

Artificial Intelligence, Machine Learning and Deep Learning With Python

Bensearch solutions

contact@bensearch-solutions.com

June 14, 2024



Bensearch
solutions
Global training & education services



Bensearch
solutions
Global training & education services



Contexte actuel :

- *Forte demande en compétences en IA, ML et Deep Learning.*
- *Jeunes diplômés cherchant à se préparer aux opportunités pro*

Objectifs de la formation :

- *Fournir une introduction complète à l'IA, ML et Deep Learning.*
- *Acquérir les compétences fondamentales en Python pour l'IA.*
- *Comprendre les concepts et les algorithmes clés de l'IA et du ML.*
- *Exploiter les bibliothèques populaires: Scikit Learn et TensorFlow*
- *Appliquer les connaissances à des projets pratiques.*

Résultats attendus :

- *Compétences solides en IA, ML et Deep Learning avec Python.*
- *Capacité à résoudre des pbs réels en utilisant des techniques d'IA.*
- *Préparation pour des opportunités professionnelles dans le domaine*

Plan général de la présentation (AI and ML)

- ① Artifical Intelligence and Machine Learning 1
 - Fondamentaux de l'IA et du Machine Learning
 - Technical Approaches of AI in Machines Learning
 - Défis et évaluation des modèles
- ② Artifical Intelligence and Machine Learning 2
 - Explicabilité de l'IA et IML
 - Techniques d'explicabilité XAI et IML
 - Statistiques, causalité, et évaluation des modèles
- ③ Artifical Intelligence and Machine Learning 3
 - Défis de l'IA et application étroite
 - Apprentissage supervisé et non supervisé
 - Explicabilité de l'IA (XAI) et traitement du langage naturel (NLP)
- ④ Artifical Intelligence and Machine Learning 4 : Cas Pratiques
 - Analyse de données avec Power BI
 - Chatbots avec Azure et Google Dialogflow
 - Apprentissage automatique avec Scikit-Learn

Plan général de la présentation (Python)

- ① Python for Machine Learning 1 : Initialization
- ② Python for Machine Learning 2 : Advanced
- ③ Deep Learning and Neural Network

Artificial Intelligence and Machine Learning 1

AI / ML

Introduction à l'IA et au Machine Learning

- Qu'est ce que l'Intelligence Artificielle?
- Histoire de l'Intelligence Artificielle
- Machine Learning
- L'Intelligence Artificielle en robotique
- Introduction au Big Data et son intégration avec l'IA
- Éviter les pièges et travailler avec les données
- Principes du Machine Learning
- Avantages et inconvénients de l'IA et du ML
- Conclusion

Questions clés pour comprendre l'impact quotidien de l'IA



Figure: Reconnaissance faciale

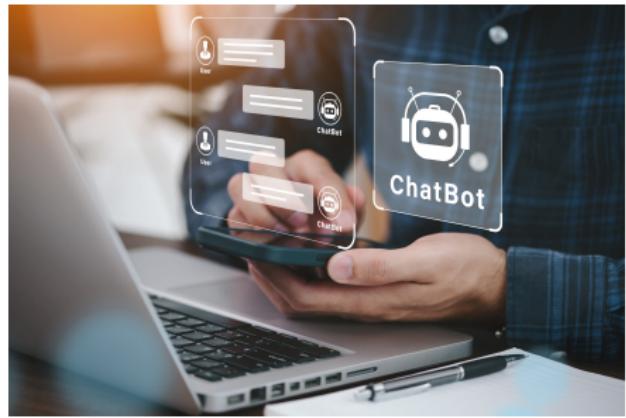


Figure: Application de chatbot

Artificial Intelligence, Machine Learning and Deep Learning With Python

└ Introduction à l'IA et au Machine Learning

└ Qu'est ce que l'IA?

└ Questions clés pour comprendre l'impact

Questions clés pour comprendre l'impact quotidien de l'IA



Figure: Reconnaissance faciale



Figure: Application de chatbot

- **La reconnaissance faciale** utilise des algorithmes d'apprentissage automatique (ML) que nous explorerons dans cette formation pour extraire des caractéristiques faciales uniques, telles que la distance entre les yeux, la forme du nez, etc. Ces caractéristiques sont ensuite comparées à une base de données pour trouver une correspondance.
- **Les chatbots** utilisent des techniques d'IA telles que le traitement du langage naturel (NLP) et l'apprentissage automatique pour comprendre les messages des utilisateurs, générer des réponses appropriées et améliorer leur performance au fil du temps.

Questions clés pour comprendre l'impact quotidien de l'IA



Figure: Classification d'emails



Figure: Suggestion d'amis facebook

Artificial Intelligence, Machine Learning and Deep Learning With Python

└ Introduction à l'IA et au Machine Learning

└ Qu'est ce que l'IA?

└ Questions clés pour comprendre l'impact

Questions clés pour comprendre l'impact quotidien de l'IA

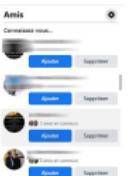


Figure: Suggestion d'amis facebook

- **La classification d'emails** utilise des techniques d'IA, comme l'apprentissage supervisé, pour prédire les catégories des nouveaux messages en se basant sur des ensembles de données étiquetés (Grand ensemble d'emails classifiés comme **INBOX** ou **SPAM**).
- **La suggestion d'amis Facebook** utilise des algorithmes d'apprentissage automatique pour analyser les profils d'utilisateurs, leurs interactions et leurs intérêts communs afin de proposer des suggestions pertinentes de connexions amicales sur la plateforme.

Questions clés pour comprendre l'impact quotidien de l'IA



Figure: Voiture autonome



Figure: Composantes

Comment les voitures autonomes sont-elles capables de détecter et de réagir aux obstacles sur la route de manière autonome?

Artificial Intelligence, Machine Learning and Deep Learning With Python

└ Introduction à l'IA et au Machine Learning

└ Qu'est ce que l'IA?

└ Questions clés pour comprendre l'impact

Questions clés pour comprendre l'impact quotidien de l'IA



Figure: Voiture autonome

Figure: Composantes

Comment les voitures autonomes sont-elles capables de détecter et de réagir aux obstacles sur la route de manière autonome?

Les voitures autonomes utilisent des capteurs tels que des caméras, des lidars et des radars pour détecter les obstacles sur la route.

Les données fournis par ces capteurs sont traitées par des algorithmes de ML qui reconnaissent les objets et prennent des décisions en temps réel, comme le freinage ou l'évitement d'obstacles. L'IA permet aux voitures autonomes d'apprendre et de s'améliorer avec l'expérience pour une conduite plus sûre et efficace.

Questions clés pour comprendre l'impact quotidien de l'IA



Figure: Assistant personnel intelligent de MS



Figure: Applications de reconnaissance vocale

The screenshot shows a Google search results page for the query "creation site immobilier". The results are personalized for "agence immobilière". The first result is from "Creation Site Immobilier" and includes a snippet: "Créer son site immobilière en ligne avec un budget de 25€". A red arrow points from the text "Personnalisation des résultats" to this snippet. Other results include "Creation Site Immobilier: N°1 sur la création de site internet..." and "Creation Site Immobilier pour agence immobilière - Immence".

Figure: Résultat de recherche personnalisée



Figure: Application de traduction instantanée

Artificial Intelligence, Machine Learning and Deep Learning With Python

└ Introduction à l'IA et au Machine Learning

└ Qu'est ce que l'IA?

└ Questions clés pour comprendre l'impact

Questions clés pour comprendre l'impact quotidien de l'IA



Figure: Assistant personnel intelligent de MS



Figure: Résultat de recherche personnalisée



Figure: Applications de reconnaissance vocale



Figure: Application de traduction instantanée

- **L'assistant personnel de Microsoft utilise le NLP et le ML, pour comprendre les requêtes des utilisateurs et fournir des réponses et des actions pertinentes. Il s'améliore également en apprenant des interactions passées.**
- **Les applications de reconnaissance vocale utilisent le ML pour convertir la parole en texte, utilisés dans des applications, comme les assistants vocaux.**
- **Les applications de traduction instantanée utilisent des techniques d'IA, telles que les réseaux de neurones, pour traduire automatiquement du texte ou de la parole d'une langue à une autre en temps réel.**
- **Les résultats de recherche personnalisés utilisent le ML pour analyser les préférences et le comportement de l'utilisateur, afin de fournir des résultats de recherche plus pertinents et personnalisés en fonction de ses intérêts et de son historique.**

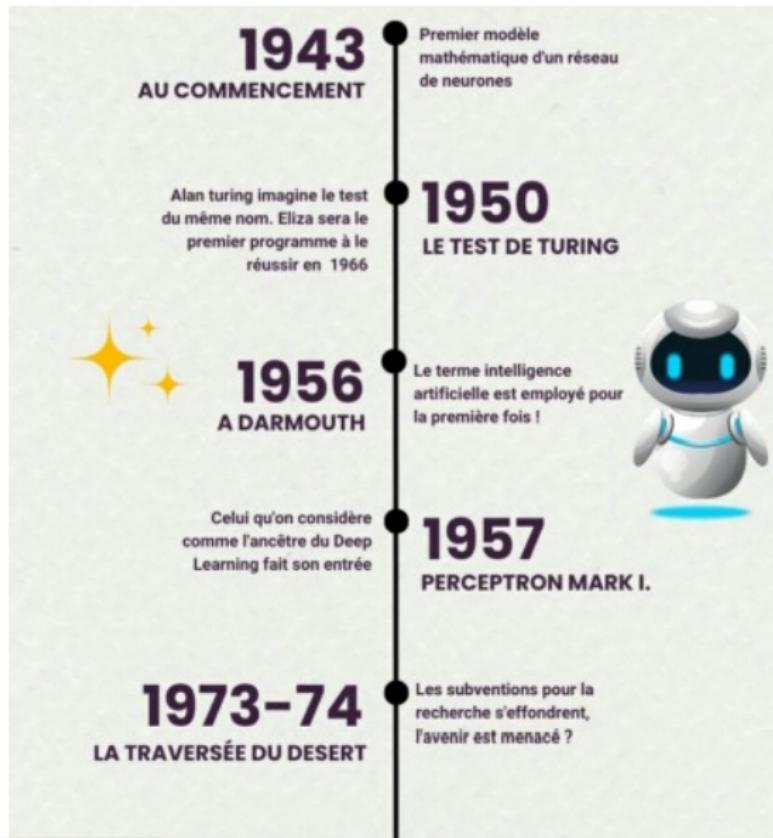
Comment définiriez-vous l'IA en vos propres termes ?

INTELLIGENCE ARTIFICIELLE : DÉFINITION

Définition de l'IA selon quelques auteurs

- **John McCarthy (1956)** : "L'IA est la science et l'ingénierie de la création de machines intelligentes, qui ont la capacité de réaliser des tâches qui nécessitent normalement l'intelligence humaine."
- **Stuart Russell et Peter Norvig (2009)** : "L'IA est le domaine de l'informatique qui traite de la création et de l'étude de machines qui peuvent effectuer des tâches qui, si elles étaient accomplies par des êtres humains, nécessiteraient de l'intelligence."
- **Marvin Minsky (1968)** : "L'IA est la construction de programmes informatiques qui peuvent accomplir des tâches qui, lorsqu'elles sont accomplies par des personnes, demandent de l'intelligence."

Les dates clés de l'Intelligence Artificielle



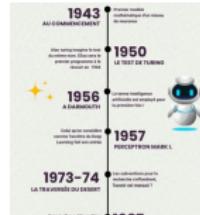
Artificial Intelligence, Machine Learning and Deep Learning With Python

└ Introduction à l'IA et au Machine Learning

└ Histoire de l'IA

└ Les dates clés de l'Intelligence Artificielle

Les dates clés de l'Intelligence Artificielle



- **1965** : Joseph Weizenbaum crée ELIZA, un programme de traitement du langage naturel qui simule une conversation psychothérapeutique.
- **1969** : Marvin Minsky et Seymour Papert publient le livre "Perceptrons", qui met en évidence les limites du perceptron et freine temporairement le développement des réseaux de neurones.
- **1973** : James Lighthill publie le rapport Lighthill, critiquant les progrès insuffisants de l'IA et entraînant une période de désintérêt et de financement réduit pour le domaine.
- **1974** : Ted Shortliffe développe MYCIN, un système expert utilisé pour le diagnostic et le traitement des infections bactériennes.

Les dates clés de l'Intelligence Artificielle



Artificial Intelligence, Machine Learning and Deep Learning With Python

└ Introduction à l'IA et au Machine Learning

└ Histoire de l'IA

└ Les dates clés de l'Intelligence Artificielle

Les dates clés de l'Intelligence Artificielle



- **2014** : Les réseaux de neurones récurrents (RNN) deviennent populaires pour le traitement des séquences, ouvrant la voie à des applications telles que la traduction automatique et la génération de texte.
- **2018** : GPT-2 (Generative Pre-trained Transformer 2), un modèle basé sur l'apprentissage non supervisé, fait sensation en générant du texte de manière presque indiscernable de l'écriture humaine.
- **2020** : L'IA de génération de texte, GPT-3, développée par OpenAI, impressionne par sa capacité à produire du texte créatif et convaincant dans une variété de domaines.
- **2022** : OpenAI lance DALL·E, un modèle de génération d'images capable de créer des illustrations à partir de descriptions textuelles, repoussant les limites de la créativité des machines.

Exercice sur l'histoire de l'IA



Pourquoi c'est à nos jours que l'IA fait la une de l'actualité, alors que les fondements mathématiques des réseaux de neurones remontent à 1943 ?

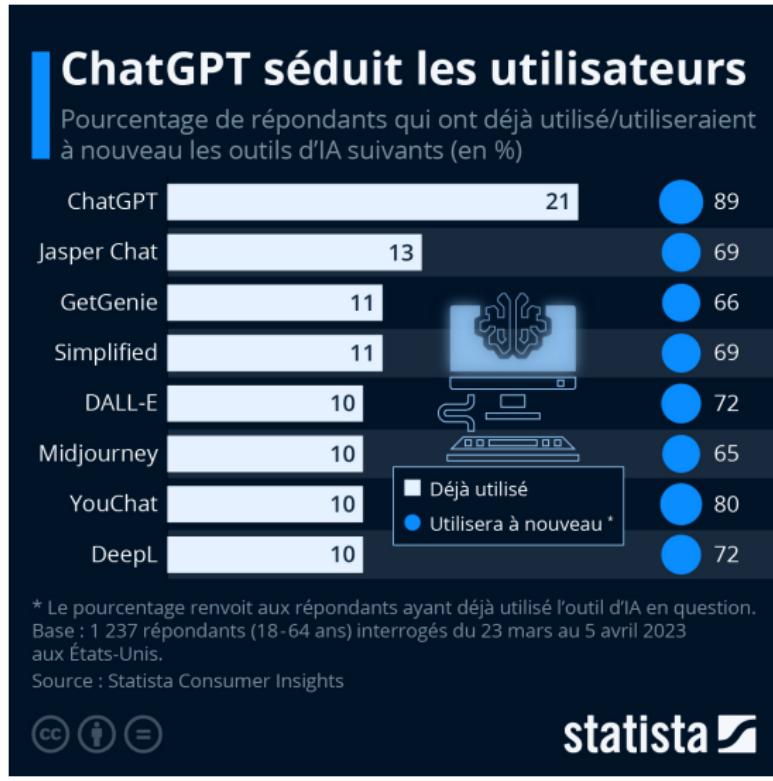
- └ Introduction à l'IA et au Machine Learning
 - └ Histoire de l'IA
 - └ Exercice sur l'histoire de l'IA

Pourquoi c'est à nos jours que l'IA fait la une de l'actualité, alors que les fondements mathématiques des réseaux de neurones remontent à 1943 ?

L'IA fait aujourd'hui la une de l'actualité pour plusieurs raisons clés:

- **Progrès techniques** : *Les récentes avancées en matière de puissance de calcul, de stockage de données et d'algorithmes ont permis de développer des systèmes d'IA beaucoup plus performants.*
- **Données massives** : *L'explosion du volume de données disponibles, grâce à l'Internet et aux appareils connectés, alimente l'entraînement des modèles d'IA.*
- **Applications concrètes** : *L'IA peut maintenant être déployée dans de nombreux domaines (santé, transport, finance, etc.), démontrant son utilité pratique.*
- **Prise de conscience publique** : *Les médias et le grand public s'intéressent davantage à l'IA et à son impact potentiel, alimentant le débat.*

Les Intelligences Artificielles populaires



Artificial Intelligence, Machine Learning and Deep Learning With Python

└ Introduction à l'IA et au Machine Learning

└ Histoire de l'IA

└ Les Intelligences Artificielles populaires

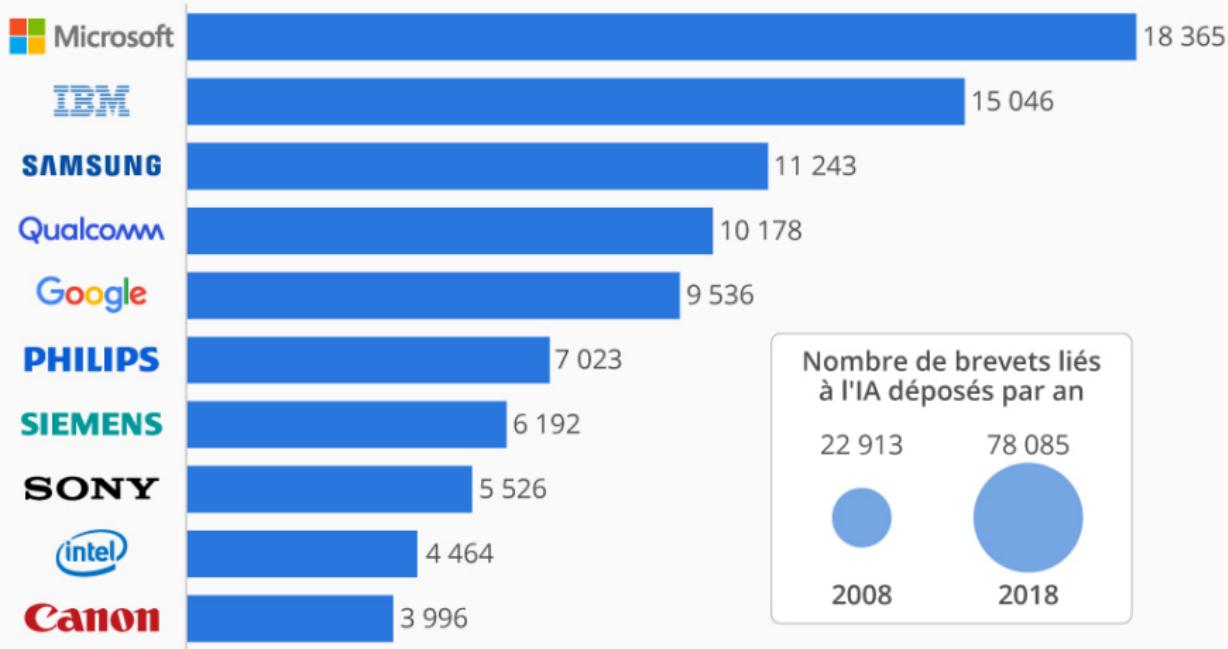


- **TCHATGPT** : *TCHATGPT est un modèle de langage développé par OpenAI. Il est conçu pour répondre de manière interactive aux questions et aux conversations.*
- **Jasper Chat** : *Jasper Chat est un assistant vocal développé par Jasper Technologies. Il est utilisé pour la réservation de rendez-vous, la gestion des tâches,...*
- **Getgenie** : *Getgenie est un assistant vocal développé par Genie AI. Il est conçu pour aider les utilisateurs à gérer leurs e-mails en utilisant des commandes vocales, notamment la recherche, le tri et la réponse aux messages.*
- **Simplified** : *Simplified est une IA développée par Simplified AI. Elle est utilisée pour automatiser les tâches de service client, en fournissant des réponses automatisées aux requêtes des clients et en les redirigeant vers les ressources appropriées.*

La course vers les brevets

La course aux brevets dans l'intelligence artificielle

Entreprises possédant le plus de brevets liés à l'intelligence artificielle *



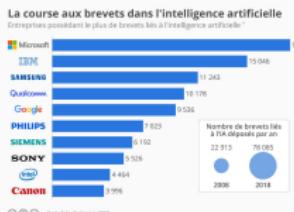
Artificial Intelligence, Machine Learning and Deep Learning With Python

└ Introduction à l'IA et au Machine Learning

└ Histoire de l'IA

└ La course vers les brevets

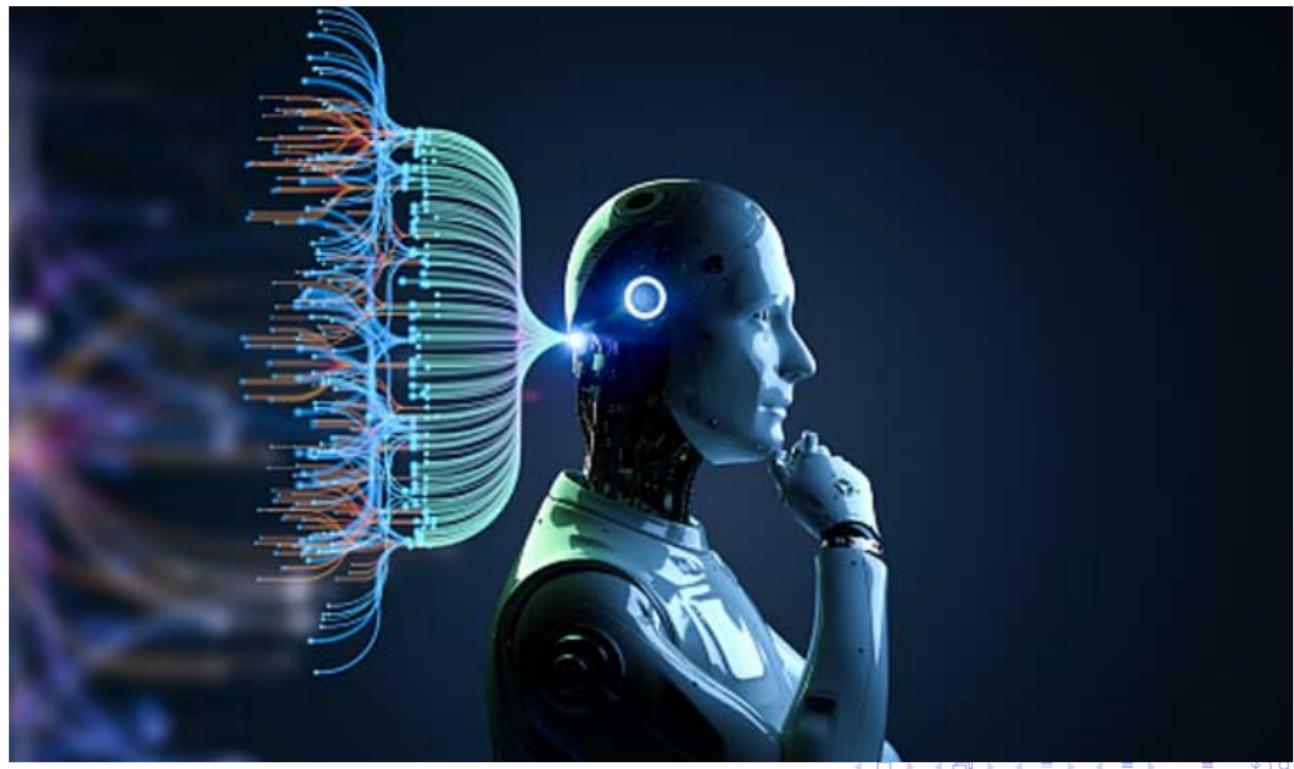
La course vers les brevets



- La course aux brevets en intelligence artificielle (IA) reflète l'importance stratégique de cette technologie.
- Les principaux acteurs (États-Unis, Chine, Europe, etc.) investissent massivement dans la R&D en IA, cherchant à protéger leurs innovations par des brevets.
- Cette course aux brevets soulève des questions éthiques et de politique publique sur l'accessibilité et l'utilisation responsable de l'IA.

Future de l'IA

Que penser de son impact future dans notre société?



Artificial Intelligence, Machine Learning and Deep Learning With Python

└ Introduction à l'IA et au Machine Learning

└ Histoire de l'IA

└ Future de l'IA

Future de l'IA
Que penser de son impact future dans notre société?



- *Amélioration des capacités de traitement du langage naturel, ce qui permettra une communication plus fluide entre les humains et les machines, facilitant ainsi l'interaction avec les systèmes d'IA dans divers domaines tels que les assistants virtuels et les traducteurs automatiques.*
- *Avancées dans la compréhension et la génération de contenu multimédia: création artistique, divertissement, réalité virtuelle.*
- *Améliorations significatives d'efficacité, de précision et de prise de décision.*
- *Systèmes plus autonomes et capables de s'adapter à des situations changeantes.*
- *Systèmes d'IA respectant les normes éthiques et les valeurs humaines, confiance et acceptation sociale de ces technologies.*

Qu'est ce que le Machine Learning

www.sales-hacking.com

Intelligence Artificielle

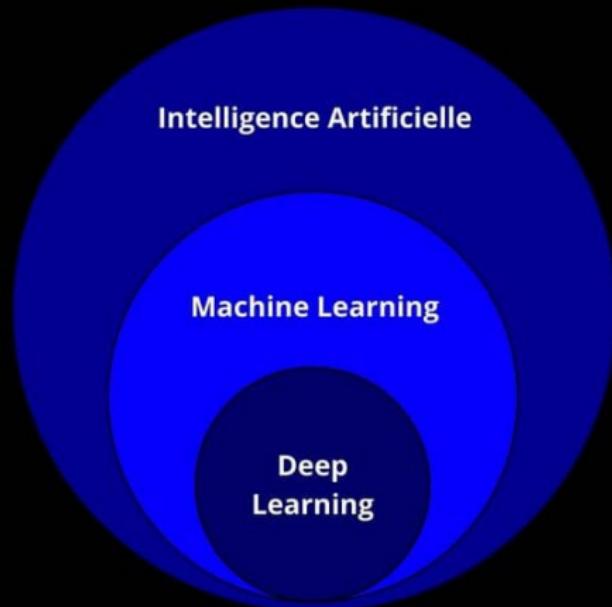
Une science qui vise à faire en sorte que les machines pensent et agissent comme des humains.

Machine Learning

L'objectif est de permettre aux ordinateurs d'effectuer des tâches sans programmation explicite.

Deep Learning

Un sous-ensemble du Machine Learning basé sur les réseaux neuronaux.



