entraînement du Modèle
6. Prédiction avec le Modèle
7. Évaluation du Modèle
8. Interprétation des Résultats

3. Étapes de la Régression Linéaire

Importation des Bibliothèques Nécessaires

1. **Pandas** : Manipulation et analyse de données.
2. **NumPy** : Calculs numériques et manipulation de tableaux.
3. **Scikit-learn** : Bibliothèque pour le machine learning.
   - train_test_split : Divise les données en ensembles d'entraînement et de test.
   - LinearRegression : Crée un modèle de régression linéaire.
   - model.fit : Entraîne le modèle.
   - model.predict : Fait des prédictions avec le modèle.
   - mean_squared_error : Calcule l'erreur quadratique moyenne. r2_score : Calcule le coefficient de détermination.

3. Étapes de la Régression Linéaire

Chargement des Données & Séparation des Données :

- Chargez vos données dans un DataFrame et préparez-les pour l'analyse.

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

- Divisez vos données en ensembles d'entraînement et de test pour valider le modèle.

```
X = df[[‘feature1', feature2’, ' feature1’,…]]
y = df[label']

# Séparation des données
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)
```

3. Étapes de la Régression Linéaire

Choix, entrainement du modèle et prédiction

- Instanciez le modèle de régression linéaire

```
model = LinearRegression()
```

- Entraînez le modèle sur les données d'entraînement.

```
model.fit(X_train, y_train)
```

- Utilisez le modèle pour faire des prédictions sur les données de test.

```
y_pred = model.predict(X_test)
```

3. Étapes de la Régression Linéaire

Évaluation du modèle.

Pour évaluer la performance d'un modèle de régression linéaire, on utilise les métriques suivantes :

- Coefficient de détermination (R^2) : mesure la proportion de la variance de Yqui est expliquée par X

```
r2 = r2_score(y_test, y_pred)
```

- Erreur quadratique moyenne (MSE) : mesure la moyenne des carrés des erreurs.

```
mse = mean_squared_error(y_test, y_pred)
```

- Erreur absolue moyenne (MAE) : mesure la moyenne des valeurs absolues des erreurs.

```
mae = mean_absolute_error(y_test, y_pred)
```

3. Étapes de la Régression Linéaire

Interprétation des Résultats

- Coefficients de régression (pente et intercept) : Les coefficients de régression représentent la pente et l'ordonnée à l'origine de la droite de régression. Ils indiquent comment la variable indépendante influence la variable dépendante. Par exemple, un coefficient positif indique une relation positive, tandis qu'un coefficient négatif indique une relation négative.
- Coefficient de détermination (R^2) : Le coefficient de détermination mesure la proportion de la variance de la variable dépendante qui est expliquée par le modèle. Plus il est proche de 1, plus le modèle est capable d'expliquer la variation des données observées.

3. Étapes de la Régression Linéaire

Interprétation des Résultats

- Erreur quadratique moyenne (RMSE) ou Erreur absolue moyenne (MAE) : Ces mesures quantifient l'écart entre les valeurs prédites par le modèle et les valeurs réelles dans l'ensemble de données. Elles fournissent une indication de l'erreur moyenne du modèle.
- Analyse des résidus : L'examen des résidus (différences entre les valeurs observées et prédites) est crucial pour évaluer si les hypothèses de la régression linéaire sont respectées. Des résidus normalement distribués et homogènes suggèrent que le modèle est approprié.
- Validation croisée : La validation croisée est utilisée pour évaluer les performances prédictives du modèle en testant sa capacité à généraliser sur des données non vues. Une faible erreur de validation croisée indique une bonne capacité de généralisation.

4. Applications de la Régression Linéaire

- **Prévision des Ventes** : Utiliser des données historiques pour prédire les ventes futures en fonction de différents paramètres tels que les dépenses publicitaires, les promotions, les saisons, etc.
- **Estimation des Prix de l'Immobilier** : Évaluer les prix des biens immobiliers en fonction de caractéristiques telles que la surface habitable, le nombre de chambres, l'emplacement, l'année de construction, etc.
- **Analyse des Tendances Économiques**: Étudier et prédire les tendances économiques comme le PIB, les taux de chômage, les taux d'inflation, etc., en utilisant des indicateurs économiques et des données historiques.
- **Analyse de la Satisfaction Client** : Comprendre les facteurs influençant la satisfaction des clients en analysant les réponses aux enquêtes et les commentaires des clients.

4. Applications de la Régression Linéaire

- **Études de Marché :** Analyser les données démographiques et comportementales pour comprendre les préférences des consommateurs et prévoir les comportements d'achat.
- **Analyse des Performances Scolaires** : Étudier les facteurs qui influencent les performances des étudiants dans le but d'améliorer les méthodes d'enseignement et les programmes scolaires.
- **Analyse Financière:** Prévoir les performances financières des entreprises ou des portefeuilles d'investissement en fonction de divers indicateurs financiers.
- **Modélisation de la Demande Énergétique** : Prédire la consommation d'énergie en fonction de variables telles que la température, le jour de la semaine, la production industrielle, etc.

5. La régression logistique

- **Définition**

La **régression logistique** est une méthode statistique et d'apprentissage automatique largement utilisée pour les problèmes de classification binaire, c'est-à-dire utilisée pour prédire une variable binaire en fonction d'une ou plusieurs variables indépendantes.

Elle est basée sur la fonction logistique, qui transforme une combinaison linéaire des caractéristiques d'entrée en une probabilité entre 0 et 1, représentant la probabilité qu'une observation appartienne à une classe spécifique.

5. La régression logistique

Définition

- Soit Y la variable à prédire (variable expliquée)
- $X = (X_1, X_2, \dots, X_J)$ les variables prédictives (variables explicatives).
- La variable Y prend deux modalités possibles {1, 0}
- Les variables Xj sont exclusivement continues ou binaires.
- Y aléatoire et Xi non aléatoires

On cherche à expliquer la survenue d'un évènement
On cherche la probabilité de succès
On travaille en terme d'espérance

5. La régression logistique

Notation

On note:

- $(Y, X_1, X_2, \dots, X_k)$ les variables de la population dont on extrait un échantillon de n individus i.
- (y_i, x_i) est le vecteur des réalisations de (Y_i, X_i)
- K variables explicatives
- Y= $f(x_1, \dots\dots, x_k)$
- f ne peut être une fonction linéaire car Y ne prend que deux valeur

5. La régression logistique

Fonction Logistique : La fonction logistique est une fonction sigmoïde qui prend une entrée réelle et la transforme en une probabilité entre 0 et 1.

Elle est exprimée par :

$$f(x) = \frac{\exp(x)}{1 + \exp(x)} = p \qquad 0<f(x)<1$$

5. La régression logistique

Estimation des Coefficients : Les coefficients ((b_0) et (b_1)) de la régression logistique sont estimés à partir des données d'entraînement en utilisant une méthode appelée estimation de maximum de vraisemblance. Cette méthode cherche à trouver les valeurs de coefficients qui maximisent la probabilité de l'ensemble des observations données 1.

6 Classification supervisée

1. Introduction

Définition:

Classer un document c'est tout simplement mettre une étiquette sur son contenu , lui faisant ainsi appartenir à un groupe ou classe bien définie.

Définition

Un classificateur d'un document est une fonction booléenne (f) qui associe automatiquement un document (D) à la classe (C) :

$$f : D \rightarrow C$$

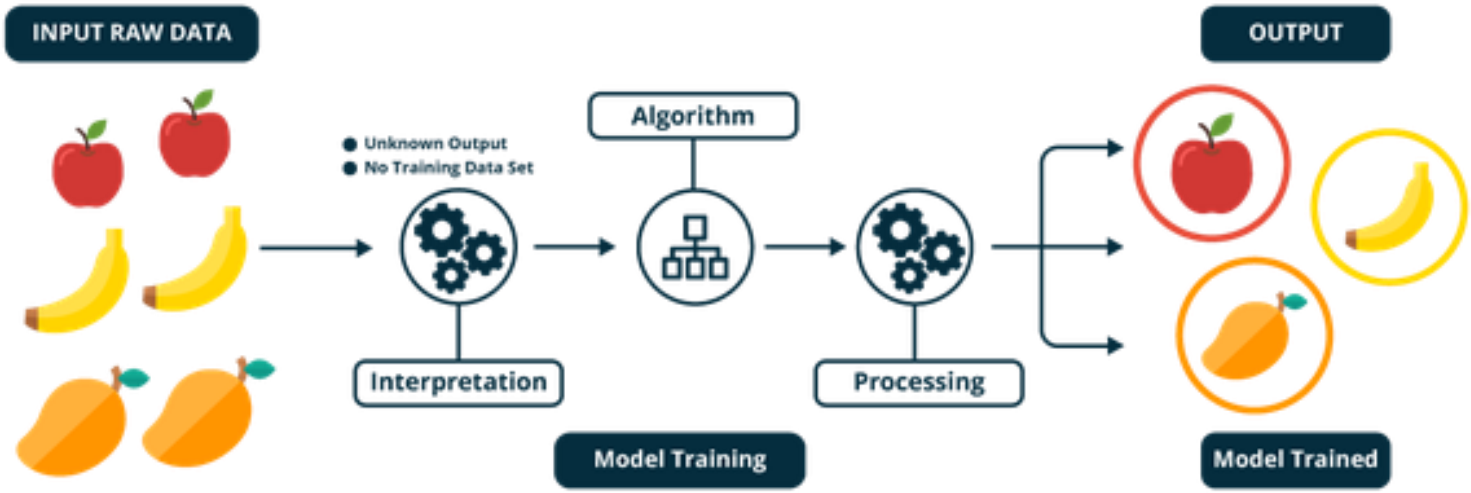
2. Formes de classification

- **Classification Binaire** : Il s'agit tout simplement d'une classification à 2 classes. Par exemple, un système de détection de SPAM classifie un e-mail comme étant soit un "SPAM", soit un "NON-SPAM".
- **Classification Multi-Classe** : Elle consiste à associer un document à une classe parmi plusieurs.
- **Classification Multi-Label** : Elle consiste à associer le texte en entrée à une ou plusieurs classes.
- **Classification en Cascade** : Il s'agit d'un classifieur composé de plusieurs classifieurs mis l'un à la suite de l'autre. Le but étant de classifier suivant des sous-classes.

