

*Cette ressource est issue d'une publication du numéro 108 de La Revue 3EI d'avril 2022. Ludovic De Matteis, élève du DER Nikola Tesla, est auteur de l'article « Personalized Human-Swarm Interaction Through Hand Motion » [6] issu de son stage à l'EPFL sur l'apprentissage des mouvements intuitifs de l'utilisateur pour le contrôle d'une flotte de drone. Steeven Janny, ancien élève du DER Nikola Tesla de l'ENS Paris-Saclay, est doctorant au laboratoire LAGEPP sur l'établissement de modèles profonds pour la simulation et la prédiction de phénomènes physiques. Solal Nathan est étudiant du M2 IA de l'Université Paris-Saclay, actuellement en stage au LISN (Laboratoire Interdisciplinaire des Sciences du Numérique) sur « Combatting Unemployment : Data, Recommendations and Fairness ». Wenqi Shu-Quartier, ancienne élève du DER Nikola Tesla, est doctorante au laboratoire Neurospin du CEA sur l'application de l'apprentissage par ordinateur à la comparaison de la maladie d'Alzheimer avec son analogue chez le chimpanzé.*

Cette ressource introduit le « Dossier Intelligence Artificielle » [7] en définissant les contours de l'IA puis, après un rappel des dates marquantes de sa courte histoire, recentre le propos sur l'apprentissage automatique en différenciant apprentissage supervisé, apprentissage non-supervisé et apprentissage par renforcement. Les réseaux de neurones et l'apprentissage par renforcement faisant l'objet d'autres ressources du dossier, cette introduction illustre l'apprentissage supervisé par deux algorithmes classiques, régression linéaire et K plus proches voisins, de même que l'apprentissage non supervisé avec l'algorithme des k-moyennes, ceci notamment afin de mettre en évidence l'importance des hyperparamètres.

## 1 – Introduction

L'intelligence artificielle (IA) a pour but de créer des entités intelligentes. Dans ce sens, il y a une poursuite de cette capacité dont l'humain est maître, à résoudre des problèmes. « La capacité à résoudre des problèmes » est d'ailleurs une définition de l'intelligence que l'on retrouve régulièrement. Il n'existe pas de définition qui fasse consensus, bien que des chercheurs aient tenté de définir formellement l'intelligence. L'humain tente de répliquer son intelligence pour pouvoir automatiser certaines tâches. Il est important de noter qu'il y a une différence forte entre l'IA et l'apprentissage automatique (machine learning). L'apprentissage automatique est une forme d'IA qui consiste en un système qui s'améliore grâce à l'expérience alors que l'IA peut être un simple ensemble de règles et d'heuristiques.

## 2 – Qu'est-ce que l'intelligence artificielle ?

L'intelligence artificielle (IA) est un domaine de l'informatique et des mathématiques rassemblant un ensemble de techniques algorithmiques et de théories permettant de réaliser des machines imitant l'intelligence humaine. Son but est de reproduire l'intelligence afin d'être capable de résoudre des problèmes complexes. Cet objectif passe par la modélisation de l'intelligence humaine en tant que phénomène, comme cela pourrait être fait dans le domaine de la physique, de la chimie ou de la biologie. L'intelligence artificielle est un domaine en pleine expansion, trouvant sa théorie

et ses applications dans de très nombreux domaines, comme la théorie des probabilités, les neurosciences, la robotique, la théorie des jeux, la santé et les transports.

L'intelligence artificielle est généralement classée comme forte ou faible.

- L'IA forte a pour objectif de développer une machine disposant de capacités cognitives proches de l'humain. Cette machine serait capable de trouver seule une solution à un problème qui lui est encore inconnu. Bien que ce soit un sujet de recherche actif, l'IA forte n'existe pas à l'heure actuelle.
- L'IA faible, quant à elle consiste à développer des machines permettant de répondre à un problème précis. Elle dispose alors de données et d'aide humaine, et doit être capable de simuler le raisonnement humain, sans pour autant comprendre les phénomènes mis en œuvre.

Le domaine de l'intelligence artificielle est scindé en plusieurs sous-domaines imbriqués (présentés figure 1). L'apprentissage automatique est un sous-domaine de l'intelligence artificielle consistant à apprendre par l'expérience ou par une base de données des règles implicites pour répondre à un problème donné. Ce domaine s'oriente spécialement autour de l'analyse statistique de données d'entraînement. Dans le cadre de l'apprentissage automatique, nous verrons qu'il existe de nombreux algorithmes utilisant des modèles mathématiques variés. Le réseau de neurones est un de ces modèles, certainement le plus répandu et celui utilisé dans les domaines les plus diverses. L'apprentissage profond est un ensemble de techniques tirant parti des réseaux de neurones pour résoudre des problèmes complexes. Ces techniques sont très utilisées, notamment dans le domaine du traitement d'image, le traitement de séries temporelles (reconnaissance vocale...).

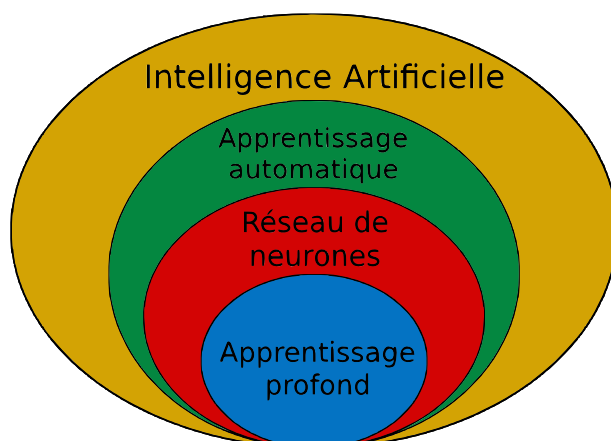


Figure 1 : Schéma de décomposition du domaine de l'intelligence artificielle et de ces sous-domaines

Toutes les techniques d'intelligence artificielle ne sont pas forcément de l'apprentissage automatique. Il est notamment possible de fournir directement à la machine les règles de prédiction sous forme d'une suite de conditions « si-alors ». Ces systèmes, appelés systèmes experts ont été très utilisés durant les années 1980 pour résoudre, avec une grande précision, des tâches données. Dans cet article ne nous intéresserons pas à ces systèmes experts et considérerons uniquement les techniques d'apprentissage automatique.

### 3 – Historique de l'intelligence artificielle

L'idée d'intelligence artificielle prend ses sources dans la première moitié du 20<sup>e</sup> siècle. Dans l'article *Computing Machinery and Intelligence*, publié en 1950, le mathématicien britannique Alan Turing pose la question « les machines peuvent-elles penser ? ». Dans ce même article, il décrit le comportement des machines (ordinateurs au sens moderne du terme) comme un « jeu de

l'imitation » du raisonnement humain. Il définit entre autres le test de Turing, censé déterminer si la machine parvient à simuler le comportement humain. À la même époque et indépendamment, le mathématicien américain Warren Weaver publie un mémo sur la traduction automatique des langues mentionnant l'idée qu'une machine pourrait reproduire un comportement relevant de l'intelligence humaine. À cette même époque, le neurologue américain Warren McCulloch publie un article sur le fonctionnement des neurones et modélise le fonctionnement d'un neurone par un circuit électrique simple. Cet événement marque le début de la recherche sur les réseaux de neurones artificiels. Cependant, les limites technologiques ne permettent pas de forts développements des réseaux de neurones avant de nombreuses années.