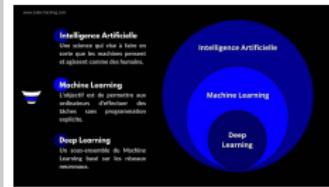
Artificial Intelligence, Machine Learning and Deep Learning With Python

└ Introduction à l'IA et au Machine Learning

└ Qu'est ce que le Machine Learning?

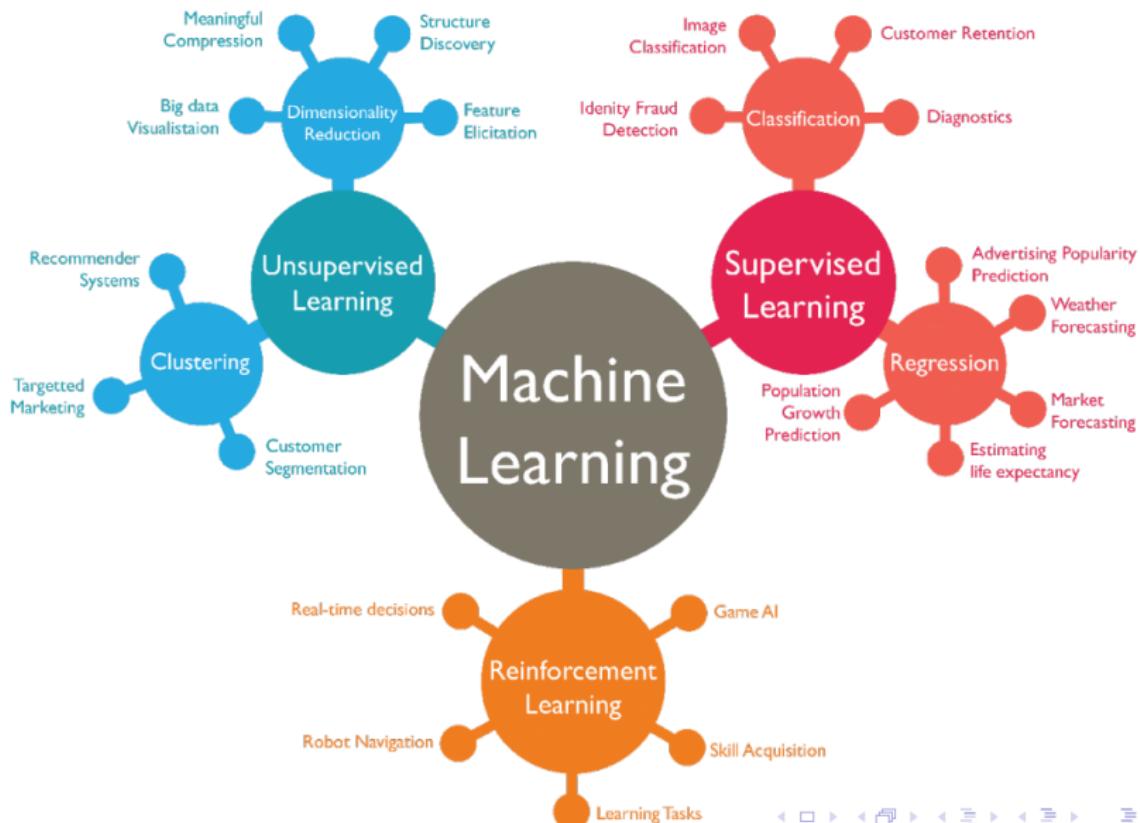
└ Qu'est ce que le Machine Learning

Qu'est ce que le Machine Learning



- *Le Machine Learning peut être défini comme étant une technologie d'IA permettant aux machines d'apprendre sans avoir été préalablement programmées spécifiquement à cet effet.*
- *Ledit apprentissage est fait sur les données, à titre d'illustration une banque disposant d'informations pertinentes sur des personnes ayant emprunté de l'argent autrefois est en mesure d'entraîner une machine à détecter à priori si quelqu'un est un "**GOOD or BAD credit risk**"*
- *C'est donc tout processus par lequel un système améliore ses performances à partir de l'expérience issue des données.*

Approches techniques de l'IA utilisées dans le ML



Artificial Intelligence, Machine Learning and Deep Learning With Python

Introduction à l'IA et au Machine Learning

Approches techniques de l'IA utilisées dans le ML

Approches techniques de l'IA utilisées dans le ML



On distingue trois grandes catégories de ML

• Supervised Learning (Apprentissage supervisé)

- Entraînement à partir de données étiquetées*
- La régression pour prédiction de valeurs continues et la classification de catégories*
- Généralisation des schémas pour prédire de nouvelles données non étiquetées*

• Unsupervised Learning (Apprentissage non supervisé)

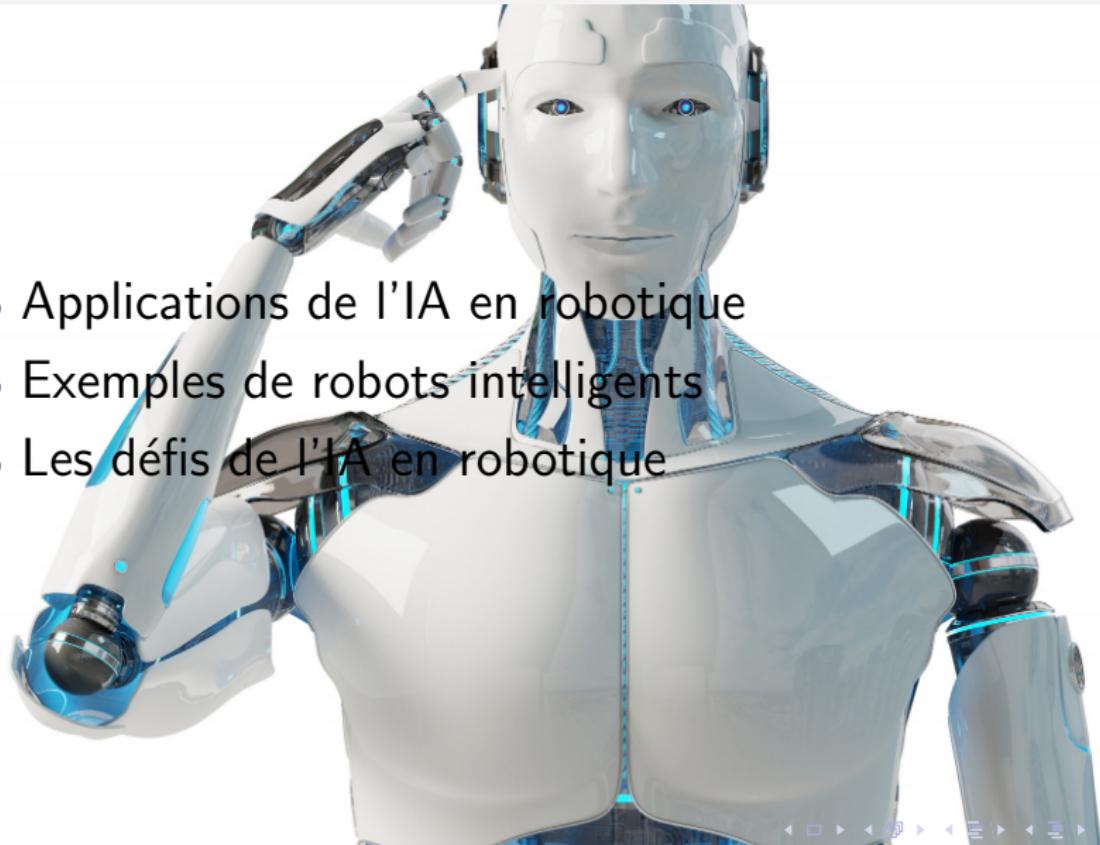
- Découverte de structures dans les données non étiquetées*
- Réduction de la dimensionnalité (visualisation, analyse)*
- Clustering pour regrouper les exemples similaires*

• Reinforcement Learning (Apprentissage par renforcement)

- Interaction avec un environnement dynamique*
- Actions suivies de récompenses ou de pénalités*
- Objectif d'optimiser les actions pour maximiser les récompenses*

L'IA en robotique

- Applications de l'IA en robotique
- Exemples de robots intelligents
- Les défis de l'IA en robotique



Applications de l'IA en robotique

- Industrie : robot de gestion de chaîne d'assemblage...
- Armée : drone, robot-espion, robot-mule...
- Sécurité : vidéosurveillance...
- Santé : échographie, chirurgie assistée...
- Aérospatial : robot explorateur de la NASA...
- Transport : voiture autonome...
- Usage domestique : robot aspirateur, robot tondeuse...
- Accompagnement : jouet automatisé, robot humanoïde...
- Informatique : chatbot, assistant vocal...

Exemples de Robots intelligents

- Atlas : Robot humanoïde développé par Boston Dynamics, connu pour sa mobilité et sa dextérité avancées.
- Asimo : Robot humanoïde développé par Honda, célèbre pour sa marche bipède, sa capacité à monter les escaliers et ses mouvements semblables à ceux d'un humain.
- iCub : Robot humanoïde à source ouverte conçu pour étudier la cognition et le développement humains.
- Nao : Petit robot humanoïde créé par SoftBank Robotics, utilisé dans l'éducation, la recherche et le divertissement, capable d'interagir avec les humains par la parole et les gestes.

Atlas : Un robot humanoïde aux capacités impressionnantes

Le Big Data et son intégration avec l'IA

- Les Big Data sont les traces numériques (données) qui se génèrent dans l'ensemble de l'écosystème numérique.
- Les Big Data sont des actifs d'information à haut volume, haute vélocité et haute variété.
- Les principales caractéristiques des Big Data sont les quatre V

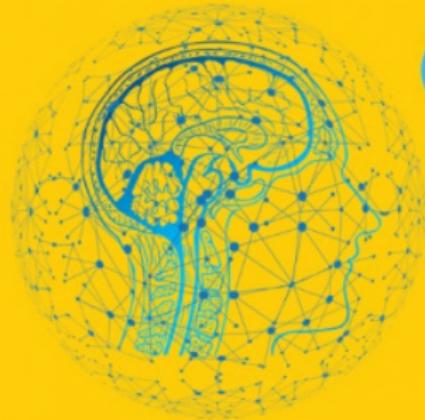


Impact des Big Data

- Les IoT génèrent continuellement d'énormes volumes de données.
- L'analyse des Big Data aide les entreprises à obtenir des informations à partir des données collectées par les appareils IoT.



Comment le Big Data alimente l'IA?

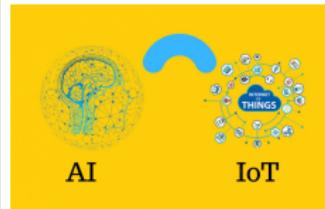


AI



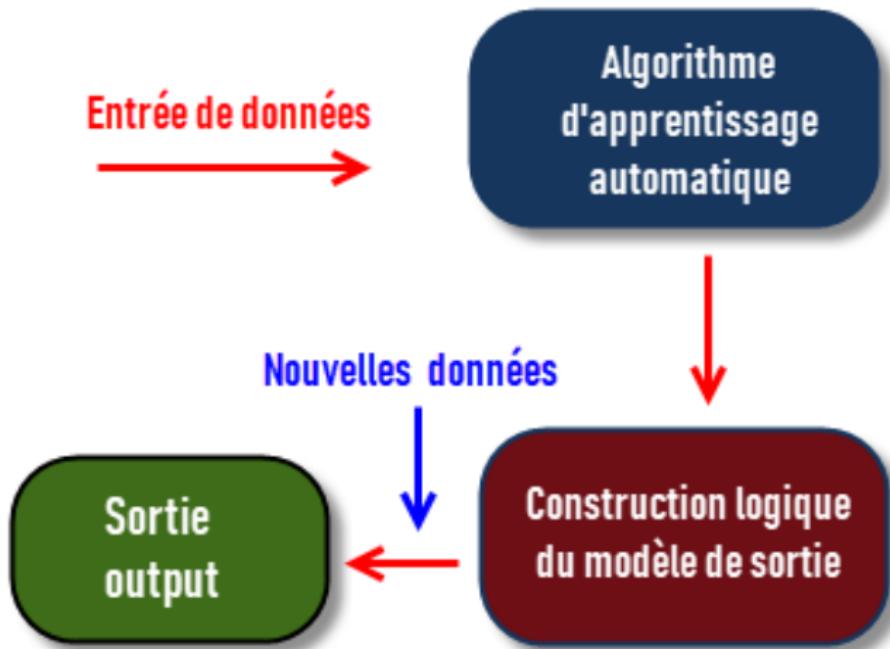
IoT

- └ Le Big Data et son intégration avec l'IA
 - └ Impact des Big Data
 - └ Comment le Big Data alimente l'IA?



- **Entraînement des modèles :**
 - *Fournit des ensembles de données massifs.*
 - *Permet aux modèles d'apprendre à reconnaître les relations.*
- **Amélioration de la précision :**
 - *Le plus de données améliore la précision des modèles.*
 - *Diversité des sources.*
- **Détection de tendances et de modèles cachés :**
 - *Analyse de grandes quantités de données.*
 - *Identifie des corrélations et des insights précieux.*
- **Personnalisation et recommandations :**
 - *Analyse des préférences et des comportements des utilisateurs.*
 - *Fournit des recommandations personnalisées.*
- **Prise de décision basée sur les données :** *informations objectives et décisions éclairées*

Principe du Machine Learning



Les influences majeures du Machine Learning

Le ML est motivé et soutenu par :

- ① La théorie formelle de la statistique
- ② L'accélération du développement des ordinateurs
- ③ Le défi, dans de nombreux domaines, de corpus de données toujours plus grands
- ④ L'accent mis sur la quantification dans une variété toujours plus large de disciplines

Artificial Intelligence, Machine Learning and Deep Learning With Python

└ Le Big Data et son intégration avec l'IA

└ Les influences

└ Les influences majeures du Machine Learning

Les influences majeures du Machine Learning

Le ML est motivé et ou soutenu par :

- ─ La théorie formelle de la statistique
- ─ L'accélération du développement des ordinateurs
- ─ Le défi, dans de nombreux domaines, de corpus de données toujours plus grands
- ─ L'accent mis sur la quantification dans une variété toujours plus large de disciplines

Le Machine learning ou apprentissage statistique est un champ d'étude de l'IA qui se fonde sur des approches statistiques pour donner aux ordinateurs la capacité d'apprendre à partir de données.

Comment résoudre un problème avec une approche Data?

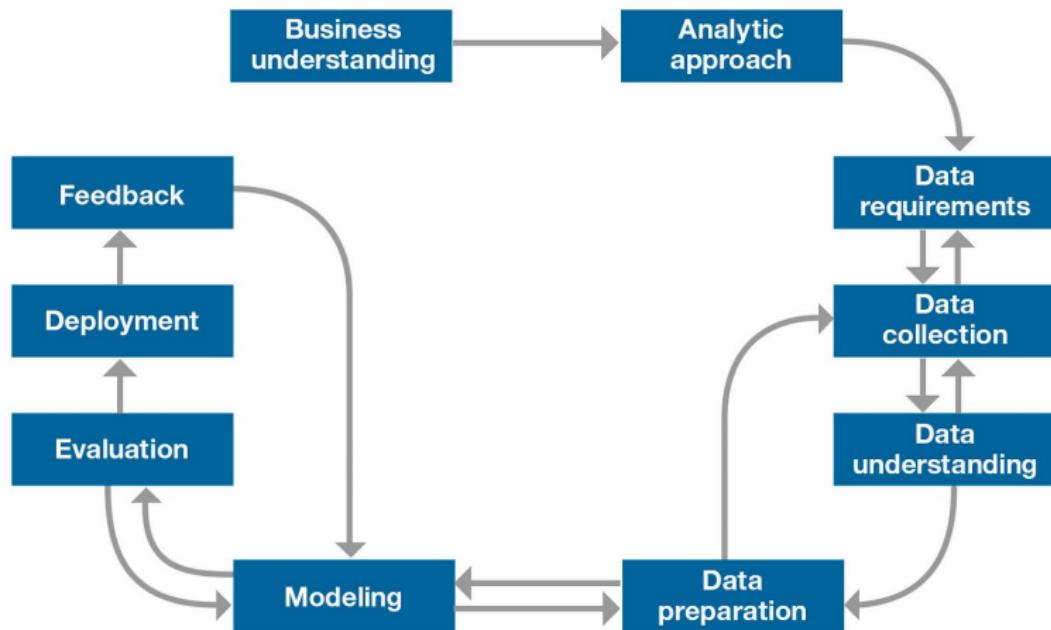


Figure: Data science live cycle

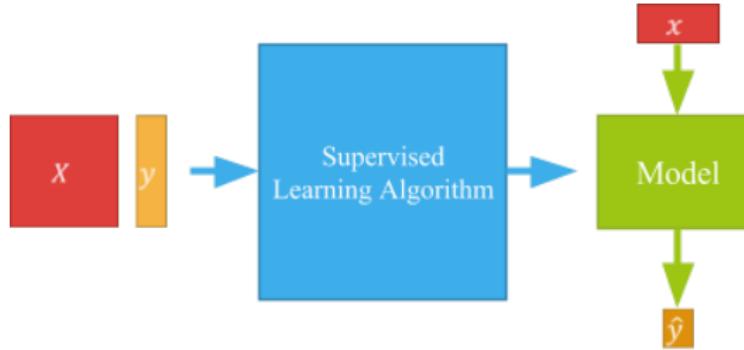
Apprentissage Supervisé et non-supervisé

- Apprentissage supervisé

- Objectif : apprendre une fonction f prédisant une variable Y à partir de features X .
- Données : ensemble d'apprentissage (X_i, Y_i)

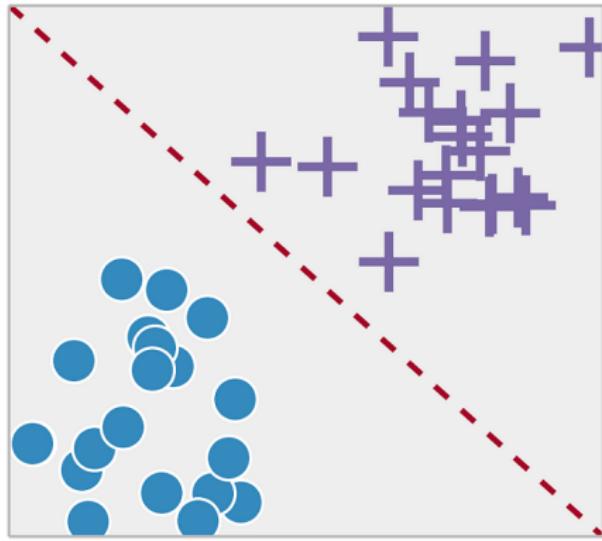
- Apprentissage non-supervisé

- Objectif: découvrir une structure au sein d'un ensemble d'individus (X_i)
- Data : Learning set (X_i)

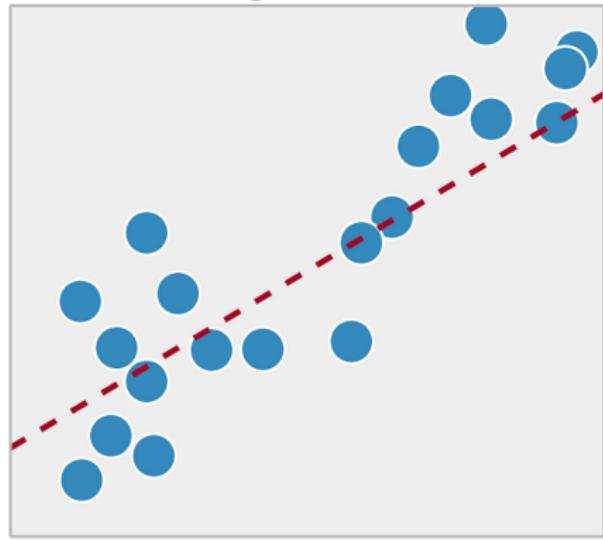


Apprentissage supervisé

Classification



Regression



Quiz: Classification ou Regression?

- 1- Vous avez un large inventaire d'articles identiques. Vous voulez prédire combien de ces articles se vendront au cours des 3 prochains mois.
- 2- Vous devez examiner les comptes de vos clients et décider pour chacun d'entre eux s'ils ont été piratés ou compromis.
- 3- Prédiction du churn d'une entreprise
- 4- Prédire si un prospect deviendra client
- 5- Prédire le chiffre d'affaire d'une entreprise dans 10 ans

Artificial Intelligence, Machine Learning and Deep Learning With Python

- └ Le Big Data et son intégration avec l'IA
 - └ Types de Machine Learning - Quiz
 - └ Quiz: Classification ou Regression?

Quiz: Classification ou Regression?

- 1- Vous avez un large inventaire d'articles identiques. Vous voulez prédire combien de ces articles se vendront au cours des 3 prochains mois.
- 2- Vous devez examiner les comptes de vos clients et décider pour chacun d'entre eux s'ils ont été piratés ou compromis.
- 3- Prédiction du churn d'une entreprise
- 4- Prédire si un prospect deviendra client
- 5- Prédire le chiffre d'affaire d'une entreprise dans 10 ans

- 1- *Il s'agit d'un problème de régression, car l'objectif est de prédire une valeur numérique continue dans le futur.*
- 2- *Détection de piratage ou de compromission des comptes : C'est un problème de classification. La réponse est binaire, c'est-à-dire qu'elle peut être catégorisée en deux classes distinctes.*
- 3- *Prédiction du churn (taux de désabonnement) d'une entreprise : C'est un problème de classification, car l'objectif est de prédire si un client donnera son avis positif ou négatif*
- 4- *Prédiction de conversion d'un prospect en client : C'est un problème de classification, car il s'agit de prédire si un prospect deviendra un client ou non. La réponse est binaire: conversion ou non conversion.*
- 5- *Prédiction du chiffre d'affaires d'une entreprise dans 10 ans : Il s'agit d'un problème de régression, car l'objectif est de prédire une valeur numérique continue pour une période de temps spécifique.*

Exercise : Study case

Given a case study: pricing apartments based on a real estate website.

- House descriptions with their price
- Predicting house prices from their description
- Use case: finding houses that are cheap compared to market value

Question 1

What kind of problem is it?

- a) A supervised problem
- b) An unsupervised problem
- c) A classification problem
- d) A regression problem

Select all answers that apply

Artificial Intelligence, Machine Learning and Deep Learning With Python

└ Le Big Data et son intégration avec l'IA
 └ Types de Machine Learning - Quiz
 └ Question 1

Question 1

What kind of problem is it?

- a) A supervised problem
- b) An unsupervised problem
- c) A classification problem
- d) A regression problem

Select all answers that apply

- a) *It's a supervised learning problem because we have houses description and for each of them we have the corresponding price.*
- d) *It's a regression problem because the value that we're trying to predict (price) is a continue value.*

Question 2

What are the features?

- a) The number of rooms might be a feature
- b) The post code of the house might be a feature
- c) The price of the house might be a feature

Select all answers that apply

Artificial Intelligence, Machine Learning and Deep Learning With Python

- └ Le Big Data et son intégration avec l'IA
 - └ Types de Machine Learning - Quiz
 - └ Question 2

Question 2

What are the features?

- a) The number of rooms might be a feature
- b) The post code of the house might be a feature
- c) The price of the house might be a feature

Select all answers that apply

- a) *To know the features, ask you the question "What is pertinent to explain the house price?". In this case it's the number of rooms.*
- b) *The post code of the house is also important because it provides geographic informations.*

Question 3

What is the target variable?

- a) The full text description is the target
- b) The price of the house is the target
- c) Only house descriptions with no price mentioned are the target

Select a single answer

Artificial Intelligence, Machine Learning and Deep Learning With Python

- └ Le Big Data et son intégration avec l'IA
 - └ Types de Machine Learning - Quiz
 - └ Question 3

Question 3

What is the target variable?