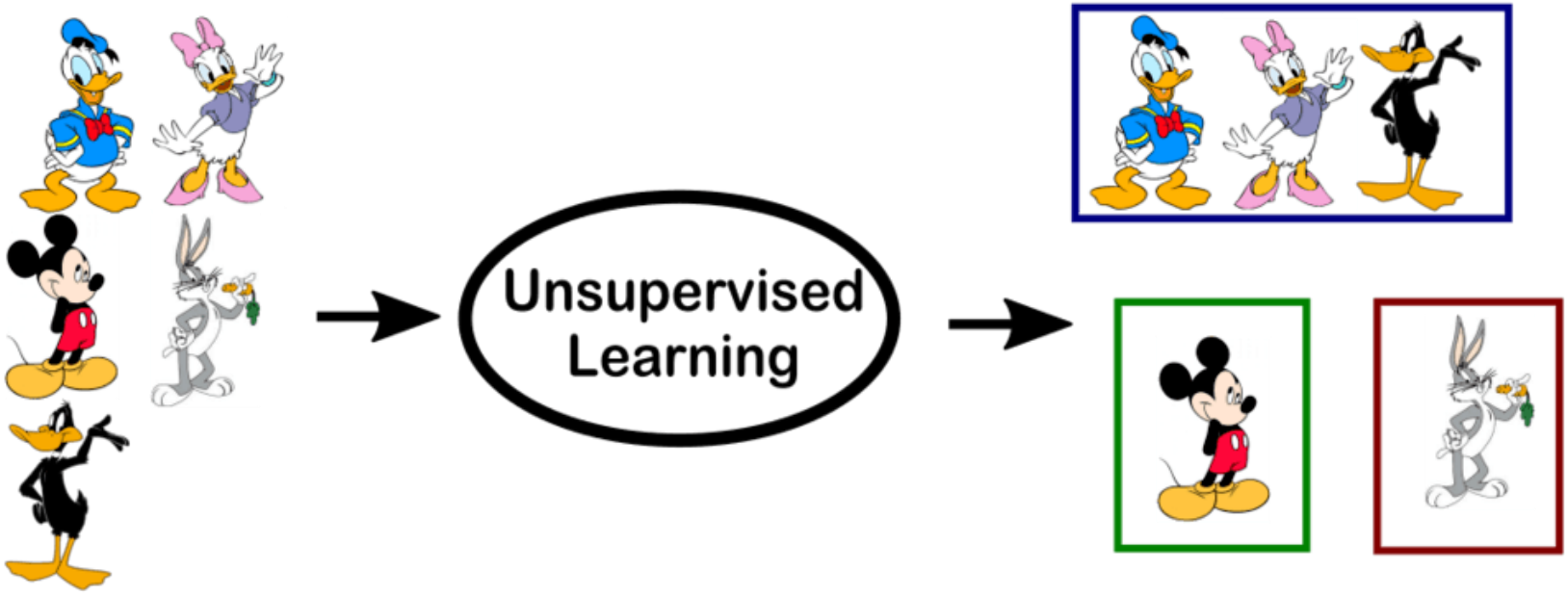
3. Types de classification

1. **La classification non supervisée (clustering)** : Elle consiste à apprendre à classer sans supervision. Au début du processus on ne dispose ni de la définition des classes, ni du nombre. C'est l'algorithme de classification qui va déterminer ces informations.
2. **La classification supervisée (catégorisation)** : Contrairement à l'apprentissage non supervisé, on commence ici par un ensemble de classes connues et définies à l'avance. On dispose aussi d'une sélection initiale de données dont la classification est connue. Ces données sont supposées indépendantes et identiquement distribuées. Elles nous servent pour l'apprentissage de l'algorithme. L 'algorithme réalise donc la classification selon le modèle qu'il a appris

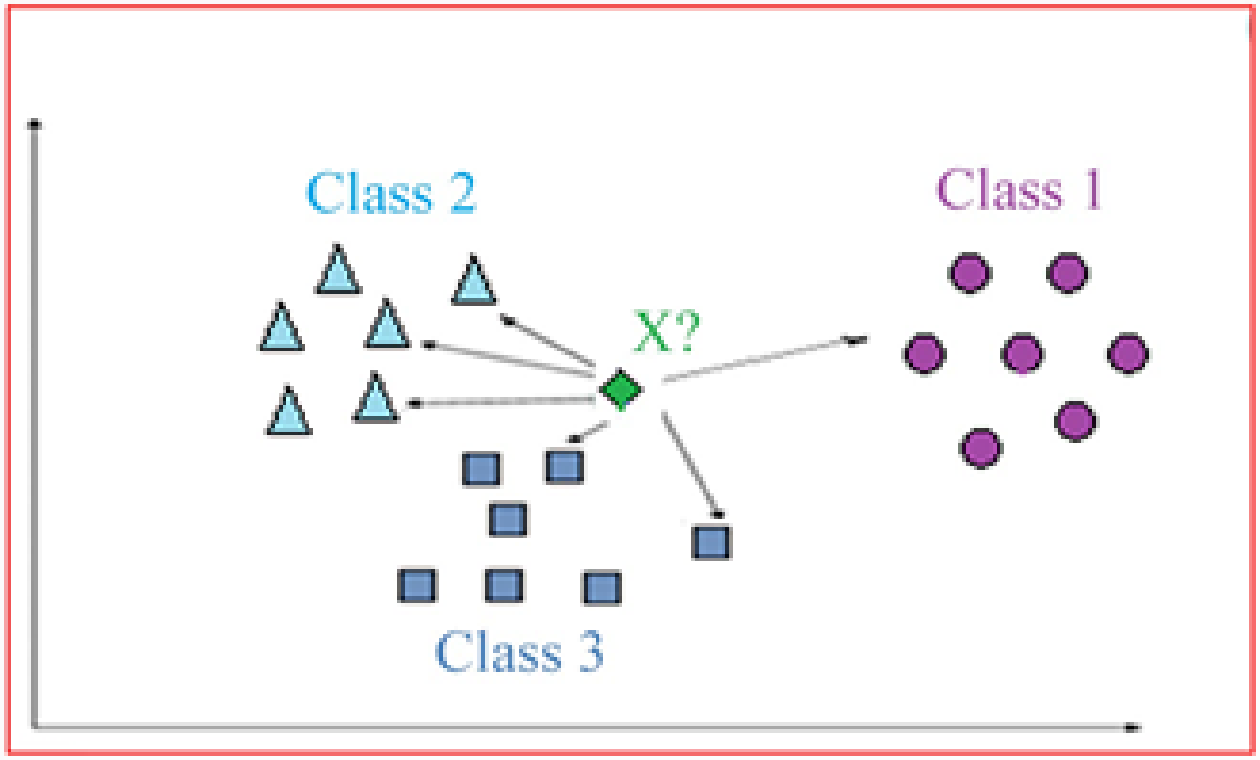
3. Types de classification



3. Types de classification



3. Types de classification



4. Étapes de la Classification Supervisée

1.Préparation des Données :

- **Collecte des données :** Recueillir un ensemble de données avec des caractéristiques (features) et des étiquettes (labels).
- **Prétraitement des données :** Nettoyage des données, gestion des valeurs manquantes, normalisation/standardisation des caractéristiques, etc.
- **Séparation des données :** Division des données en ensembles d'entraînement et de test.

2.Sélection du Modèle :

- Choisir un algorithme de classification supervisée approprié (comme la régression logistique, les k-plus proches voisins (k-NN), les machines à vecteurs de support (SVM), les arbres de décision, etc.).

4. Étapes de la Classification Supervisée

3. Entraînement du Modèle :

- Utiliser l'ensemble d'entraînement pour ajuster les paramètres du modèle afin de minimiser l'erreur de prédiction.

4.Évaluation du Modèle :

- Utiliser l'ensemble de test pour évaluer la performance du modèle en termes de précision, rappel, score F1, etc.

5.Prédiction :

- Utiliser le modèle entraîné pour prédire les étiquettes des nouvelles instances.

5. Algorithmes de classification

Par groupe de 5 personnes, étudiez chacun des algorithmes suivants :

- ID3
- Naive Bayésienne
- KNN
- SVM

Présentez le principe de chaque algorithme, son utilisation et une application théorique.