L'intelligence artificielle devient officiellement reconnue comme un domaine scientifique en 1956 lors d'une conférence au Dartmouth College aux Etats-Unis. Un des premiers programmes d'intelligence artificielle est présenté en 1959 par Arthur Samuel, un informaticien américain. Il présente un programme jouant aux dames, capable de s'améliorer au fil des parties, battant le quatrième meilleur joueur des Etats-Unis. Le terme d'apprentissage automatique (machine learning) est utilisé pour la première fois lors de la présentation de ce programme.

Cependant, après les années 1950 et 1960, marquées par un fort engouement pour le domaine de l'IA, les avancées n'aboutissent pas et les gouvernements réduisent les financements pour les projets d'intelligence artificielle, privilégiant des domaines donnant des débouchés concrets plus rapidement. Cette période de ralentissement du domaine appelée « l'Hiver de l'IA » durera jusqu'aux années 1980.

Le développement des performances informatiques et l'essor d'Internet au début des années 1990 permet de relancer le domaine de l'intelligence artificielle. L'arrivée de grandes quantités de données (Big Data) permet un entraînement plus précis des systèmes d'IA. Ces systèmes sont alors développés plus globalement et utilisés dans de nombreux domaines comme les diagnostics médicaux. C'est également à cette époque que se développent sérieusement les premiers algorithmes d'apprentissage profond (deep learning), en particulier les réseaux de neurones artificiels.

L'intelligence artificielle profite d'une véritable sortie médiatique à partir de 1997 quand l'ordinateur Deep Blue, développé par IBM, bat le champion du monde d'échecs Garry Kasparov. Depuis cette époque, l'IA s'intègre dans la société, poussée notamment par l'augmentation de la connectivité informatique mondiale. Les projets de recherche en intelligence artificielle se multiplient, devenant plus complexes et s'étendant à de nouveaux domaines, comme la vision par ordinateur et la robotique.

Certaines organisations telles que ImageNet, Human Connectome Project et OpenIA contribuent au développement de l'IA en proposant des bases de données et des algorithmes de bases permettant à tous d'appréhender les concepts de l'intelligence artificielle. De 2010 à 2017, ImageNet organise le concours ILSVRC (ImageNet Large Scale Visual Recognition Challenge), consistant en la conception de logiciels classifiant des objets et des scènes dans des images naturelles. Ces compétitions auront permis de fournir de très importants jeux de données encore utilisés pour tester de nouveaux algorithmes d'apprentissage automatique.

Depuis les années 2010, les réseaux de neurones reviennent sur le devant de la scène, poussés par l'augmentation de la puissance de calcul et par la mise à disposition de bases de données plus importantes, favorisant l'utilisation des algorithmes d'apprentissage profond. Les techniques d'apprentissage profond imposent rapidement leur suprématie dans le domaine de la classification d'image. Basé également sur des réseaux de neurones comme modèles de prédiction, l'ordinateur

AlphaGo bat en 2017 le champion du monde de jeu de Go (alors considéré comme le jeu le plus complexe). L'ordinateur AlphaZero (successeur d'AlphaGo) est alors capable dès fin 2017 de surpasser les joueurs humains des jeux les plus complexes (Go, Échecs, Shogi...) et ceci en apprenant uniquement seul.

De nos jours, l'intelligence artificielle trouve des applications dans des domaines très variés. On retrouve ainsi ces algorithmes utilisés pour le traitement d'image, le traitement automatique du langage (chatbot, GPT3...), la reconnaissance vocale (Alexa, Siri, Cortana, Rasa...) et pour la traduction automatique (pourtant hors de portée des méthodes traditionnelles). De nombreux algorithmes sont également utilisés pour de la prédiction d'évolution temporelle de phénomènes complexes (cours de la bourse, épidémiologie...).

## 4 – Qu'est-ce que l'apprentissage automatique ?

### 4.1 - Définition

Le domaine de l'intelligence artificielle a pour objectif de parvenir à simuler l'intelligence humaine et en particulier l'apprentissage de nombreuses tâches. Deux méthodes sont alors possibles pour apprendre :

- L'apprentissage par cœur consiste à mémoriser explicitement tous les exemples possibles afin de pouvoir les restituer ;
- L'apprentissage par généralisation a pour objectif d'extraire des règles implicites à partir d'une quantité d'exemples afin de les réappliquer à de nouvelles situations jamais rencontrées. L'apprentissage par cœur est relativement aisé pour une machine à condition de disposer des exemples. En revanche, l'apprentissage par généralisation est difficile car il demande d'extraire des règles qui ne sont pas explicitement mentionnées dans les exemples. Ce défi constitue le cœur l'apprentissage automatique.

Comme expliqué précédemment, l'apprentissage automatique est un champ de l'IA porté sur l'analyse statistiques de données d'apprentissage. Historiquement, cette branche est définie comme le développement de machines capables d'apprendre sans avoir été explicitement programmées à apprendre une tâche.

### 4.2 - Exemple



Figure 2 : Exemple de chiffres manuscrits

Extrait de la base de données MNIST, couramment utilisée en apprentissage automatique

Un exemple répandu d'apprentissage automatique est la reconnaissance automatique des chiffres manuscrits. Des exemples de ces chiffres sont présentés figure 1. Ces caractères présentent une grande diversité en forme, orientation, épaisseur de traits... Il est difficile dans ce cas d'écrire une liste explicite de règles permettant de discriminer les chiffres. On va donc chercher à mettre en œuvre un algorithme permettant d'extraire des règles implicites en se basant sur des données d'exemples. Ces règles peuvent par la suite être appliquées sur de nouveaux chiffres afin de les reconnaître.

### 4.3 - Les phases de l'apprentissage automatique

De manière générale, les algorithmes d'apprentissage automatique se séparent en plusieurs phases.

- Phase d'entraînement (ou d'apprentissage) : le modèle choisi est soumis à un grand nombre d'exemples significatifs. Le système cherche alors à apprendre des règles implicites en se basant sur ces données (appelées données d'entraînement). Cette phase d'entraînement précède généralement l'utilisation du modèle, bien que certains systèmes continuent d'apprendre indéfiniment s'ils disposent d'un retour sur les résultats (on appelle cela de l'apprentissage en ligne).
- Phase d'inférence : Le modèle entraîné peut être utilisé sur de nouvelles entrées. Les entrées fournies lors de la phase d'inférence peuvent être traitées même si elles n'ont pas été vues par le modèle lors de la phase d'apprentissage. En effet, grâce à l'extraction de règles implicites, le modèle peut se généraliser à des entrées inconnues.