- a) The full text description is the target
- b) The price of the house is the target
- c) Only house descriptions with no price mentioned are the target

Select a single answer

- b) *The price of the house is the target. It's the variable that we're trying to predict.*
- c) *House descriptions with no price mentioned are the set of independant variable (features) generally assign in a matrix X.*

Question 4

What is a record (a sample, instance)? (observation)

- a) Each house description is a record
- b) Each house price is a record
- c) Each kind of description (e.g., house size) is a record

Select a single answer

Artificial Intelligence, Machine Learning and Deep Learning With Python

- └ Le Big Data et son intégration avec l'IA
 - └ Types de Machine Learning - Quiz
 - └ Question 4

a) *Each house description is a record, only.*

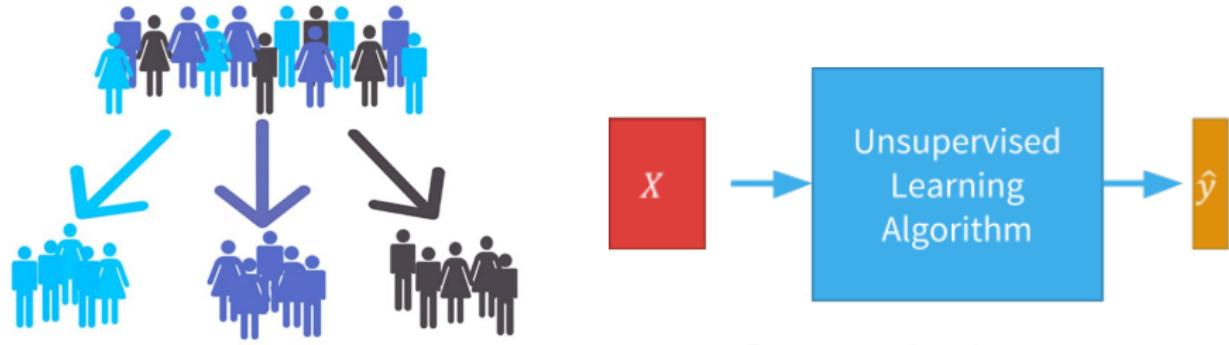
Question 4

What is a record (a sample, instance)? (observation)

- a) Each house description is a record
- b) Each house price is a record
- c) Each kind of description (e.g., house size) is a record

Select a single answer

Apprentissage non-supervisé (clustering)



Principe du clustering

3 Clusters constitués à partir d'un dataset hétérogène

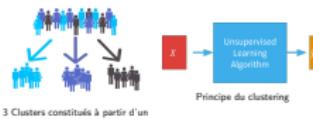
Artificial Intelligence, Machine Learning and Deep Learning With Python

Le Big Data et son intégration avec l'IA

Types de Machine Learning - Quiz

Apprentissage non-supervisé (clustering)

Apprentissage non-supervisé (clustering)



3 Clusters constitués à partir d'un dataset hétérogène

Principe du clustering

- *Le clustering est une technique d'apprentissage non supervisé qui regroupe des données similaires en ensembles distincts appelés clusters. Il aide à découvrir des structures et des relations inhérentes aux données sans avoir d'étiquettes prédéfinies.*
- *Imaginons que nous ayons un ensemble de données contenant des informations sur des clients d'un site de commerce électronique, comme leur âge, leur revenu, et leurs habitudes d'achat. Nous pourrions utiliser le clustering pour regrouper ces clients en segments homogènes en fonction de leurs caractéristiques similaires. Nous pourrions obtenir différents clusters représentant des profils de clients distincts. Par exemple, nous pourrions identifier un cluster de jeunes consommateurs à faible revenu, un autre cluster de clients aisés et fréquents, et un troisième cluster de clients plus âgés et prudents.*

Reduction de la dimension



Figure: Recommendation

Artificial Intelligence, Machine Learning and Deep Learning With Python

- └ Le Big Data et son intégration avec l'IA
 - └ Types de Machine Learning - Quiz
 - └ Reduction de la dimension

Reduction de la dimension



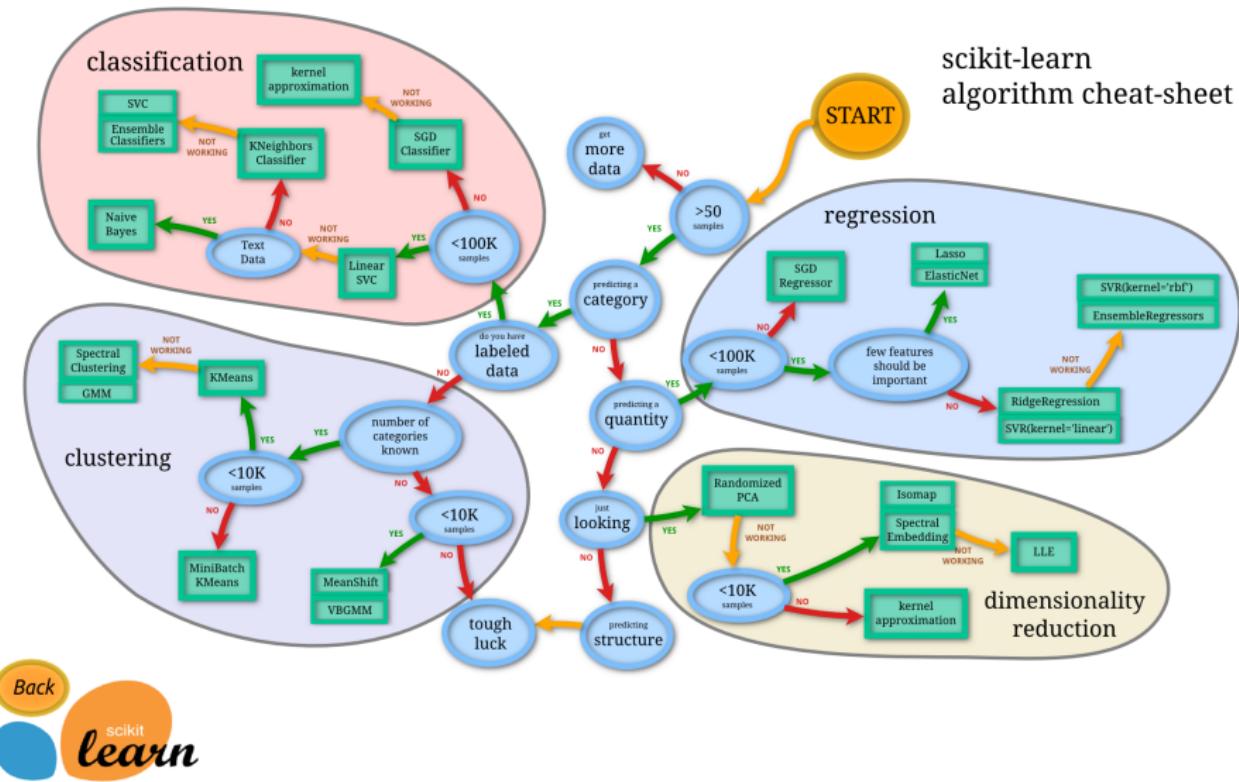
Figure: Recommendation

- *Les plateformes comme Netflix utilise la réduction de la dimensionnalité, notamment la factorisation matricielle, pour recommander des films en se basant sur ceux que vous avez déjà visionnés.*
- *Cette technique permet de représenter les films et les utilisateurs dans un espace de dimensions réduites, où les similarités peuvent être calculées plus efficacement. En utilisant ces représentations réduites, Netflix peut recommander des films similaires à ceux que vous avez aimés.*

Réduction de la Dimensionnalité

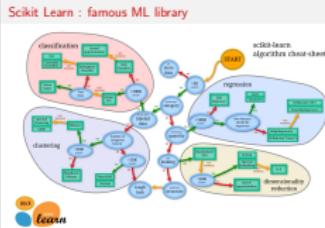
- Technique utilisée pour réduire le nombre de features d'un dataset.
- Objectifs de la réduction de la dimensionnalité :
 - Réduire la complexité du modèle et le temps de calcul.
 - Éliminer les redondances et le bruit dans les données.
 - Visualiser les données dans un espace de dimension inférieure.
- Méthodes courantes de réduction de la dimensionnalité :
 - Analyse en composantes principales (PCA) : transforme les variables d'origine en un nouvel ensemble de variables non corrélées appelées CP.
 - Sélection de caractéristiques : sélectionne un sous-ensemble de caractéristiques les plus informatives.
 - Manifold Learning : trouve des représentations non linéaires des données dans un espace de dimension inférieure.

Scikit Learn : famous ML library



Artificial Intelligence, Machine Learning and Deep Learning With Python

- └ Le Big Data et son intégration avec l'IA
 - └ Types de Machine Learning - Quiz
 - └ Scikit Learn : famous ML library



- *Scikit-learn : bibliothèque open-source pour l'apprentissage automatique en Python.*
- *Large gamme d'outils et d'algorithmes pour la classification, la régression, le clustering, etc.*
- *Prise en charge des tâches de prétraitement de données, de sélection de modèles, etc.*
- *Régression linéaire : modélisation des relations linéaires entre les variables.*
- *SVM (Support Vector Machines) : classification et régression basées sur la recherche de la meilleure séparation entre les classes.*

Travailler avec les données et éviter les pièges

Collecte et préparation des données

- ① Collecte des données de qualité et représentatives
 - Déterminez les informations que vous voulez collecter
 - Définissez la méthode de collecte des données
- ② Exploration des données (Data mining)
 - Compréhension du problème (activité, objectif)
 - Compréhension des données
- ③ Nettoyage des données
 - Gestion des erreurs de saisie
 - Gestion des doublons
 - Gestion des valeurs manquantes et des données aberrantes
- ④ Les bonnes pratiques de normalisation et transformation des données

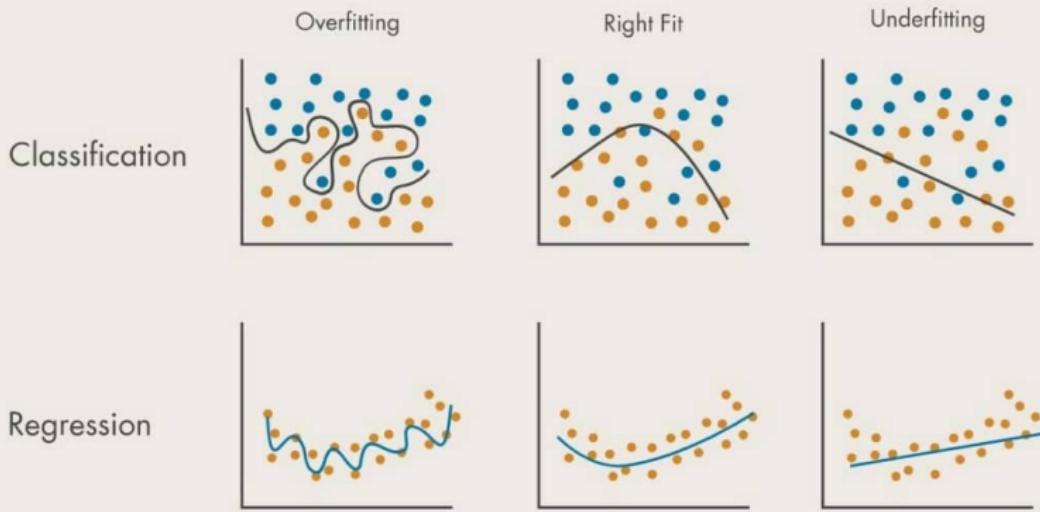
Travailler avec les données et éviter les pièges

Sélection des caractéristiques et modélisation

- ① Feature selection (features pertinentes pour le modèle)
- ② Choix et évaluation des modèles
 - Sélection du modèle approprié en fonction du problème et des data
 - L'utilisation de la validation croisée pour évaluer le modèle
 - L'interprétation des métriques d'évaluation telles que la précision, le rappel, le F1-score, etc.
- ③ Gestion du déséquilibre des classes
 - L'identification et la gestion du déséquilibre des classes dans les problèmes de classification
 - L'utilisation de techniques de suréchantillonnage, de sous-échantillonnage ou d'ajustement des poids des classes pour traiter ce déséquilibre

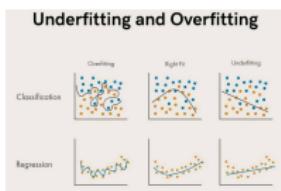
Travailler avec les données et éviter les pièges

Underfitting and Overfitting



Artificial Intelligence, Machine Learning and Deep Learning With Python

- └ Le Big Data et son intégration avec l'IA
 - └ Eviter les pièges en Machine Learning
 - └ Travailler avec les données et éviter les pièges

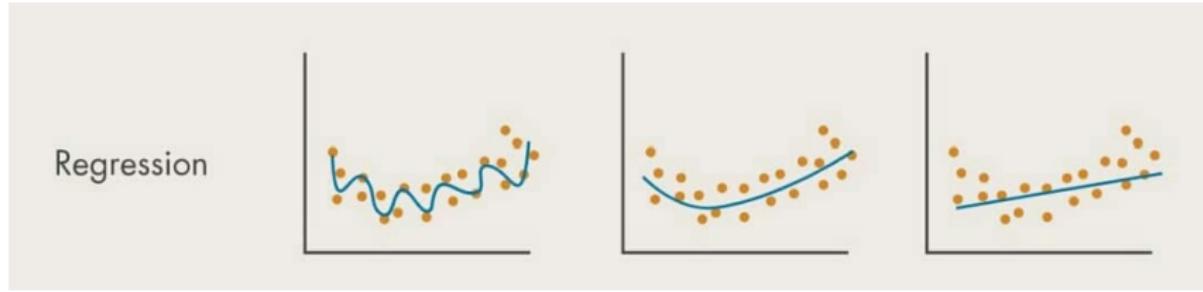


- Overfitting (Surapprentissage) en régression :
 - Modèle trop complexe par rapport au training set.
 - Excellente performance sur le trainset, mais mauvaise sur de nouvelles données.
 - Le modèle "mémorise" les données d'entraînement spécifiques au lieu de généraliser les tendances sous-jacentes.
 - Peut se produire lorsque le modèle a un nombre excessif de paramètres par rapport à la taille des données d'entraînement.
- Underfitting (Sous-apprentissage) en régression :
 - Le modèle est trop simple pour capturer les tendances sous-jacentes des données.
 - Performance médiocre à la fois sur les train et new data.
 - Le modèle ne parvient pas à saisir les relations complexes entre les variables.
 - Peut se produire lorsque le modèle est trop limité en termes de capacité ou que le trainset insuffisantes.

Travailler avec les données et éviter les pièges

Gestion de l'Overfitting et de l'Underfitting en régression

- **Overfitting** : Réduire la complexité du modèle, utiliser la régularisation et la validation croisée.
- **Underfitting** : Augmenter la complexité du modèle et ajouter de nouvelles caractéristiques.



Artificial Intelligence, Machine Learning and Deep Learning With Python

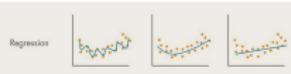
Le Big Data et son intégration avec l'IA

Eviter les pièges en Machine Learning

Travailler avec les données et éviter les pièges

Travailler avec les données et éviter les pièges
Gestion de l'Overfitting et de l'Underfitting en régression

- **Overfitting** : Réduire la complexité du modèle, utiliser la régularisation et la validation croisée.
- **Underfitting** : Augmenter la complexité du modèle et ajouter de nouvelles caractéristiques.



Gestion de l'overfitting en régression :

- Utiliser des techniques de régularisation telles que la régression ridge ou la régression Lasso pour limiter la complexité du modèle.
- Collecter davantage de données d'entraînement pour fournir une meilleure représentation de la variabilité des données réelles.
- Utiliser la validation croisée pour évaluer la performance du modèle.

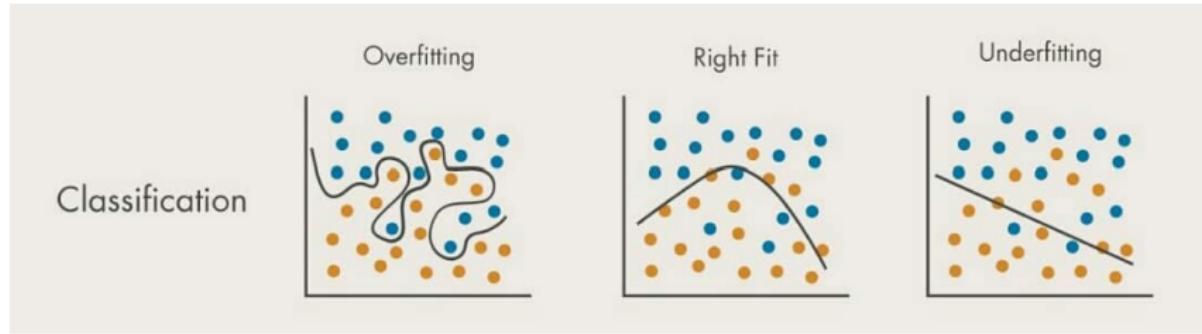
Gestion de l'underfitting en régression :

- Utiliser un modèle plus complexe comme une régression polynomiale pour capturer des relations non linéaires entre les variables.
- Ajouter de nouvelles caractéristiques ou transformer les caractéristiques existantes pour fournir plus d'informations.
- Augmenter la capacité du modèle en augmentant le nombre de paramètres ou de couches dans un réseau de neurones.

Travailler avec les données et éviter les pièges

Gestion de l'Overfitting et de l'Underfitting en classification

- **Overfitting** : Utiliser la régularisation, la validation croisée et l'ajustement des hyperparamètres pour limiter la complexité du modèle.
- **Underfitting** : Augmenter la complexité du modèle en ajoutant de nouvelles caractéristiques ou en utilisant des modèles plus sophistiqués.



Artificial Intelligence, Machine Learning and Deep Learning With Python

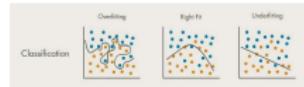
Le Big Data et son intégration avec l'IA

Eviter les pièges en Machine Learning

Travailler avec les données et éviter les pièges

Travailler avec les données et éviter les pièges Gestion de l'Overfitting et de l'Underfitting en classification

- **Overfitting** : Utiliser la régularisation, la validation croisée et l'ajustement des hyperparamètres pour limiter la complexité du modèle.
- **Underfitting** : Augmenter la complexité du modèle en ajoutant de nouvelles caractéristiques ou en utilisant des modules plus sophistiqués.



- Gestion de l'overfitting en classification :
 - Utiliser des techniques de régularisation comme la pénalisation L1 ou L2 pour réduire la complexité du modèle.
 - Appliquer des techniques de réduction de dimensionnalité pour sélectionner les caractéristiques les plus informatives.
 - Utiliser l'augmentation de données pour augmenter la taille et la diversité de l'ensemble d'entraînement.
- Gestion de l'underfitting en classification :
 - Utiliser un modèle plus complexe comme les machines à vecteurs de support (SVM) non linéaires ou les réseaux de neurones profonds.
 - Ajouter des caractéristiques supplémentaires ou des transformations des caractéristiques existantes pour améliorer la représentation des données.
 - Ajuster les paramètres du modèle pour augmenter sa capacité et permettre une meilleure adaptation aux données.

Quelques modèles de Machine Learning



- └ Le Big Data et son intégration avec l'IA
 - └ Quelques modèles de ML
 - └ Quelques modèles de Machine Learning



- **Source de connaissances :**

- Humain : *expérience, observation, interaction avec l'environnement.*
- Machine : *algorithmes et modèles statistiques apprenant à partir de données.*

- **Capacité d'abstraction :**

- Humain : *capacité naturelle à généraliser et à appliquer des connaissances à de nouvelles situations.*
- Machine : *nécessite souvent un grand nombre de données étiquetées pour généraliser.*

- **Adaptabilité et flexibilité :**

- Humain : *capacité à apprendre de nouvelles tâches avec peu d'exemples.*
- Machine : *nécessite un entraînement spécifique pour chaque tâche et peut avoir du mal à se généraliser.*

Modèles de Régression

Régression Linéaire

Modélisation: relation entre les variables explicatives et la variable cible

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_n x_n + \epsilon$$

- y : Variable dépendante à prédire
- x_1, x_2, \dots, x_n : Variables indépendantes (caractéristiques)
- $\beta_0, \beta_1, \beta_2, \dots, \beta_n$: Coefficients de régression à estimer
- ϵ : Terme d'erreur aléatoire

Régression Logistique

$$P(y = 1) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_n x_n)}}$$

- $P(y = 1)$: Probabilité que la variable dépendante y prenne la valeur 1
- x_1, x_2, \dots, x_n : features.

Artificial Intelligence, Machine Learning and Deep Learning With Python

- └ Le Big Data et son intégration avec l'IA
 - └ Quelques modèles de ML
 - └ Modèles de Régression

Modèles de Régression

Régression Linéaire

Modélisation: relation entre les variables explicatives et la variable cible

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_n x_n + \epsilon$$

- y : Variable dépendante à prédire
- x_1, x_2, \dots, x_n : Variables indépendantes (caractéristiques)
- $\beta_0, \beta_1, \beta_2, \dots, \beta_n$: Coefficients de régression à estimer
- ϵ : Terme d'erreur aléatoire

Régression Logistique

$$P(y = 1) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_n x_n)}}$$

- $P(y = 1)$: Probabilité que la variable dépendante y prenne la valeur 1
- x_1, x_2, \dots, x_n : features.

Régression Linéaire: La régression linéaire modélise la relation entre les variables explicatives et la variable cible à l'aide d'une équation linéaire. Par exemple, on peut utiliser la régression linéaire pour prédire le prix d'une maison en fonction de sa superficie, du nombre de chambres, de l'année de construction, etc.

Régression Logistique: La régression logistique estime la probabilité qu'une variable dépendante binaire (0 ou 1) prenne la valeur 1 en fonction des variables explicatives. Par exemple, on peut utiliser la régression logistique pour prédire la probabilité qu'un client achète un produit en fonction de son âge, de son revenu, de ses intérêts, etc.

Arbre de Décision

- Modèle d'apprentissage automatique non linéaire et non paramétrique
- Représente une série de décisions basées sur les features
- Utilisé pour la classification et la régression

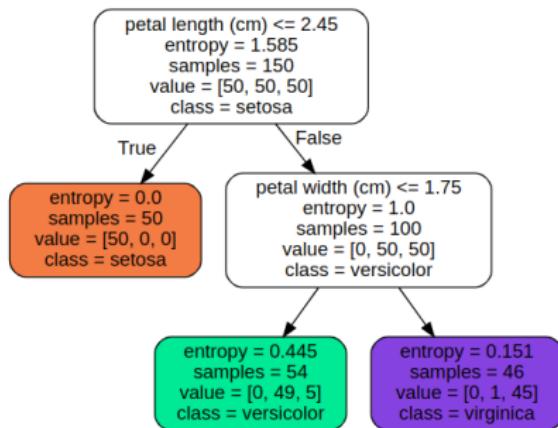


Figure: Classification tree

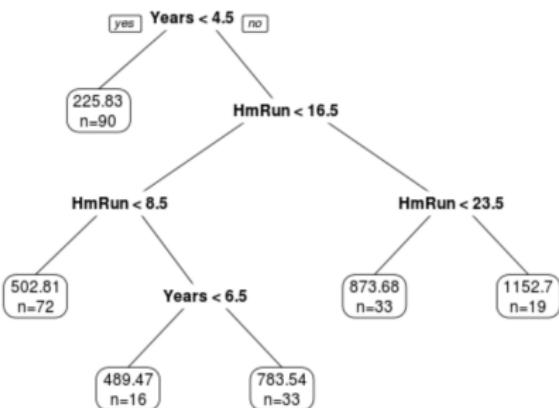


Figure: Regression tree

2024-06-14

Artificial Intelligence, Machine Learning and Deep Learning With Python

- Le Big Data et son intégration avec l'IA
 - Quelques modèles de ML
 - Arbre de Décision

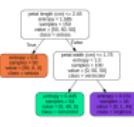
2024-06-14

Artificial Intelligence, Machine Learning and Deep Learning With Python

- Le Big Data et son intégration avec l'IA
 - Quelques modèles de ML
 - Arbre de Décision

Arbre de Décision

- Modèle d'apprentissage automatique non linéaire et non paramétrique
- Représente une série de décisions basées sur les features
- Utilisé pour la classification et la régression



Arbre de Décision

- Modèle d'apprentissage automatique non linéaire et non paramétrique
- Représente une série de décisions basées sur les features
- Utilisé pour la classification et la régression

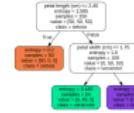


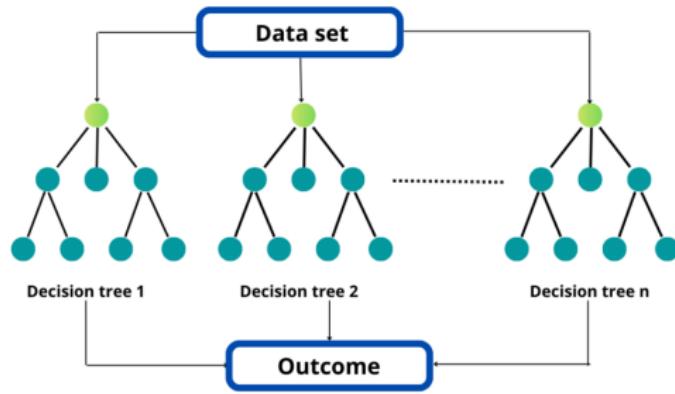
Figure: Classification tree

Figure: Regression tree

L'arbre de décision est un modèle qui utilise une structure en forme d'arbre pour prendre des décisions ou faire des prédictions. Il est construit en partitionnant les données en fonction des caractéristiques jusqu'à une condition d'arrêt. Il est interprétable et sensible aux variations des données d'entraînement.

Forêt Aléatoire (Random Forest)

- Basée sur l'ensemble de plusieurs arbres de décision
- Chaque arbre est construit sur un sous-ensemble aléatoire des données d'entraînement et des caractéristiques
- Les prédictions finales sont obtenues en agrégant les prédictions de chaque arbre (majorité pour la classification, moyenne pour la régression)



Machines à Vecteurs de Support (SVM)

- Modèle d'apprentissage automatique supervisé
- Utilisé pour la classification et la régression
- Trouve un hyperplan optimal qui sépare les données de différentes classes ou estime une fonction pour la régression
- Maximise la marge entre les données et l'hyperplan pour une meilleure généralisation

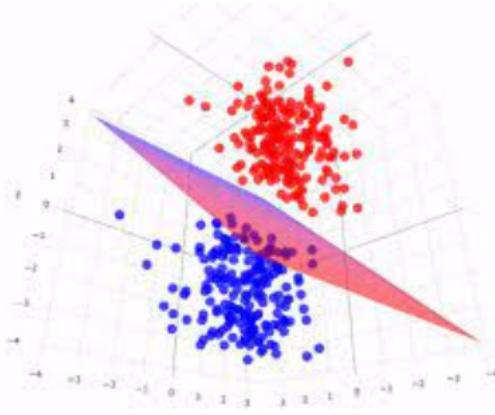


Figure: Support Vector Classifier

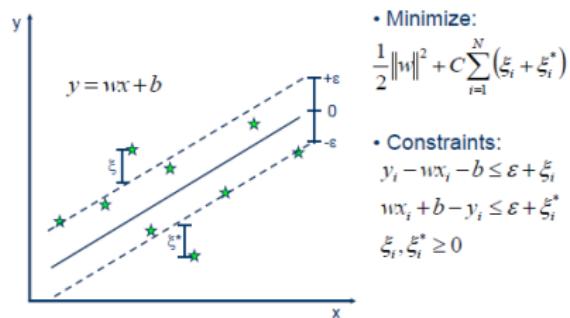
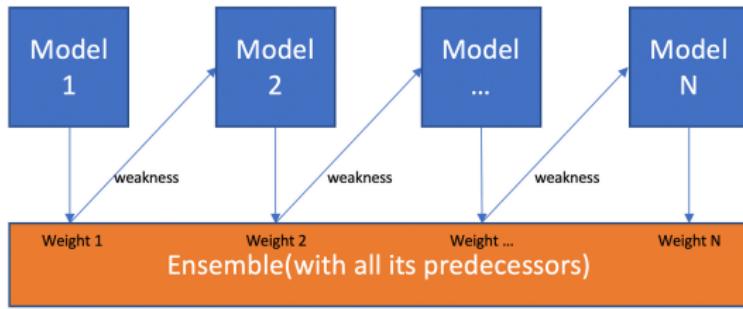


Figure: Support Vector Régresseur

Boosting

- Technique de ML utilisée pour améliorer la performance des modèles
- Combinaison **séquentielle** de modèles faibles pour former un fort
- Chaque modèle faible est entraîné à se concentrer sur les échantillons mal classés par les modèles précédents
- Les prédictions finales sont obtenues en agrégant les prédictions de chaque modèle faible (pondération selon leur performance)

Model 1,2,..., N are individual models (e.g. decision tree)



Boosting vs Bagging

Boosting

- Combinaison séquentielle de modèles faibles
- Chaque modèle se concentre sur les échantillons mal classés par les modèles précédents
- Biais réduit, forte capacité de généralisation
- Sensible aux données d'entraînement bruitées ou aberrantes

Bagging

- Combinaison parallèle de modèles indépendants
- Chaque modèle est entraîné sur un sous-ensemble aléatoire des données d'entraînement
- Réduction de la variance, faible risque de surajustement
- Moins sensible aux données bruitées ou aberrantes

Évaluation des modèles

- L'évaluation des modèles est essentielle pour mesurer leur performance et prendre des décisions informées.
- Métriques de performance couramment utilisées :
 - Précision : mesure la proportion de prédictions positives correctes.
 - Rappel : mesure la proportion de vrais positifs identifiés.
 - F-mesure : combine la précision et le rappel en une seule métrique.
 - Exactitude : mesure la proportion de prédictions correctes dans l'ensemble des données.
 - Courbe ROC : représente la sensibilité (rappel) en fonction de la spécificité.
- Techniques d'évaluation :
 - Validation croisée : divise les données en ensembles d'entraînement et de test pour évaluer la performance.
 - Holdout : divise les données en un ensemble d'entraînement et un ensemble de test.
 - Bootstrap : utilise des échantillons bootstrap pour estimer la performance du modèle.