6. Évaluation des modèles de classification

Matrice de contingence ou de confusion : Cette technique utilise un corpus étiqueté de documents pour lequel on connaît la vraie catégorie de chaque document, et le résultat obtenu par le classifieur.

Pour un corpus, on construit la matrice de contingence pour chaque classe , qui fournit 4 informations essentielles :

- Vrai Positif (VP) : documents correctement classés ;
- Vrai Négatif(VN) : documents correctement non classés ;
- Faux Positif (FP) : documents incorrectement classés ;
- Faux Négatif (FN) : documents incorrectement non classés

6. Évaluation des modèles de classification

- **Matrice de confusion d'une classe**

Catégorie C_i		Jugement expert	
		Oui	Non
Jugement classifieur	Oui	**VPi**	**FPi**
	Non	**FNi**	**VNi**

- **Matrice de confusion de plusieurs classes**

Classifieur \ Expert	$C = \{c_1, c_2, \ldots\ldots, c_{\lvert C\rvert}\}$	C_i	$\neg C_i$
	C_i	$VP = \sum_{i=1}^{\lvert C\rvert} VP_i$	$FP = \sum_{i=1}^{\lvert C\rvert} FP_i$
	$\neg C_i$	$FN = \sum_{i=1}^{\lvert C\rvert} FN_i$	$VN = \sum_{i=1}^{\lvert C\rvert} VN_i$

6. Évaluation des modèles de classification

Le Rappel : Elle est la proportion de documents correctement classés par le système par rapport à tous les documents de la classe Ci, elle mesure la capacité d'un système de classification à détecter les documents correctement classés . Elle est donnée par la relation :

$$\mathbf{Rappel}(C_i) = \frac{\mathbf{Nombredededocumentsbienclassésdans C}_i}{\mathbf{Nombredededocumentsdela C}_i} = \frac{\mathbf{VP}_i}{\mathbf{VP}_i + \mathbf{FN}_i}$$

6. Évaluation des modèles de classification

La précision : est la proportion de documents correctement classés parmi ceux classés par le système dans Ci .Elle mesure la capacité d'un système de classification à ne pas classer un document dans une classe, un document qui ne l'est pas. Comme elle peut aussi être interprétée par la probabilité conditionnelle qu'un document choisi aléatoirement dans la classe soit bien classé par le classifieur. Elle est donnée par la relation

$$\text{Précision}(C_i) = \frac{\text{Nombredededocumentsbienclassésdans}C_i}{\text{Nombredededocumentsclassésdans}C_i} = \frac{VP_i}{VP_i + FP_i}$$

6. Évaluation des modèles de classification

- **Le bruit** est le pourcentage de textes incorrectement associés à une classe par le système. Il est donné par la formule :

$$\textbf{Bruit(B)} = \textbf{1-Précision(P)} = \frac{\textbf{FP}_i}{\textbf{VP}_i + \textbf{FP}_i}$$

- **Le silence** est le pourcentage de textes à associer à une classe incorrectement non classés par le système. Il est donné par la formule

$$\textbf{Silence(S)} = \textbf{1-Rappel(R)} = \frac{\textbf{FN}_i}{\textbf{VP}_i + \textbf{FN}_i}$$

6. Évaluation des modèles de classification

- La **F-mesure** est le plus usuel des indicateurs. Elle prend en compte la valeur relative de la précision et du rappel.Elle est calculée par la formule :

$$\text{F-measure} = \frac{2 \times \text{précision} \times \text{rappel}}{\text{précision} + \text{rappel}}$$

- **Le taux de succès** est le rapport entre les documents bien classés sur le nombre total des documents du corpus.Il se calcule par la formule :

$$\text{Taux succès} = \frac{VP + VN}{VP + FP + VN + FN}$$

6. Évaluation des modèles de classification

- **Taux d'erreur :** est le rapport entre les documents mal classés sur le nombre total des documents du corpus. Il se calcule par la formule :

$$\text{Taux d'erreur} = 1 - \text{Taux succès} = \frac{FP + FN}{VP + FP + VN + FN}$$

7. Cas d'Utilisation Courants

1.Détection de Fraude : Classifier les transactions bancaires en "frauduleuses" ou "non frauduleuses".

2.Diagnostic Médical : Prédire si un patient a une maladie particulière basée sur les symptômes et les résultats des tests.

3.Filtrage de Courriel : Classifier les e-mails en "spam" ou "non spam".

4.Reconnaissance d'Image : Identifier des objets dans des images, comme classifier les images de vêtements dans différentes catégories (t-shirts, pantalons, chaussures, etc.).

8. Cas Pratique

1. Prédiction de sentiment sur les produits
2. Prédiction du classement de l'équipe nationale du football du Cameroun.
3. Prédiction des maladies en fonction des symptômes.
4. Classement des mails en spam ou non
5. Prédiction des réponses à un message

7

Les méthodes De clustering

1. Introduction

ID	Âge	Taille (cm)	Poids (kg)	Moyenne des notes
I1	20	170	65	06
I2	22	160	55	19
I3	21	175	75	12
I4	23	180	85	09
I5	20	165	60	04
I6	21	172	68	17
I7	22	178	70	18
I8	23	167	58	16
I9	24	182	80	12
I10	25	170	65	08

Selon les critères de votre choix, regrouper ces individus en 2 groupes puis 3 groupes

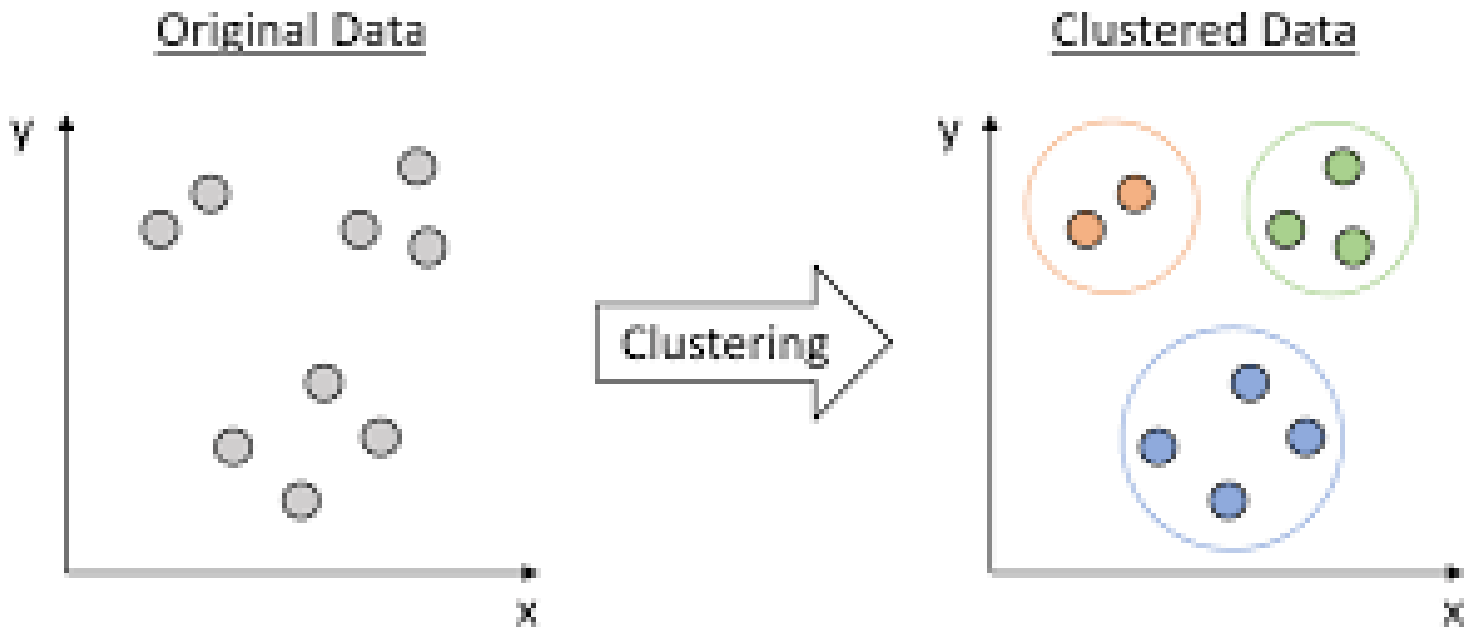
1. Introduction

Définition:

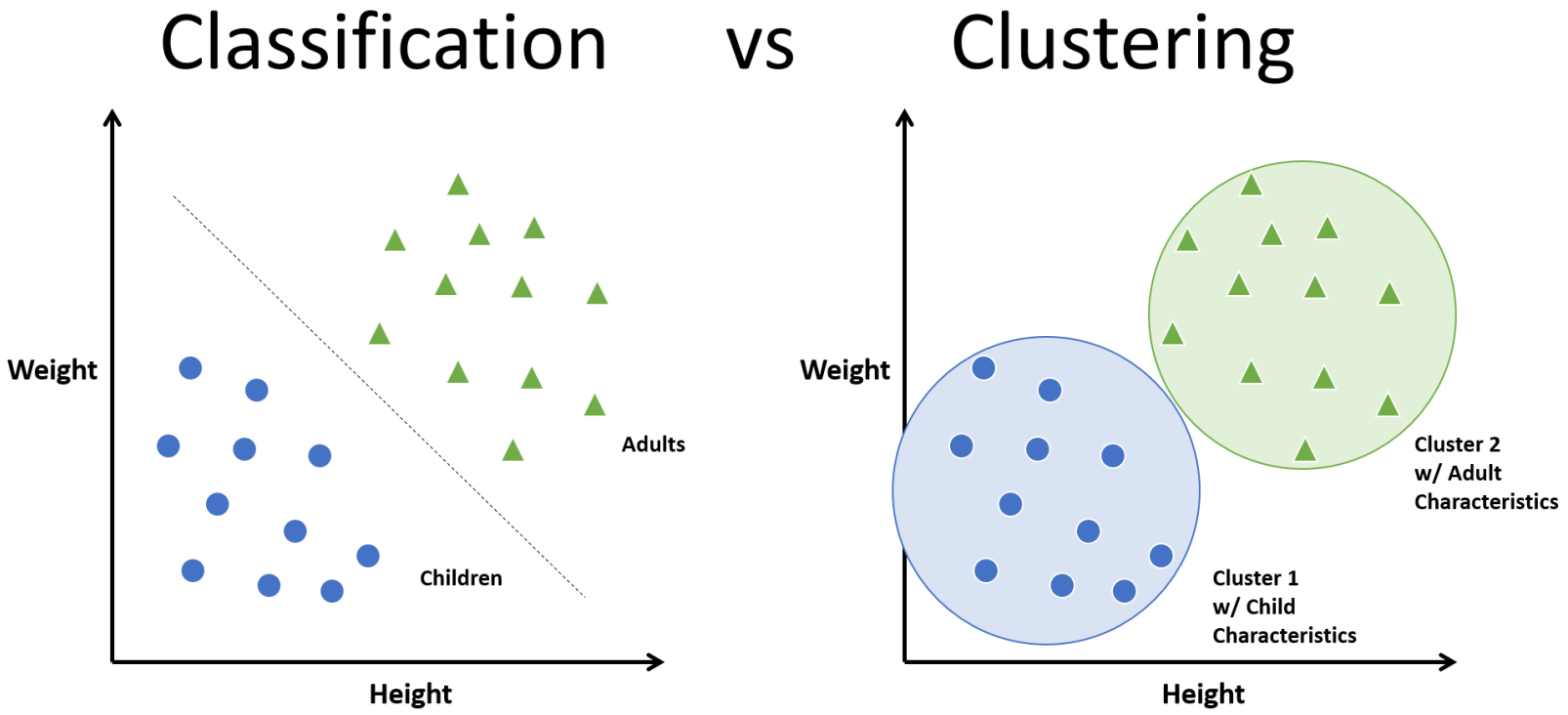
Le clustering est une technique d'apprentissage automatique non supervisée qui consiste à diviser un ensemble de données en groupes homogènes, appelés clusters.

Chaque cluster regroupe des éléments plus similaires entre eux qu'avec ceux des autres clusters**.**

1. Introduction



1. Introduction



2. Applications du Clustering

- **Segmentation de clientèle** : Identifier des groupes de clients aux comportements d'achat similaires pour des campagnes de marketing ciblées.
- **Analyse de réseaux sociaux** : Détecter des communautés dans les réseaux sociaux.
- **Détection d'anomalies** : Identifier des transactions frauduleuses ou des pannes dans les systèmes.
- **Biologie/Génomique** : Regrouper des gènes ou des protéines en fonction de leurs caractéristiques.
- **Compression de données** : Réduire la taille des données en les regroupant.
- **Recommandation de contenu** : Regrouper des articles, des films ou des produits similaires pour faire des recommandations personnalisées.

3. Types de Clustering

1. Clustering Partitionnel
2. Clustering Hiérarchique:
3. Clustering Basé sur la Densité :
4. Clustering Basé sur les Modèles :

4. Étude de l’algorithme Kmeans

Présentation

- K-means est l'un des algorithmes de clustering les plus simples et les plus populaires.
- Il partitionne les données en k clusters en minimisant la somme des distances au carré entre les points et le centre du cluster.

4. Étude de l’algorithme Kmeans

Étapes de l'algorithme K-means:

1. Choisir le nombre de clusters k.
2. Initialiser aléatoirement k centres de clusters.
3. Assigner chaque point au centre de cluster le plus proche.
4. Recalculer les centres des clusters.
5. Répéter les étapes 3 et 4 jusqu'à convergence.

4. Étude de l'algorithme Kmeans

Exemple

Soit A={1,2,3,6,7,8,13,15,17} un ensemble des données numériques.

Créer 3 clusters à partir de A

4. Étude de l'algorithme Kmeans

- On prend 3 objets au hasard. Supposons que c'est 1, 2 et 3. ça donne :

  C1={1}, M1=1,
  C2={2}, M2=2,
  C3={3} et M3=3

- Chaque objet O est affecté au cluster au milieu duquel, O est le plus proche.

  6 est affecté à C3 car ,
  dist(M3,6)<dist(M2,6) et dist(M3,6)<dist(M1,6)

On a :

- C1={1}, M1=1
- C2={2}, M2=2
- C3={3, 6,7,8,13,15,17}, M3=69/7=9.86

4. Étude de l'algorithme Kmeans

- dist(3,M2)<dist(3,M3)→ 3 passe dans C2. Tous les autres objets ne bougent pas. C1={1}, M1=1, C2={2,3},M2=2.5, C3={6,7,8,13,15,17} et M3= 66/6=11

- dist(6,M2) < dist(6,M3) → 6 passe dans C2. Tous les autres objets ne bougent pas. C1={1}, M1=1, C2={2,3,6}, M2=11/3=3.67, C3={7,8,13,15,17}, M3= 12
- dist(2,M1)<dist(2,M2) → 2 passe en C1. dist(7,M2)<dist(7,M3)→ 7 passe en C2. Les autres ne bougent pas. C1={1,2}, M1=1.5, C2={3,6,7}, M2=5.34, C3= {8,13,15,17}, M3=13.25

4. Étude de l'algorithme Kmeans

dist(3,M1)<dist(3,M2)→ 3 passe en 1.
dist(8,M2)<dist(8,M3)→ 8 passe en 2
C1={1,2,3}, M1=2,
C2={6,7,8}, M2=7,
C3={13,15,17}, M3=15
Plus rien ne bouge, on s'arrete car l'algorithme converge. On a donc les classes finales suivantes :

- **C1={1,2,3},**
- **C2={6,7,8}**
- **C3={13,15,17}**

4. Étude de l'algorithme Kmeans

Exercice : Utilisation de l'algorithme Kmeans

Soit l'ensemble D des entiers suivants : D= { 2, 5, 8, 10, 11, 18, 20 } . On veut répartir les données de D en trois (3) clusters, en utilisant l'algorithme Kmeans. La distance d entre deux nombres a et b est calculée ainsi : d(a , b) = |a - b| (la valeur absolue de a moins b)

1. Appliquez l'algorithme Kmeans en choisissant comme centres initiaux des 3 clusters respectivement : 8, 10 et 11. Montrez toutes les étapes de votre calcul.
2. Donnez le résultat final et précisez le nombre d'itérations qui ont été nécessaires.
3. Peut-on avoir un nombre d'itérations inférieur pour ce problème ? Discutez.

4. Étude de l'algorithme Kmeans

Quelques limites de cet algorithme sont :

- Un nombre K grand peut conduire à un partitionnement trop fragmenté des données. Ce qui empêchera de découvrir des patterns intéressants dans les données.
- Par contre, un nombre de clusters trop petit, conduira à avoir, potentiellement, des cluster trop généralistes contenant beaucoup de données. Dans ce cas, on n'aura pas de patterns "fins" à découvrir.
- Pour un même jeu de données, il n'existe pas un unique clustering possible. La difficulté résidera donc à choisir un nombre de cluster K qui permettra de mettre en lumière des patterns intéressants entre les données. **Malheureusement il n'existe pas de procédé automatisé pour trouver le bon nombre de clusters.**

5. Bibliothèques Python pour le Clustering

Scikit-Learn : Fournit une implémentation de base et performante pour les principaux algorithmes de clustering.

Algorithmes : K-means, Mean Shift, DBSCAN, Agglomerative Clustering, Spectral Clustering.

```
from sklearn.cluster import KMeans
```

HDBSCAN : Implémente l'algorithme HDBSCAN (Hierarchical Density-Based Spatial Clustering of Applications with Noise), une extension du DBSCAN.

```
import hdbscan
```

6. Cas Pratique

Contexte : Une entreprise de vente au détail souhaite segmenter ses clients pour mieux cibler ses campagnes marketing.

Données :

- **Nombre de commandes**
- **Montant total dépensé**

Objectif : Utiliser le clustering pour identifier des groupes de clients ayant des comportements d'achat similaires.

Tâches :

1. Générer un dataset de taille 50 000
2. Visualiser les données puis réaliser le nettoyage
3. Transformer les données avec StandardScaler
4. Appliquer l'algorithme Kmeans sur les données
5. Générer un graphique faisant ressortir tous les clusters
6. Interpréter

8

Techniques d'Optimisation d’un modèle

1. Introduction à l'Optimisation de Modèle

L'optimisation de modèles en machine learning consiste à ajuster le modèle et ses paramètres pour améliorer sa performance sur des données de test.

