Résumé

Introduction Big Data/No-sql & Machine Learning

Définitions

Big Data

- Fort **volume** de donnée.

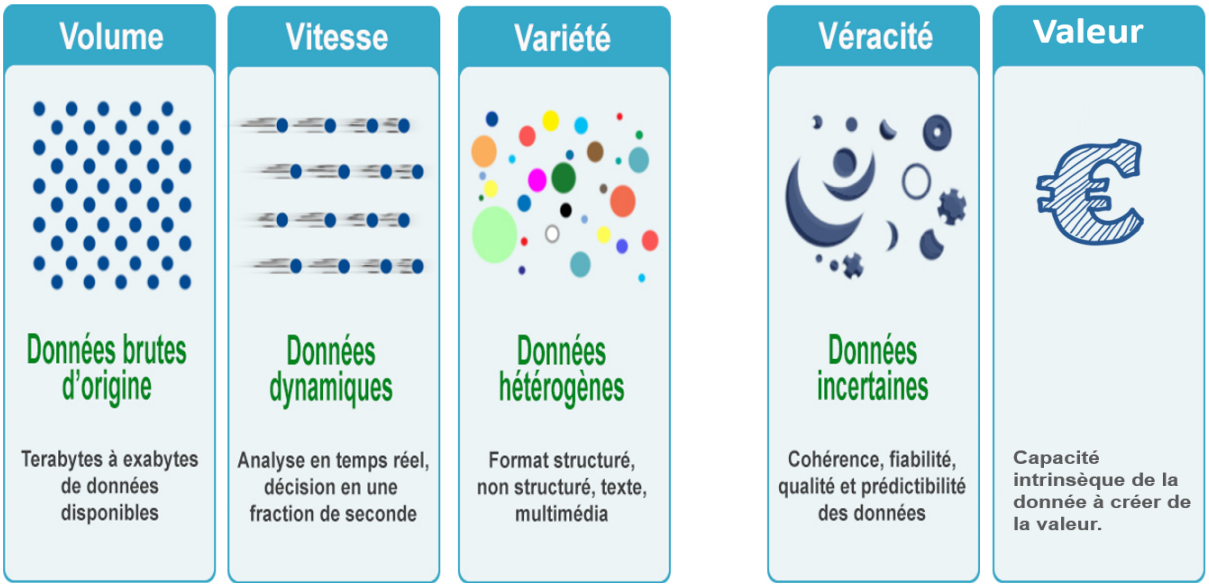
- Données **hétérogènes** (concerne plusieurs domaines : réseau social, médecine, physique, …).

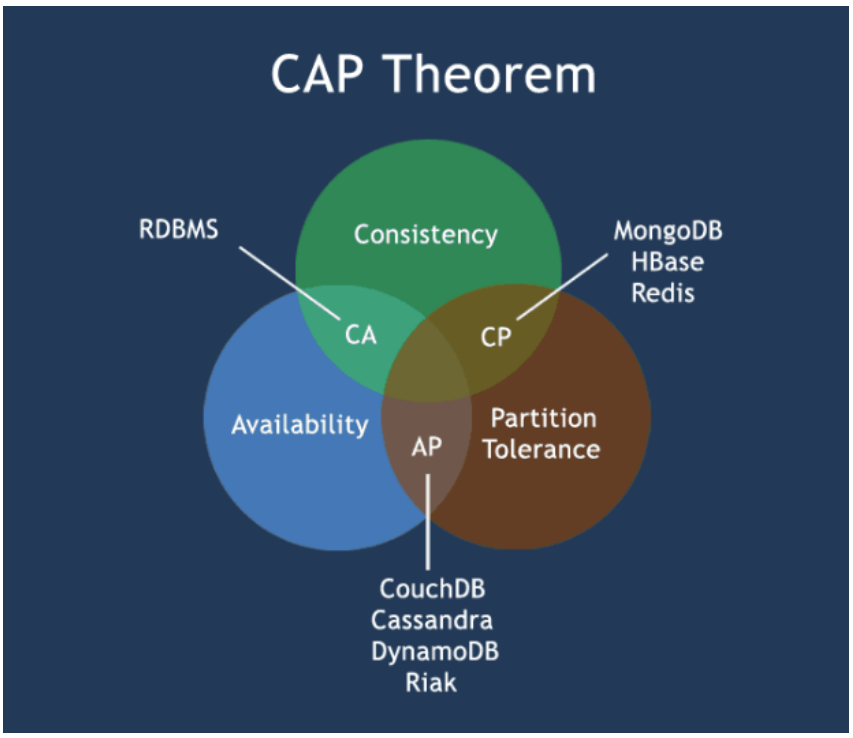
- **Rapidité** de traitement nécessaire à un algorithme de traitement de données pour fournir une réponse en moins de 100ms (se rappeler de l'exemple de la bonne publicité ciblée à afficher à l'utilisateur : contacter le serveur web, retrouver et analyser les données d'un utilisateur donné, trouver la bonne publicité à afficher, la renvoyer au client) => d'où le besoin de gérer des bases de données hétérogènes (no-sql) sur plusieurs serveurs et de faire du calcul distribué face aux nombres de connexions (hadoop, spark).