Conclusion

- Le Machine Learning est une approche d'intelligence artificielle qui permet aux ordinateurs d'apprendre à partir des données sans être explicitement programmés.
- Le processus de Machine Learning comprend :
 - ① Collecte des données d'entraînement.
 - ② Sélection du modèle approprié.
 - ③ Entraînement du modèle sur les données d'entraînement.
 - ④ Évaluation de la performance du modèle sur des données de test.
 - ⑤ Utilisation du modèle entraîné pour faire des prédictions sur de nouvelles données.
- Les algorithmes d'apprentissage automatique peuvent être supervisés ou non supervisés.
- L'objectif du Machine Learning est de généraliser à partir des données d'entraînement afin de faire des prédictions précises sur de nouvelles données non vues auparavant.

Artificial Intelligence and Machine Learning 2



AI / ML

Plan de présentation

- Introduction
 - Contexte et objectifs
- XAI et IML : Explicabilité dans l'IA et le ML
 - Comprendre XAI et IML
 - Difficultés liées à l'isolation de la contribution d'une variable
 - Modèle de boîte noire 101
- KNIME pour XAI et IML
 - Présentation de KNIME
 - Utilisation de KNIME pour l'explicabilité et l'interprétabilité
 - Exemples d'utilisation de KNIME dans XAI et IML
- Techniques de XAI : Explications globales
 - Obtenir des explications globales
 - Techniques : importance des variables, arbres de décision, règles d'association, etc.
 - Avantages et limitations des techniques d'explication globale

Vue d'ensemble des concepts clés

- Qu'est-ce que l'Explainable AI (XAI) et l'Interprétabilité en ML (IML)?
- Les défis de l'isolement de la contribution d'une variable
- Modèle de boîte noire 101 : Comprendre les modèles opaques
- KNIME pour XAI et IML : Une introduction à l'outil
- Techniques d'Explainable AI (XAI) : explications globales, explications locales
- Techniques d'Interprétabilité en Machine Learning (IML) : visualisation, importance des variables, etc.
- Conception expérimentale et contrôles statistiques : l'importance de la planification et des comparaisons de modèles
- Conditionnel Probabilité et théorème de Bayes : Mise à jour des probabilités
- Prédiction et preuve avec les statistiques bayésiennes : Introduction aux statistiques bayésiennes
- Modélisation causale : Modélisation d'équations structurelles (SEM), 

Explainable AI (XAI) et Interprétabilité en ML (IML)

- L'Explainable AI (XAI) et l'Interprétabilité en ML (IML) visent à rendre les modèles de ML plus compréhensibles et expliquables pour les humains.
- L'IML : compréhension des décisions prises par les modèles, alors que le XAI vise à expliquer le fonctionnement interne des modèles.
- L'importance de l'IML et du XAI :
 - Gagner la confiance des utilisateurs et des parties prenantes.
 - Déetecter les biais et les erreurs de modélisation.
- Techniques d'IML et de XAI :
 - Visualisation des caractéristiques importantes.
 - Interprétation des poids des modèles linéaires.

Artificial Intelligence, Machine Learning and Deep Learning With Python

└ Artificial Intelligence and Machine Learning 2

└ XAI et IML : Explicabilité dans l'IA et le ML

└ Explainable AI (XAI) et Interprétabilité en ML (IML)

Explainable AI (XAI) et Interprétabilité en ML (IML)

- L'Explainable AI (XAI) et l'Interprétabilité en ML (IML) visent à rendre les modèles de ML plus compréhensibles et expélioables pour les humains.
- L'IML : compréhension des décisions prises par les modèles, alors que le XAI vise à expliquer le fonctionnement interne des modèles.
- L'importance de l'IML et du XAI:
 - Gagner la confiance des utilisateurs et des parties prenantes.
 - Déetecter les biais et les erreurs de modélisation.
- Techniques d'IML et de XAI :
 - Visualisation des caractéristiques importantes.
 - Interprétation des poids des modèles linéaires.

- L'importance de l'IML et du XAI :

- Gagner la confiance des utilisateurs et des parties prenantes.
 - Déetecter les biais et les erreurs de modélisation.
 - Se conformer aux réglementations et aux normes éthiques.
 - Faciliter la résolution des problèmes lorsque les modèles produisent des résultats inattendus ou incorrects.

- Techniques d'IML et de XAI :

- Visualisation des caractéristiques importantes.
 - Interprétation des poids des modèles linéaires.
 - Méthodes d'interprétabilité spécifiques à certains algorithmes (des tree).
 - Utilisation de modèles explicatifs, tels que les réseaux de neurones à propagation avant avec des couches interprétables.

Les défis de l'isolement de la contribution d'une variable

- Les défis courants de l'isolation de la contribution d'une va en ML :
 - Corrélations
 - Interactions
 - Non-linéarité
 - Multicollinearité
- Méthodes pour aborder ces défis :
 - Analyse de sensibilité : évaluer l'impact d'une variable en la modifiant de manière contrôlée tout en maintenant les autres variables constantes.
 - Décomposition de la variance : attribuer une part de la variance expliquée à chaque variable.
 - Méthodes d'importance de variable : estimer l'importance relative des variables en utilisant des techniques telles que les arbres de décision ou les coefficients de régression.

Artificial Intelligence, Machine Learning and Deep Learning With Python

└ Artificial Intelligence and Machine Learning 2

└ XAI et IML : Explicabilité dans l'IA et le ML

└ Les défis de l'isolement de la contribution d'une variable

- Corrélations : les variables sont souvent corrélées entre elles, ce qui rend difficile de déterminer la contribution spécifique d'une variable sans tenir compte des autres.
- Interactions : les variables peuvent interagir de manière complexe, ce qui rend difficile d'attribuer une contribution individuelle à chaque va.
- Non-linéarité : les relations entre les variables et la variable cible peuvent être non linéaires, ce qui complique l'isolement des contributions individuelles.
- Multicollinéarité : lorsque plusieurs variables sont fortement corrélées, il peut être difficile de distinguer leur contribution individuelle.

Les défis de l'isolement de la contribution d'une variable

- Les défis courants de l'isolation de la contribution d'une va en ML :
 - Corrélations
 - Interactions
 - Non-linéarité
 - Multicollinearité
- Méthodes pour aborder ces défis :
 - Analyse de sensibilité : évaluer l'impact d'une variable en la modifiant de manière contrôlée tout en maintenant les autres variables constantes.
 - Décomposition de la variance : attribuer une part de la variance expliquée à chaque variable.
 - Méthodes d'importance de variable : estimer l'importance relative des variables en utilisant des techniques telles que les arbres de décision ou les coefficients de régression.

Modèle de boîte noire 101 : Modèles opaques



- Modèles d'IA dont les mécanismes internes sont difficiles à interpréter.
- Modèles très performants en termes de prédiction, mais souvent difficile à comprendre.
- Les modèles opaques (DNN ou SVM) peuvent avoir des millions de paramètres et des architectures complexes.
- L'opacité de ces modèles pose des défis en termes de transparence, d'éthique et d'acceptabilité sociale de l'IA.
- Malgré leur complexité, XAI et IML tentent de les comprendre.

KNIME pour XAI et IML : Une introduction à l'outil

- KNIME est une plateforme open-source d'analyse des données et de construction de workflows.
- Il offre une interface intuitive et conviviale pour créer, exécuter et partager des workflows d'analyse de données.
- KNIME propose également une large gamme de modules et d'extensions pour XAI et l'IML.
- Avec KNIME, vous pouvez appliquer des techniques d'XAI pour comprendre comment les modèles prennent des décisions et les expliquer de manière compréhensible.

Artificial Intelligence, Machine Learning and Deep Learning With Python

└ Artificial Intelligence and Machine Learning 2

└ XAI et IML : Explicabilité dans l'IA et le ML

└ KNIME pour XAI et IML : Une introduction à

...
...

KNIME pour XAI et IML : Une introduction à l'outil

- KNIME est une plateforme open-source d'analyse des données et de construction de workflows.
- Il offre une interface intuitive et conviviale pour créer, exécuter et partager des workflows d'analyse de données.
- KNIME propose également une large gamme de modules et d'extensions pour XAI et l'IML.
- Avec KNIME, vous pouvez appliquer des techniques d'XAI pour comprendre comment les modèles prennent des décisions et les expliquer de manière compréhensible.

- Vous pouvez également utiliser des modules d'IML pour visualiser et interpréter les résultats de vos modèles, tels que l'importance des variables, les poids des coefficients, etc.
- KNIME offre une flexibilité et une extensibilité importantes, vous permettant d'adapter facilement vos workflows aux besoins spécifiques de votre projet.

Techniques XAI : explications globales, explications locales

- Les techniques XAI sont des méthodes utilisées pour expliquer les décisions prises par les modèles d'IA.
- Les explications globales fournissent une vue d'ensemble du modèle et de ses principaux facteurs de décision, permettant de comprendre le comportement général du modèle.
- Les techniques d'explication globale incluent des méthodes telles que les diagrammes d'importance de variables, les graphiques de dépendance partielle et les cartes de chaleur.

Artificial Intelligence, Machine Learning and Deep Learning With Python

└ Artificial Intelligence and Machine Learning 2

└ XAI et IML : Explicabilité dans l'IA et le ML

└ Techniques XAI : explications globales,

Techniques XAI : explications globales, explications locales

- Les techniques XAI sont des méthodes utilisées pour expliquer les décisions prises par les modèles d'IA.
- Les explications globales fournissent une vue d'ensemble du modèle et de ses principaux facteurs de décision, permettant de comprendre le comportement général du modèle.
- Les techniques d'explication globale incluent des méthodes telles que les diagrammes d'importance de variables, les graphiques de dépendance partielle et les cartes de chaleur.

- Les explications locales se concentrent sur une prédiction spécifique et fournissent une explication détaillée de la contribution de chaque variable à cette prédiction.
- Les techniques d'explication locale incluent des méthodes telles que les perturbations de variable, les méthodes de désensibilisation et les arbres de décision locaux.
- Compréhension fine des facteurs qui ont conduit à une prédiction spécifique, ce qui peut aider à détecter les biais, les erreurs ou les cas extrêmes.

Techniques IML : visualisation, importance des variables

- La visualisation permet d'explorer et de comprendre les relations entre les variables et les résultats du modèle.
- Les graphiques, les diagrammes et les cartes peuvent être utilisés pour représenter les données de manière intuitive et faciliter l'interprétation.
- L'importance des variables est une autre technique d'IML qui permet d'identifier les variables qui ont le plus d'influence sur les prédictions du modèle.

Artificial Intelligence, Machine Learning and Deep Learning With Python

└ Artificial Intelligence and Machine Learning 2

└ XAI et IML : Explicabilité dans l'IA et le ML

└ Techniques IML : visualisation, importance des

- Les méthodes d'importance des variables incluent l'analyse de sensibilité, le calcul des coefficients ou des poids des variables et les techniques d'élagage.
- En combinant la visualisation des données, l'importance des variables et d'autres techniques d'IML, on peut obtenir une compréhension approfondie du modèle de ML et de ses mécanismes de décision.

Techniques IML : visualisation, importance des variables

- La visualisation permet d'explorer et de comprendre les relations entre les variables et les résultats du modèle.
- Les graphiques, les diagrammes et les cartes peuvent être utilisés pour représenter les données de manière intuitive et faciliter l'interprétation.
- L'importance des variables est une autre technique d'IML qui permet d'identifier les variables qui ont le plus d'influence sur les prédictions du modèle.

Conception expérimentale et contrôles statistiques

- Planification : définition claire des objectifs de l'expérience, choix des features à mesurer, échantillonnage.
- Les comparaisons de modèles
- Des techniques statistiques telles que les tests d'hypothèse, l'ANOVA et les tests de comparaison multiple
- Il est également important de prendre en compte les biais potentiels

Artificial Intelligence, Machine Learning and Deep Learning With Python

└ Artificial Intelligence and Machine Learning 2

└ XAI et IML : Explicabilité dans l'IA et le ML

└ Conception expérimentale et contrôles

- Une planification et des contrôles appropriés permettent de minimiser les erreurs expérimentales, d'obtenir des résultats précis et de fournir des informations utiles pour prendre des décisions éclairées en matière de modélisation.
- En fin de compte, la conception expérimentale et les contrôles statistiques renforcent la validité et la crédibilité des résultats obtenus à partir des modèles d'IA.

- Planification : définition claire des objectifs de l'expérience, choix des variables à mesurer, échantillonnage.
- Les comparaisons de modèles
- Des techniques statistiques telles que les tests d'hypothèse, l'ANOVA et les tests de comparaison multiple
- Il est également important de prendre en compte les biais potentiels

Probabilité Conditionnelle et théorème de Bayes

- Elle est notée $\mathbb{P}(A|B)$ et se calcule suivant la relation

$$\mathbb{P}(A|B) = \frac{\mathbb{P}(A \cap B)}{\mathbb{P}(B)}$$

- Le théorème de Bayes permet de mettre à jour les probabilités conditionnelles lorsque de nouvelles informations sont disponibles.
- La formule de Bayes est donnée par

$$\mathbb{P}(A|B) = \frac{\mathbb{P}(B|A) * \mathbb{P}(A)}{\mathbb{P}(B)}$$

- Le théorème de Bayes est largement utilisé dans les domaines de la statistique, du ML et de l'IA pour la prise de décision.
- Il permet de mettre à jour les probabilités a priori en fonction des nouvelles observations, ce qui permet d'obtenir des estimations plus précises et fiables.

Introduction aux statistiques bayésiennes

- Contrairement aux statistiques fréquentistes, les statistiques bayésiennes considèrent les paramètres comme des v.a avec des dp.
- L'analyse bayésienne commence par une distribution a priori pour les paramètres, puis met à jour cette distribution en utilisant le théorème de Bayes pour obtenir une distribution a posteriori.
- La distribution a posteriori est utilisée pour effectuer des prédictions et des estimations, en tenant compte de l'incertitude associée aux résultats.
- Les statistiques bayésiennes permettent également de comparer des modèles en utilisant des facteurs de Bayes, qui intègrent la vraisemblance des données et la complexité des modèles.
- Les statistiques bayésiennes offrent une approche cohérente et intuitive pour la prise de décision en intégrant les connaissances préalables et les données observées.
- Cependant, l'analyse bayésienne peut nécessiter des techniques d'inférence approximative, telles que les chaînes de MCMC.



Statistique Bayésienne vs. Statistique Fréquentiste

Statistique Bayésienne

- Les paramètres sont considérés comme des variables aléatoires
- On utilise des distributions de probabilité a priori pour les paramètres
- La conclusion est une distribution de probabilité a posteriori des paramètres
- Permet d'incorporer des connaissances préalables sur les paramètres

Statistique Fréquentiste

- Les paramètres sont considérés comme des quantités fixes inconnues
- On utilise des estimateurs ponctuels et des intervalles de confiance
- La conclusion est une estimation ponctuelle des paramètres
- Se concentre sur la répétition d'expériences pour faire des inférences

Artificial Intelligence, Machine Learning and Deep Learning With Python

└ Artificial Intelligence and Machine Learning 2

└ XAI et IML : Explicabilité dans l'IA et le ML

└ Statistique Bayésienne vs. Statistique Fréquentiste

La principale différence entre l'approche bayésienne et l'approche fréquentiste réside dans la façon de considérer les paramètres du modèle statistique.

En statistique bayésienne, les paramètres sont vus comme des variables aléatoires pour lesquelles on peut définir des distributions de probabilité a priori. L'inférence se fait alors en mettant à jour ces distributions a priori en utilisant les données observées pour obtenir des distributions a posteriori des paramètres.

En statistique fréquentiste, les paramètres sont considérés comme des quantités fixes inconnues. L'inférence se fait en utilisant des estimateurs ponctuels et des intervalles de confiance, basés sur la répétition d'expériences.

L'approche bayésienne permet d'incorporer des connaissances préalables sur les paramètres, tandis que l'approche fréquentiste se concentre uniquement sur les données observées.

Statistique Bayésienne vs. Statistique Fréquentiste

Statistique Bayésienne

- Les paramètres sont considérés comme des variables aléatoires
- On utilise des distributions de probabilité a priori pour les paramètres
- La conclusion est une distribution de probabilité a posteriori des paramètres
- Permet d'incorporer des connaissances préalables sur les paramètres

Statistique Fréquentiste

- Les paramètres sont considérés comme des quantités fixes inconnues
- On utilise des estimateurs ponctuels et des intervalles de confiance
- La conclusion est une estimation ponctuelle des paramètres
- Se concentre sur la répétition d'expériences pour faire des inferences

Modélisation causale : Modélisation d'équations structurelles (SEM) et réseaux bayésiens

- La modélisation causale vise à comprendre les relations de cause à effet entre les variables dans un système.
- Les équations structurelles (SEM) et les réseaux bayésiens sont deux approches couramment utilisées en modélisation causale.

Modélisation d'équations structurelles (SEM)

- Les SEM sont des modèles qui représentent les relations causales entre les variables à l'aide d'équations mathématiques.
- Les SEM permettent de tester des hypothèses causales.

Réseaux bayésiens

- Les réseaux bayésiens utilisent des graphes probabilistes pour représenter les relations causales entre les variables.
- Les réseaux bayésiens intègrent également des connaissances a priori et des distributions de probabilité pour quantifier l'incertitude dans la modélisation causale.

Artificial Intelligence, Machine Learning and Deep Learning With Python

└ Artificial Intelligence and Machine Learning 2

└ XAI et IML : Explicabilité dans l'IA et le ML

└ Modélisation causale : Modélisation d'équations

Modélisation causale : Modélisation d'équations structurelles (SEM) et réseaux bayésiens

- La modélisation causale vise à comprendre les relations de cause à effet entre les variables dans un système.
- Les équations structurelles (SEM) et les réseaux bayésiens sont deux approches couramment utilisées en modélisation causale.

Modélisation d'équations structurelles (SEM)

- Les SEM sont des modèles qui représentent les relations causales entre les variables à l'aide d'équations mathématiques.
- Les SEM permettent de tester des hypothèses causales.

Réseaux bayésiens

- Les réseaux bayésiens utilisent des graphes probabilistes pour représenter les relations causales entre les variables.
- Les réseaux bayésiens intègrent également des connaissances a priori et des distributions de probabilité pour quantifier l'incertitude dans la modélisation causale.

Ces deux approches de modélisation causale, les SEM et les réseaux bayésiens, permettent de faire des prédictions causales, d'identifier les variables clés et de tester les hypothèses causales dans un système.

Les SEM se concentrent sur la représentation des relations causales à l'aide d'équations mathématiques, tandis que les réseaux bayésiens utilisent une approche graphique probabiliste qui intègre les connaissances a priori.

Bien que différentes dans leur mise en œuvre, ces deux méthodes visent toutes deux à comprendre et à modéliser les liens de cause à effet entre les variables d'un système complexe.

Statistiques et tests d'hypothèse

- La p -value est utilisée pour évaluer les preuves contre l'hypothèse H_0 .
- Une p -value faible indique une preuve solide contre H_0 .
- La p -value seule ne fournit pas d'informations sur la taille de l'effet ou la signification pratique des résultats.
- L'interprétation doit tenir compte de la p -value et de l'importance clinique ou scientifique des résultats.

- La corrélation mesure la relation statistique entre deux variables.
- La causalité nécessite des preuves supplémentaires, comme des expérimentations contrôlées.
- Les études observationnelles fournissent des indications de relation causale, mais ne prouvent pas définitivement l'existence de la causalité.

Détection de la multicolinéarité et stratégies associées

Détection de la multicolinéarité :

- Calcul des coefficients de corrélation.
- Variance inflation factor (VIF).
- Analyse des valeurs propres.

Stratégies pour traiter la multicolinéarité :

- Supprimer les variables fortement corrélées.
- Combinaison de variables corrélées.
- Utilisation de techniques de régularisation (régression ridge, régression lasso).

Remarque : Les approches de détection et de gestion de la multicolinéarité doivent être adaptées au contexte spécifique de l'analyse et aux objectifs de modélisation.

Induction, déduction, falsification et contrefactuel : Importance dans l'évaluation du modèle

- **Induction** : Permet de généraliser à partir des observations spécifiques et de tirer des conclusions sur des situations non observées.
- **Déduction** : Utilisation de règles logiques pour déduire des conséquences spécifiques à partir de propositions générales. Peut aider à tester les prédictions d'un modèle.
- **Falsification** : Processus de recherche d'observations ou de tests qui pourraient réfuter ou invalider un modèle. Contribue à la rigueur et à la fiabilité d'un modèle.
- **Contrefactuel** : Étudie les effets hypothétiques d'un changement dans les conditions ou les variables. Permet d'évaluer l'impact causal et de comparer différentes situations.

L'utilisation appropriée de ces concepts contribue à une évaluation rigoureuse et solide des modèles, en tenant compte à la fois des preuves empiriques et des raisonnements logiques.

Utilité décroissante des valeurs p avec un nombre croissant de paramètres de modèle : Explication

- Lorsque le nombre de paramètres d'un modèle statistique augmente, les valeurs p associées à ces paramètres ont tendance à diminuer.
- Les valeurs p mesurent la significativité statistique d'un paramètre en évaluant la probabilité d'obtenir des résultats aussi extrêmes que ceux observés, sous l'hypothèse nulle.
- Cependant, avec un nombre croissant de paramètres, il y a une augmentation du nombre de tests statistiques effectués, ce qui peut conduire à une augmentation des chances de trouver des associations significatives par pur hasard.
- Ce phénomène est connu sous le nom d'utilité décroissante des valeurs p et est causé par le problème de multiplicité des tests.
- Plus il y a de paramètres à tester, plus il est probable d'obtenir des associations significatives par simple hasard, même en l'absence de relations réelles entre les variables.
- Cela souligne l'importance de prendre en compte le contexte

Évaluation des performances du modèle dans l'EDA

- L'évaluation des performances du modèle est une étape essentielle dans l'exploration de données, visant à évaluer la qualité et l'adéquation d'un modèle pour représenter les données.

Approche statistique traditionnelle

- Les statistiques traditionnelles se concentrent sur la précision et la validité des modèles, en utilisant des méthodes rigoureuses basées sur des hypothèses et des tests statistiques.
- Les statistiques mettent l'accent sur la généralisation des résultats, la réduction de l'incertitude et la déduction causale.

Approche d'exploration de données

- L'exploration de données se concentre sur la découverte de nouvelles connaissances, l'identification de modèles intéressants et la génération d'hypothèses.
- L'exploration de données privilégie l'exploration des données brutes, l'utilisation de techniques non paramétriques et l'incorporation de

Artificial Intelligence, Machine Learning and Deep Learning With Python

└ Artificial Intelligence and Machine Learning 2

└ XAI et IML : Explicabilité dans l'IA et le ML

└ Évaluation des performances du modèle dans l'EDA

Évaluation des performances du modèle dans l'EDA

- L'évaluation des performances du modèle est une étape essentielle dans l'exploration de données, visant à évaluer la qualité et l'adéquation d'un modèle pour représenter les données.

Approche statistique traditionnelle

- Les statistiques traditionnelles se concentrent sur la précision et la validité des modèles, en utilisant des méthodes rigoureuses basées sur des hypothèses et des tests statistiques.
- Les statistiques mettent l'accent sur la généralisation des résultats, la réduction de l'incertitude et la déduction causale.

Approche d'exploration de données

- L'exploration de données se concentre sur la découverte de nouvelles connaissances, l'identification de modèles intéressants et la génération d'hypothèses.
- L'exploration de données priviliege l'exploration des données brutes, l'utilisation de techniques non paramétriques et l'incorporation de l'expertise humaine.

Les statistiques traditionnelles et l'exploration de données ont souvent des objectifs et des philosophies contradictoires dans l'évaluation des performances du modèle.

Les statistiques se concentrent davantage sur la validité et la généralisation des résultats, tandis que l'exploration de données se concentre sur la découverte de nouvelles connaissances et l'identification de modèles intéressants, même si cela implique une certaine incertitude.

Cependant, il est important de reconnaître que ces deux approches sont complémentaires et peuvent être utilisées conjointement pour une évaluation plus complète des performances du modèle dans le cadre de l'exploration de données.

Conclusion

- Récapitulatif des concepts clés abordés dans les applications pratiques de l'IA et du Machine Learning
- Invitation aux questions et discussions

Artificial Intelligence and Machine Learning 3



AI / ML

Plan de présentation

- Examen des défis de l'IA
- Application de l'IA étroite à une décision
- Définition de deux approches efficaces utilisées face à l'IA
- Examen de l'apprentissage supervisé et non supervisé
- Explication du harcèlement par l'IA
- Identification de trois concepts sur lesquels repose la justice distributive
- Avantages et limites de XAI
- Humains par rapport aux ordinateurs
- Exemples commerciaux de XAI
- Investir dans XAI

Plan de présentation

- Transformateurs en PNL
- Formation des transformateurs et leur architecture
- Grands modèles de langage
- Présentation des arbres de décision
- Présentation de l'algorithme C5.0
- Présentation des arbres de classification
- Présentation des arbres de régression
- NPL et transformateurs
- BERT et apprentissage par transfert
- Architecture de transformateur et BERT
- Classification de texte

Examen des défis de l'IA

- Manque de transparence des modèles d'IA
- Biais et résultats discriminatoires
- Problèmes d'éthique et de confidentialité
- Impact sur l'emploi
- Besoin de big data
- Sécurité des systèmes d'IA



L'éthique de l'IA sous la loupe de l'UNESCO

└ Artificial Intelligence and Machine Learning 3

└ Examen des défis de l'IA

└ L'éthique de l'IA sous la loupe de l'UNESCO

En 2021, l'UNESCO a adopté une Recommandation sur l'éthique de l'IA, établissant un cadre mondial pour guider le développement et l'utilisation éthique de l'IA. Cette résolution aborde des questions clés telles que

- le respect des droits de l'homme,
- la transparence des systèmes d'IA, et
- la nécessité d'une gouvernance inclusive.