Cela inclut :

1. le choix des bonnes caractéristiques,
2. la sélection des bons hyperparamètres,
3. et l'évaluation correcte des performances.

1. Introduction à l'Optimisation de Modèle

Le But de l'optimisation

- L'objectif ultime de l'optimisation en machine Learning est **d'améliorer la performance des modèles**.

- Cette performance peut être mesurée par divers indicateurs, tels que la **précision** dans le cas de la classification ou le **taux d'erreur dans le cas de la prédiction.**

- L'optimisation vise à trouver les valeurs des paramètres du modèle qui minimisent une certaine fonction de coût, laquelle mesure l'écart entre les prédictions du modèle et les valeurs réelles.

1. Introduction à l'Optimisation de Modèle

Le But de l'optimisation

1. **Minimisation de la fonction de coût** : L'optimisation vise à minimiser la fonction de coût, qui est une représentation mathématique de l'erreur entre les prédictions du modèle et les données réelles..
2. **Amélioration des performances** : En minimisant la fonction de coût, le modèle s'ajuste de manière à améliorer ses performances, ce qui se traduit par des prédictions plus précises et fiables.

2. Techniques d'Optimisation

En utilisant le jeu de données sur le cancer du sein fourni par **scikit-learn**, développer un modèle basé sur l'algorithme KNN qui doit prédire la classe (bénigne ou maligne) d'une tumeur à partir de ses caractéristiques.

1. Importer les bibliothèques suivantes:
   - **from sklearn.datasets import load_breast_cancer**
   - **from sklearn.neighbors import KNeighborsClassifier**
2. Charger les données puis les prétraiter.
3. Subdiviser les données en test et entrainement
4. Choisir le modèle puis l'entrainer
5. Imprimer la précision du modèle.

2. Techniques d'Optimisation

1. Normalisation et Standardisation

La normalisation et la standardisation sont des techniques utilisées pour mettre les caractéristiques sur une même échelle.

```
import pandas as pd
from sklearn.preprocessing import StandardScaler, MinMaxScaler

# Chargement des données
data = pd.read_csv('data.csv')

# Normalisation des données avec Min-Max Scaler
scaler_minmax = MinMaxScaler()
data_normalized = scaler_minmax.fit_transform(data)

# Standardisation des données avec StandardScaler
scaler_standard = StandardScaler()
data_standardized = scaler_standard.fit_transform(data)
```

2. Techniques d'Optimisation

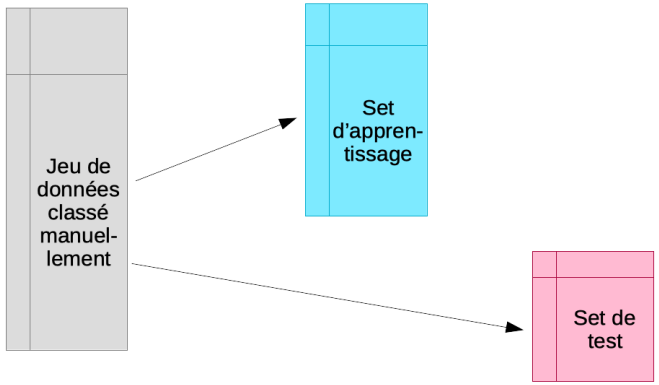
2. Ingénierie des Caractéristiques

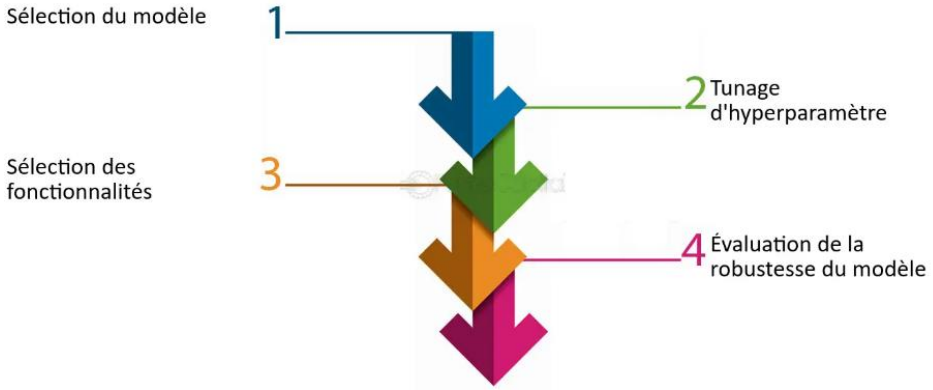
L'ingénierie des caractéristiques consiste à créer de nouvelles caractéristiques à partir des données existantes.

```python
import pandas as pd

# Chargement des données
data = pd.read_csv('data.csv')

# Extraction de nouvelles caractéristiques
data['new_feature'] = data['feature1'] * data['feature2']

# Suppression des caractéristiques redondantes
data = data.drop(columns=['feature1', 'feature2'])
```

2. Techniques d'Optimisation

3. Validation Croisée (Cross-validation)





- La validation croisée est une technique qui consiste à diviser les données en plusieurs sous-échantillons.

- Le modèle est entraîné sur une partie des échantillons et testé sur les échantillons restants. Cela permet d'évaluer la performance du modèle de manière plus fiable.

2. Techniques d'Optimisation

3. Validation Croisée (Cross-validation)

- **Exemple** : Optimisation du modèle pour une classification binaire sur un jeu des données quelconque (Voir code)

2. Techniques d'Optimisation

4. Recherche d'Hyperparamètres:

Elle est une étape cruciale dans le développement de modèles de machine learning. Elle consiste à trouver les meilleures combinaisons de paramètres pour optimiser les performances du modèle.
Il existe deux méthodes principales pour la recherche d'hyperparamètres :

- **Grid Search** (Recherche par grille) est une technique systématique de recherche des meilleures combinaisons de valeurs d'hyperparamètres dans un espace de recherche défini. Cette méthode évalue toutes les combinaisons possibles des hyperparamètres spécifiés.
- **Random Search** (Recherche aléatoire) est une méthode où les combinaisons d'hyperparamètres sont choisies de manière aléatoire dans un espace de recherche défini. Contrairement à Grid Search, Random Search ne teste pas toutes les combinaisons possibles, mais un nombre fixé de combinaisons aléatoires.

2. Techniques d'Optimisation

4. Recherche d'Hyperparamètres:

Comparaison entre Grid Search et Random Search

	Avantages	Inconvénients
Grid Search	Évalue toutes les combinaisons possibles, ce qui garantit que le meilleur ensemble d'hyperparamètres est trouvé dans l'espace de recherche défini.	Peut être très coûteux en temps de calcul si l'espace de recherche est grand.
Random Search	Plus efficace en termes de temps de calcul, surtout lorsque l'espace de recherche est large et que certaines combinaisons d'hyperparamètres sont peu influentes.	Ne garantit pas de trouver le meilleur ensemble d'hyperparamètres car il ne teste pas toutes les combinaisons possibles.

2. Techniques d'Optimisation

4. Recherche d'Hyperparamètres:

Grid Search : Exemple pour le modèle KNN

1. Vous définissez une grille de valeurs possibles pour chaque hyperparamètre. Par exemple, pour un modèle KNN, vous pourriez définir une grille comme suit :
   - Nombre de voisins (n_neighbors): [3, 5, 7, 9]
   - Distance de calcul (metric): ['euclidean', 'manhattan']

2. Grid Search testera toutes les combinaisons possibles de ces valeurs
   - n_neighbors = 3 avec metric = 'euclidean'
   - n_neighbors = 3 avec metric = 'manhattan'n_
   - neighbors = 5 avec metric = 'euclidean'
   - Etc.

2. Techniques d'Optimisation

4. Recherche d'Hyperparamètres:

Random Search : Exemple pour le modèle KNN

1. Pour le modèle KNN, vous définissez des plages ou des distributions de valeurs possibles :

- Nombre de voisins (n_neighbors): distribution uniforme entre 1 et 20
- Distance de calcul (metric): liste ['euclidean', 'manhattan', 'chebyshev']

2. Random Search choisira des valeurs au hasard dans ces plages ou listes.

2. Techniques d'Optimisation

5. Sélection de caractéristiques (Feature Selection)

- La sélection de caractéristiques est une étape cruciale dans le processus de création de modèles d'apprentissage automatique.
- Elle consiste à sélectionner un sous-ensemble des **variables pertinentes** (caractéristiques) pour l'entraînement de votre modèle.
- Cela peut aider à améliorer les performances du modèle, réduire le temps de calcul et éviter le surapprentissage (overfitting).

2. Techniques d'Optimisation

5. Sélection de caractéristiques (Feature Selection)

Méthodes de Sélection de Caractéristiques

Méthodes Basées sur des Statistiques :

- **Variance Threshold** : Élimine les caractéristiques avec une variance en dessous d'un certain seuil.
- **Chi-Square Test** : Évalue l'indépendance de deux événements.
- **ANOVA F-test** : Compare les moyennes de différentes groupes.