- Cycle de la donnée :

- Différents métiers liés à plusieurs domaines : business, mathématiques et informatiques.

- La règle des **3V/5V** :





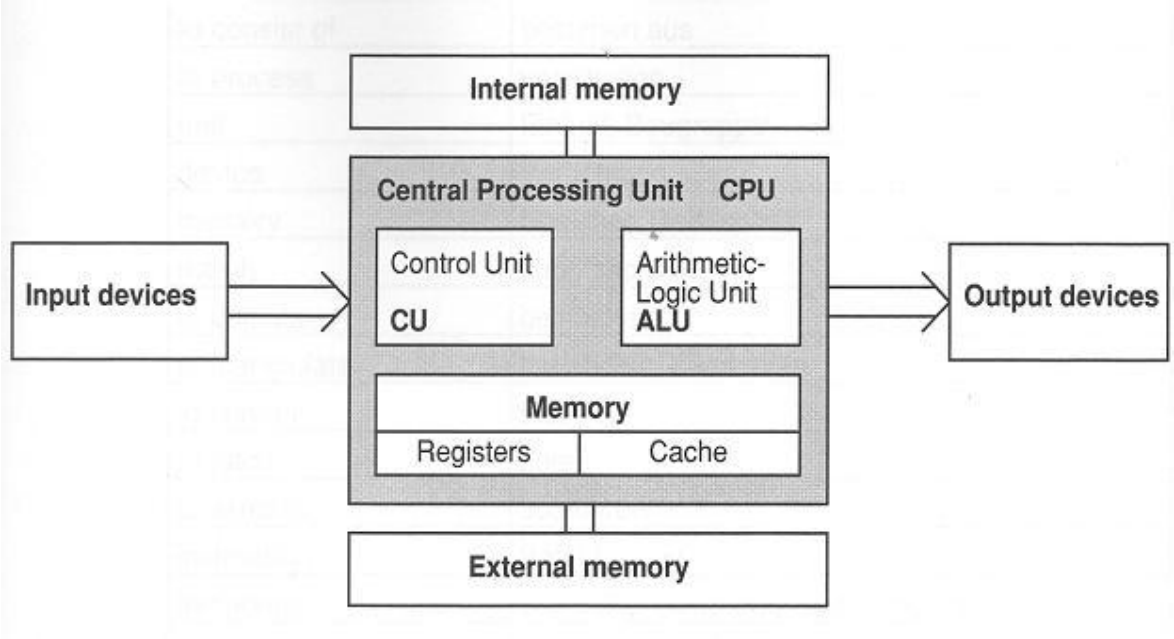
Atomicity
Consistency
Isolation
Durability

vs

Basically **A**vailable
Soft State
Eventually Consistent

ACID	BASE
Strong consistency for transactions highest priority Availability less important Pessimistic Rigorous analysis Complex mechanisms	Availability and scaling highest priorities Weak consistency Optimistic Best effort Simple and fast

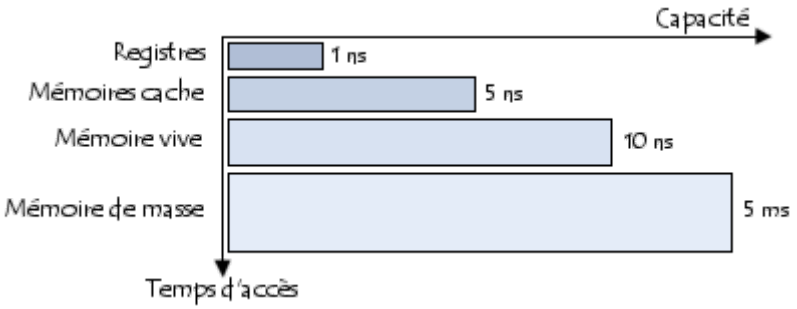
Fonctionnement d'un processeur



- **Unité de contrôle** : définit les opérations à réaliser.

- **Unité arithmétique et logique** : effectue les opérations calculatoires.