La course aux brevets en IA soulève des défis par rapport à ces principes éthiques, en termes d'accessibilité et d'utilisation responsable de cette technologie. Les États membres de l'UNESCO sont appelés à mettre en œuvre ces recommandations dans leurs politiques nationales.

Application de l'IA étroite à une décision

- L'IA étroite ou IA faible, se réfère à des systèmes d'IA conçus pour effectuer des tâches spécifiques avec une grande précision.
- Ces systèmes d'IA sont utilisées en reconnaissance d'image, traduction automatique, détection de fraude, etc.
- L'IA étroite est utilisée dans les systèmes de recommandation personnalisée ou les chatbots de service client.
- Les systèmes d'IA étroite utilisent des algorithmes sophistiqués et des techniques de ML.
- Limites: incapacité à reproduire pleinement les capacités cognitives humaines (compréhension du contexte, de flexibilité et de raisonnement abstrait).

Définition de deux approches efficaces utilisées face à l'IA

Apprentissage supervisé

- Utilise des données étiquetées
- Prédit des résultats pour de nouvelles données
- Exemples : classification, prédiction

Apprentissage non supervisé

- Utilise des données non étiquetées
- Découvre des structures, des modèles ou des relations
- Exemples : regroupement, réduction de dimensionnalité

Supervised Learning

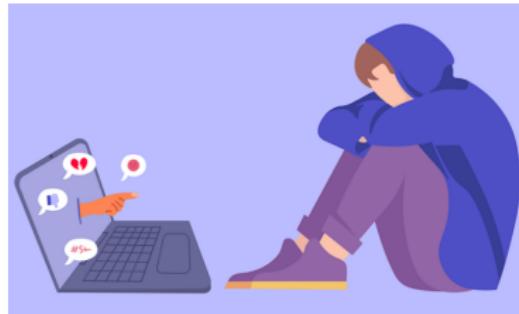
X ₁	X ₂	X ₃	X _p	Y

Un-Supervised Learning

X ₁	X ₂	X ₃	X _p	Y

Explication du harcèlement par l'IA

- Génération de contenu haineux
- Ciblage et traque en ligne
- Biais et discrimination



- Les modèles d'IA peuvent être exploités pour générer automatiquement du contenu haineux et offensant.
- Les algorithmes d'IA peuvent être utilisés pour cibler et harceler des individus en ligne.
- Les biais et préjugés présents dans les données d'entraînement peuvent conduire à une discrimination automatique.

Concepts clés de la justice distributive

Égalité

- Principe fondamental de la justice distributive
- Répartition équitable des ressources et des avantages
- Élimination de la discrimination basée sur des features personnelles

Besoins

- Prise en compte des besoins réels des individus
- Attribution des ressources en fonction des besoins urgents ou essentiels
- Exemple : priorité d'accès aux soins pour les personnes malades

Mérite

- Distribution des ressources en fonction du mérite ou de la contribution
- Récompense des gens pour leur travail, compétences ou contribution
- Exemple : rémunération plus élevée pour les compétences spécialisées ou les emplois valorisés

Artificial Intelligence, Machine Learning and Deep Learning With Python

└ Artificial Intelligence and Machine Learning 3

└ Examen des défis de l'IA

└ Concepts clés de la justice distributive

Concepts clés de la justice distributive

Égalité

- Principe fondamental de la justice distributive
- Répartition équitable des ressources et des avantages
- Elimination de la discrimination basée sur des特征 personnelles

Besoins

- Prise en compte des besoins réels des individus
- Attribution des ressources en fonction des besoins urgents ou essentiels
- Exemple : priorité d'accès aux soins pour les personnes malades

Mérite

- Distribution des ressources en fonction du mérite ou de la contribution
- Récompense des gens pour leur travail, compétences ou contribution
- Exemple : rémunération plus élevée pour les compétences spécialisées ou les emplois valorisés

La justice distributive est un concept clé en philosophie politique et en éthique, qui traite de la répartition équitable des ressources, des avantages et des charges au sein d'une société. Trois principes fondamentaux sont généralement considérés comme les piliers de la justice distributive : l'égalité, les besoins et le mérite. Ces principes peuvent parfois entrer en conflit et nécessitent un équilibre dans leur application.

XAI (IA explicative)

Rendre les décisions des IA explicables pour les utilisateurs humains.

- Avantages de XAI :
 - Compréhension des décisions des modèles d'IA.
 - Détection des biais et des erreurs dans les prédictions.
 - Renforcement de la confiance des utilisateurs dans les systèmes d'IA.
- Limites de XAI :
 - Certaines IA sont intrinsèquement opaques et difficiles à expliquer.
 - XAI peut ne pas refléter pleinement la complexité du modèle.
 - Explicabilité et performance des modèles délicat à atteindre.
- Exemples commerciaux de XAI :
 - Systèmes de recommandation personnalisée expliquant les suggestions.
 - Modèles d'IA pour la détection de fraude expliquant les classifications.
 - Chatbots explicatifs expliquant le raisonnement derrière leurs réponses.
- Importance d'investir dans XAI :
 - Renforcement de la transparence et de la responsabilité des IA.
 - Réduction des risques liés aux décisions automatisées.
 - Adoption plus large et acceptation sociale de l'IA.

Transformateurs en PNL

Architecture du transformateur

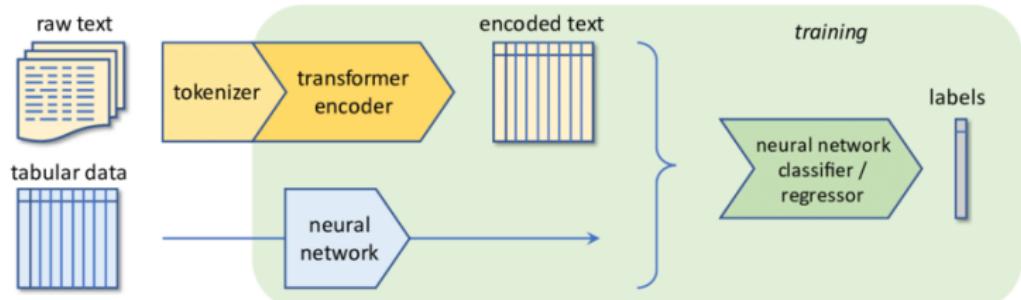
- Les couches d'attention capturent les relations entre les mots.
- Pas besoin de modèles récurrents ou de convolutions.

Formation des transformateurs

- Apprentissage supervisé avec des tâches spécifiques.
- Entraînement sur de grandes quantités de données annotées.

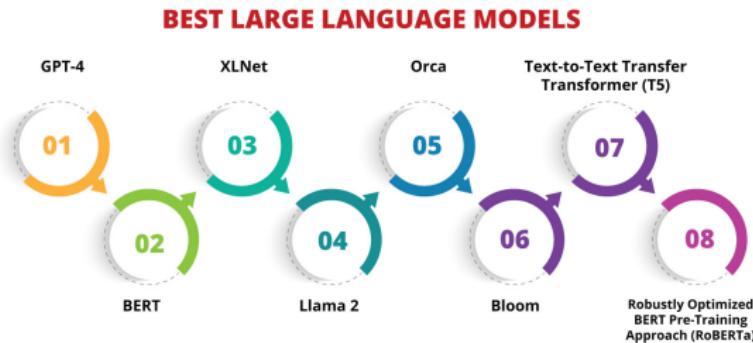
Applications en PNL

- Traduction automatique
- Génération de texte et Compréhension du langage



Grands modèles de langage

- Les grands modèles de langage révolutionnent le PNL.
- Basés sur les transformateurs, des architectures de réseaux np.
- GPT (Generative Pre-trained Transformer) : Apprentissage non supervisé, excellente génération de texte.
- BERT (Bidirectional Encoder Representations from Transformers) : Apprentissage supervisé, diverses tâches de PNL.
- T5 (Text-to-Text Transfer Transformer) : toutes les tâches de PNL.



NLP et transformateurs

Traitement du langage naturel (NLP)

- IA qui se concentre sur l'interaction ordinateurs - langage h.
- Traduction, génération de texte, compréhension du langage

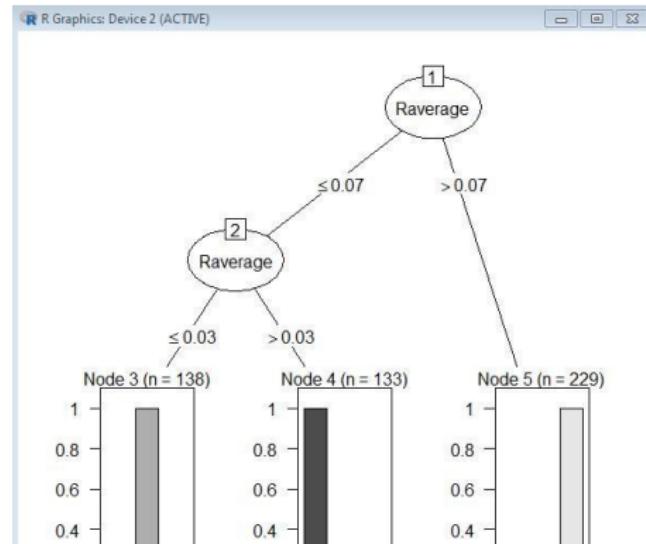
Transformateurs

- Prise en compte des dépendances dans les séquences.
- Élimine le besoin de modèles récurrents ou de convolutions



Présentation de l'algorithme C5.0

- Collecte des données étiquetées pour la classification.
- Construction d'un arbre de décision en sélectionnant la meilleure variable d'entrée à chaque nœud.
- Prédiction des classes pour les exemples non étiquetés.
- Élagage de l'arbre pour améliorer la généralisation.



Artificial Intelligence and Machine Learning 4



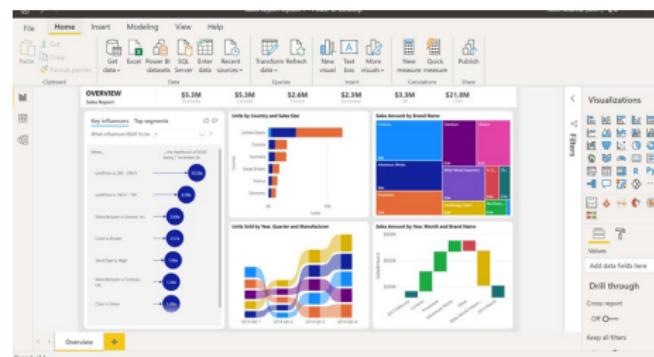
AI / ML

Plan de présentation

- Power BI
 - Analyse d'une variable unique et relations entre les variables
 - Utilisation de visuels d'IA pour poser des questions de simulation
 - Création et partage d'analyses
- Chatbots avec Azure
 - Introduction aux chatbots
 - Terminologie et architecture des chatbots
 - Conception d'un chatbot et amélioration des actions
- Chatbots via Google Dialogflow
 - Blocs de construction Dialogflow
 - Configuration d'un compte Dialogflow
 - Création des intentions
- Machine Learning avec Scikit-Learn
 - Pourquoi utiliser Scikit-learn
 - Apprentissage supervisé ou non supervisé
 - Clustering K-means
 - Analyse en composantes principales (ACP)

Power BI : Business Intelligence pour tous

- Power BI, plateforme de BI développée par Microsoft.
- Importer, visualiser et analyser les data interactivement.
- Data Analyst, Responsable d'entreprise ou utilisateurs.
- Transformer et modéliser les data.
- Créer des tableaux de bord interactifs et attrayants pour partager vos analyses..



Analyse d'une variable unique

- Une fois les données importées, vous pouvez effectuer des analyses sur une variable unique.
- Représenter graphiquement la distribution de la variable.
- Créer des histogrammes, diagrammes en boîte, des diagrammes à barres, etc.
- Ces visualisations vous aident à comprendre les tendances, les valeurs aberrantes et les diverses caractéristiques de la variable.

Mesure des relations entre les variables

- Les diagrammes de dispersion (scatter plots).
- Analyse statistique (coefficients de corrélation) pour quantifier la force et la direction des relations entre les variables.
- Comprendre les dépendances entre les différentes variables et à identifier les facteurs qui influencent vos données.
- En explorant les relations entre les variables, vous pouvez prendre des décisions éclairées et développer des modèles prédictifs plus précis.

Utilisation de visuels d'IA pour poser des questions de simulation

- Power BI intègre des fonctionnalités d'IA pour vous permettre de poser des questions de simulation à vos données.
- Vous pouvez utiliser des visuels d'IA, tels que les cartes de tendance temporelle prédictive, pour obtenir des prévisions et des simulations basées sur vos données historiques.
- Ces visuels vous aident à explorer différents scénarios et à évaluer l'impact de différentes variables sur vos résultats.
- Par exemple, vous pouvez simuler l'effet d'une augmentation des prix sur les ventes ou visualiser l'évolution attendue d'une métrique au fil du temps.
- En utilisant ces visuels d'IA, vous pouvez prendre des décisions plus éclairées et anticiper les résultats potentiels de vos actions.

Modèle de série chronologique

Qu'est-ce qu'une série chronologique ?

- Une série chronologique est une collection de données organisées selon un ordre temporel.
- Chaque observation est associée à une date ou un instant précis.
- Les séries chronologiques peuvent être univariées ou multivariées.

Modélisation de séries chronologiques

- L'objectif de la modélisation de séries chronologiques est de comprendre les motifs, les tendances et les comportements dans les données temporelles.
- Les modèles de séries chronologiques utilisent des équations mathématiques pour décrire et prévoir les données.
- Les techniques de modélisation: ARIMA, modèles de lissage exponentiel, les réseaux de neurones récurrents (RNN) et les modèles basés sur les états cachés.

Modèles mathématiques de série chronologique

Composants d'une série chronologique

- Tendance : évolution à long terme de la série
- Saisonnalité : variations périodiques (jour, mois, trimestre, etc.)
- Cycle : fluctuations à moyen terme
- Résidu : composante aléatoire non expliquée

Modèle général : $Y_t = f(t) + s(t) + c(t) + \epsilon_t$, où :

- Y_t : valeur de la série au temps t
- $f(t)$: composante de tendance
- $s(t)$: composante saisonnière
- $c(t)$: composante cyclique

Exemples de modèles spécifiques

- Modèle autorégressif (AR)
- Modèle à moyenne mobile (MA)
- Modèle ARIMA (autorégression intégrée à moyenne mobile)

Artificial Intelligence, Machine Learning and Deep Learning With Python

└ AI and ML 4 : Cas Pratique

└ Power BI

└ Modèles mathématiques de série chronologique

Modèles mathématiques de série chronologique

Composants d'une série chronologique

- Tendance : évolution à long terme de la série
- Saisonalité : variations périodiques (jour, mois, trimestre, etc.)
- Cycle : fluctuations à moyen terme
- Résidu : composante aléatoire non expliquée

Modèle général : $Y_t = f(t) + s(t) + c(t) + \epsilon_t$, où :

- Y_t : valeur de la série au temps t
- $f(t)$: composante de tendance
- $s(t)$: composante saisonnière
- $c(t)$: composante cyclique

Exemples de modèles spécifiques

- Modèle autoregressif (AR)
- Modèle à moyenne mobile (MA)
- Modèle ARIMA (autorégression intégrée à moyenne mobile)

Les modèles mathématiques de série chronologique sont utilisés pour analyser et prédire le comportement des données temporelles. Ils se basent sur des équations qui décrivent la structure et la dynamique des données dans le temps. Cette formalisation mathématique permet d'étudier les différents composants d'une série chronologique et d'appliquer des méthodes statistiques avancées pour la modélisation et la prévision.

Mise en œuvre de l'analyse de séries chronologiques

Visualisation des séries chronologiques

- Importez vos données de séries chronologiques dans Power BI.
- Utilisez des visualisations appropriées pour afficher les tendances, les motifs saisonniers et les variations.

Prévision des séries chronologiques

- Utilisez les équations mathématiques des modèles de séries chronologiques pour générer des prévisions.
- Les modèles ARIMA, par exemple, sont basés sur les équations AR (AutoRegressive), MA (Moving Average) et I (Integrated) pour capturer les motifs, les tendances et les erreurs.

Analyse des séries chronologiques

- Utilisez les fonctions DAX pour calculs et analyses des séries chrono.
- Les moyennes mobiles, les décompositions saisonnières et les tests stat

Python 1 : Initiation



{OOP}

BeautifulSoup



python

Plan de présentation

- Apprentissage automatique avec Python
- Découverte de modèles avec des règles d'association
- Réseaux de neurones en Python
- Choix d'un réseau de neurones
- Les éléments constitutifs des réseaux de neurones
- Construction de votre réseau
- Formation de votre réseau
- Création d'un affichage de segments classificateur

Python



Plateforme de distribution des IDE et Librairies Python



Environnement Python



Platform

IDE

Library



Outils de développement - Code Python



Matplotlib
Visualisation et affichage graphique



Pandas
Manipulation de données



NumPy
Manipulation de tableaux multidimensionnels



Scikit-learn
Algorithme pour Machine Learning

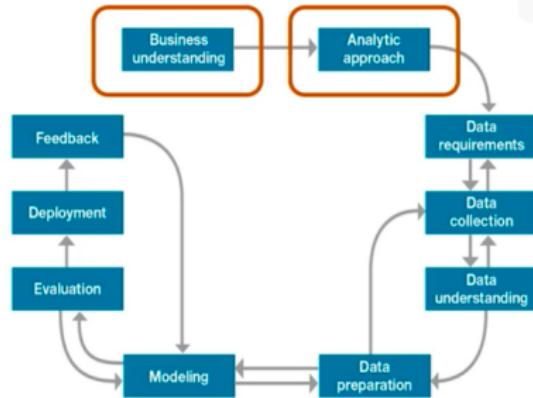


TensorFlow
Machine Learning et CNN

Librairies
Appelés en début de programme pour utiliser des fonctions prédéfinies

From Understanding to Approach

Supposons que nous souhaitons automatiser le processus de détermination de la cuisine d'un plat ou d'une recette donnée.



Business understanding

- What is the problem that you are trying to solve?*



Analytic approach

- How can you use data to answer the question?*

Cuisines



Figure: Atayef and Ma'mul -
Balha's Pastry



Figure: Avgolemono Soup and
Grilled Chicken



Figure: Bacon and cheese



Figure: Baguette french toast

Business understanding : Automating Cuisine Identification

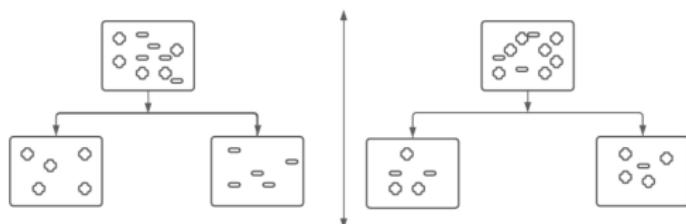
- Definition of the problem and its importance.
 - Identifying the key stakeholders and their objectives.
 - Gathering domain knowledge and understanding business requirements.
- 2- En regardant le diagramme, nous repérons deux caractéristiques remarquables de la méthodologie de la DS. Lesquelles?
- 3- Pouvons-nous prédire la cuisine d'un plat donné en utilisant uniquement le nom du plat ?
- 4- Et en utilisant uniquement l'apparence ? Est-il possible de prédire la cuisine d'un plat donné ?

Automatiser le processus de détermination de la cuisine d'un plat donné n'est donc pas un problème simple.

- 5- Que dire de la détermination de la cuisine d'un plat en fonction de ses ingrédients ?

Analytic Approach: Automating Cuisine Identification

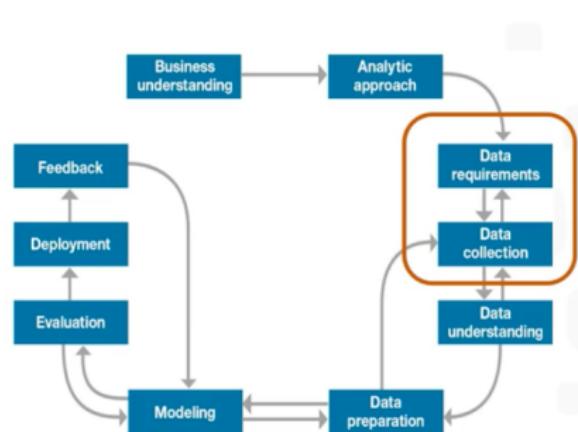
- Questions to consider:
 - ① What are the available data sources?
 - ② Which features can be extracted from the data?
 - ③ Are there existing models or algorithms that can be leveraged?
 - ④ How can the accuracy of the predictions be evaluated?
- Informative Decision Tree:



Conclusion: The goal of this stage is expressing the problem in the context of statistical and ml techniques

Requirements

From Requirements to Collection



Data Requirements

- What are data requirements?

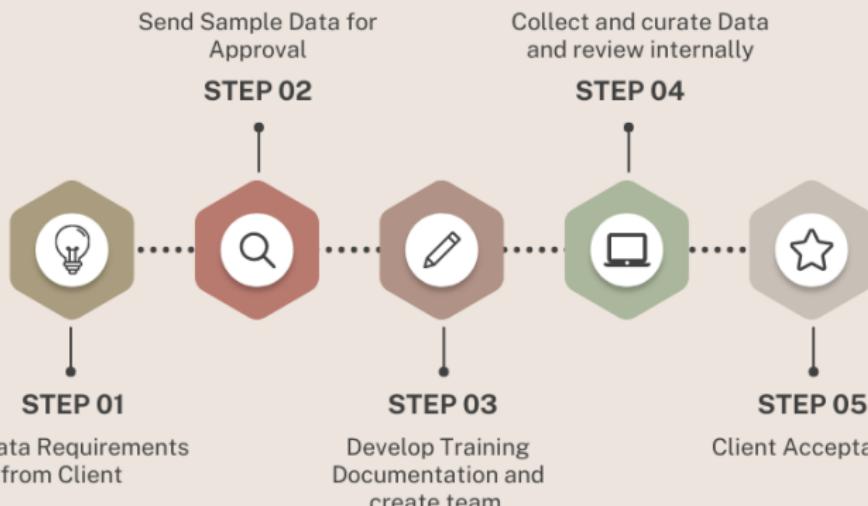


Data Collection

- What occurs during data collection?

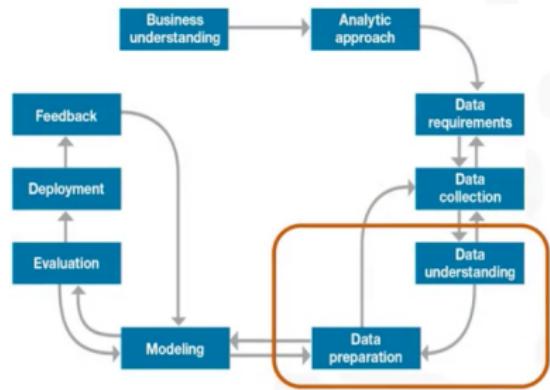
Data collection for Machine Learning

Data Collection Workflow



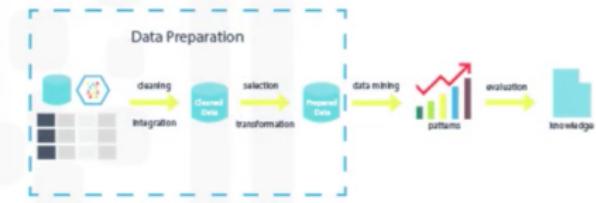
Data understanding

From Understanding to Preparation



Data understanding

- What does it mean to "prepare" or "clean" data?



Data preparation

- What are ways in which data is prepared?

The data we need to answer the question, **can we automate the process of determining the cuisine of a given recipe?**, is readily available. Researcher **Yong-Yeol Ahn** scraped tens of thousands of food recipes (cuisines and ingredients).

Data understanding

Import libraries and download Data

```
1 import pandas as pd # import library to read data
2 pd.set_option('display.max_columns', None)
3 import numpy as np # import numpy library
4 import re # import library for regular expression
5 recipes = pd.read_csv("https://cf-courses-data.s3.us.
6   cloud-object-storage.appdomain.cloud/
7   IBMDeveloperSkillsNetwork-DS0103EN-SkillsNetwork/
8   labs/Module%202/recipes.csv") # 30 s
```

```
1 recipes.head()
2 recipes.shape
3 ingredients = list(recipes.columns.values)
```

Data understanding

Description du jeu de données

Notre jeu de données est composé de 57 691 recettes. Chaque ligne représente une recette, et pour chaque recette, la cuisine correspondante est documentée, ainsi que la présence de 384 ingrédients dans la recette. Nous savons qu'une recette de sushi de base comprend les ingrédients suivants :

- Riz
- Sauce soja
- Wasabi
- Du poisson/légumes au choix

Data Preparation

Frequency Table

This stage involves exploring the data further and making sure that it is in the right format for the machine learning algorithm that we selected in the analytic approach stage.

```
recipes[ "country" ].value_counts()
```

By looking at the table, we can make the following observations:

- Cuisine column is labeled as Country, which is inaccurate.
- Cuisine names are not consistent as not all of them start with an uppercase first letter.
- Some cuisines are duplicated as variation of the country name, such as Vietnam and Vietnamese.
- Some cuisines have very few recipes.

Data Preparation

Let's fix these problems

- ① Fix the name of the column showing the cuisine

```
1 column_names = recipes.columns.values
2 column_names[0] = "cuisine"
3 recipes.columns = column_names
```

- ② Make all the cuisine names lowercase

```
1 recipes["cuisine"] = recipes["cuisine"].str.lower()
()
```

- ③ Make the cuisine names consistent.

```
recipes.loc[recipes["cuisine"] == "austria", "cuisine"]
    ] = "austrian"
recipes.loc[recipes["cuisine"] == "belgium", "cuisine"]
    ] = "belgian"
recipes.loc[recipes["cuisine"] == "china", "cuisine"]
    = "chinese"
```

Data Preparation

Let's fix these problems

- Convert all Yes's to 1's and the No's to 0's

```
1 recipes = recipes.replace(to_replace="Yes", value  
    =1)  
2 recipes = recipes.replace(to_replace="No", value  
    =0)
```

- Recipes that contain rice and soy and wasabi and seaweed

```
1 check_recipes = recipes.loc[(recipes["rice"] == 1)  
    & (recipes["soy_sauce"] == 1) & (recipes["  
        wasabi"] == 1) & (recipes["seaweed"] == 1)  
    ]  
2  
3 check_recipes
```

Based on the results of the above code, can we classify all recipes that contain rice and soy and wasabi and seaweed as Japanese recipes? Why?