2. Techniques d'Optimisation

5. Sélection de caractéristiques (Feature Selection)

Méthodes Basées sur des Modèles :

- **Régression Logistique avec Régularisation L1 (Lasso)** : Sélectionne les caractéristiques en forçant les coefficients des caractéristiques non pertinentes à zéro.
- **Random Forests** : Utilise l'importance des caractéristiques pour sélectionner les plus pertinentes.

Méthodes de Sélection Itérative :

- **Recursive Feature Elimination (RFE)** : Sélectionne les caractéristiques en construisant un modèle et en éliminant les caractéristiques les moins importantes par itérations successives.
- **Forward/Backward Feature Selection** : Ajoute ou retire des caractéristiques itérativement en fonction d'un critère de performance.

2. Techniques d'Optimisation

5. Sélection de caractéristiques (Feature Selection)

Exemple pratique : Voir code

2. Techniques d'Optimisation

Exercice Pratique :

Prenez un dataset de classification binaire et appliquez les techniques d'optimisation mentionnées ci-dessus sur un modèle de votre choix.

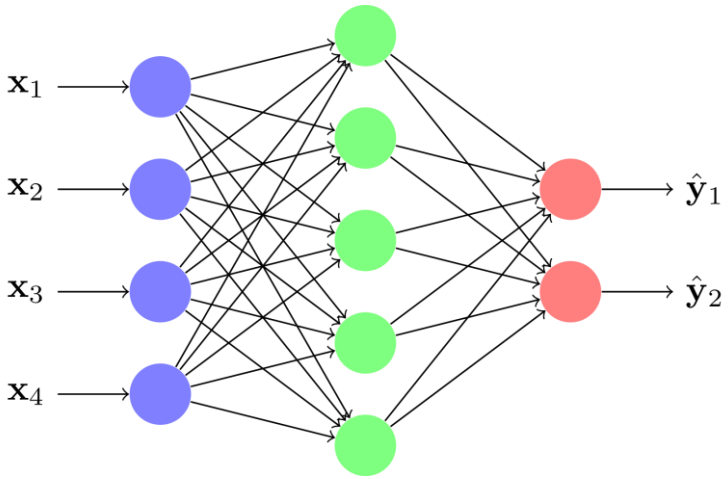
9

Réseaux de Neurones et Deep Learning

1. Introduction aux Réseaux de Neurones

Définition:

- Les **réseaux de neurones** sont des modèles computationnels inspirés du cerveau humain, composés de neurones interconnectés.
- Chaque neurone reçoit des entrées pondérées, les somme et applique une fonction d'activation pour générer une sortie.



1. Introduction aux Réseaux de Neurones

Définition:

Définition

Un **neurone formel** est un modèle mathématique inspiré du fonctionnement des neurones biologiques. Il est composé de trois éléments principaux :

1. **Une entrée** : la valeur d'une caractéristique du problème à résoudre.
2. **Une fonction d'activation** : une fonction qui transforme l'entrée en une sortie.
3. **Une sortie** : la valeur du neurone.

1. Introduction aux Réseaux de Neurones

Définition:

La **fonction d'activation** est un élément important du neurone formel. Elle permet au neurone de traiter l'entrée et de produire une sortie qui soit significative.

Il existe de nombreux types de fonctions d'activation, telles que :

Fonction
▪ la fonction sigmoïde
▪ la fonction tangente hyperbolique
▪ la fonction ReLU
▪ la fonction softmax

1. Introduction aux Réseaux de Neurones

Définition:

Fonction sigmoïde

- Formule : $f(x) = \frac{1}{1+e^{-x}}$
- La fonction sigmoïde prend en entrée un nombre réel et renvoie une sortie dans l'intervalle (0, 1).
- Elle est couramment utilisée dans les couches de sortie des réseaux de neurones pour effectuer une classification binaire, où elle peut être interprétée comme une probabilité.
- Cependant, elle a tendance à souffrir du problème de disparition des gradients (vanishing gradients) lors de l'entraînement de réseaux de neurones profonds, ce qui limite sa capacité à modéliser des relations complexes.

1. Introduction aux Réseaux de Neurones

Fonction tangente hyperbolique

- Formule : $f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$
- La fonction tanh est similaire à la fonction sigmoïde, mais elle renvoie une sortie dans l'intervalle (-1, 1), ce qui la rend centrée autour de zéro.
- Elle est souvent utilisée dans les couches cachées des réseaux de neurones et contribue à résoudre le problème de la symétrie des poids dans les réseaux profonds.
- Cependant, elle peut également souffrir du problème de disparition des gradients pour les réseaux très profonds.

1. Introduction aux Réseaux de Neurones

Fonction Rectified Linear Unit (ReLU)

- Formule : $f(x) = \max(0, x)$
- La fonction ReLU est devenue très populaire en raison de sa simplicité et de ses avantages en termes d'entraînement.
- Elle renvoie zéro si l'entrée est négative et l'entrée elle-même si elle est positive, ce qui la rend non linéaire.
- La fonction ReLU aide à résoudre le problème de disparition des gradients et accélère la convergence de l'entraînement.
- Cependant, elle peut également souffrir du problème de "dying ReLU", où certains neurones peuvent devenir inactifs et ne pas apprendre.

1. Introduction aux Réseaux de Neurones

Fonction softmax

- Formule : Pour un vecteur de scores (ou logits) z de dimension N (où N est le nombre de classes), la fonction softmax calcule les probabilités associées à chaque classe en utilisant la formule suivante : $\text{softmax}(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{N} e^{z_j}}$
- La fonction softmax transforme les scores bruts en une distribution de probabilité sur les classes.
- Chaque composante $\text{softmax}(z)_i$ représente la probabilité que la classe i soit la classe correcte.
- La fonction normalise les scores en exponentiellement accentuant les scores plus élevés.
- Elle garantit que la somme des probabilités pour toutes les classes est égale à 1, ce qui facilite la sélection de la classe la plus probable.

1. Introduction aux Réseaux de Neurones

Un premier modèle : le perceptron

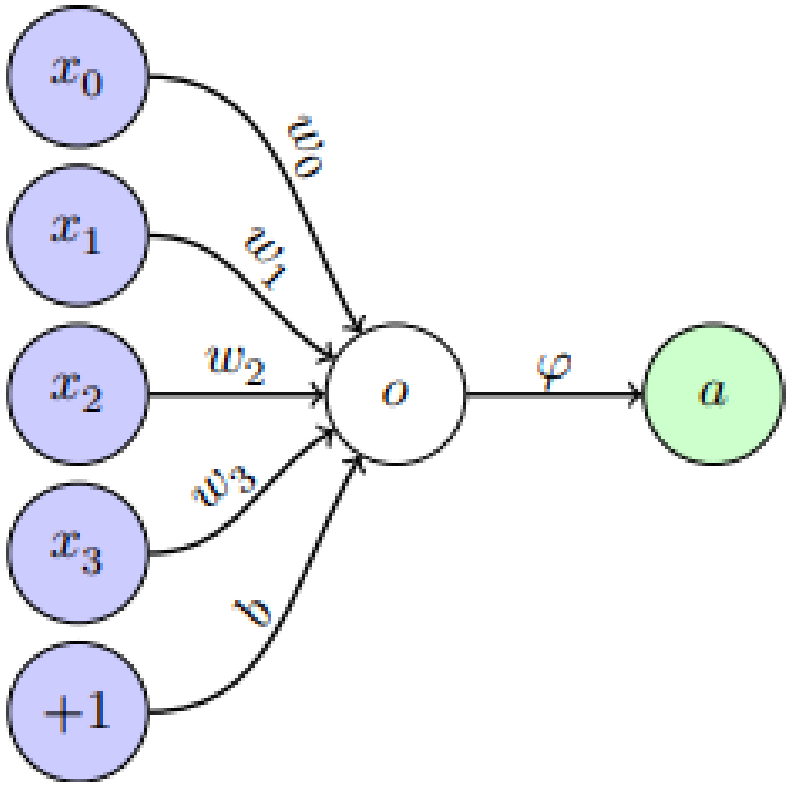
Dans la terminologie des réseaux de neurones, un neurone est une fonction paramétrée qui prend un vecteur x en entrée et sort une valeur unique a comme suit :

$$a = \varphi(\underbrace{wx + b}_{o})$$

où les paramètres du neurone sont ses poids stockés dans w. et un terme de biais b, et φ est une fonction d'activation qui est choisie a priori.