- Il existe des **registres** (64bits) et des **caches** (quelques Ko à plusieurs Mo) par processeur. La **mémoire vive** et le **disque dur** ou **SSD** sont partagés pour tous les processeurs de la machine. Pour du calcul distribué, chaque machine a sa mémoire.



- Une machine possédant 4 processeurs peut contenir 8 coeurs lorsque les processeurs sont hyperthreadés. Il y a 4 coeurs physiques et 8 logiques, comme s'il y avait 8 coeurs. Sur cette machine, 8 programmes peuvent s'exécuter en parallèle.

Programmation

- On parle de **programmation séquentielle** pour exécuter les tâches d'un programme séquentiellement, dans l'ordre d'écriture, ligne à ligne. On raisonne en terme de boucle et d'instruction qui s'exécutent ligne à ligne.

- On parle de **programmation parallèle** lorsque l'on parallélise le code afin d'exploiter les différents processeurs d'une même machine afin d'exécuter simultanément plusieurs tâches.

- On peut utiliser des cartes graphiques (ou GPU) pour paralléliser les calculs.

- Par opposition à la programmation séquentielle, on parle de **programmation fonctionnelle** ou l'on réfléchit en terme de fonction s'appliquant à un tableau (ou autre structure de données) (map/reduce).

- **Programmation synchrone** : on effectue les opérations dans l'ordre. S'il y a un évènement bloquant on attend.

- **Programmation asynchrone** ou **programmation évènementiel** : utilisé notamment pour les serveurs web (nodejs) où il y a beaucoup d'attentes bloquantes dues à l'intervention de l'utilisateur ; dès lors on pourra paralléliser massivement pour répondre à plusieurs connexions entrantes donc à plusieurs utilisateurs sans attendre la réponse de chacun.

- Les **langages de programmation compilés** (C, C++, C#, …) utilisent un compilateur pour traduire le code haut niveau en langage bas niveau ou machine (assembleur) avant de le traduire en langage binaire. Le compilateur réalise des opérations d'optimisation.

- Les **langages de programmation interprétés** tels que Python ou Bash sont typiquement des langages de script pour exécuter des commandes ligne à ligne

- Des langages comme Java ou Scala utilisent une **machine virtuelle**, il s'agit d'une surcouche entre le code haut niveau et le code machine. Après la compilation, on génère un code machine (bytecode pour la JVM) avant de traduire ce code en binaire. Cette surcouche rajoute du temps à l'exécution mais la machine virtuelle est utile pour le calcul distribué sur plusieurs machines avec des processeurs différents : inutile de se soucier d'une compilation spécifique à chaque machine, la JVM se charge de transformer le bytecode en binaire.

- Lors de la création d'un processus, un espace mémoire est réservé en mémoire RAM. Celui-ci est partagé entre plusieurs threads.

- Il y a 2 sortes de mémoires en programmation : la **pile** pour les appels de fonctions (sauvegarde du contexte) et le **tas** pour la création des listes, tableaux, etc.

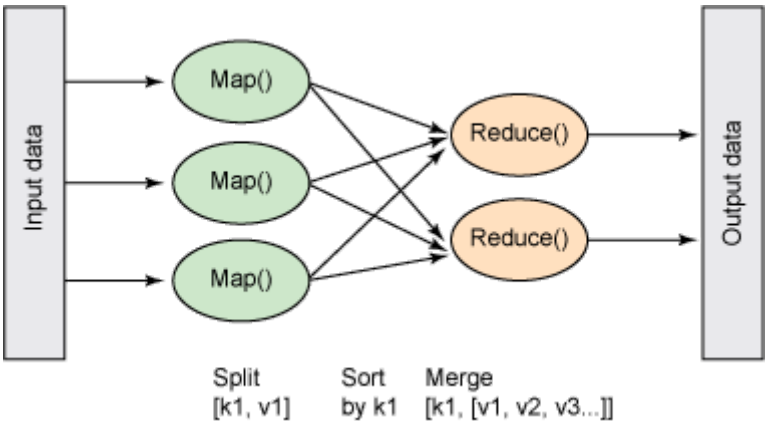
- D'un point de vue conceptuel, il existe différents types d'automates pour différentes classes de calcul. Les automates à états finis, les automates à piles (utilisé pour les appels de fonctions récursives) et les machines de Turing qui coïncide avec la classe des fonctions calculables (implémentée par les ordinateurs).

Map/Reduce

- **MapReduce** est un patron d'architecture de développement informatique, inventé par Google, dans lequel sont effectués des calculs parallèles, et souvent distribués, de données potentiellement très volumineuses, typiquement supérieures en taille à 1 téraoctet.