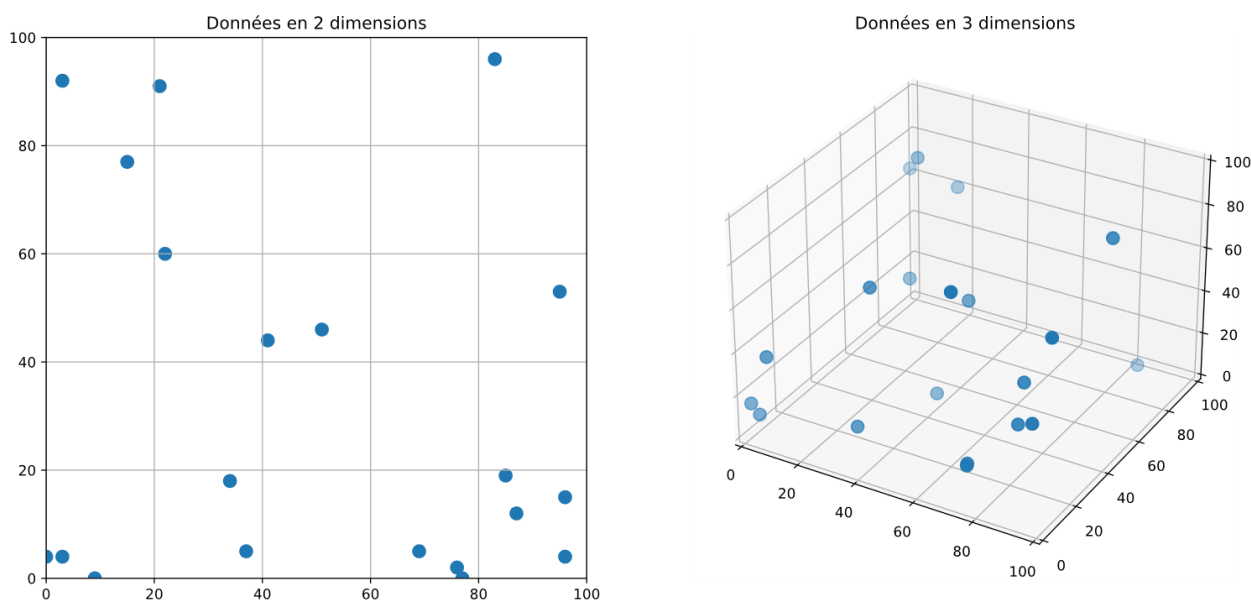
### 4.4 - Applications et limites

En se libérant de l'explicitation de règles, l'apprentissage automatique peut s'étendre à des problèmes complexes dans de nombreux domaines. Des algorithmes d'apprentissage automatique peuvent alors être trouvés notamment dans le domaine de la santé avec l'aide au diagnostic médical et la création de vaccins (en 2019, SAM met au point un vaccin contre la grippe), dans le domaine de la robotique avec de la vision par ordinateur ou des commandes vocales, dans l'industrie avec de la détection de panne et de la gestion de systèmes. On peut également citer les algorithmes d'apprentissage automatique utilisés sur certains systèmes de voitures autonomes, les algorithmes présents dans les jeux (AlphaZero, Stockfish, DeepBlue, AlphaGo...) ou encore des systèmes utilisés en astronomie pour améliorer les observations.

Cependant, bien que l'apprentissage automatique soit flexible et permettent de résoudre des problèmes complexes, il présente quelques limites dont il faut avoir conscience :

- La précision des prédictions n'est jamais de 100 %, il existe donc une probabilité (même faible) de se tromper un jour, ce qui peut être bloquant pour certaines applications ;
- De nombreux algorithmes agissent de manière opaque, il est difficile pour un utilisateur de comprendre comment le système abouti à une prédiction, même si la structure du modèle est connue. Un enjeu majeur de l'IA est de permettre l'explicabilité des modèles ;
- La correction et la détection d'erreur est difficile dans la plupart des cas, ce qui rend parfois le développement et l'entraînement du modèle difficile. Cette limite est directement liée à la question d'explicabilité ;
- La plupart des algorithmes d'apprentissage automatique sont de complexité exponentielle. Ainsi, il peut être nécessaire de faire quelques approximations sur le problème, permettant de gagner en efficacité, au prix de la précision du modèle ;

- Les performances du modèle sont fortement dépendantes des données d'entraînement, ce qui demande au concepteur de choisir des données suffisamment représentatives pour que le modèle soit généralisable ;
- Un problème récurrent dans l'apprentissage automatique est le fléau de la dimension. Il s'agit d'un phénomène arrivant lorsque l'on cherche à analyser des données de grande dimension. La figure 3 illustre ce problème pour des données en 2 et 3 dimensions. Lorsque la dimension de l'espace augmente, son volume croit rapidement, les données qui étaient avant représentatives deviennent éparées et un plus grand nombre de données est alors nécessaire. Ce problème demande alors l'utilisation lorsque c'est possible de méthode de réduction de dimensionnalité (PCA, ICA, NMF...).



*Figure 3 : Illustration du problème du fléau de la dimension  
Pour avoir une bonne représentation statistique il faut un plus grand nombre de données en 3 dimensions*

Il ne faut pas non plus omettre les nombreuses questions éthiques autour de l'apprentissage automatique et même plus généralement autour de l'intelligence artificielle. On peut citer pour cela l'exemple de l'algorithme COMPAS, utilisé par les autorités américaines pour évaluer le risque de récidive des criminels. En 2016, une enquête de l'organisation ProPublica a montré que l'algorithme était biaisé, jugeant les Afro-Américains avec un risque bien plus élevé. Cet exemple montre que les algorithmes d'apprentissage automatique peuvent donner des modèles biaisés et restent dépendant du concepteur et des données accessibles.

## 5 – Les types d'apprentissage automatique

### 5.1 - Apprentissage supervisé

On parle d'apprentissage supervisé lorsque l'on dispose de données d'entraînement étiquetées, c'est à dire dont on connaît la sortie voulue. En notant les  $N$  entrées  $x_i$  et les sorties cibles associées, on dispose de l'ensemble de données  $D = \{x_i, y_i\}_{i \in [1, N]}$ . L'objectif est d'entraîner le modèle choisi pour qu'il puisse prédire correctement la sortie pour des entrées non étiquetées.

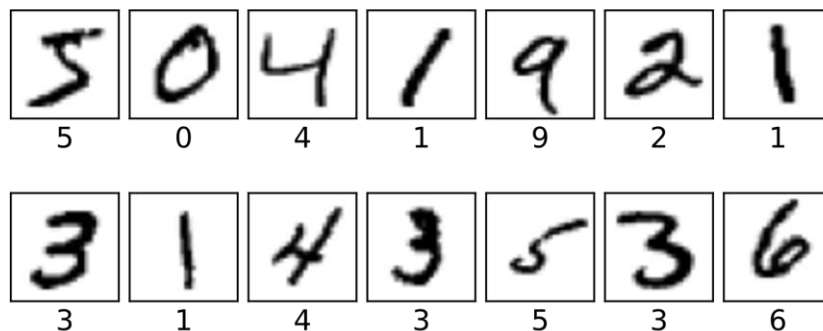


Figure 4 : Ensemble de données étiquetées  
Chaque image d'entrée est associée avec la prédiction voulue

L'apprentissage supervisé est généralement utilisé pour de la régression ou de la classification :

- La régression est utilisée lorsque la sortie à prédire peut prendre des valeurs continues, il s'agit d'une variable réelle. Exemple : un algorithme prédisant la consommation électrique d'une installation ou un algorithme prédisant le cours d'actions en bourse ;
- La classification est une tâche consistant à choisir une classe (valeur) parmi toutes celles possibles Exemple : Un algorithme prédisant le chiffre manuscrit sur l'image d'entrée ou un algorithme classifiant une tumeur comme « bénigne » ou « maligne ».

Il est important de noter que ces deux familles, classification et régression, ne sont pas exhaustives et qu'il existe des algorithmes réalisant des prédictions parmi des vecteurs de grande dimension, entre régression et classification. Les exemples les plus communs de modèles entre régression et classification se retrouvent dans le traitement automatique du langage où les mots peuvent être modélisés comme des combinaisons de lettres, donnant alors lieux à des vecteurs de très grande dimension.

Parmi les algorithmes d'apprentissage supervisé classiques, on peut citer l'algorithme des k plus proches voisins, la régression linéaire, la régression logistique et les modèles tels que les réseaux de neurones, les arbres de décision, les forêts aléatoires et les machines à support de vecteur.

## 5.2 - Apprentissage non-supervisé

On parle cette fois d'apprentissage non supervisé si les données ne sont pas étiquetées. On dispose donc de données d'entrée dont on ne connaît pas la sortie associée. L'ensemble de données est donc  $D = \{x_i\}_{i \in [1, N]}$  et l'objectif du système est d'identifier des caractéristiques communes aux données d'entraînement.