Data Preparation

Learn the data better

- ➊ Let's count the ingredients across all recipes.

```
1 ing = recipes.iloc[:, 1: ].sum(axis=0) #sum column
2
3 # define each column as a pandas series
4 ingredient = pd.Series(ing.index.values, index =
5     np.arange(len(ing)))
6 count = pd.Series(list(ing), index = np.arange(len(
7     ing)))
8
9 # create the dataframe
10 ing_df = pd.DataFrame(dict(ingredient = ingredient
11     , count = count))
12 ing_df = ing_df[["ingredient", "count"]]
13 print(ing_df.to_string())
```

Data Preparation

Learn the data better

- ① What are the 3 most popular ingredients?

```
1 ing_df.sort_values(["count"], ascending=False,  
2     inplace=True)  
2 ing_df.reset_index(inplace=True, drop=True)  
3  
4 print(ing_df)
```

- ② However, note that there is a problem with the above table. There are 40,000 American recipes in our dataset, which means that the data is biased towards American ingredients. **Let's create a profile for each cuisine.**

```
1 cuisines = recipes.groupby("cuisine").mean()  
2 cuisines.head()
```

Data Preparation

Learn the data better

- ① What are the 3 most popular ingredients?

```
1 ing_df.sort_values(["count"], ascending=False,  
                     inplace=True)  
2 ing_df.reset_index(inplace=True, drop=True)  
3  
4 print(ing_df)
```

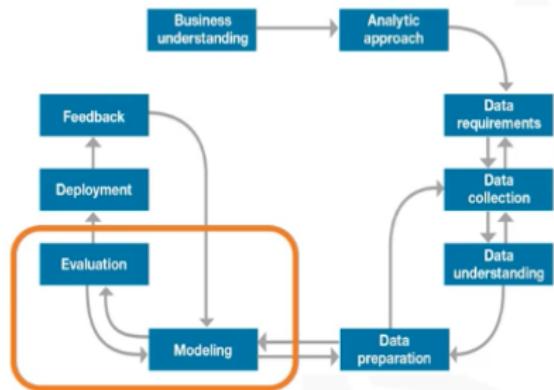
Data Preparation

Learn the data better: Top four ingredients in each cuisine

```
1 num_ingredients = 4 # number of top ingredients
2 # function that prints the top ingredients for each c
3 def print_top_ingredients(row):
4     print(row.name.upper())
5     row_sorted = row.sort_values(ascending=False)
6         *100
7     top_ingredients = list(row_sorted.index.values
8         )[0:num_ingredients]
9     row_sorted = list(row_sorted)[0:
10         num_ingredients]
11
12     for ind, ingredient in enumerate(
13         top_ingredients):
14         print("%s (%d%%)" % (ingredient,
15             row_sorted[ind]), end=' ')
16
17 create_cuisines_profiles = cuisines.apply(
18     print_top_ingredients, axis=1)
```

Data Modelling

From Modeling to Evaluation



Modeling

- In what way can the data be visualized to get to the answer that is required?*



Evaluation

- Does the model used really answer the initial question or does it need to be adjusted?*

Download and install dependencies to build decision trees

```
# import decision trees scikit-learn libraries
%matplotlib inline
from sklearn import tree
from sklearn.metrics import accuracy_score,
    confusion_matrix

import matplotlib.pyplot as plt

# If running locally, you can try using the
# graphviz library but we'll use sklearn's
# plot_tree method
# !conda install python-graphviz --yes
# from sklearn.tree import export_graphviz

import itertools
```

Creating Decision Trees for Asian and Indian Cuisines

- In this project, we are focusing on creating a decision tree for recipes from Asian (Korean, Japanese, Chinese, Thai) and Indian cuisines.
- The reason for this approach is that the decision tree algorithm may not perform well when the dataset is biased towards one cuisine, such as American cuisines.
- Instead of excluding the American cuisines from our analysis, we have chosen to build decision trees for different subsets of the data.
- By focusing on specific subsets, we can better capture the unique characteristics and features of each cuisine, leading to more accurate and meaningful decision trees.

Building Decision Trees for Asian and Indian Cuisines

```
# select subset of cuisines
asian_indian_recipes = recipes[recipes.cuisine.isin([
    "korean",
    "japanese",
    "chinese",
    "thai",
    "indian"])]  
  
cuisines = asian_indian_recipes["cuisine"]
ingredients = asian_indian_recipes.iloc[:,1:]  
  
bamboo_tree = tree.DecisionTreeClassifier(max_depth=3)
bamboo_tree.fit(ingredients, cuisines)  
  
print("Decision tree model saved to bamboo_tree!")
```

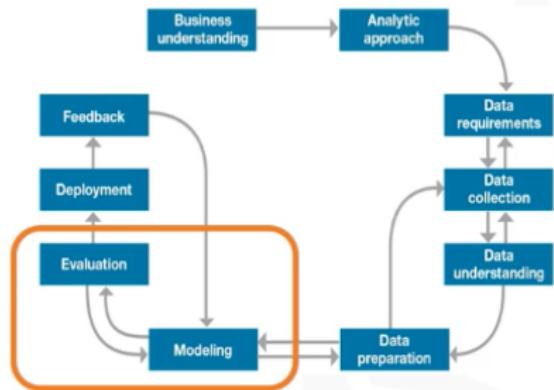
Visualizing Decision Tree for Asian and Indian Cuisines

```
1 plt.figure(figsize=(40,20))
2 _ = tree.plot_tree(bamboo_tree,
3 feature_names = list(ingredients.columns.values),
4 class_names=np.unique(cuisines),filled=True, node_ids=
5     True, impurity=False, label="all", fontsize=20,
6     rounded = True)
7 plt.show()
```

- The decision tree provides insights into the distinguishing features of different cuisines.
- Based on the tree, we can make predictions about the cuisine of a recipe based on its ingredients.
- For example, if a recipe contains cumin and fish but no yogurt, it is most likely a Thai recipe.
- On the other hand, if a recipe contains cumin but no fish and no soy sauce, it is most likely an Indian recipe.

Model Evaluation

From Modeling to Evaluation



Modeling

- In what way can the data be visualized to get to the answer that is required?*



Evaluation

- Does the model used really answer the initial question or does it need to be adjusted?*

Evaluating the Model for Asian and Indian Cuisines

- To evaluate our model for Asian and Indian cuisines, we will split our dataset into a training set and a test set.
- This will allow us to assess the performance of the decision tree model and compare the predicted cuisines to the actual cuisines.
- Let's start by creating a new dataframe called "bamboo" that contains only the data pertaining to the Asian and Indian cuisines.

```
bamboo = recipes[recipes.cuisine.isin(["korean", "japanese", "chinese", "thai", "indian"])]  
bamboo["cuisine"].value_counts()
```

- By creating this new dataframe, we can focus our analysis on the specific cuisines of interest.
- Next, we will proceed to split the dataset into a training set and a test set for model evaluation.

Evaluating the Model for Asian and Indian Cuisines

Split data into training and test set

- Let's remove 30 recipes from each cuisine to use as the test set

```
1 sample_n = 30 # set sample size
2 random.seed(1234) # set random seed
3 bamboo_test = bamboo.groupby("cuisine", group_keys
4     =False).apply(lambda x: x.sample(sample_n))
5 # ingredients
6 bamboo_test_ingredients = bamboo_test.iloc[:,1:]
7 # corresponding cuisines or labels
bamboo_test_cuisines = bamboo_test["cuisine"]
```

- Let's create the training set by removing the test set

```
1 bamboo_test_index = bamboo.index.isin(bamboo_test.
    index)
2 bamboo_train = bamboo[~bamboo_test_index]
3 bamboo_train_ingredients = bamboo_train.iloc[:,1:]
4 bamboo_train_cuisines = bamboo_train["cuisine"]
```



Evaluating the Model for Asian and Indian Cuisines

Let's build the decision tree using the training set

- Model building

```
1 bamboo_train_tree = tree.DecisionTreeClassifier(  
    max_depth=15)  
2 bamboo_train_tree.fit(bamboo_train_ingredients ,  
    bamboo_train_cuisines)
```

- Let's plot the decision tree and explore it.

```
1 plt.figure(figsize=(40,20)) # customize according  
    to the size of your tree  
2 _ = tree.plot_tree(bamboo_train_tree ,  
3 feature_names=list(bamboo_train_ingredients .  
    columns.values) ,  
4 class_names=np.unique(bamboo_train_cuisines) ,  
    filled=True , node_ids=True , impurity=False ,  
    label="all" , fontsize=10 , rounded= True)  
5 plt.show()
```

Evaluating the Model for Asian and Indian Cuisines

Let's test our model on the test data.

- Prediction of test data sample

```
1 bamboo_pred_cuisines = bamboo_train_tree.predict(  
    bamboo_test_ingredients)
```

- How well the decision tree is able to correctly classify the recipes?

```
1 test_cuisines = np.unique(bamboo_test_cuisines)  
2 bamboo_confusion_matrix = confusion_matrix(  
    bamboo_test_cuisines, bamboo_pred_cuisines,  
    labels = test_cuisines)  
3 title = 'Bamboo Confusion Matrix'  
4 cmap = plt.cm.Blues  
5 plt.figure(figsize=(8, 6))  
6 bamboo_confusion_matrix =(bamboo_confusion_matrix.  
    astype('float') / bamboo_confusion_matrix.sum(  
    axis=1)[:, np.newaxis]) * 100  
7 plt.imshow(bamboo_confusion_matrix, interpolation=  
    'nearest', cmap=cmap)
```



Evaluating the Model for Asian and Indian Cuisines

Analyse de la Matrice de Confusion

	Chinese	Indian	Korean	Thai	Total
Chinese	60%	0%	37%	3%	100%
Indian	3%	77%	13%	7%	100%
Korean	-	-	-	-	100%
Thai	-	-	-	-	100%

- 60% des recettes chinoises dans `bamboo_test` ont été correctement classées par notre arbre de décision.
- 37% des recettes chinoises ont été mal classées comme coréennes et 3% comme indiennes.
- 77% des recettes indiennes dans `bamboo_test` ont été correctement classées.
- 13% des recettes indiennes ont été mal classées comme coréennes et 7% comme thaïlandaises.

Important Considerations

- Please note that because decision trees are created using random sampling of the datapoints in the training set, then you may not get the same results every time you create the decision tree even using the same training set.
- The performance should still be comparable though! So don't worry if you get slightly different numbers in your confusion matrix than the ones shown above.

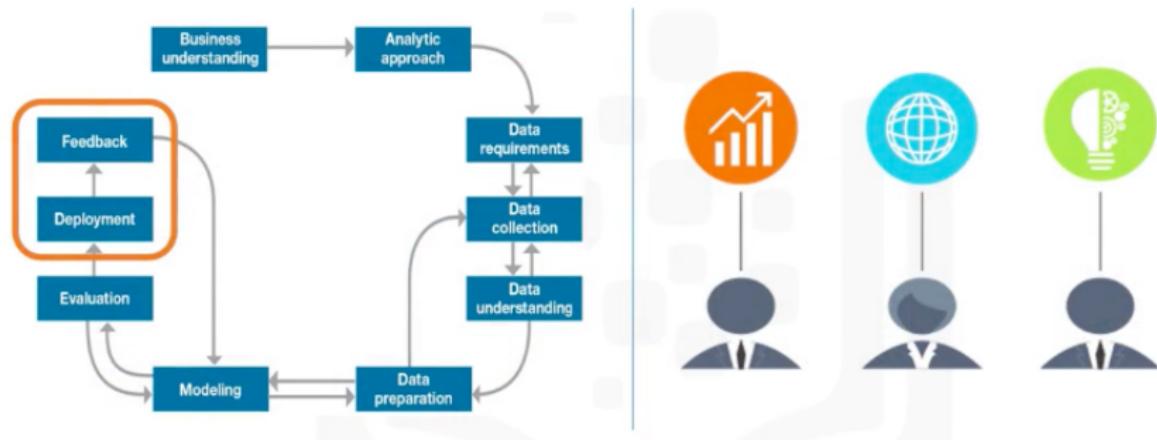
Confusion Matrix Analysis

Using the reference confusion matrix:

- ① How many Japanese recipes were correctly classified by our decision tree?
- ② How many Korean recipes were misclassified as Japanese?
- ③ What cuisine has the least number of recipes correctly classified by the decision tree using the reference confusion matrix?

Model deployment

From Deployment to Feedback



Python for Machine Learning 2: Advanced



Introduction aux chatbots

- Présentation des chatbots et de leur utilité dans les applications pratiques
- Compréhension des composants clés d'un chatbot (interfaces, traitements, etc.)
- Exemples d'utilisation de chatbots dans différents domaines (service client, support technique, etc.)
- Introduction aux plateformes de développement de chatbots (Dialogflow, Microsoft Bot Framework, etc.)

Conception et amélioration des chatbots

- Méthodologies de conception de chatbots conversationnels
- Utilisation des techniques de NLP pour améliorer la compréhension et la génération de réponses
- Optimisation de l'expérience utilisateur des chatbots
- Méthodes d'évaluation et d'amélioration des performances des chatbots

Introduction à Scikit-Learn

Qu'est-ce que Scikit-Learn?



- Scikit-Learn est une bibliothèque de Machine Learning en Python.
- Elle fournit des outils simples et efficaces pour le prétraitement des données, la construction de modèles, l'évaluation des performances et bien plus encore.
- Scikit-Learn est open-source, largement utilisé dans l'industrie et la recherche, et bénéficie d'une grande communauté de développeurs.

Principales fonctionnalités de Scikit-Learn

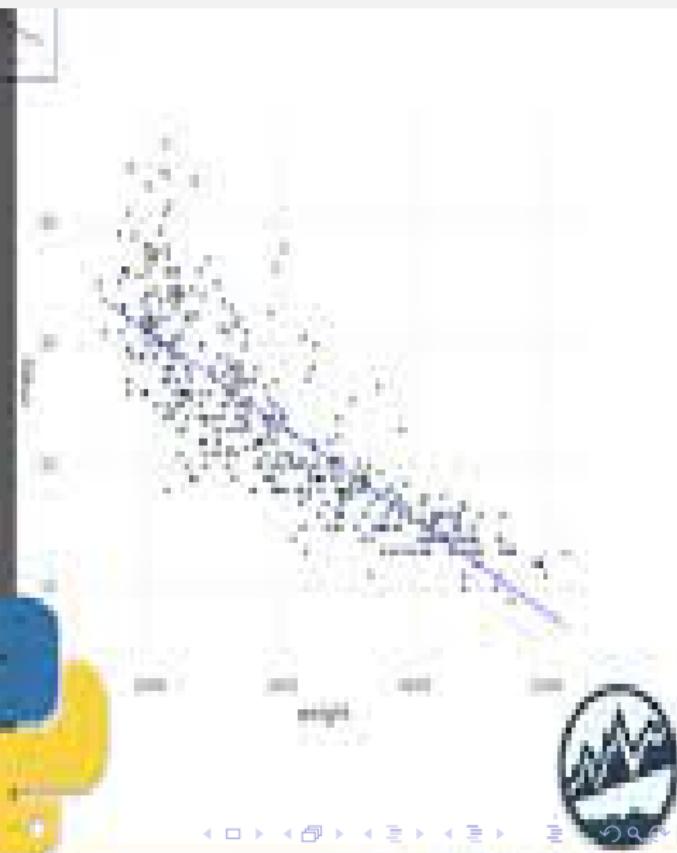
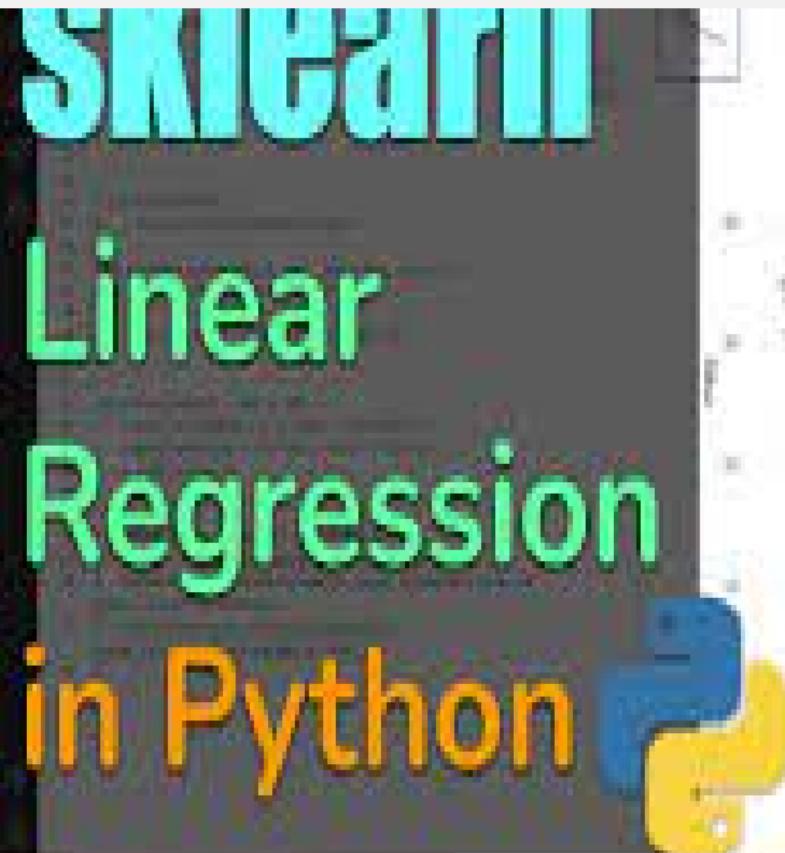
Scikit-Learn offre une large gamme de fonctionnalités le ML :

- Prétraitement des données : normalisation, encodage, réduction de dimension ...
- Apprentissage supervisé : classification, régression, etc.
- Apprentissage non supervisé : regroupement, détection d'anomalies, etc.
- Évaluation des performances : métriques, validation croisée, etc.
- Sélection de modèles : recherche d'hyperparamètres, validation, etc.
- Intégration facile avec d'autres bibliothèques Python telles que NumPy et Pandas.

Régression linéaire et logistique avec Scikit-Learn

- Utilisation de Scikit-Learn pour effectuer des tâches de régression linéaire et logistique
- Méthodes d'évaluation des modèles de régression et de classification
- Optimisation des hyperparamètres des modèles avec Scikit-Learn

Utilisation de Scikit-Learn pour la régression



Régression linéaire

Introduction

- La régression linéaire est une technique couramment utilisée pour modéliser la relation entre une variable dépendante continue et une ou plusieurs variables indépendantes.
- L'objectif de la régression linéaire est d'ajuster une ligne droite (ou un hyperplan dans le cas de variables multiples) aux données observées afin de pouvoir prédire la valeur de la variable dépendante pour de nouvelles observations.

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_n x_n + \epsilon$$

- y : Variable dépendante à prédire
- x_1, x_2, \dots, x_n : Variables indépendantes (caractéristiques)
- $\beta_0, \beta_1, \beta_2, \dots, \beta_n$: Coefficients de régression à estimer
- ϵ : Terme d'erreur aléatoire

Forme matricielle de la régression linéaire

La forme matricielle de la régression linéaire permet de représenter les calculs de manière plus concise et efficace. Voici la formulation en utilisant des matrices :

$$Y = X\beta + \epsilon$$

- Y est un vecteur de taille $n \times 1$ contenant les valeurs de la variable dépendante,
- X est une matrice de taille $n \times (p + 1)$ contenant les valeurs des variables indépendantes, y compris une colonne de 1 pour le terme d'interception,
- β est un vecteur de taille $(p + 1) \times 1$ contenant les coefficients de régression,
- ϵ est un vecteur de taille $n \times 1$ contenant les erreurs résiduelles.

Estimation des coefficients de régression

Pour estimer les coefficients de régression β , nous utilisons la méthode des moindres carrés ordinaires (OLS) qui minimise la somme des carrés des résidus :

$$\hat{\beta} = (X^T X)^{-1} X^T Y$$

où $\hat{\beta}$ est l'estimation des coefficients de régression.

Une fois que nous avons estimé les coefficients, nous pouvons utiliser le modèle pour prédire les valeurs de la variable dépendante pour de nouvelles observations en utilisant la formule :

$$\hat{Y} = X\hat{\beta}$$

Exemple de Régression Linéaire avec Scikit-Learn

```
1 import numpy as np
2 from sklearn.linear_model import LinearRegression
3 X = np.array([[1, 1], [1, 2], [2, 2], [2, 3]])
4 # y = 1 * x_0 + 2 * x_1 + 3
5 y = np.dot(X, np.array([1, 2])) + 3
6 reg = LinearRegression().fit(X, y)
7 reg.score(X, y)
8 reg.coef_
9 reg.intercept_
10 reg.predict(np.array([[3, 5]]))
```

Artificial Intelligence, Machine Learning and Deep Learning With Python

└ Python for Machine Learning 2 : Avancé

└ Machine Learning avec Scikit-Learn

└ Exemple de Régrression Linéaire avec Scikit-Learn

Exemple de Régression Linéaire avec Scikit-Learn

```
import numpy as np
from sklearn.linear_model import LinearRegression
X = np.array([[1, 1], [1, 2], [2, 2], [2, 3]])
# y = 1 + x_0 + 2 * x_1 + 3
y = np.dot(X, np.array([1, 2])) + 3
reg = LinearRegression().fit(X, y)
reg.score(X, y)
reg.coef_
reg.intercept_
reg.predict(np.array([[3, 5]]))
```

- *Ici on a défini une matrice X représentant les caractéristique des individus auquel on a associé pour chaque individu un label y.*
- On a par la suite utilisé le modèle de regression lineaire LinearRegression de scikit learn pour entraîner les données.
- Les coefficients du modèle ont été estimés et le modèle utilisé pour faire des prévisions.

Introduction au clustering

- Clustering : Technique d'apprentissage **non supervisé**
- Objectif : Regrouper les données en clusters homogènes
- Pas de **variables cibles** à prédire
- Se base sur les **similarités** entre les données
- Algorithmes de clustering populaires :
 - K-means
 - Hierarchical Clustering
 - DBSCAN
- Applications :
 - Segmentation de marché
 - Analyse de la clientèle
 - Détection d'anomalies
- Défis :
 - Choix du bon nombre de clusters
 - Sensibilité aux valeurs aberrantes
 - Interprétation des clusters

Clustering avec K-means

- K-means : Algorithme de **clustering** non supervisé
- Objectif : Partitionner les données en **K** clusters
- Étapes de l'algorithme :
 - ① Initialisation aléatoire de **K** centres de clusters
 - ② Assignation de chaque donnée au cluster avec le centre le plus proche
 - ③ Calcul des nouveaux centres de clusters (mean des data des clusters)
 - ④ Répétition des étapes 2 et 3 jusqu'à convergence (centres de clusters stables)
- Critère d'optimalité : Minimiser la **somme des carrés des distances** entre les données et leur centre de cluster

Artificial Intelligence, Machine Learning and Deep Learning With Python

- └ Python for Machine Learning 2 : Avancé
 - └ Machine Learning avec Scikit-Learn
 - └ Clustering avec K-means

Clustering avec K-means

- K-means : Algorithm de **clustering** non supervisé
- Objectif : Partitionner les données en **K** clusters
- Étapes de l'algorithme:
 - ➊ Initialisation aléatoire de **K** centres de clusters
 - ➋ Attribuation de chaque donnée au cluster avec le centre le plus proche
 - ➌ Calcul des nouveaux centres de clusters (mean des data des clusters)
 - ➍ Répétition des étapes 2 et 3 jusqu'à convergence (centres de clusters stables)
- ▼ Critère d'optimalité : Minimiser la **somme des carrés des distances** entre les données et leur centre de cluster

- **Avantages :**

- *Simple à mettre en œuvre*
- *Efficace pour des données sphériques*
- *Rapide pour de grands jeux de données*

- **Inconvénients :**

- *Sensible à l'initialisation des centres*
- *Performances dégradées pour des clusters de tailles/formes différentes*

Clustering avec K-means: Détailles Mathématiques

- Soit un ensemble de n données $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n$ dans un espace \mathbb{R}^d .
- L'objectif est de partitionner ces données en K clusters.
- Soit $\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_K$ les centres des K clusters.
- La fonction de coût à minimiser est la somme des carrés des distances entre chaque donnée et son centre de cluster le plus proche :

$$J = \sum_{i=1}^n \min_{1 \leq j \leq K} \|\mathbf{x}_i - \mathbf{c}_j\|^2 \quad (1)$$

- L'algorithme K-means procède de manière itérative pour minimiser J :
 - ➊ Initialisation aléatoire des centres de clusters $\{\mathbf{c}_j\}_{j=1}^K$.
 - ➋ Répéter jusqu'à convergence :
 - Étape d'assignation : Affecter chaque donnée \mathbf{x}_i au cluster j dont le centre \mathbf{c}_j est le plus proche.
 - Étape de mise à jour : Calculer les nouveaux centres de clusters comme la moyenne des données appartenant à chaque cluster.