- **Hadoop** est une implémentation open source en Java du MapReduce distribué par la fondation Apache. Les deux caractéristiques principales de Hadoop sont le framework MapReduce et le Hadoop Distributed File System. Le HDFS permet de distribuer les données et de faire des traitements performants sur ces données grâce au MapReduce en distribuant une opération sur plusieurs nœuds afin de la paralléliser.

- **Spark** est une amélioration d'Hadoop codé en Scala. Au lieu d'utiliser la mémoire morte, il utilise la mémoire vive pour accélérer le traitement des données par 10 environ.



Base de données relationnelle

- **Base de données relationnelle** : base de données où l'information est organisée dans des tableaux à deux dimensions appelés des relations ou tables. Selon ce modèle relationnel, une base de données consiste en une ou plusieurs relations. Les lignes de ces relations sont appelées des nuplets. Les colonnes sont appelées des attributs. Les logiciels qui permettent de créer, utiliser et maintenir des bases de données relationnelles sont des systèmes de gestion de base de données relationnels. Pratiquement tous les systèmes relationnels utilisent le langage SQL pour interroger les bases de données. Ce langage permet de demander des opérations d'algèbre relationnelle telles que l'intersection, la sélection et la jointure.

- Pour stocker de manière efficace l'information sans redondance, on utilise les **formes normales** :

    - 1ère forme normale : une donnée doit être atomique (si l'on a ID/Moto : 1/Yamaha,Sym on écrira plutôt 1/Yamaha et 2/Sym).

    - 2ème forme normale : tous les attributs non-primaire doivent se référer à l'ensemble des attributs formants la clé primaire, sinon on peut couper en plusieurs tables (par exemple, si l'on a id_student et id_prog comme clés primaires et name_student et name_prog comme attributs, on pourra séparer en une table student et une table prog).

    - 3ème forme normale : s'il y a des redondances entre les attributs, on peut séparer en plusieurs tables (par exemple, la ville qui dépend du code postal qui dépend de l'id de la personne ; on créera plutôt une nouvelle table ville avec le code postal comme clé primaire).

No-sql

- **NoSQL** désigne une famille de systèmes de gestion de base de données (SGBD) qui s'écarte du paradigme classique des bases relationnelles. L'explicitation du terme la plus populaire de l'acronyme est Not only SQL. L'architecture machine en clusters induit une structure logicielle distribuée fonctionnant avec des agrégats répartis sur différents serveurs permettant des accès et modifications concurrentes mais imposant également de remettre en cause de nombreux fondements de l'architecture SGBD relationnelle traditionnelle, notamment les propriétés ACID (opération atomique, consistance des données).

- **Apache Kafka** est un projet open-source agent de messages développé par l'Apache Software Foundation écrit en Scala. Le projet vise à fournir un système unifié, temps réel à latence faible pour la manipulation de flux de données en temps réel. La conception est fortement influencée par les transaction logs.

- **HBase** est un système de gestion de base de données non-relationnelles distribué, écrit en Java, disposant d'un stockage structuré pour les grandes tables. C'est une base de données orientée colonnes.

- **Pig** est une plateforme haut niveau pour la création de programme MapReduce utilisé avec Hadoop. Le langage de cette plateforme est appelé le Pig Latin. Pig Latin s'abstrait du langage de programmation Java MapReduce et se place à un niveau d'abstraction supérieure, similaire à celle de SQL pour systèmes SGBDR. Pig Latin peut être étendue en utilisant UDF (User Defined Functions) que l'utilisateur peut écrire en Java, en Python, en JavaScript, en Ruby ou en Groovy et ensuite être utilisé directement au sein du langage.

- **Apache Hive** est une infrastructure d'entrepôt de donnée intégrée sur Hadoop permettant l'analyse, le requétage via un langage proche syntaxiquement de SQL ainsi que la synthèse de données.

- **Apache Cassandra** est un système de gestion de base de données (SGBD) de type NoSQL conçu pour gérer des quantités massives de données sur un grand nombre de serveurs, assurant une haute disponibilité en éliminant les points individuels de défaillance. Il permet une répartition robuste sur plusieurs centres de données, avec une réplication asynchrone sans master et une faible latence pour les opérations de tous les clients. C'est une base de données orientée colonnes.

- **Neo4j** est un système de gestion de base de données au code source libre basée sur les graphes, développé en Java.

- **MongoDB** est un système de gestion de base de données NoSQL orientée documents, répartissable sur un nombre quelconque d'ordinateurs et ne nécessitant pas de schéma prédéfini des données. Il est écrit en C++.