Figure 5 : Ensemble de données non-étiquetées  
Les images d'entrées, extraites de la base de données MNIST Fashion, ne sont pas associées à une sortie cible

L'apprentissage non-supervisé est principalement composé des algorithmes de regroupement (clustering). Ces algorithmes cherchent à séparer les données d'entrée en un nombre donné de groupes. Chaque élément du groupe doit avoir des caractéristiques proches de celles des éléments du même groupe mais des caractéristiques relativement éloignées de celles des autres groupes. Ces algorithmes regroupent donc les entrées en familles afin de les étiqueter automatiquement. Par exemple un algorithme de regroupement peut être utilisé pour regrouper des patients dans le but de prévoir les réactions possibles à certains traitements.

Parmi les algorithmes d'apprentissage non-supervisé les plus courants, on peut citer l'algorithme des k-moyennes, l'analyse en composante principale, le DBSCAN, la décomposition en valeurs singulières et certains réseaux de neurones.

### 5.3 - Apprentissage par renforcement

L'apprentissage par renforcement consiste à apprendre en interagissant avec son environnement. Un système de récompenses permet de renforcer les bons choix du modèle et pénaliser les mauvais choix. Le modèle est initialisé aléatoirement et adapte sa stratégie en fonction de son expérience durant la phase d'entraînement. Il s'agit d'un fonctionnement itératif. À chaque itération le modèle réalise une tâche et un score lui est attribué en fonction de ses performances. Les paramètres du modèle sont ensuite légèrement modifiés et la tâche est répétée. Un meilleur score oriente le choix des paramètres vers certaines valeurs, assurant que le modèle se rapproche à terme de résultats satisfaisants. Le choix des récompenses et pénalités à appliquer est donc très important car il conditionne les résultats du modèle final.

Les algorithmes génétiques, régulièrement utilisés sont un type particulier d'algorithme d'apprentissage par renforcement, se basant sur le principe de la sélection naturelle et de la génétique. Lors de la phase d'apprentissage, de nombreux individus sont créés. Tous sont basés sur le même modèle mais avec des paramètres différents et génèrent donc des comportements différents. Tous les individus sont soumis à la même tâche et un score leur est attribué en fonction de leurs performances. Les meilleurs individus sont sélectionnés et la génération d'après est créée à partir d'une mutation de ces derniers, c'est à dire un léger changement de leurs paramètres. Les individus deviennent alors plus performants à mesure des générations.

Le principe de l'apprentissage par renforcement a notamment été utilisé pour entraîner l'ordinateur AlphaGo, ayant vaincu le champion du monde du jeu de Go en 2019. Le modèle avait alors la possibilité de jouer contre lui-même ou contre d'autres machines afin d'apprendre la stratégie à adopter. On peut aussi appliquer ces algorithmes pour l'entraînement d'un modèle pour la navigation de voiture autonome. Le modèle de comportement est ici complexe et peut être modélisé par le biais d'un réseau de neurone. L'utilisation d'apprentissage profond (réseaux de neurones) est aujourd'hui majoritaire pour l'apprentissage par renforcement en raison de la complexité des modèles. L'apprentissage par renforcement est également couramment utilisé pour des problèmes complexes d'optimisation sous contrainte, en attribuant des récompenses en fonction du critère d'optimisation et des pénalités selon les contraintes voulues.

L'apprentissage par renforcement est une méthode d'apprentissage automatique permettant d'entraîner un modèle en le faisant interagir avec son environnement. L'agent dispose d'un modèle initial qui doit être optimisé et est placé au sein de son environnement (généralement simulé). Il doit alors atteindre son objectif en faisant certaines actions, basée sur des observations, provenant généralement de capteurs. Un système de récompense et de pénalité permet de guider l'agent vers les bonnes actions à réaliser et d'améliorer le modèle utilisé pour la prise de décision. Une autre ressource de de dossier [7] détaille les différentes méthodes d'apprentissage par renforcement.



## 5.4 - Choix d'un modèle de machine learning

Nous avons vu qu'il existe une multitude d'algorithmes d'apprentissage automatique, il est donc question maintenant de se demander lequel utiliser. Un premier critère de choix concerne le type d'apprentissage voulu ou possible. Ainsi, on choisira de faire de l'apprentissage supervisé, de l'apprentissage non supervisé ou encore de l'apprentissage par renforcement en fonction de ce qui est possible pour la tâche voulue et en fonction des données à disposition. Cependant, il reste dans chaque catégorie une multitude d'algorithmes possibles. Il a été démontré qu'il n'existe pas d'algorithme performant mieux que les autres pour tous les problèmes. Ce théorème appelé « No Free Lunch Theorem » déclare que tous les algorithmes d'optimisation ont des performances égales si elles sont moyennées sur tous les problèmes possibles. Autrement dit, si un algorithme A offre de meilleurs résultats qu'un algorithme B sur une tâche, alors il existe obligatoirement au moins une tâche pour laquelle les résultats de A seront moins bons que ceux donnés par B. Cela empêche donc de choisir systématiquement le même algorithme pour toutes les tâches. En effet, type d'apprentissage et le modèle sont choisis en fonction des hypothèses réalisées pour un problème donné. Ces hypothèses ne seront pas validées pour toutes les tâches possibles et le modèle sera alors plus ou moins correct en fonction du problème traité. Une certaine expertise est donc nécessaire pour choisir un modèle à partir de la tâche considérée, des données disponibles et des contraintes qui s'appliquent sur le système (mémoire, puissance de calcul...). De nouvelles méthodes, telles que AutoML, permettent d'aider le concepteur dans le choix d'un modèle adapté à son problème.

Il est important de réévaluer les hypothèses pour chaque problème considéré et d'en déduire quel algorithme sera plus performant dans le cas étudié. Par exemple, pour un problème donné d'apprentissage supervisé, on peut faire l'hypothèse que la sortie prend des valeurs continues et choisir un algorithme de régression ou bien faire l'hypothèse que les valeurs sont discrètes et suffisamment peu nombreuses pour utiliser un algorithme de classification. Dans ce second cas, on peut également supposer que le modèle de classification est suffisamment linéaire pour utiliser l'algorithme des k plus proches voisins ou au contraire opter pour un modèle plus complexe basé sur un réseau de neurones. En somme, le concepteur d'un système d'apprentissage automatique doit analyser les données à disposition et l'objectif voulu pour formuler des hypothèses cohérentes et qui permettront de choisir un algorithme.

Dans les parties suivantes, nous présenterons des algorithmes classiques d'apprentissage supervisé et non-supervisé.

## 6 – Apprentissage supervisé

### 6.1 - Régression linéaire

#### 6.1.1 - Principe de fonctionnement

L'objectif de la régression linéaire est d'exprimer une variable de sortie  $y$  en fonction de la variable d'entrée  $x$  de manière linéaire, c'est à dire  $y = ax + b$ . Ce modèle a donc deux paramètres A et B, dont il faut trouver les valeurs optimales durant la phase d'apprentissage. Plusieurs techniques existent pour estimer ces paramètres, les plus répandues étant la méthode des moindres carrés, la méthode des déviations et la méthode du maximum de vraisemblance.

La régression linéaire est un algorithme d'apprentissage supervisé, on dispose alors de N couples entrée-sortie constituant l'ensemble de données  $D = \{x_i, y_i\}_{i \in [1, N]}$ . Ce sont ces données connues qui vont permettre d'estimer les paramètres du modèle. Nous allons prendre ici l'exemple de la

méthode des moindres carrés, plus répandue que les autres. Le principe des algorithmes de régression linéaire reste similaire pour les autres méthodes.