Scikit-Learn pour effectuer le clustering avec K-means

Traning and Prediction

```
# Importation des bibliotheques necessaires
import numpy as np
from sklearn.cluster import KMeans
from sklearn.datasets import make_blobs

# Generation d'un jeu de donnees artificiel
X, y = make_blobs(n_samples=500, n_features=2, centers=4, random_state=42)

# Initialisation du modele K-means avec 4 clusters
kmeans = KMeans(n_clusters=4, random_state=42)
# Entrainement du modele sur les donnees
kmeans.fit(X)

# Recuperation des predictions (indices des clusters)
labels = kmeans.labels_
```

Artificial Intelligence, Machine Learning and Deep Learning With Python

└ Python for Machine Learning 2 : Avancé

└ Machine Learning avec Scikit-Learn

└ Scikit-Learn pour effectuer le clustering avec

- Importation des bibliothèques nécessaires :
 - NumPy : manipulations numériques
 - Scikit-Learn : modèle de clustering K-means
- Génération d'un jeu de données artificiel avec 4 clusters :
 - Utilisation de la fonction `make_blobs` de Scikit-Learn
 - Génération de 500 données à 2 caractéristiques
- Initialisation du modèle K-means avec 4 clusters :
 - Paramétrage du nombre de clusters à 4
 - Fixation d'un état aléatoire pour la reproductibilité
- Entraînement du modèle sur les données :
- Récupération des prédictions (indices des clusters)

Scikit-Learn pour effectuer le clustering avec K-means
Training and Prediction

```
# Importation des bibliothèques nécessaires
import numpy as np
from sklearn.cluster import KMeans
from sklearn.datasets import make_blobs

# Génération d'un jeu de données artificiel
X, y = make_blobs(n_samples=500, n_features=2, centers=4,
                   random_state=42)

# Initialisation du modèle K-means avec 4 clusters
kmeans = KMeans(n_clusters=4, random_state=42)
# Entraînement du modèle sur les données
kmeans.fit(X)

# Récupération des prédictions (indices des clusters)
labels = kmeans.labels_
```

Scikit-Learn pour effectuer le clustering avec K-means

Affichage des résultats

```
# Affichage des résultats
import matplotlib.pyplot as plt
plt.figure(figsize=(8, 6))
plt.scatter(X[:, 0], X[:, 1], c=labels, cmap='viridis')
plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.
            cluster_centers_[:, 1], c='red', s=200, alpha=0.5)
plt.title('Clustering K-means')
plt.xlabel('feature 1')
plt.ylabel('feature 2')
plt.show()
```

Artificial Intelligence, Machine Learning and Deep Learning With Python

└ Python for Machine Learning 2 : Avancé

└ Machine Learning avec Scikit-Learn

└ Scikit-Learn pour effectuer le clustering avec

Scikit-Learn pour effectuer le clustering avec K-means
Affichage des résultats

```
# affichage des résultats
import matplotlib.pyplot as plt
plt.figure(figsize=(8, 8))
plt.scatter(X[:, 0], X[:, 1], c=labels, cmap='viridis')
plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1], c='red', s=200, alpha=0.5)
plt.title('Clustering K-means')
plt.xlabel('feature 1')
plt.ylabel('feature 2')
plt.show()
```

Affichage graphique des résultats :

- Utilisation de Matplotlib pour visualiser les clusters
- Représentation des centres de clusters en rouge

Évaluation des résultats de clustering

- **Inertie** (somme des carrés des distances) :

$$\text{Inertie} = \sum_{i=1}^n \min_{1 \leq j \leq K} \|\mathbf{x}_i - \mathbf{c}_j\|^2$$

- Critère d'optimalité de l'algorithme K-means
 - Doit être minimisé pour obtenir des clusters compacts
- **Silhouette Score** :

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

- Mesure la cohésion et la séparation des clusters
 - Valeurs entre -1 et 1, plus proche de 1 meilleur
- **Indice de Calinski-Harabasz** : $\text{CH} = \frac{\text{SSB}/(K-1)}{\text{SSW}/(n-K)}$
 - Rapport entre la variance inter-clusters et intra-clusters
 - Plus la valeur est élevée, meilleur est le clustering

Artificial Intelligence, Machine Learning and Deep Learning With Python

└ Python for Machine Learning 2 : Avancé

└ Machine Learning avec Scikit-Learn

└ Évaluation des résultats de clustering

Évaluation des résultats de clustering

- Inertie (somme des carrés des distances) :

$$\text{Inertie} = \sum_{i=1}^n \min_{1 \leq j \leq K} \|x_i - c_j\|^2$$

- Critère d'optimalité de l'algorithme K-means

- Doit être minimisé pour obtenir des clusters compacts

- Silhouette Score :

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

- Mesure la cohésion et la séparation des clusters

- Valeurs entre -1 et 1, plus proche de 1 meilleur

- Indice de Calinski-Harabasz : $CH = \frac{\sum_{k=1}^K n_k (c_k - \bar{c})^2}{\sum_{k=1}^K \sum_{i \in C_k} (x_i - c_k)^2}$

- Rapport entre la variance inter-clusters et intra-clusters

- Plus la valeur est élevée, meilleur est le clustering

Ces métriques permettent d'évaluer la qualité du clustering effectué par l'algorithme K-means :

- L'inertie mesure la compacité des clusters et doit être minimisée.
- Le Silhouette Score évalue la cohésion et la séparation des clusters.
- L'indice de Calinski-Harabasz compare la variance inter-clusters et intra-clusters.

Vous pouvez utiliser ces métriques pour choisir le nombre optimal de clusters ou comparer différents résultats de clustering.

Évaluation des résultats de clustering précédent

```
# Calcul de l'inertie
inertia = kmeans.inertia_
print(f"Inertie du modèle : {inertia:.2f}")

# Calcul du Silhouette Score
from sklearn.metrics import silhouette_score
silhouette = silhouette_score(X, labels)
print(f"Silhouette Score : {silhouette:.3f}")

# Calcul de l'indice de Calinski-Harabasz
from sklearn.metrics import calinski_harabasz_score
ch_index = calinski_harabasz_score(X, labels)
print(f"Indice de Calinski-Harabasz : {ch_index:.2f}")
```

Artificial Intelligence, Machine Learning and Deep Learning With Python

Python for Machine Learning 2 : Avancé

Machine Learning avec Scikit-Learn

Évaluation des résultats de clustering précédent

Évaluation des résultats de clustering précédent

```
# Calcul de l'inertie
inertia = kmeans.inertia_
print(f"Inertie du modèle : {inertia:.2f}")

# Calcul du Silhouette Score
from sklearn.metrics import silhouette_score
silhouette = silhouette_score(X, labels)
print(f"Silhouette Score : {silhouette:.3f}")

# Calcul de l'indice de Calinski-Harabasz
ch_score = calinski_harabasz_score(X, labels)
print(f"Indice de Calinski-Harabasz : {ch_score:.2f}")
```

- **Inertie** : 327,66
 - Valeur relativement faible, indiquant des clusters compacts
 - Suggère que le modèle a bien capturé la structure des données
- **Silhouette Score** : 0,784 proche de 1
 - Indiquant une bonne cohésion et séparation des clusters
 - Les points sont bien assignés à leurs clusters respectifs
- **Indice de Calinski-Harabasz** : 314,87
 - Valeur élevée, suggérant une bonne séparation des clusters
 - Le modèle a réussi à bien différencier les 4 groupes de données
- **Conclusion** :
 - Le modèle semble avoir bien fonctionné sur ces données
 - Les métriques d'évaluation indiquent des clusters compacts, bien séparés et cohérents
 - $K = 4$ semble approprié pour cette structure de données

Analyse en composantes principales (ACP)

- Introduction à l'analyse en composantes Principales (ACP) comme une technique de réduction de dimension
- Explication du concept de variance expliquée par les composantes principales
- Utilisation de Scikit-Learn pour effectuer l'ACP
- Visualisation des données réduites avec l'ACP

Plan de présentation

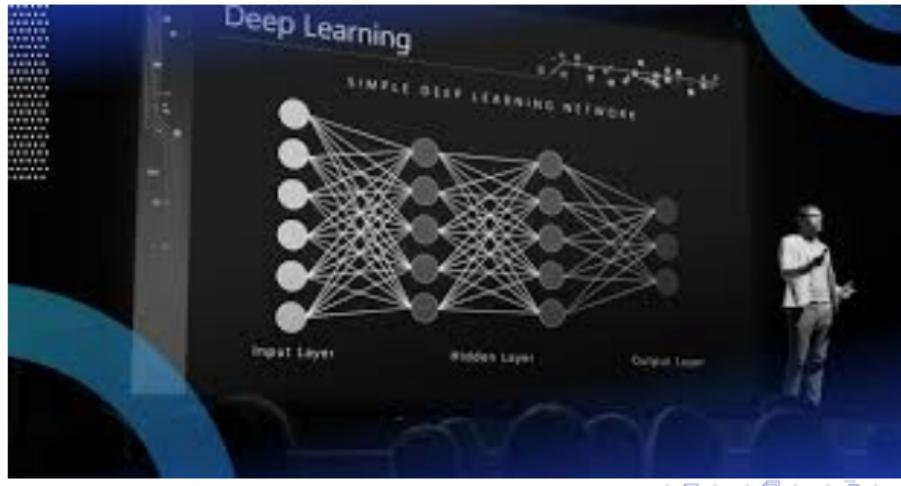
- Régression
- Régression logistique
- Classification des données avec régression logistique
- PNL avancée avec Python pour l'apprentissage automatique
- Révision des bases de la PNL
- Word2Vec
- Doc2Vec
- Réseaux de neurones récurrents
- Comparaison des techniques avancées de PNL sur un problème de ML
- PNL avec Python pour l'apprentissage automatique essentiel
- Formation en PNL de base
- Nettoyage de données supplémentaires
- Vectorisation des données brutes
- Ingénierie des fonctionnalités
- Création de classificateurs d'apprentissage automatique

Deep Learning and Neural Network



Plan de présentation

- Apprentissage profond
- Optimisation et réglage du modèle
- Création d'applications de Deep Learning avec TensorFlow
- Réseaux de neurones
- Réseaux de neurones récurrents
- Réseaux de neurones et réseaux de neurones convolutifs



Introduction aux Réseaux de Neurones

- Les réseaux de neurones sont une technique avancée de Machine Learning.
- Ils sont inspirés par le fonctionnement du cerveau humain, où les neurones interagissent pour effectuer des tâches complexes.
- Les réseaux de neurones sont composés de couches de neurones interconnectés.
- Chaque neurone reçoit des entrées, effectue un calcul et produit une sortie.
- Les sorties des neurones sont transmises aux neurones de la couche suivante, permettant ainsi la propagation de l'information dans le réseau.

Réseaux de neurones: Définition

Biological neuron and Perceptrons

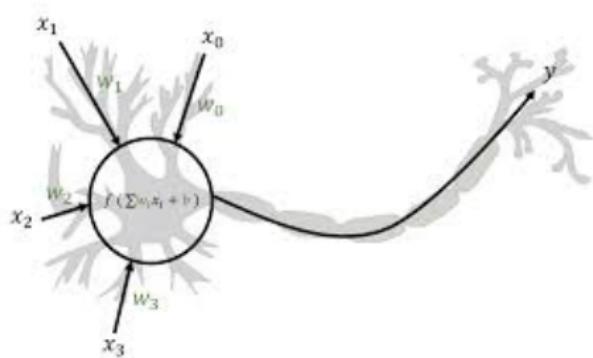
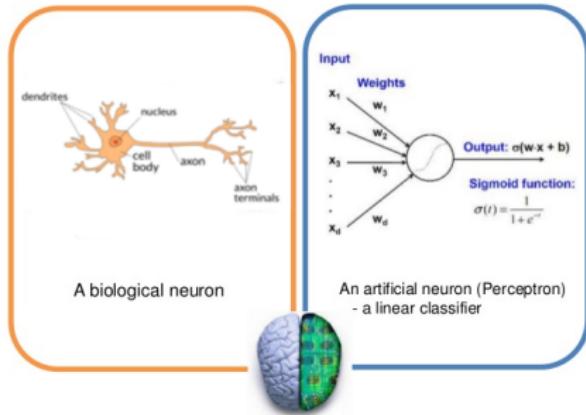


Figure: Modélisation Math du RN

Figure: Réseau de neurone biologique vs Perceptron

Réseaux de neurones: Définition

Neurone artificielle

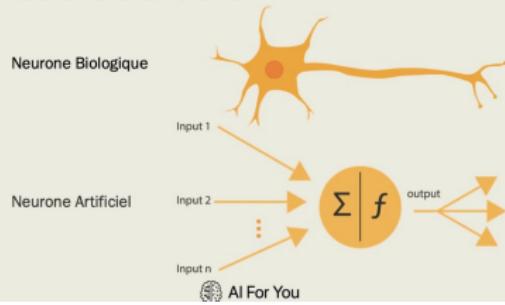


Figure: Neurone biologique et artificiel

Couche d'un neurone

Rendement estimé

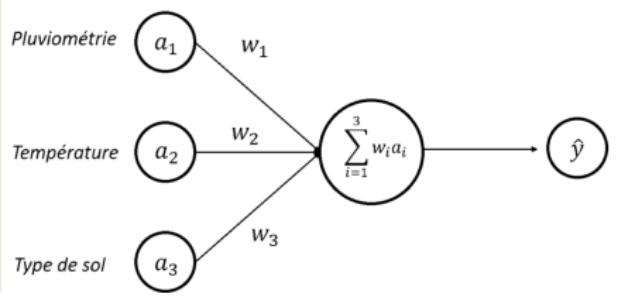


Figure: Architecture d'un perceptron

Applications des Réseaux de Neurones

- Les réseaux de neurones ont des applications dans de nombreux domaines :
 - Vision par ordinateur : Reconnaissance d'images, détection d'objets, segmentation d'images, etc.
 - Traitement du langage naturel : Traduction automatique, compréhension du langage, génération de texte, etc.
 - Reconnaissance de la parole : Reconnaissance vocale, synthèse vocale, etc.
 - Prévisions et prédictions : Prévisions météorologiques, prédictions financières, recommandations personnalisées, etc.
 - Sciences de la santé : Diagnostic médical, analyse d'images médicales, découverte de médicaments, etc.
- Les réseaux de neurones offrent des avantages tels que la capacité à apprendre à partir de données non structurées et à détecter des motifs complexes.

Modélisation du perceptron et comparaison au neurone biologique

Modélisation du perceptron

- Entrées x_1, x_2, \dots, x_n
- Poids synaptiques w_1, w_2, \dots, w_n
- Fonction de sommation $\sum_{i=1}^n w_i x_i$
- Fonction d'activation $f(x)$ (seuil, sigmoïde, tangente hyperbolique)
- Sortie $y = f(\sum_{i=1}^n w_i x_i)$

Comparaison au neurone biologique

- Entrées = signaux provenant des dendrites
- Poids synaptiques = forces de connexion entre les neurones
- Sommation des signaux = intégration des signaux au niveau du corps cellulaire
- Fonction d'activation = potentiel d'action généré par le neurone
- Sortie = axone transmettant le signal à d'autres neurones

Artificial Intelligence, Machine Learning and Deep Learning With Python

└ Python for Machine Learning 2 : Avancé

└ Deep Learning and Neural Network

└ Modélisation du perceptron et comparaison au neurone biologique

Modélisation du perceptron et comparaison au neurone biologique

Modélisation du perceptron

- Entrées x_1, x_2, \dots, x_n
- Poids synaptiques w_1, w_2, \dots, w_n
- Fonction de sommation $\sum_{i=1}^n w_i x_i$
- Fonction d'activation $f(x)$ (ouai, sigmoïde, tangente hyperbolique)
- Sortie $y = f(\sum_{i=1}^n w_i x_i)$

Comparaison au neurone biologique

- Entrées = signaux provenant des dendrites
- Poids synaptiques = forces de connexion entre les neurones
- Sommation des signaux = intégration des signaux au niveau du corps cellulaire
- Fonction d'activation = potentiel d'action générée par le neurone
- Sortie = axone transmettant le signal à d'autres neurones

Le perceptron est un modèle mathématique simple de neurone artificiel, inspiré du fonctionnement du neurone biologique. Bien que simplifié, il constitue la base des réseaux de neurones artificiels, qui sont devenus des outils puissants en apprentissage automatique et en intelligence artificielle.

Limites du perceptron

- Capacité de représentation limitée aux fonctions linéairement séparables
- Incapacité à résoudre des problèmes complexes comme la reconnaissance d'images
- Besoin de réseaux de neurones multicouches pour surmonter ces limitations

Architecture des réseaux de neurones

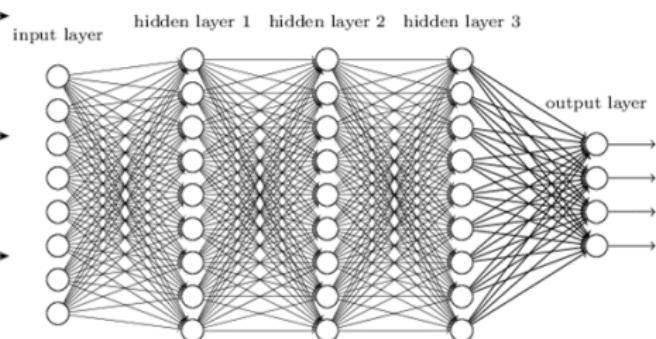
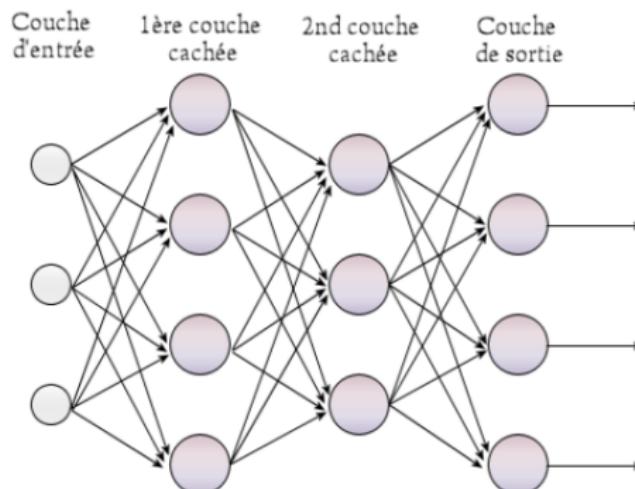


Figure: Perceptron multicouches

Figure: Perceptron à 4 couches

Artificial Intelligence, Machine Learning and Deep Learning With Python

- Python for Machine Learning 2 : Avancé
 - Deep Learning and Neural Network
 - Architecture des réseaux de neurones

Architecture des réseaux de neurones

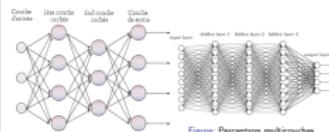


Figure: Perceptron à 4 couches

Figure: Perceptron multicouches

- Couche d'entrée :
 - Reçoit les données brutes d'un objet à classifier
 - Chaque nœud de cette couche correspond à une caractéristique
- Première couche cachée :
 - Transforme les entrées en une représentation abstraite
 - Chaque nœud calcule une combinaison linéaire pondérée des entrées, suivie d'une fonction d'activation non linéaire
- Deuxième couche cachée :
 - Extrait des caractéristiques de plus haut niveau à partir de la représentation intermédiaire de la première couche cachée
 - Applique le même processus de combinaison linéaire et de fonction d'activation non linéaire
- Couche de sortie :
 - Les résultats finaux sont les probabilités d'appartenance à différentes classes

Apprentissage et rétropropagation

1. Propagation avant (Forward propagation)

$$z = \sum_{i=1}^n w_i \cdot x_i + b$$

où w_i sont les poids associés aux entrées x_i , b est le biais du neurone.
Ensuite, une fonction d'activation non linéaire $f(z)$ est appliquée pour obtenir la sortie du neurone :

$$a = f(z)$$

La sortie a est ensuite transmise aux neurones de la couche suivante.

2. Calcul de l'erreur :

Pour un problème de régression, la (MSE) est souvent utilisée :

$$E = \frac{1}{2m} \sum_{i=1}^m (y_i - a_i)^2$$

Apprentissage et rétropropagation

3. Rétropropagation (Backpropagation) : Les gradients sont utilisés pour mettre à jour les poids et les biais lors de l'étape d'optimisation.

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial z_j} \cdot \frac{\partial z_j}{\partial w_{ij}}$$

4. Mise à jour des poids et des biais :

$$w_{ij} \leftarrow w_{ij} - \alpha \cdot \frac{\partial E}{\partial w_{ij}}$$

$$b_j \leftarrow b_j - \alpha \cdot \frac{\partial E}{\partial b_j}$$

5. Itérations : Le processus de propagation avant, de rétropropagation et de mise à jour des poids et des biais est répété pour chaque lot (mini-batch) d'exemples d'entraînement.

Types de réseaux de neurones

- Réseaux de neurones multicouches (MLP) : Utilisés pour la classification et la régression.
- Réseaux de neurones récurrents (RNN) : Traite les données séquentielles avec des boucles de rétroaction.
- Réseaux de neurones convolutifs (CNN) : Analyse les images et les données spatiales.
- Réseaux de neurones LSTM : RNN avec capacité de mémorisation à long terme.
- Réseaux de neurones générateurs adverses (GAN) : Utilisés pour générer des données synthétiques.

Types de réseaux de neurones

Réseaux de neurones multicouches (MLP)

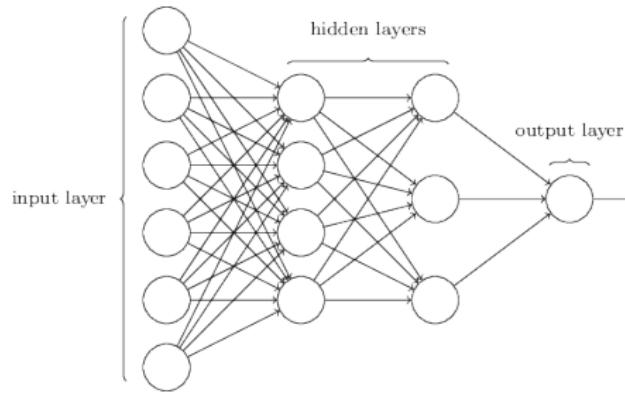


Figure: MLP pour Regression

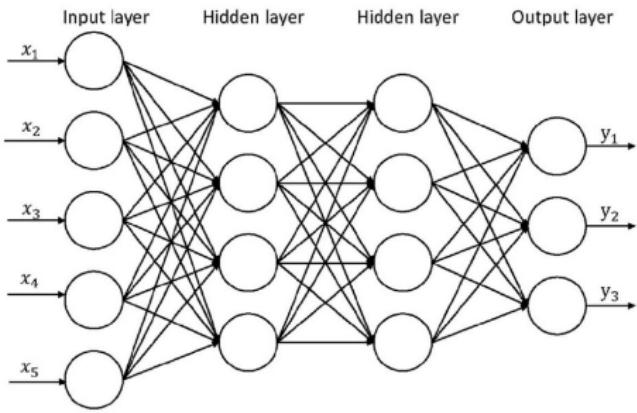


Figure: MLP pour Classification

Types de réseaux de neurones

Réseaux de neurones récurrents (RNN)

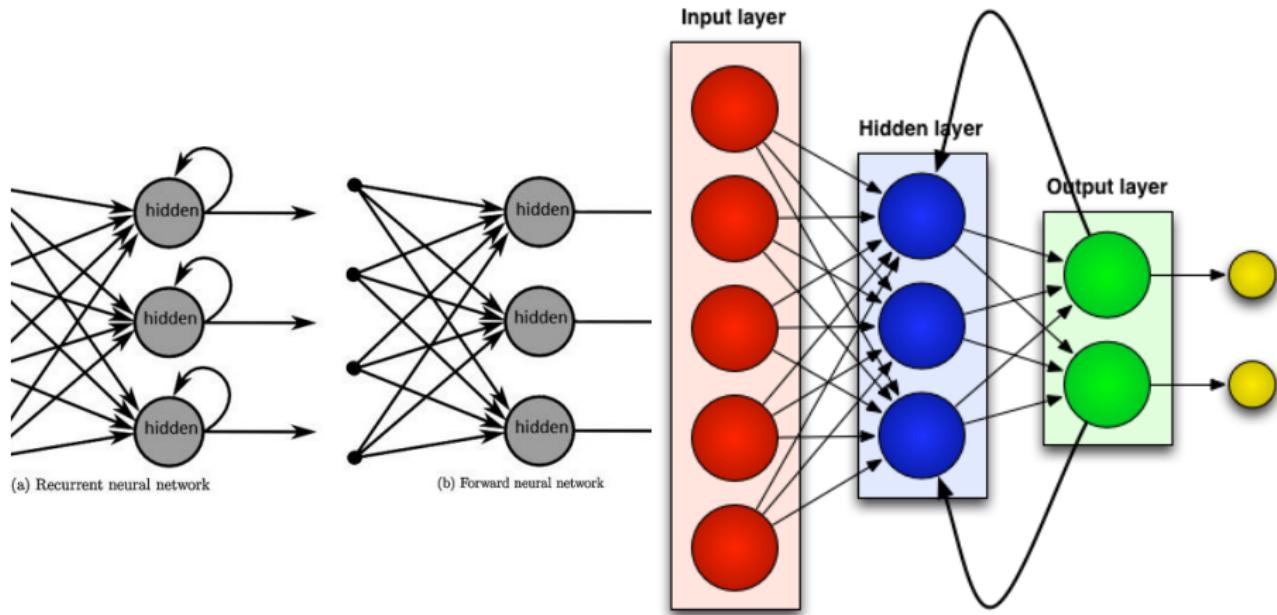


Figure: Traite les données séquentielles avec des boucles de rétroaction.

Types de réseaux de neurones

Réseaux de neurones convolutifs (CNN)

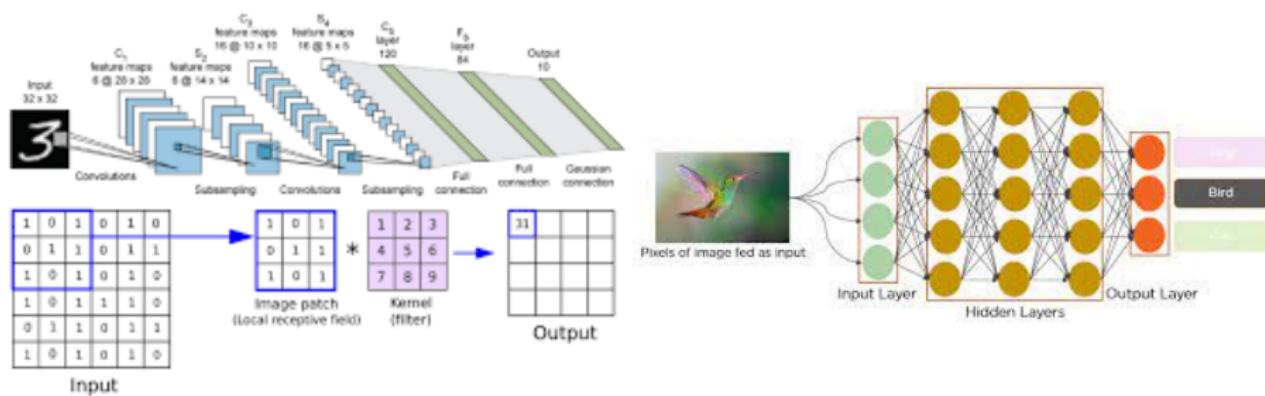


Figure: Analyse les images et les données spatiales.