Machine learning

L'apprentissage automatique/machine learning concerne la conception, l'analyse, le développement et l'implémentation de méthodes permettant à une machine d'évoluer par un processus systématique, et ainsi de remplir des tâches difficiles ou problématiques par des moyens algorithmiques plus classiques. L'analyse peut concerner des graphes, arbres, ou courbes (par exemple, la courbe d'évolution temporelle d'une mesure ; on parle alors de données continues, par opposition aux données discrètes associées à des attributs-valeurs classiques). Un exemple possible d'apprentissage automatique est celui de la classification : étiqueter chaque donnée en l'associant à une classe.

Les grandes familles de problèmes sont :

- La **réduction de dimension** est un procédé permettant de réduire le nombre de variables dans un problème en ne gardant que les variables principales.

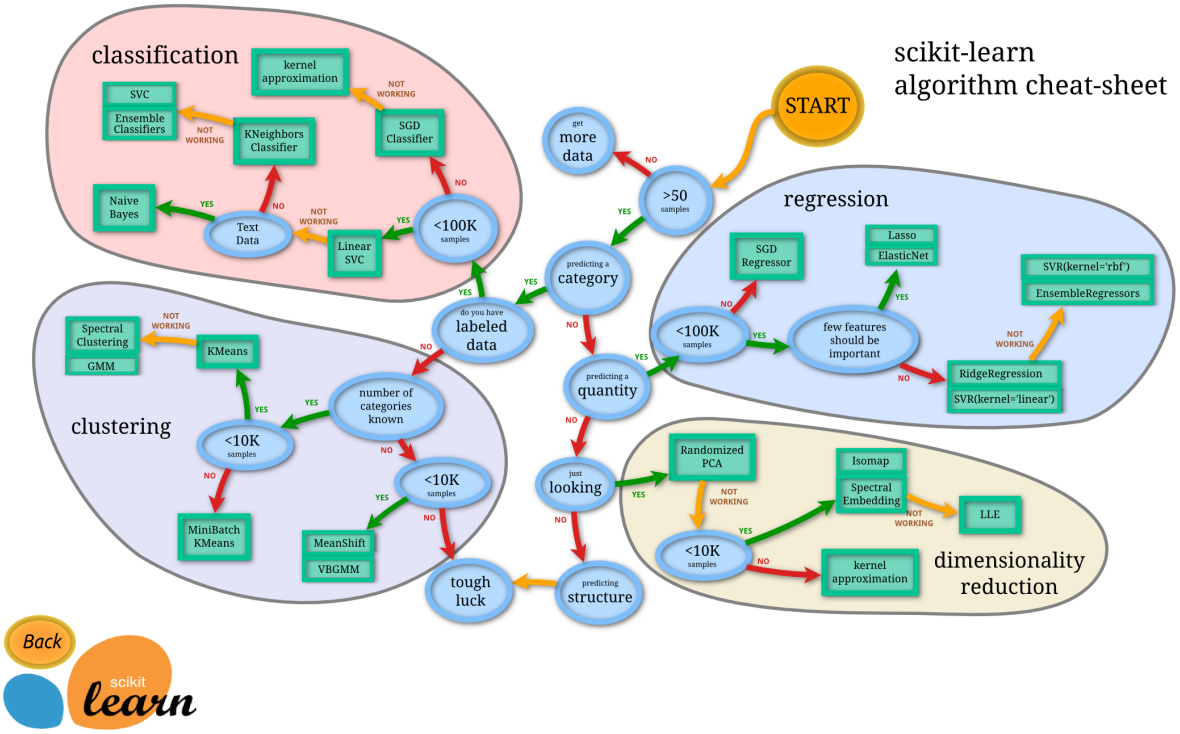
- La **régression** : quand il s'agit de prédire une quantité.

- La **classification** : étiqueter chaque donnée en l'associant à une classe de manière **supervisée** (on connaît les labels des classes pendant l'apprentissage).

- Le partitionnement de données (ou data **clustering** en anglais) est une des méthodes d'analyse des données. Elle vise à diviser un ensemble de données en différents « paquets » homogènes, en ce sens que les données de chaque sous-ensemble partagent des caractéristiques communes, qui correspondent le plus souvent à des critères de proximité (similarité) que l'on définit en introduisant des mesures et classes de distance entre objets.
  Pour obtenir un bon partitionnement, il convient d'à la fois de minimiser l'inertie intra-classe pour obtenir des grappes (cluster en anglais) les plus homogènes possibles et de maximiser l'inertie inter-classe afin d'obtenir des sous-ensembles bien différenciés.

- **Problèmes d'optimisation** : L'optimisation est une branche des mathématiques cherchant à modéliser, à analyser et à résoudre analytiquement ou numériquement les problèmes qui consistent à minimiser ou maximiser une fonction sur un ensemble.