Dans le cas de la méthode des moindres carrés, on cherche les paramètres A et B permettant de minimiser une fonction coût  $C = \sum_{i=1}^N (y_i - (ax_i + b))^2$ . Cette fonction correspond à la somme des écarts au carré entre les prédictions et les valeurs attendues. Ces écarts à minimiser sont appelés résidus.

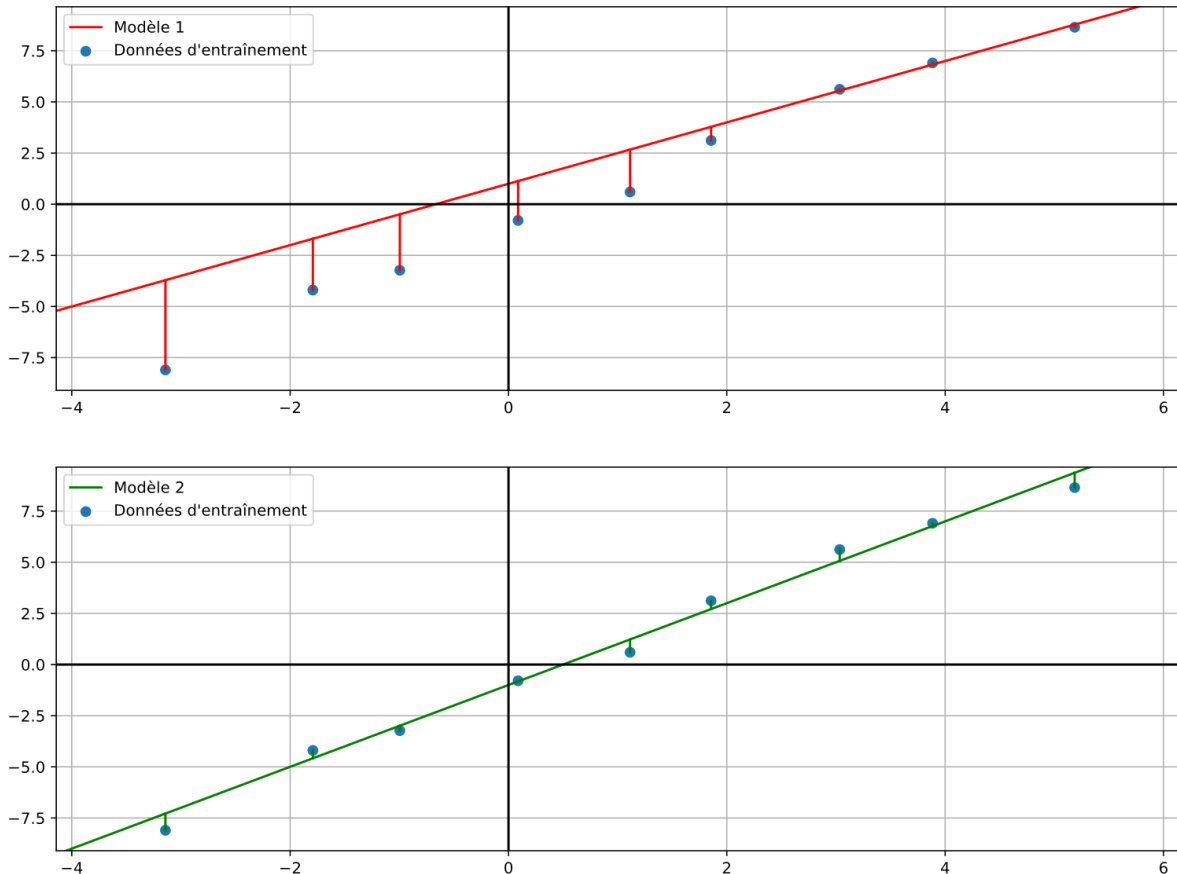


Figure 6 : Deux modèles de régression linéaire

Le premier modèle présente des écarts importants entre les valeurs prédites et attendues tandis que le second minimise les carrés de ces écarts

On peut calculer le coefficient de régression linéaire pour estimer si deux variables peuvent être liées linéairement. Ce coefficient est calculé à partir des écarts types  $\sigma_x$  et  $\sigma_y$  des variables et à partir de la covariance entre les variables d'entrée et de sortie :  $R = \frac{cov(x,y)}{\sigma_x\sigma_y}$  (Si le coefficient est proche en valeur absolue de 1, cela signifie que les variables peuvent être liées linéairement. Il s'agit donc d'un moyen pour vérifier que la régression linéaire est un choix cohérent de modèle.

### 6.1.2 - Applications

Les applications de la régression linéaire sont nombreuses. On peut par exemple imaginer une mesure du courant traversant une résistance pour différentes valeurs de tension appliquée à ses bornes. Le mesure de courant peut être bruitée, empêchant de déterminer directement la valeur de la résistance traversée. On peut utiliser une régression linéaire pour estimer la valeur de cette résistance en minimisant l'écart entre le courant mesuré et la valeur estimé par le modèle  $\$U=RI\$$ . Cette même méthode s'applique pour estimer des paramètres dans de nombreux domaines.

En pratique, il existe de nombreux logiciels permettant de réaliser une régression linéaire. La plupart des tableurs disposent de cette fonctionnalité et certains langages de programmation proposent des bibliothèques permettant de faire cette régression (scikit learn pour Python, opérateur `\` pour Matlab...).

### 6.1.3 - Limites

Bien qu'il ait l'avantage d'être facilement et rapidement implémentable, le modèle de régression linéaire présente de nombreuses limites. Tout d'abord, comme nous l'avons vu, cette méthode est limitée à l'estimation de modèles linéaires, ce qui restreint énormément les applications possibles. Le principe de la régression linéaire peut cependant s'étendre à des régressions polynomiales. Ces dernières sont plus complexes à mettre en œuvre car elles demandent plus de paramètres (et de choisir le nombre de paramètres) et font face aux problèmes de sur-apprentissage et de sous-apprentissage, qui seront abordés plus loin dans cette ressource.

D'autre part, l'estimation obtenue est sensible au bruit de mesure introduit dans les données d'apprentissage. Ainsi, une donnée aberrante (très éloignée du modèle à estimer) influera sur les estimations des paramètres. Cette influence est plus ou moins importante selon la méthode de régression linéaire utilisée. En effet, la méthode des moindres carrés considère le carré des résidus et une valeur aberrante décalera donc plus le modèle que dans le cas de la méthode des moindres déviations (qui utilise les valeurs absolues des résidus). Certaines méthodes permettent d'intégrer des connaissances a priori, limitant davantage l'effet de valeurs aberrantes.

Afin de réduire l'effet du bruit sur l'estimation, il est important d'avoir un nombre suffisamment important de données, ce qui contraint également le protocole expérimental.

## 6.2 - Algorithmes des K plus proches voisins

### 6.2.1 - Principe

L'algorithme des K plus proches voisins, comme la régression linéaire, est un algorithme d'apprentissage supervisé, on dispose donc de l'ensemble de données  $D = \{x_i, y_i\}_{i \in [1, N]}$ . En revanche, il s'agit ici d'un modèle non paramétrique, c'est à dire que le modèle ne dispose pas de paramètres dont la valeur serait à optimiser (nous verrons plus tard que K n'est pas un paramètre mais un hyperparamètre). Il s'agit également d'un algorithme de classification, les sorties  $y_i$  correspondent alors aux classes possibles (et non pas des valeurs continues comme pour la régression).

L'entrée  $x_i$  est un vecteur de dimension  $p$ , contenant les différentes variables associées à une entrée. Pour chaque nouvelle entrée  $u$ , on mesure la distance entre  $u$  et les entrées  $x_i$  pour tout  $i \in [1, N]$ . On sélectionne ensuite les K éléments les plus proches (distance plus faible) de l'entrée. La classe prédite par l'algorithme correspond alors à la classe la plus fréquente dans les K plus proches voisins sélectionnés.