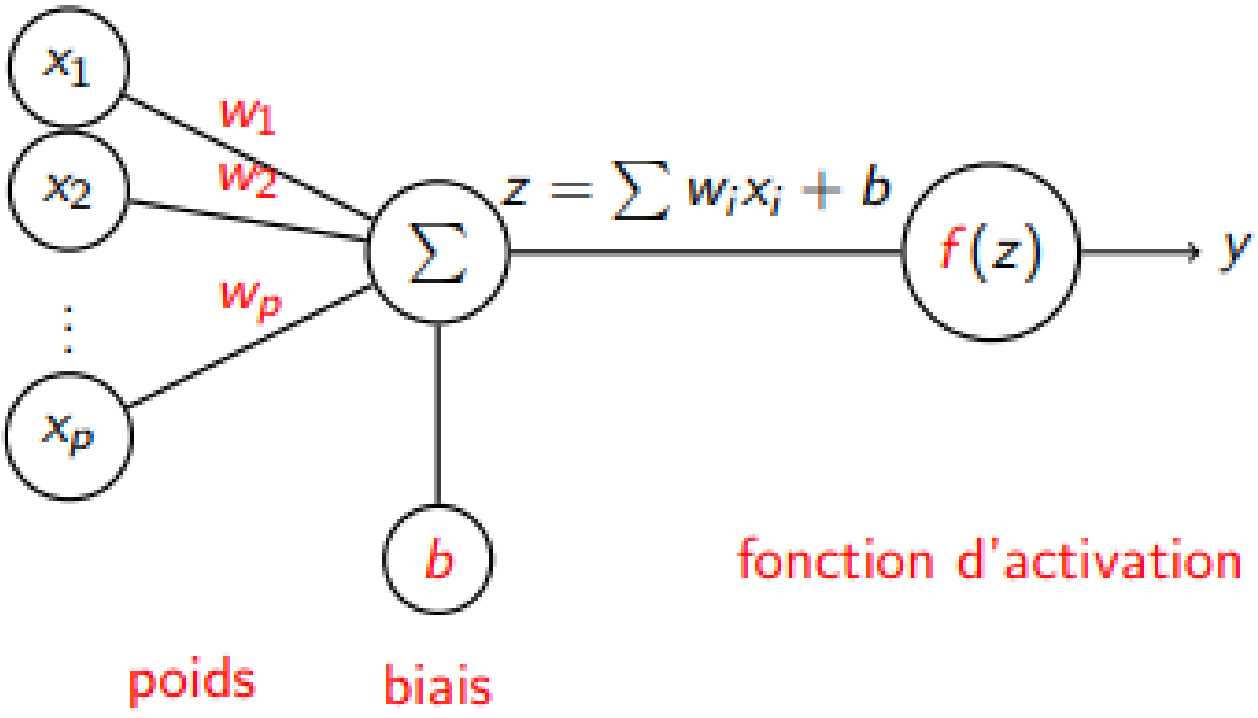
1. Introduction aux Réseaux de Neurones

Un premier modèle : le perceptron



1. Introduction aux Réseaux de Neurones

Un premier modèle : le perceptron



1. Introduction aux Réseaux de Neurones

Exercice :

Considérons un perceptron qui prend deux entrées binaires, x1 et x2, et produit une sortie binaire. Voici les poids et le biais du perceptron : w1 = 1, w2 = −1, b = 0.5. La fonction d'activation utilisée est la fonction échelon (step function) : elle renvoie 1 si la somme pondérée des entrées est supérieure ou égale au seuil, sinon elle renvoie 0.

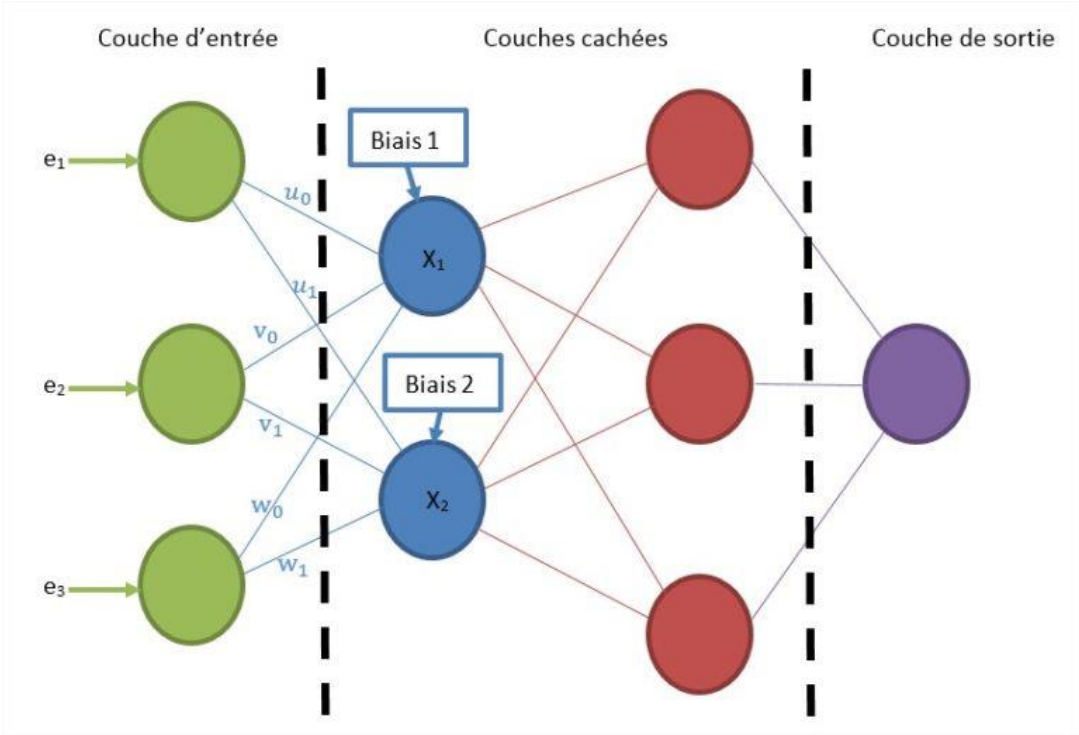
1. Écrivez la formule mathématique pour le calcul de la sortie y du perceptron en fonction des entrées x1, x2, des poids w1, w2 et du biais b.
2. Appliquez la formule pour calculer la sortie y du perceptron lorsque x1 = 1 et x2 = 0.
3. Écrire un code python pour ce perceptron puis les tester avec les valeurs données.

2. Les réseaux de neurones multicouches

Définition

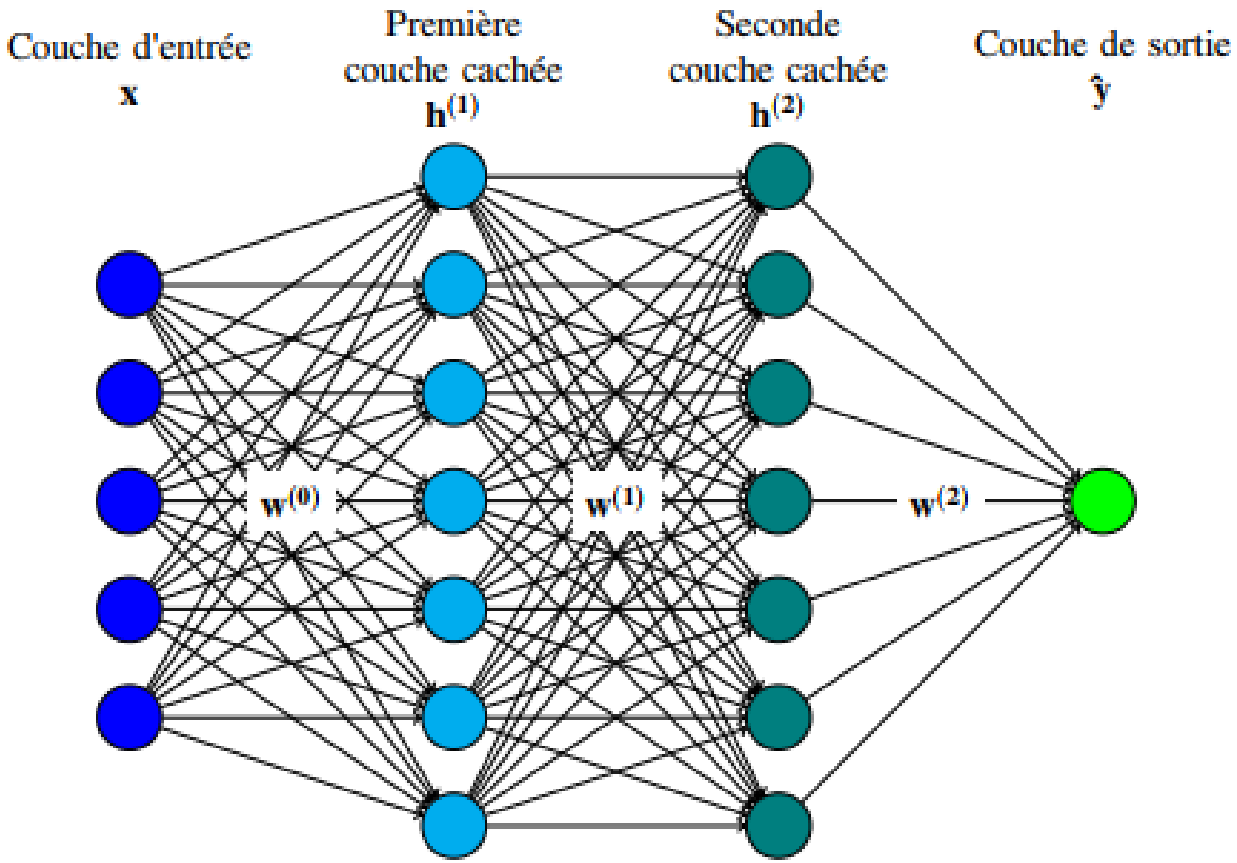
Les réseaux de neurones multicouches perceptron (MLP) sont une extension des perceptrons. Ils sont composés de plusieurs couches de perceptrons, chacune connectée à la couche précédente et à la couche suivante. Les MLP peuvent apprendre des modèles plus complexes que les perceptrons simples, ce qui les rend plus performants pour résoudre des problèmes de classification et de régression plus complexes.