- La **visualisation** des données est un ensemble de méthodes de représentation graphique, en deux ou trois dimension, utilisant ou non de la couleur. On peut par exemple utiliser la bibliothèque **Matplotlib** en Python ou **D3.js** qui est une bibliothèque graphique JavaScript qui permet l'affichage de données numériques sous une forme graphique et dynamique.



Outils

- **Scikit-learn** est une bibliothèque libre Python dédiée à l'apprentissage automatique. Elle comprend notamment des fonctions pour estimer des forêts aléatoires, des régressions logistiques, des algorithmes de classification, et les machines à vecteurs de support. Elle est conçue pour s'harmoniser avec des autres bibliothèques libre Python, notamment NumPy, Pandas et SciPy.

- **TensorFlow** est un outil open source d'apprentissage automatique développé par Google. Les autres frameworks spécialisés pour les réseaux de neurones sont **Theano** et **Torch** avec **Keras** en sur-couche.

Réduction de dimension

- **L'analyse en composantes principales** (ACP ou PCA en anglais pour principal component analysis) est une méthode de la famille de l'analyse des données et plus généralement de la statistique multivariée, qui consiste à transformer des variables liées entre elles (dites « corrélées » en statistique) en nouvelles variables décorrélées les unes des autres. Ces nouvelles variables sont nommées « composantes principales », ou axes principaux. Elle permet de réduire le nombre de variables et de rendre l'information moins redondante. Il s'agit d'une approche à la fois géométrique (les variables étant représentées dans un nouvel espace, selon des directions d'inertie maximale) et statistique (la recherche portant sur des axes indépendants expliquant au mieux la variabilité — la variance — des données). Lorsqu'on veut compresser un ensemble de N variables aléatoires, les n premiers axes de l'analyse en composantes principales sont un meilleur choix, du point de vue de l'inertie ou de la variance.

- L’ACP s’utilise pour des données quantitatives, **l'Analyse des Correspondances Multiples** s’utilise pour des variables qualitatives.

- **L’analyse discriminante linéaire** (aussi LDA, en anglais : linear discriminant analysis) fait partie des techniques d’analyse discriminante prédictive. Il s’agit d’expliquer et de prédire l’appartenance d’un individu à une classe (groupe) prédéfinie à partir de ses caractéristiques mesurées à l’aide de variables prédictives. La variable à prédire est forcément catégorielle (discrète). Les variables prédictives sont a priori toutes continues. L’analyse discriminante linéaire peut être comparée aux méthodes supervisées développées en apprentissage automatique et à la régression logistique développée en statistique.

Régression

- Un modèle de **régression linéaire** est un modèle de régression qui cherche à établir une relation linéaire entre une variable, dite expliquée, et une ou plusieurs variables, dites explicatives. Parmi les modèles de régression linéaire, le plus simple est l'ajustement affine. Celui-ci consiste à rechercher la droite permettant d'expliquer le comportement d'une variable statistique y comme étant une fonction affine d'une autre variable statistique x. Le modèle de régression linéaire est souvent estimé par la méthode des moindres carrés mais il existe aussi de nombreuses autres méthodes pour estimer ce modèle. On peut par exemple estimer le modèle par maximum de vraisemblance ou encore par inférence bayésienne.

- Un **réseau de neurones artificiels** est un ensemble d'algorithmes dont la conception est à l'origine très schématiquement inspirée du fonctionnement des neurones biologiques, et qui par la suite s'est rapproché des méthodes statistiques. Ils peuvent être utilisés pour des tâches de classification ou de régression.

- Les **machines à vecteurs de support** ou séparateurs à vaste marge (en anglais support vector machine, SVM) sont un ensemble de techniques d'apprentissage supervisé destinées à résoudre des problèmes de discrimination et de régression. Les SVM sont une généralisation des classifieurs linéaires. La frontière de séparation est linéaire mais on peut utiliser un noyau (kernel) pour changer d’espace de représentation et rendre séparable linéairement les données.

Classification

- La **régression logistique** est un modèle de régression binomiale. La variable a prédire vaut soit 0 soit 1. Il s’agit donc d’un problème de classification supervisée à 2 classes.

- En intelligence artificielle, la méthode des **k plus proches voisins** est une méthode d’apprentissage supervisé. En abrégé k-NN ou KNN, de l'anglais k-nearest neighbor. Dans ce cadre, on dispose d’une base de données d'apprentissage constituée de N couples « entrée-sortie ». Pour estimer la sortie associée à une nouvelle entrée x, la

méthode des k plus proches voisins consiste à prendre en compte (de façon identique) les k échantillons d'apprentissage dont l'entrée est la plus proche de la nouvelle entrée x, selon une distance à définir. Par exemple, dans un problème de classification, on retiendra la classe la plus représentée parmi les k sorties associées aux k entrées les plus proches de la nouvelle entrée x.