Types de réseaux de neurones

Réseaux de neurones LSTM et GAN

Long / Short Term Memory (LSTM)

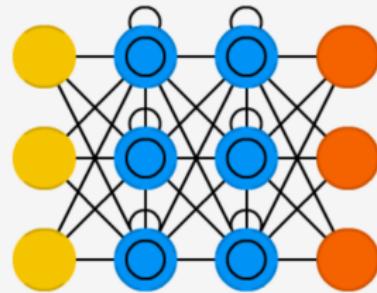


Figure: RNN avec capacité de mémorisation à long terme

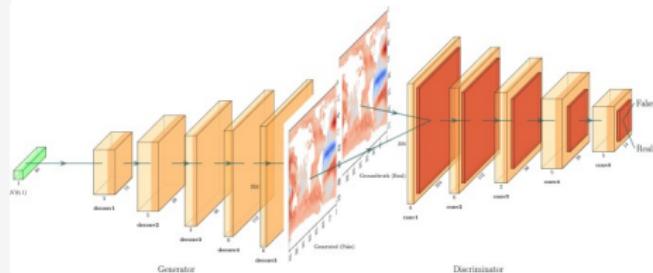


Figure: GAN pour générer des données synthétiques

Entraînement et évaluation des réseaux de neurones

- Préparation des données :
 - Collecte et préparation des données train, val, test.
 - Division des données en ensembles distincts.
- Définition de l'architecture du réseau de neurones :
 - Choix du type de réseau de neurones.
 - Définition des couches, des neurones, des fonctions d'activation, etc.
- Entraînement du réseau :
 - Propagation avant pour obtenir les prédictions.
 - Calcul de l'erreur et rétropropagation pour ajuster les poids et les biais.
 - Mise à jour des poids et des biais avec un algorithme d'optimisation.
 - Répétition pour plusieurs époques.
- Évaluation du modèle :
 - Évaluation des performances sur l'ensemble de validation.
 - Calcul des métriques d'évaluation.
 - Ajustements sur l'architecture ou les paramètres si nécessaire.

Principe du Machine Learning

- Systèmes experts
- Réseaux de neurones
- Logique floue
- Méthodes probabilistes

Systèmes experts

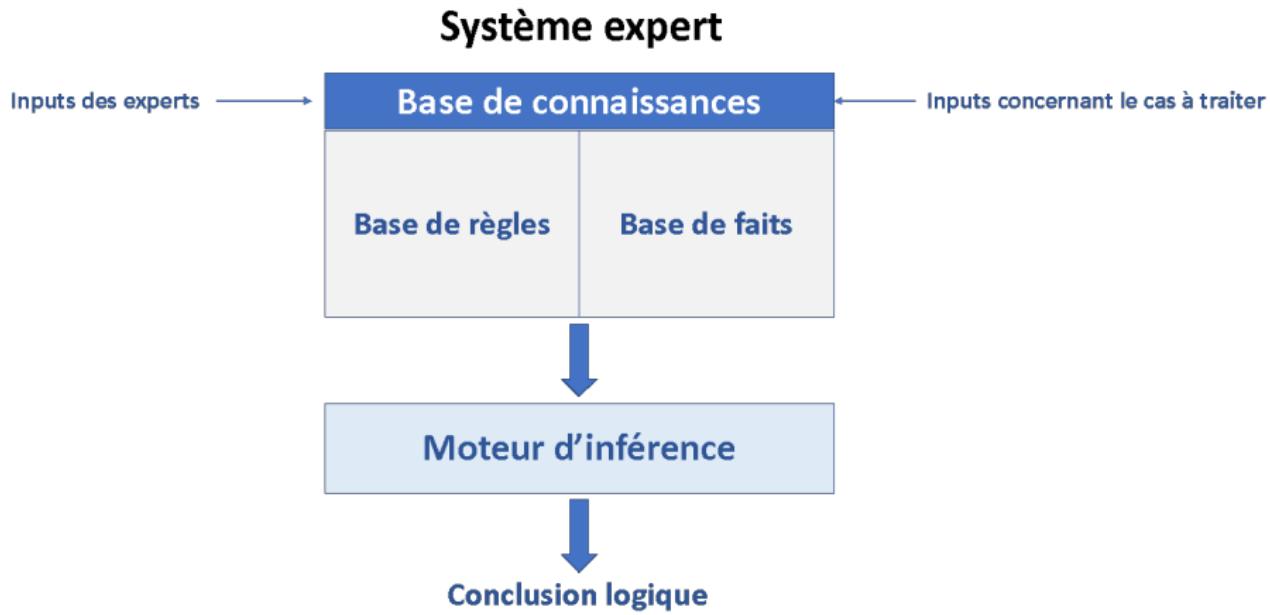
But

- Modélisation (logique) d'un expert humain
- D'une tâche de résolution d'un problème
- Explications sur les raisonnements

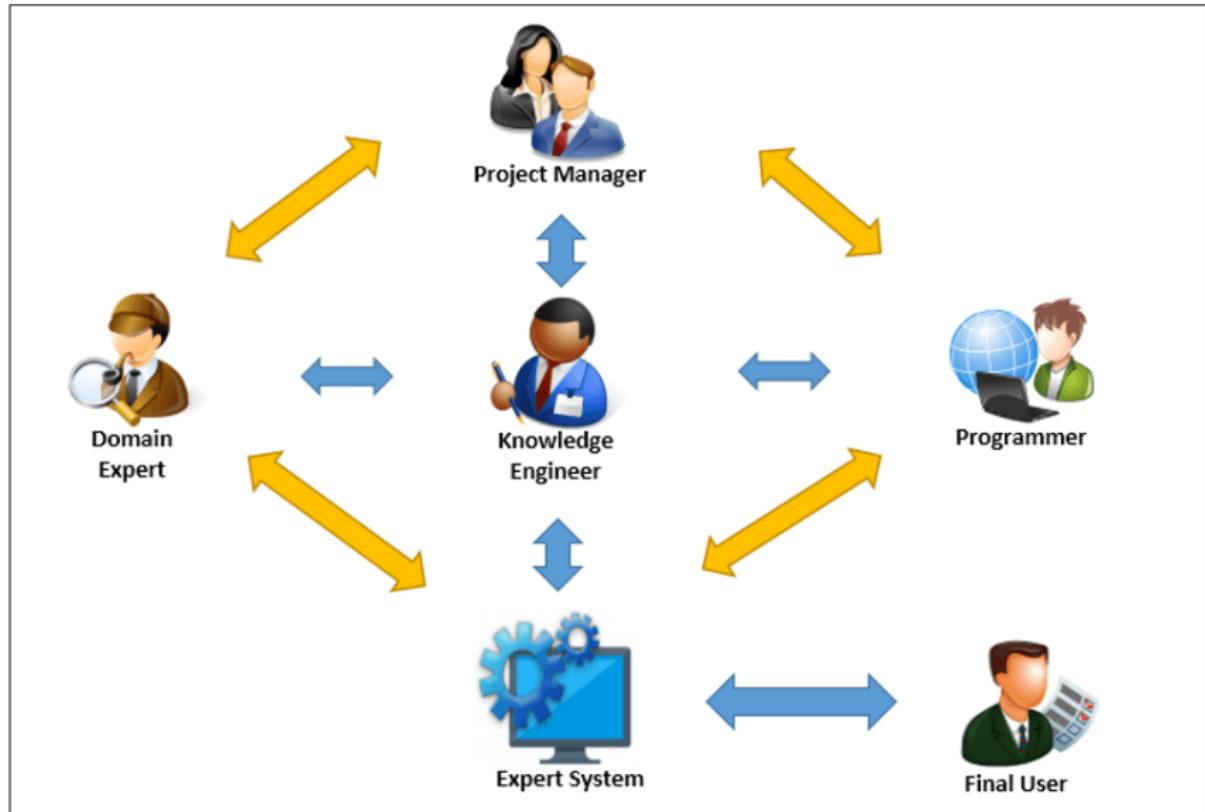
Composition

- Base de connaissance
- Moteur d'inférence

Systèmes experts



Systèmes experts



Entraînement et évaluation des réseaux de neurones

- Évaluation du modèle :
 - Évaluation des performances sur l'ensemble de validation.
 - Calcul des métriques d'évaluation.
 - Ajustements sur l'architecture ou les paramètres si nécessaire.
- Test final du modèle :
 - Évaluation des performances sur l'ensemble de test.
 - Calcul des métriques finales.
- Réglage des hyperparamètres :
 - Expérimentation avec différents hyperparamètres.
 - Utilisation de techniques comme la validation croisée.
- Répétition des étapes 3 à 6 jusqu'à atteindre les performances souhaitées.

Défis et tendances des réseaux de neurones

Défis

- ➊ Surapprentissage : Les réseaux de neurones sont sensibles au surapprentissage, ce qui entraîne une mauvaise généralisation.
- ➋ Pénurie de données : L'entraînement des réseaux de neurones profonds nécessite de grands ensembles de données étiquetés, qui ne sont pas toujours disponibles.
- ➌ Interprétabilité : Les réseaux de neurones sont souvent considérés comme des boîtes noires, ce qui rend difficile l'interprétation de leurs décisions.
- ➍ Exigences en calcul : L'entraînement de réseaux de neurones complexes peut être intensif en calcul et prendre beaucoup de temps.

Défis et tendances des réseaux de neurones

Tendances

- Architectures d'apprentissage profond : La tendance vers des architectures de réseaux de neurones plus profondes et plus complexes, telles que les réseaux neuronaux convolutifs (CNN) et les réseaux neuronaux récurrents (RNN).
- Apprentissage par transfert : Exploiter les modèles pré-entraînés et l'apprentissage par transfert pour améliorer les performances sur de nouvelles tâches avec des données limitées.
- IA explicative : Développer des méthodes pour interpréter et expliquer les décisions prises par les réseaux de neurones, augmentant ainsi leur transparence et leur fiabilité.
- Accélération matérielle : L'utilisation de matériels spécialisés, tels que les GPU et les TPU, pour accélérer l'entraînement et l'inférence des réseaux de neurones.

Techniques d'IML

① Méthodes basées sur les caractéristiques :

- Importance des caractéristiques : Évaluation de l'importance des caractéristiques pour comprendre leur impact sur les prédictions du modèle.
- Règles d'association : Identification de règles logiques pour expliquer les décisions du modèle.

② Méthodes basées sur les modèles simplifiés :

- Arbres de décision : Construction d'arbres de décision pour représenter les règles de décision du modèle.
- Modèles linéaires : Utilisation de modèles linéaires simplifiés pour interpréter les relations entre les caractéristiques et les prédictions.

③ Méthodes basées sur la perturbation des données :

- Perturbation des caractéristiques : Modification des valeurs des caractéristiques pour évaluer leur impact sur les prédictions.
- Échantillonnage de sous-populations : Création de sous-populations de données pour comprendre les différences de prédictions.

Choix de la technique

- Le choix de la technique d'IML dépend du contexte d'application et des objectifs de l'interprétation.
- Il est important de prendre en compte les compromis entre la précision, la complexité et la compréhensibilité de la technique.
- Certaines techniques peuvent être plus adaptées pour des modèles spécifiques ou des types de données particuliers.

Conclusion

- Les techniques d'IML offrent des moyens de comprendre et d'expliquer les décisions des modèles d'apprentissage automatique.
- Il existe différentes approches, chacune ayant ses avantages et ses limites.
- Le choix de la technique d'IML dépend du contexte et des objectifs spécifiques.

Méthodes basées sur les modèles linéaires

Les modèles linéaires offrent des possibilités d'interprétation grâce aux coefficients ou aux poids

- Coefficients des variables : Les modèles linéaires attribuent des coefficients à chaque variable pour indiquer leur influence sur la prédiction.
- Importance des variables : Il est possible d'évaluer l'importance des variables en se basant sur les coefficients. Les variables avec des coefficients plus élevés ont un impact plus important sur la prédiction.
- Perturbation des variables : On peut perturber les valeurs des variables une à une et observer l'impact sur la prédiction. Cela permet de quantifier l'influence de chaque variable.
- Échantillonnage de sous-populations : On peut créer des sous-populations de données en modifiant les valeurs d'un groupe de variables. Cela permet de comprendre comment ces groupes de variables affectent les prédictions.

Deep Learning comme une extension des réseaux de neurones

- Le Deep Learning étend les capacités des réseaux de neurones avec des architectures profondes.
- Il permet d'apprendre des représentations hiérarchiques des données et a révolutionné de nombreux domaines de l'intelligence artificielle.
- Les architectures populaires telles que les CNN, les RNN, les GAN et les Transformers ont des applications spécifiques.
- Le Deep Learning nécessite souvent de grandes quantités de données et des ressources de calcul significatives.
- Cependant, il a le potentiel de résoudre des problèmes complexes et de réaliser des avancées majeures dans de nombreux domaines.

Traitement du langage naturel avec les réseaux de neurones

- Présentation de l'utilisation des réseaux de neurones pour le traitement du langage naturel (NLP)
- Explication des modèles de NLP tels que les réseaux de neurones récurrents (RNN) et les transformers
- Exemples d'applications de NLP : classification de texte, génération de texte, compréhension de texte, etc.

TensorFlow et Keras

- Présentation des frameworks TensorFlow et Keras pour le développement de modèles de Deep Learning
- Explication de l'utilisation de TensorFlow pour la définition des graphes de calcul
- Présentation de Keras comme une interface conviviale pour la construction de modèles de Deep Learning

Qu'est-ce que TensorFlow ?

- TensorFlow est une bibliothèque open source d'apprentissage automatique développée par Google
- Elle permet de créer et d'exécuter des modèles de Deep Learning
- Utilise des graphes de flux de données pour le calcul parallèle et distribué
- Flexible et peut être déployé sur CPU, GPU, ou appareils mobiles
- Concurrent de Keras, CNTK, Caffe2 mais avec un objectif plus global

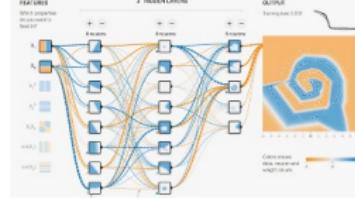


Comment fonctionne TensorFlow ?

- Crée un graphe de calcul avec des nœuds (opérations) et des arêtes (données)
- Permet un calcul parallèle et distribué pour accélérer l'entraînement
- Offre des possibilités de modélisation de haut niveau, de calcul flexible sur GPU et de visualisation
- Développé à l'origine par l'équipe Google Brain pour la recherche en apprentissage automatique

How TensorFlow works

- CPU
 - Multiprocessor
 - AVX-based acceleration
 - GPU part in chip
 - OpenMP
- GPU
 - CUDA (Nvidia) → cuDNN
 - OpenCL (AMD) → ComputeCPP / ROCm
- TPU (1st, 2nd gen.)
 - ASIC for accelerating matrix calculation
 - In-house development by Google



Comment utiliser TensorFlow ?

- Importer la bibliothèque dans votre code (Python, C++, C#)

```
1 import tensorflow as tf
```

- Créer une session, ajouter des couches pour construire le graphe de calcul
- Vérifier que tous les nœuds sont correctement connectés avant d'exécuter le graphe
- Utiliser TensorFlow comme n'importe quelle autre bibliothèque de votre langage de programmation

Avantages et inconvénients de TensorFlow

- Avantages :

- Flexibilité, capacité de calcul parallèle et distribué
- Permet d'apprendre à partir de données et faire des prédictions
- Vaste écosystème avec de nombreux outils et bibliothèques complémentaires

- Inconvénients :

- Courbe d'apprentissage plus élevée que d'autres outils
- Nécessite une expertise en apprentissage automatique
- Peut être complexe à mettre en place pour des projets simples

Conclusion

- TensorFlow est une puissante bibliothèque open source d'apprentissage automatique
- Elle offre une grande flexibilité et des performances élevées grâce au calcul parallèle et distribué
- Bien que plus complexe que certains outils, TensorFlow est idéal pour les projets d'apprentissage profond sophistiqués
- Avec son vaste écosystème, TensorFlow s'impose comme l'un des principaux outils du domaine de l'IA

Keras

Principales caractéristiques

① Abstraction de haut niveau

- API intuitive utilisée pour construire des modèles de deep learning
- Abstraction des détails techniques, permettant de se concentrer sur la conception du modèle

② Modularité et extensibilité

- Architecture modulaire permettant de construire facilement des modèles personnalisés
- Possibilité d'étendre Keras avec de nouvelles couches, fonctions d'activation, etc.

③ Prise en charge multi-backends

- Peut utiliser différents backends comme TensorFlow, CNTK et Theano
- Offre une flexibilité dans le choix de l'infrastructure de deep learning

④ Rapidité de prototypage

- Syntaxe simple et intuitive
- Favorise l'expérimentation et l'exploration de nouvelles architectures

Artificial Intelligence, Machine Learning and Deep Learning With Python

Approche technique de l'IA

Deep Learning and Neural Network Principle

Keras

Keras Principales caractéristiques

- ➊ Abstraction de haut niveau
 - API intuitive utilisée pour construire des modèles de deep learning
 - Abstraction des détails techniques, permettant de se concentrer sur la conception du modèle
- ➋ Modularité et extensibilité
 - Architecture modulaire permettant de construire facilement des modèles personnalisés
 - Possibilité d'étendre Keras avec de nouvelles couches, fonctions d'activation, etc.
- ➌ Prise en charge multi-backends
 - Peut utiliser différents backends comme TensorFlow, CNTK et Theano
 - Offre une flexibilité dans le choix de l'infrastructure de deep learning
- ➍ Rapidité de prototypage
 - Syntaxe simple et intuitive
 - Favorise l'expérimentation et l'exploration de nouvelles architectures

Keras offre une API de haut niveau qui abstrait les détails techniques du deep learning, permettant aux développeurs de se concentrer sur la conception du modèle plutôt que sur les implémentations complexes. La structure modulaire de Keras permet de construire facilement des modèles personnalisés en assemblant des briques de base. De plus, Keras est extensible, permettant aux développeurs d'ajouter de nouvelles fonctionnalités. Keras est capable d'utiliser différents backends de deep learning comme TensorFlow, CNTK et Theano. Cela offre une grande flexibilité aux développeurs dans le choix de l'infrastructure la plus adaptée à leurs besoins. Grâce à sa syntaxe simple et intuitive, Keras permet un prototypage rapide de modèles de deep learning. Cela facilite l'expérimentation et l'exploration de nouvelles architectures neuronales.

Keras

Exemple d'utilisation pour la classification d'images MNIST

```
1 from keras.models import Sequential
2 from keras.layers import Dense, Flatten
3 from keras.datasets import mnist
4
5 # Chargement des donnees MNIST
6 (X_train, y_train), (X_test, y_test)=mnist.load_data()
7
8 # Normalisation des donnees
9 X_train = X_train / 255.0
10 X_test = X_test / 255.0
11
12 # Construction du modele
13 model = Sequential()
14 model.add(Flatten(input_shape=(28, 28)))
15 model.add(Dense(128, activation='relu'))
16 model.add(Dense(10, activation='softmax'))
```

Artificial Intelligence, Machine Learning and Deep Learning With Python

└ Approche technique de l'IA

└ Deep Learning and Neural Network Principle

└ Keras

Keras

Exemple d'utilisation pour la classification d'images MNIST

```
from keras.models import Sequential
from keras.layers import Dense, Flatten
from keras.datasets import mnist

# Chargement des données MNIST
(X_train, y_train), (X_test, y_test) = mnist.load_data()

# Normalisation des données
X_train = X_train / 255.0
X_test = X_test / 255.0

# Construction du modèle
model = Sequential()
model.add(Flatten(input_shape=(28, 28)))
model.add(Dense(128, activation='relu'))
model.add(Dense(10, activation='softmax'))
```

Cet exemple montre comment utiliser Keras pour construire, entraîner et évaluer un modèle de classification d'images MNIST. Le code commence par charger les données MNIST, les normalise, puis construit un modèle séquentiel avec une couche d'aplatissement, une couche dense avec une activation ReLU, et une couche de sortie avec une activation softmax. Le modèle est ensuite compilé avec l'optimiseur Adam et la fonction de perte de l'entropie croisée catégorielle. Enfin, le modèle est entraîné pendant 10 époques avec un batch size de 32, et évalué sur l'ensemble de test.

Keras

Exemple d'utilisation pour la classification d'images MNIST

```
# Compilation du modèle
model.compile(optimizer='adam', loss='
    categorical_crossentropy', metrics=['accuracy'])

# Entrainement du modèle
model.fit(X_train, y_train, epochs=10, batch_size=32,
           validation_data=(X_test, y_test))

# Evaluation du modèle
loss, accuracy = model.evaluate(X_test, y_test)
print('Test loss:', loss)
print('Test accuracy:', accuracy)
```

Avantages et Utilisation

- TensorFlow et Keras offrent de nombreux avantages pour le développement de modèles de Deep Learning :
 - Grande flexibilité et extensibilité.
 - Prise en charge du calcul distribué.
 - Documentation complète et communauté active.
 - Intégration avec d'autres outils et bibliothèques de Machine Learning.
- Ils sont utilisés dans l'industrie et la recherche pour une variété d'applications de Deep Learning, y compris la vision par ordinateur, le traitement du langage naturel, la recommandation et bien d'autres.

Application du NLP : Etiquetage Morpho-syntaxique

Qu'est-ce que GitHub ?

Avant d'aborder ce cas pratique, voyons faisons d'abord une introduction à GitHub.

- GitHub est une plateforme de développement collaboratif basée sur Git.
- Elle permet de stocker, gérer et partager des projets de développement.
- GitHub fournit des fonctionnalités telles que la gestion des versions, le suivi des problèmes et la collaboration entre les membres de l'équipe.

Fonctionnalités de GitHub

- Gestion des dépôts : Créez des dépôts Git pour stocker et organiser vos projets.
- Collaboration : Travaillez avec d'autres développeurs en partageant des dépôts et en fusionnant des modifications.
- Problèmes : Utilisez les problèmes de GitHub pour gérer et suivre les tâches, les bogues et les demandes de fonctionnalités.
- Actions : Automatisez vos workflows de développement avec GitHub Actions.
- Pages GitHub : Hébergez des sites Web statiques directement depuis vos dépôts GitHub.
- Et bien plus encore !

Création d'un compte GitHub

- ① Rendez-vous sur le site web de GitHub : <https://github.com>
- ② Cliquez sur le bouton "Sign up" pour créer un nouveau compte.
- ③ Remplissez le formulaire d'inscription avec votre nom d'utilisateur, votre adresse e-mail et votre mot de passe.
- ④ Validez votre compte en confirmant l'adresse e-mail que vous avez utilisée lors de l'inscription.

Conclusion

- GitHub est une plateforme essentielle pour la gestion de projets de développement.
- Elle offre des fonctionnalités puissantes pour le contrôle de version, la collaboration et le suivi des problèmes.
- En créant un compte GitHub, vous pouvez explorer des projets open source, contribuer à des projets existants et héberger vos propres projets.

Description, objectif et sources existantes du problème

- Description

- On souhaite lier deux parties d'un discours de langue allemande- un adverbe pronominal (PA) et une phrase précédente (antécédent).
- Le PA peut être relié à la phrase précédente (anaphore) ou à la phrase suivante (cataphore).

- Objectif

- Identifier automatiquement les PA dans un paragraphe donné.
- Déterminer s'ils font référence à un antécédent ou à un postcédant.

- Sources d'informations existantes (voir

<https://github.com/Hermann-Sockeng>)

- Utilisation d'un algorithme de résolution de l'anaphore pronominale adverbiale.

- Étapes :

- Détection de la présence d'un postcédant pour les Da-/Hier-PA.
- Identification des Wo-PA en début de phrases.

Aspects méthodologiques et collaboration

- Méthodologie
 - Tokenization du texte en phrases et en mots.
 - Utilisation de Python et du framework NLTK.
 - Collecte des adverbes pronominaux.
 - Implémentation de l'algorithme de résolution de l'anaphore pronominale adverbiale.
 - Évaluation des résultats sur des données de test fournies par le linguiste.
- Participation et responsabilité des parties
 - Collaboration entre le linguiste et l'ingénieur en calcul scientifique.
 - Le linguiste décrit et présente les algorithmes linguistiques.
 - L'ingénieur en calcul scientifique implémente ces méthodes en langage informatique.

Conclusion

- Récapitulatif des concepts clés abordés dans les techniques avancées de Machine Learning
- Invitation aux questions et discussions

Analyse des données avec Power BI

- Introduction à Power BI pour l'analyse des données
- Utilisation des fonctionnalités de Power BI pour importer et préparer les données
- Création de visualisations interactives pour l'exploration des données
- Utilisation de Power BI pour générer des insights et des rapports visuels

Utilisation de visuels d'IA pour l'analyse de données

- Présentation des visuels d'IA disponibles dans Power BI
- Utilisation de la reconnaissance d'images pour l'analyse visuelle des données
- Utilisation de la détection d'anomalies pour identifier les valeurs aberrantes
- Utilisation du service cognitif de Power BI pour l'analyse linguistique des données textuelles

Formations gratuites online

Formations gratuites à recommander

- **"Introduction à l'intelligence artificielle"** sur Coursera
- **Spécialisation "Apprentissage automatique"** sur Coursera
 - Cette série de 4 cours gratuits propose une introduction complète au ML avec des exercices pratiques.
- **Cours "Deep Learning Specialization"** sur Coursera
 - Offert par deeplearning.ai, ce programme de 5 cours explore en détail les techniques avancées de deep learning.
- **"Python for Everybody Specialization"** sur Coursera
- **Tutoriels "Machine Learning Crash Course"** de Google
- **Applied Data Science Lab** de la WorldQuant University
 - The Applied Data Science Lab is a hands-on learning experience that accommodates learners with the right amount of foundational knowledge and a commitment to success.

Artificial Intelligence, Machine Learning and Deep Learning With Python

└ Approche technique de l'IA

└ Cas pratique de NLP

└ Formations gratuites online

Formations gratuites online

Formations gratuites à recommander

- » "Introduction à l'intelligence artificielle" sur Coursera
- » Spécialisation "Apprentissage automatique" sur Coursera
 - » Cours suivis de 4 semaines, ce programme offre une introduction complète au ML avec des exercices pratiques et interactifs.
- » Cours "Deep Learning Specialization" sur Coursera
 - » Offert par deeplearning.ai, ce programme de 5 cours explore en détail les techniques avancées de deep learning.
- » "Python for Everybody Specialization" sur Coursera
- » Tutorials "Machine Learning Crash Course" de Google
- » Applied Data Science Lab de la WorldQuant University
 - » The Applied Data Science Lab is a hands-on learning experience that accommodates learners with the right amount of foundational knowledge and a commitment to success.

Après avoir suivi le matériel de formation sur l'IA, l'apprentissage automatique et le deep learning avec Python, nous vous recommandons ces cours gratuits pour approfondir vos connaissances :



BENSEARCH-SOLUTIONS

Agilité Digitale et Innovation Continue



+33 7 82 99 08 55



bensearch-solutions.com



contact@bensearch-solutions.com