2. Les réseaux de neurones multicouches



1. **Les couches d'entrée** : elles reçoivent les données du problème à résoudre.
2. **Les couches cachées** : elles traitent les données reçues des couches d'entrée et produisent des données qui sont transmises aux couches de sortie.
3. **Les couches de sortie** : elles produisent les résultats du réseau de neurones.

2. Les réseaux de neurones multicouches



2. Les réseaux de neurones multicouches

- Un réseau de neurones artificiels est composé de plusieurs couches de neurones. Chaque couche est connectée à la couche précédente et à la couche suivante

- Chaque couche a sa fonction d'activation et ainsi que la couche de sortie. Elles peuvent etre toutes différentes l'une des autres ou les mêmes.

2. Les réseaux de neurones multicouches

- Pour programmer un réseau de neurones à plusieurs couches, vous pouvez utiliser une bibliothèque comme TensorFlow ou PyTorch en Python.

- Leur rôle principal est de fournir des outils et des abstractions pour faciliter la création, l'entraînement et le déploiement de modèles de deep learning.

2. Les réseaux de neurones multicouches

Exemple :

Écrire le code d'un modèle comportant une couche d'entrée avec une dimension de 784 (qui correspond à un vecteur d'entrée de taille 28x28 pour des images), deux couches cachées avec respectivement 256 et 128 neurones, et une couche de sortie avec 10 neurones (pour la classification de chiffres de 0 à 9). Les fonctions d'activation 'relu' et 'softmax' sont utilisées respectivement pour les couches cachées et de sortie.

2. Les réseaux de neurones multicouches

```
 import tensorflow as tf
# Définition des paramètres du modèle
input_dim = 784  # Dimension de l'entrée (par exemple, pour des images 28x28)
hidden1_dim = 256  # Nombre de neurones dans la première couche cachée
hidden2_dim = 128  # Nombre de neurones dans la deuxième couche cachée
output_dim = 10    # Dimension de la sortie (par exemple, pour la classification des chiffres de 0 à 9)

# Définition du modèle
model = tf.keras.Sequential([
    tf.keras.layers.Dense(hidden1_dim, input_dim=input_dim, activation='relu'), # Première couche cachée
    tf.keras.layers.Dense(hidden2_dim, activation='relu'), # Deuxième couche cachée
    tf.keras.layers.Dense(output_dim, activation='softmax') # Couche de sortie (activation softmax pour la classification multiclasse)
])

# Affichage de l'architecture du modèle
model.summary()
```

3. Le Deep Learning

Définition

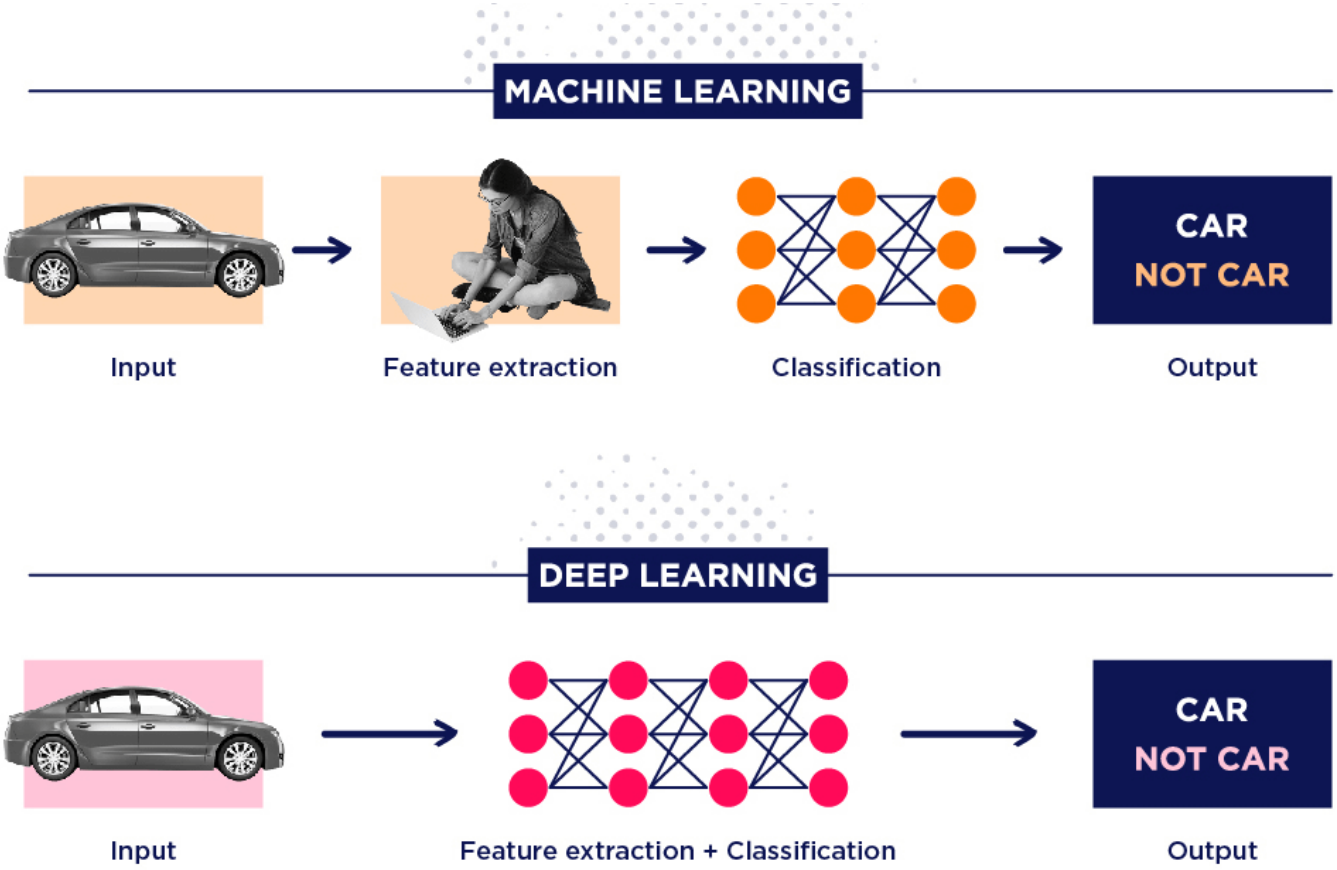
Le **deep learning**, ou **apprentissage profond** en français, est une sous-discipline de l'apprentissage automatique (machine learning) qui se concentre sur l'utilisation de réseaux de neurones artificiels pour résoudre des problèmes complexes. Il est appelé "profond" parce qu'il repose sur des architectures de réseaux de neurones profonds, avec de nombreuses couches intermédiaires (d'où le terme "profond").

4. Fonctionnement du deep learning

- Les réseaux de neurones artificiels sont entraînés sur des jeux de données étiquetés. Cela signifie que chaque exemple de données est associé à une étiquette, qui indique ce que l'exemple représente. Par exemple, un jeu de données pour la reconnaissance d'images peut contenir des images de voitures, de chats et de chiens, étiquetées en conséquence.

- Pendant l'entraînement, le réseau de neurones apprend à prédire l'étiquette correcte pour chaque exemple de données. Il le fait en ajustant les paramètres de ses neurones. Une fois que le réseau est entraîné, il peut être utilisé pour prédire l'étiquette d'un nouvel exemple de données, même si cet exemple n'a jamais été vu auparavant.

5. Machine Learning vs deep learning



5. Application du deep learning

- **Reconnaissance d'images** : le deep learning est utilisé pour
- identifier des objets dans des images, comme des voitures, des visages ou des animaux.
- **Reconnaissance vocale** : le deep learning est utilisé pour transcrire la parole en texte.
- **Traduction automatique** : le deep learning est utilisé pour traduire des textes d'une langue à une autre.
- **Traitement du langage naturel** : le deep learning est utilisé pour comprendre et générer du langage naturel.
- **Intelligence artificielle** : le deep learning est utilisé pour développer des systèmes d'intelligence artificielle capables d'apprendre et de prendre des décisions de manière autonome

6. Optimisation des réseaux des neurones

- La **Rétropropagation** : souvent appelée « backpropagation » en anglais, est l'un des algorithmes d'optimisation les plus utilisés. Il est essentiel dans l'apprentissage des réseaux de neurones, une technique cruciale en IA.
- **Descente de Gradient Stochastique (SGD)** : Cette technique consiste à mettre à jour les paramètres du modèle à chaque exemple d'entraînement. Cela rend l'entraînement plus rapide et plus efficace.
- **Adam (Adaptive Moment Estimation) :** est un algorithme d'optimisation qui adapte les taux d'apprentissage pour chaque paramètre, améliorant ainsi la convergence.
- **RMSProp (Root Mean Square Propagation):** est un autre algorithme d'optimisation qui adapte les taux d'apprentissage en divisant le taux d'apprentissage par une moyenne mobile des carrés des gradients.

10 Présentation des Projets