- Un **arbre de décision** est un outil d'aide à la décision représentant un ensemble de choix sous la forme graphique d'un arbre. Les différentes décisions possibles sont situées aux extrémités des branches (les « feuilles » de l'arbre), et sont atteints en fonction de décisions prises à chaque étape. Il s'agit d'un algorithme d'apprentissage supervisé. Principalement utilisé en classification, ils peuvent également être utilisé pour des tâches de régression.

- Les **forêts d'arbres décisionnels** (ou forêts aléatoires de l'anglais random forest classifier) ont été formellement proposées en 2001 par Leo Breiman et Adèle Cutler. Elles font partie des techniques d'apprentissage automatique. Cet algorithme combine les concepts de sous-espaces aléatoires et de bagging. L'algorithme des forêts d'arbres décisionnels effectue un apprentissage sur de multiples arbres de décision entraînés sur des sous-ensembles de données légèrement différents. Principalement utilisé en classification, ils peuvent également être utilisé pour des tâches de régression.

- Un **réseau de neurones convolutifs** ou réseau de neurones à convolution (en anglais CNN ou ConvNet pour Convolutional Neural Networks) est un type de réseau de neurones artificiels de type feed-forward, dans lequel le motif de connexion entre les neurones est inspiré par le cortex visuel des animaux. Les neurones de cette région du cerveau sont arrangés de sorte qu'ils correspondent à des régions qui se chevauchent lors du pavage du champ visuel. Leur fonctionnement est inspiré par les processus biologiques, ils consistent en un empilage multicouche de perceptrons, dont le but est de prétraiter de petites quantités d'informations. Les réseaux neuronaux convolutifs ont de larges applications dans la reconnaissance d'image et vidéo, les systèmes de recommandation et le traitement du langage naturel.

Clustering

- Le partitionnement en **k-moyennes** (ou k-means en anglais) est une méthode de partitionnement de données et un problème d'optimisation combinatoire. Étant donnés des points et un entier k, le problème est de diviser les points en k groupes, souvent appelés clusters, de façon à minimiser une certaine fonction. On considère la distance d'un point à la moyenne des points de son cluster ; la fonction à minimiser est la somme des carrés de ces distances. Les k-moyennes sont notamment utilisées en apprentissage non supervisé où l'on divise des observations en k partitions.

- **DBSCAN** (density-based spatial clustering of applications with noise) est un algorithme de partitionnement de données. Il s'agit d'un algorithme basé sur la

densité dans la mesure où il s'appuie sur la densité estimée des clusters pour effectuer le partitionnement. L'algorithme DBSCAN utilise 2 paramètres : la distance epsilon et le nombre minimum de points MinPts devant se trouver dans un rayon epsilon pour que ces points soient considérés comme un cluster. Les paramètres d'entrées sont donc une estimation de la densité de points des clusters. L'idée de base de l'algorithme est ensuite, pour un point donné, de récupérer son epsilon-voisinage et de vérifier qu'il contient bien MinPts points ou plus. Ce point est alors considéré comme faisant partie d'un cluster. On parcourt ensuite l'epsilon-voisinage de proche en proche afin de trouver l'ensemble des points du cluster.

Optimisation

- En informatique, la **programmation dynamique** est une méthode algorithmique pour résoudre des problèmes d'optimisation. La programmation dynamique consiste à résoudre un problème en le décomposant en sous-problèmes, puis à résoudre les sous-problèmes, des plus petits aux plus grands en stockant les résultats intermédiaires. La programmation dynamique s'appuie sur un principe simple, appelé le principe d'optimalité de Bellman : toute solution optimale s'appuie elle-même sur des sous-problèmes résolus localement de façon optimale. Concrètement, cela signifie que l'on peut déduire une ou la solution optimale d'un problème en combinant des solutions optimales d'une série de sous-problèmes. Les solutions des problèmes sont calculées de manière ascendante, c'est-à-dire qu'on débute par les solutions des sous-problèmes les plus petits pour ensuite déduire progressivement les solutions de l'ensemble. La méthode de programmation dynamique, comme la méthode diviser pour régner, résout des problèmes en combinant des solutions de sous-problèmes. Alors que les algorithmes diviser-pour-régner partitionnent le problème en sous-problèmes indépendants qu'ils résolvent récursivement, puis combinent leurs solutions pour résoudre le problème initial, la programmation dynamique, quant à elle, peut s'appliquer même lorsque les sous-problèmes ne sont pas indépendants, c'est-à-dire lorsque des sous-problèmes comportent des parties communes. Dans ce cas, un algorithme diviser-pour-régner fait du travail inutile en résolvant plusieurs fois le sous-problème commun. Un algorithme de programmation dynamique résout chaque sous-problème une seule fois et mémorise la réponse dans un tableau, évitant ainsi le recalcul.