On peut utiliser différentes mesures de distance (euclidienne, Manhattan...) pour cet algorithme en fonction du problème étudié.

### 6.2.2 - Applications

Un exemple célèbre et couramment utilisé pour les K plus proches voisins se base sur la base de données Iris de Fisher. En 1936, le botaniste Edgar Anderson a mesuré 150 spécimens d'iris de trois espèces variétés différentes, les Iris Setosa, les Iris Versicolor et les Iris Virginica. Pour chaque spécimen, il a mesuré la longueur et la largeur des pétales et des sépales. Dans le cadre de cet

exemple, nous allons nous intéresser uniquement à la longueur et largeur des pétales. Les entrées  $x_i$  sont donc vecteurs contenant la longueur des pétales et la largeur des pétales. Les sorties  $y_i$  sont un entier 0, 1 ou 2, correspondant à la classe du spécimen (setosa, versicolor ou virginica).

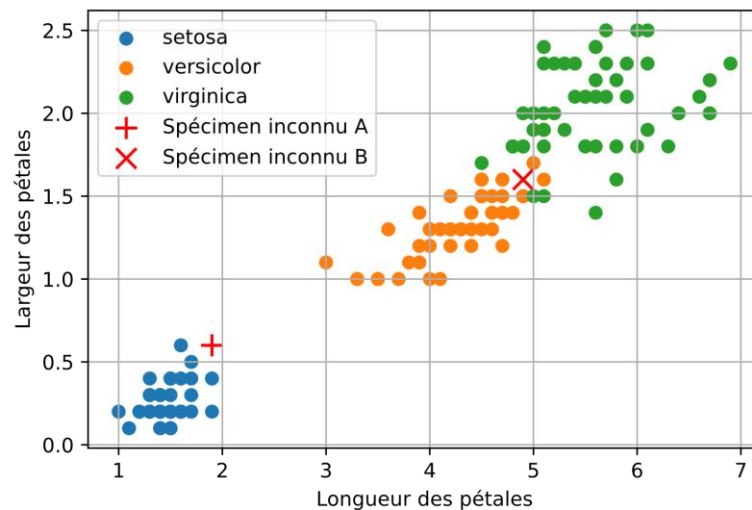


Figure 7 : Bases de données Iris de Fisher

Les 150 spécimens sont répartis en 3 variétés et positionnés en fonction de la longueur et largeur des pétales

On peut observer que les spécimens sont regroupés en fonction de leur variété. En effet, des spécimens d'une même variété ont des caractéristiques similaires. Il semble alors cohérent de prédire l'appartenance d'un spécimen à une variété selon les autres spécimens proches (au sens de la distance euclidienne). Considérant pour cet exemple deux spécimens inconnus A et B et cherchons à prédire la variété à laquelle ils appartiennent.

- Le premier spécimen, appelé A est représenté par un '+' sur le graphique. Il s'agit d'une iris avec des pétales de 1,9 cm de longueur et 0,6 cm de largeur. Cette iris a des caractéristiques plus proches de celle des spécimens d'Iris setosa que de celle des autres spécimens. On peut alors classer cette iris dans la variété setosa.
- Le second spécimen, appelé B, a des pétales de longueur 4,9 cm et de largeur 1,6 cm et est représenté par un 'x' sur le graphique. On peut alors regarder les 5 spécimens d'iris connus les plus proches du spécimen B. On peut constater que parmi ceux-là il y a 1 Iris virginica et 4 Iris versicolor. On classe alors le spécimen B dans la variété versicolor.

Nous avons donc pu classer deux spécimens inconnus en utilisant la méthode des K plus proches voisins.

Notons que nous avons utilisés ici l'algorithme dans le cadre de classification mais celui-ci peut être étendu à de la régression. Dans ce cas, les éléments ne sont pas associés à une classe mais à une valeur. La sortie de l'algorithme correspond alors à la valeur moyenne des K plus proche voisins

### 6.2.3 - Applications et limites

L'algorithme des K plus proches voisins est fréquemment utilisé en classification comme en régression. On peut alors le retrouver dans le cadre de la reconnaissance de forme avec des entrées contenant la circonférence, l'aire ou encore la signature du contour de la forme et des sorties correspondant aux différentes formes possibles. Cet algorithme a l'avantage d'être relativement robuste si suffisamment d'exemple d'entraînement sont fournis.

Cet algorithme est également facilement implémentable dans différents langages de programmation.

Le choix de la valeur de K dans l'algorithme des K plus proches voisins a une influence forte sur la prédiction faite par l'algorithme :

- K trop petit : des éléments sortant de l'ordinaire influenceront plus facilement la prédiction. La généralisation du modèle pour de nouveaux éléments sera donc moins bonne, il s'agit du problème de sur-apprentissage (overfitting). Par exemple, si l'on choisit  $k=1$ , un Iris versicolor ayant des pétales anormalement petits sera pris en compte par le modèle qui risque de mal classer une iris aux dimensions trop proche de cette valeur aberrante.
- K trop grand : Le modèle prendra en compte des données trop éloignées et la classe majoritaire sera prédite trop souvent. Il s'agit cette fois du problème de sous apprentissage, le modèle n'utilise que trop peu les données d'entraînement. Par exemple, si l'on choisit  $K=N$ , toutes les iris seront prises en compte et la classe prédite sera la même peu importe la taille des pétales.

Ainsi, le choix de la valeur de K n'est pas aisé, il ne suffit pas de la prendre grande ou de la prendre petite. Différentes méthodes existent pour avoir une idée de la valeur de K possible. Il est également important de tester l'algorithme sur un ensemble de données connues afin de vérifier le fonctionnement de ce dernier. On peut alors séparer l'ensemble de données connues  $D$  en deux ensemble, l'ensemble d'entraînement  $D_e$  et l'ensemble de test  $D_e$ . Le premier permettra de réaliser la prédiction et le second permettra de vérifier les performances de l'algorithme et d'éviter les phénomènes de sur-apprentissage et de sous-apprentissage.

Un des principaux problèmes rencontrés lors de l'implémentation de l'algorithme des K plus proches voisins est le fléau de la dimension. En effet, pour que l'algorithme fonctionne de manière optimale, il faut un nombre suffisant de données d'entraînement afin que les points étudiés soient toujours proches d'exemples connus. Lors de l'étude de problèmes de plus grande dimension, il est donc primordial d'avoir un grand nombre de données d'entraînement. Pour cette raison, l'algorithme des K plus proches voisins devient rapidement inutilisable (au-delà de 4 ou 5 dimensions, le nombre de données nécessaires devient trop important).