- Les **algorithmes génétiques** appartiennent à la famille des algorithmes évolutionnistes. Leur but est d'obtenir une solution approchée à un problème d'optimisation, lorsqu'il n'existe pas de méthode exacte (ou que la solution est inconnue) pour le résoudre en un temps raisonnable. Les algorithmes génétiques utilisent la notion de sélection naturelle et l'appliquent à une population de solutions potentielles au problème donné. La solution est approchée par « bonds » successifs, comme dans une procédure de séparation et évaluation, à ceci près que ce sont des formules qui sont recherchées et non plus directement des valeurs.

- Les **algorithmes de colonies de fourmis** sont des algorithmes inspirés du comportement des fourmis et qui constituent une famille de métaheuristiques d'optimisation. Initialement proposé par Marco Dorigo et al. dans les années 1990 pour la recherche de chemins optimaux dans un graphe, le premier algorithme s'inspire du comportement des fourmis recherchant un chemin entre leur colonie et une source de nourriture. L'idée originale s'est depuis diversifiée pour résoudre une classe plus large de problèmes et plusieurs algorithmes ont vu le jour, s'inspirant de divers aspects du comportement des fourmis.

- Le **recuit simulé** est une méthode empirique (métaheuristique) inspirée d'un processus utilisé en métallurgie. On alterne dans cette dernière des cycles de refroidissement lent et de réchauffage (recuit) qui ont pour effet de minimiser l'énergie du matériau. Cette méthode est transposée en optimisation pour trouver les extrema d'une fonction. Partant d'un état donné du système, en le modifiant, on obtient un autre état. Soit celui-ci améliore le critère que l'on cherche à optimiser et on accepte la nouvelle solution, soit celui-ci le dégrade et on le refuse avec une certaine probabilité. En acceptant un état améliorant le critère, on tend ainsi à chercher l'optimum dans le voisinage de l'état de départ. L'acceptation d'un « mauvais » état permet alors d'explorer une plus grande partie de l'espace des états et tend à éviter de s'enfermer trop vite dans la recherche d'un optimum local.

- **L'apprentissage par renforcement** fait référence à une classe de problèmes d'apprentissage automatique, dont le but est d'apprendre, à partir d'expériences, ce qu'il convient de faire en différentes situations, de façon à optimiser une récompense quantitative au cours du temps. L'agent cherche, au travers d'expériences itérées, un comportement décisionnel (appelé stratégie ou politique, et qui est une fonction associant à l'état courant l'action à exécuter) optimal, en ce sens qu'il maximise la somme des récompenses au cours du temps.

Validation des algorithmes

- La **validation croisée** (« cross-validation ») est une méthode d'estimation de fiabilité d'un modèle fondé sur une technique d'échantillonnage (en général 80% pour l'ensemble d'apprentissage et 20% pour l'ensemble de test). Cela permet d'éviter le sur-apprentissage ou l'overfitting pour ne permet pas de généraliser à de nouvelles données.

- La fonction d'efficacité du récepteur, plus fréquemment désignée sous le terme «**courbe ROC**» dite aussi caractéristique de performance (d'un test) ou courbe sensibilité/spécificité, est une mesure de la performance d'un classificateur binaire, c'est-à-dire d'un système qui a pour objectif de catégoriser des éléments en deux groupes distincts sur la base d'une ou plusieurs des caractéristiques de chacun de ces éléments. Graphiquement, on représente souvent la mesure ROC sous la forme d'une courbe qui donne le taux de vrais positifs (fraction des positifs qui sont effectivement détectés) en fonction du taux de faux positifs (fraction des négatifs qui sont détectés (incorrectement)).

- La méthode **grid-search** consiste à essayer de séquentielle plusieurs ensembles d'hyperparamètres en quadrillant l'espace des hyperparamètres par pas discret.

- L'analyse de la **complexité d'un algorithme** consiste en l'étude formelle de la quantité de ressources (par exemple de temps ou d'espace) nécessaire à l'exécution de cet algorithme. La complexité d'un algorithme s'écrit O(f(n)) ou f est une fonction et n le nombre de données en entrée de l'algorithme. La complexité peut être polynomiale (O(n), O(n²), …) ou non-polynomiale (O(n!), O(2^n), …).