### 6.3 - Autres algorithmes d'apprentissage supervisé

Il existe un grand nombre d'algorithmes d'apprentissage supervisé. Hormis les algorithmes de régression linéaire ou polynomiale et l'algorithme des K plus proches voisins, déjà mentionnés précédemment, on peut citer les suivants :

- Régression logistique : On cherche ici à prédire le comportement d'une variable de sortie  $Y$  en fonction des entrées  $X$ , comme en régression linéaire ou polynomiale. Cependant, dans le cadre de la régression logistique ordinaire,  $Y$  n'est pas un réel mais prend les valeurs 0 ou 1 (il s'agit d'une variable binaire). Elle permet d'évaluer l'effet de différentes variables explicatives sur une variable d'intérêt. Cette méthode d'apprentissage automatique est très utilisée dans le domaine du marketing pour évaluer la vente ou non de produits à la suite d'une décision ou dans le domaine médical pour évaluer la guérison ou non d'un patient.
- Arbre de classification : Les arbres de classification (ou arbre de décision) et de régression sont des méthodes permettant de créer un modèle de décision. Cette structure est constituée de nœuds racines, constituant les entrées de l'arbre, de nœuds internes, réalisant des opérations intermédiaires et de feuilles (ou nœuds terminaux), représentant la valeur des variables de sorties. Cette structure hiérarchique se base sur un enchaînement de décisions créant le passage d'un nœud à l'autre, jusqu'aux feuilles. L'arbre est généralement construit récursivement en cherchant pour chaque nœud la décision permettant de réaliser le meilleur partage de l'ensemble de données. Après une décision,

on dispose alors de plusieurs sous-ensembles de données sur lesquels on applique le même processus. Les arbres de décision ont l'avantage d'avoir un processus de décision clair, garantissant une bonne explicabilité de la solution. Ils ont longtemps été utilisés avant le développement des réseaux de neurones artificiels. Il est possible de décomposer les problèmes plus complexes en utilisant plusieurs arbres de classification, on parle alors de forêt aléatoire.

- Réseaux de neurones artificiels : Les réseaux de neurones sont des modèles complexes pouvant être utilisés en apprentissage supervisé pour de la régression et de la classification. Ces modèles sont expliqués en détail dans une autre ressource de dossier [6]. Ils sont de plus en plus utilisés dans de très nombreux domaines car ils permettent de modéliser des phénomènes complexes. On les retrouve par exemple en traitement d'image pour de la reconnaissance de motif, dans le traitement de la parole et pour l'approximation de fonctions complexes longues à calculer.
- Machine à support de vecteur (SVM) : Les SVM constituent une généralisation des classificateurs et estimateurs linéaires (tels que les modèles de régression linéaire) pour des données de plus grande dimension. Le modèle prend alors en entrée un vecteur de paramètre  $X$  et lui associe une valeur de sortie  $Y$  grâce à une fonction  $h(X)$ . Dans le cas de classification, on introduit des hyperplans séparateurs servant de critère de décision. L'espace de la valeur de sortie  $Y$  est séparé en différentes zones correspondant chacune à des prédictions. Par exemple, on prédit une classe si la valeur de  $Y$  est positive et une autre si elle est négative.

## 7 – Apprentissage non-supervisé

### 7.1 - Algorithmes des k-moyennes

#### 7.1.1 - Principe

Le premier exemple d'apprentissage non-supervisé que nous allons voir est l'algorithme des k-moyennes. Nous disposons cette fois de données  $D = \{x_i\}_{i \in [1, N]}$  non-labelisées que nous allons chercher à classer automatiquement. Les k-moyennes est un algorithme de regroupement non supervisé qui cherche à créer k groupes d'éléments de caractéristiques proches. Afin de sélectionner les éléments proches pour les rassembler en un groupe, l'algorithme cherche à minimiser la variance intra-classe, c'est à dire la distance euclidienne entre les éléments d'une classe et son barycentre. En notant  $m_1, m_2, \dots, m_k$  les barycentres des classes  $C_1, C_2, \dots, C_k$ , on cherche alors :

$$m_1, m_2, \dots, m_k \text{ tels que } \sum_{i=1}^k \sum_{j \in C_i} \|x_j - m_i\|^2 \text{ est minimal}$$

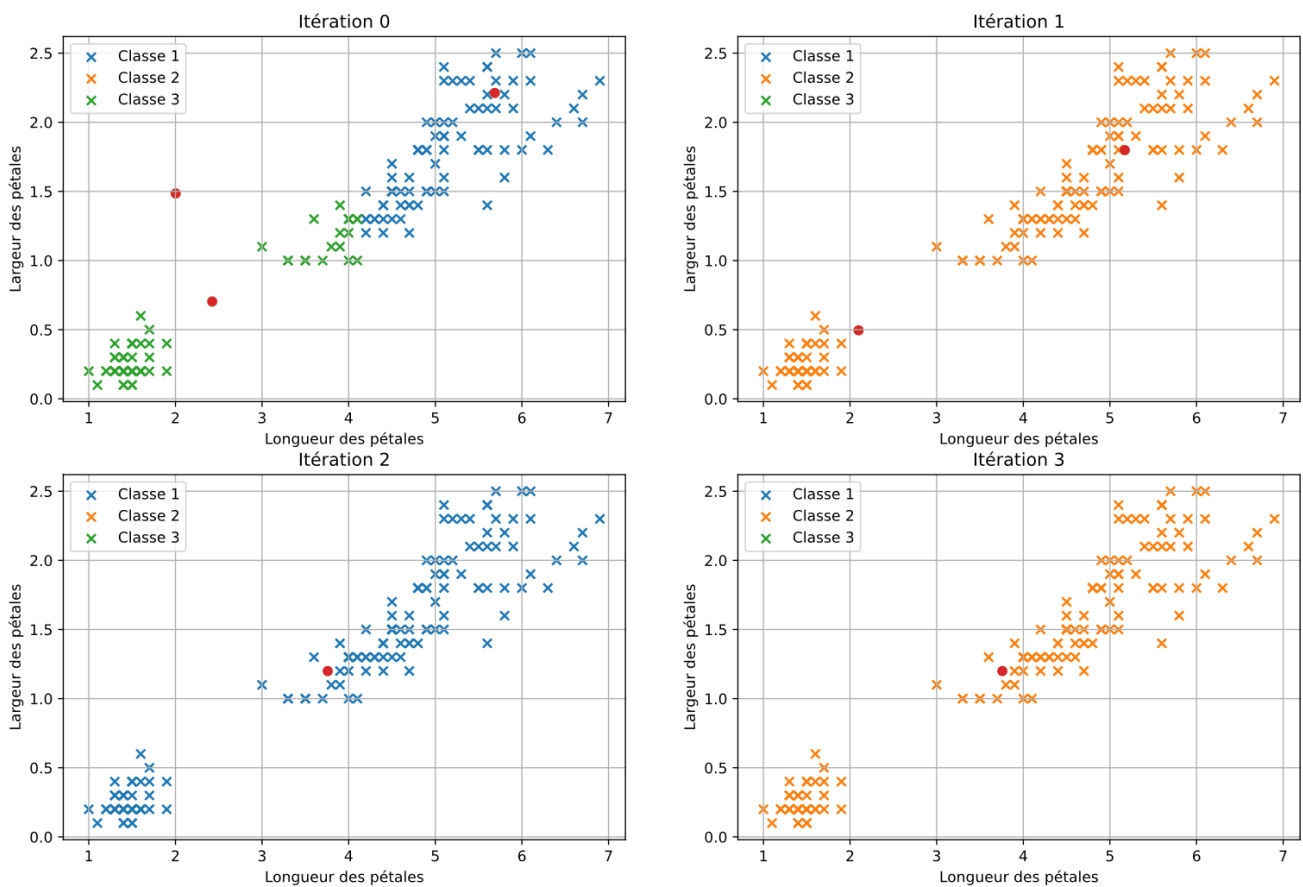
Le déroulement de l'algorithme est le suivant :

- On choisit k positions initiales correspondant au centre des classes  $C_1, C_2, \dots, C_k$ , souvent aléatoirement ;
- On associe chaque élément de la base de données  $D_e$  à la classe dont il est le plus proche. Les données sont alors réparties en k groupes (certains groupes peuvent être vides) ;
- Le barycentre des éléments de chaque groupe est recalculé et devient le nouveau centre des classes  $C_1, C_2, \dots, C_k$  ;
- On répète les étapes précédentes jusqu'à ce que les centres ne bougent plus entre deux itérations, il y a convergence.

#### 7.1.2 - Exemple

On peut utiliser à nouveau la base de données Iris de Fisher, en se plaçant cette fois dans le cas où les labels ne sont pas disponibles. On a alors les mesures de tous les spécimens mais nous ne

connaissons pas leur appartenance aux trois variétés différentes. Il est alors possible de mettre en œuvre l'algorithme des k-moyennes pour séparer les spécimens dans k classes, nous permettant alors de retrouver approximativement les appartenances aux variétés. Dans notre cas, nous allons utiliser  $k = 3$ , supposant que le nombre de variétés à extraire est connu. Il est possible d'appliquer le même algorithme pour un nombre k différent, ce qui donnera alors un nombre différent de classe.



*Figure 8 : Résultat de l'algorithme des k-moyennes sur la base de données Iris de Fisher. On utilise ici  $k=3$  et on observe que les centres ne changent pas entre les deux dernières itérations.*

On peut observer la convergence rapide de l'algorithme, les centres des classes ne bougent plus au bout de 3 itérations. On obtient un résultat semblable à celui attendu (présenté en partie Apprentissage Supervisé - K plus proches voisins). Cependant, on peut observer que les spécimens sortant de l'ordinaire (par exemple avec des pétales longs pour son espèce) se retrouvent dans la mauvaise classe car ils sont trop éloignés de la moyenne de leur espèce.

### 7.1.3 - Applications et limites

Comme nous avons pu le souligner en étudiant un exemple d'application, cet algorithme présente plusieurs limites dont il faut avoir conscience. Tout d'abord, le choix du nombre de classe k à utiliser est particulièrement important et ce choix peut souvent être difficile. Dans le cas où l'on choisit une valeur de k trop petite, pas assez de classes seront créées et des éléments seront regroupés à tort dans une même classe. À l'inverse, un choix de k trop grand créera trop de classe, séparant des éléments qui aurait dû être regroupés. Le choix de cette valeur dépend alors fortement de l'application et du cahier des charges associé. Il existe cependant des méthodes comme la méthode d'Elbow, permettant de choisir une valeur optimale de k en comparant les valeurs obtenues pour plusieurs k.

Nous avons également vu que cet algorithme ne considère pas les éléments éloignés de la moyenne de leur classe, ce qui classe donc un certain nombre d'éléments dans la mauvaise classe. Ce phénomène peut être réduit en prenant en compte plus de caractéristiques pour les éléments. Les éléments anormaux auront également tendance à faire dévier le barycentre, faisant potentiellement perdre en précision.

De plus, le choix initial des centres n'est pas toujours simple. Bien qu'une initialisation aléatoire puisse fonctionner, ce n'est pas toujours le cas et l'algorithme peut alors converger vers des groupes différents de ceux attendus.

Enfin, une des limites les plus restrictive de l'algorithme des K moyennes concerne la taille et la forme des clusters à identifier. En effet, cet algorithme est peu adapté pour des groupes de tailles différentes et ne fonctionne pas pour des groupes de formes différentes.

L'algorithme des k-moyennes est utilisé dans de nombreuses applications, comme le groupement de la clientèle permettant d'adapter les offres en se basant sur différents critères, l'extraction de couleur (par exemple sur l'histogramme d'une image) ou encore l'apprentissage semi-supervisé. En effet, il est possible d'imaginer des algorithmes à la frontière entre apprentissage supervisé et non supervisé. On dispose alors d'un ensemble de données étiquetées (en petite quantité) et de données non-étiquetées (en plus grande quantité). On peut alors créer des groupes en utilisant les techniques d'apprentissage non-supervisé puis étiqueter ces données en se basant sur la partie étiquetée de la base de données. Ce type d'apprentissage permet de limiter le travail humain nécessaire pour créer les bases de données permettant l'apprentissage supervisé et permet donc de traiter un plus grand nombre de données, améliorant de ce fait les performances des modèles.

## 7.2 - Autres algorithmes d'apprentissage non-supervisé

Il existe plusieurs autres algorithmes d'apprentissage non-supervisé, utilisés dans différentes circonstances. Il est important de noter que ces algorithmes ne sont pas tous adaptés aux mêmes applications et certains seront plus performants que d'autres pour l'application voulue. Parmi les algorithmes couramment utilisés, on peut citer les suivants :

- **Modèle de mélange de gaussien (GMM) :** Il s'agit d'un modèle statistique permettant d'estimer la distribution de variables aléatoires en la modélisant comme une somme de distributions gaussiennes de paramètres différents. L'étape d'entraînement cherche alors à déterminer la moyenne et la variance des distributions gaussiennes en utilisant un critère de maximum de vraisemblance. Ces modèles sont fréquemment utilisés dans des problèmes de classification comme la reconnaissance de l'écriture manuscrite et la segmentation d'images par leur histogramme. Le modèle de mélange de gaussien permet, contrairement à l'algorithme des K moyennes, de réaliser des groupes de tailles différentes.
- **Analyse en composante principales (PCA) :** Cet algorithme permet de réduire les dimensions d'un ensemble de données, permettant alors un traitement plus simple et plus rapide. Cette méthode crée des variables décorréelées en réalisant un changement de base. Les vecteurs de cette nouvelle base, appelés composantes principales, sont ensuite sélectionnés pour ne garder que celles portant le plus d'informations. Il est alors possible de minimiser la perte d'information tout en réduisant efficacement la dimensionnalité de l'ensemble. Cet algorithme est largement utilisé en neuroscience ou en analyse des mouvements où de nombreux signaux de grande dimensions (activation de neurones ou mouvements des membres) sont utilisés pour traduire des informations plus simples (réponse à un stimuli ou déplacement dans une direction). D'autres algorithmes de réduction de dimensionnalité



existent, comme la décomposition en valeurs singulières ou l'analyse en composantes indépendantes (ICP).

## 8 – Choix et influence des hyperparamètres

Nous allons nous intéresser dans cette partie à la notion de paramètres et d'hyperparamètres, fondamentale en apprentissage automatique. La différence entre ces deux termes est fondamentale et permet de comprendre comment les choix du concepteur influent sur le comportement final du système.

- Les hyperparamètres sont des réglages du modèle et de l'algorithme d'apprentissage, ils décrivent le déroulement de l'algorithme et sont généralement choisis par le concepteur. Ils conditionnent les limites de fonctionnement et les performances du modèle et influent sur l'efficacité de l'apprentissage.  
Exemple : Valeur de  $K$  dans l'algorithme des  $K$  plus proches voisins, nombre de couches d'un réseau de neurones, degré du modèle d'une régression polynomiale.
- Les paramètres sont des valeurs variables dans un modèle, ils définissent son comportement. Les valeurs des paramètres changent au cours de l'apprentissage afin de donner au modèle le comportement voulu, permettant de répondre à la tâche voulue. Exemple : Valeurs de  $A$  et  $B$  dans le modèle de régression linéaire, valeurs des poids et des biais d'un réseau de neurones.

Certains algorithmes d'apprentissage automatique demandent de choisir de nombreux hyperparamètres, il est donc très important de comprendre l'influence de chacun afin d'adapter les choix à l'application voulue. Il est par exemple possible de se demander quel degré choisir pour le modèle d'une régression polynomiale. Le modèle donnera des résultats généralement différents si l'on utilise un modèle linéaire ou un modèle d'ordre 3. Nous devons ici choisir un hyperparamètre, ce choix est particulier à la tâche visée et un choix sera rarement bon pour plusieurs applications différentes. Un mauvais choix d'hyperparamètre peut détériorer l'étape d'apprentissage, conduisant à des paramètres qui ne sont pas optimaux et donc à un modèle imparfait. Il est fréquent de rencontrer des problèmes de sous-apprentissage et de sur-apprentissage.

- Sous-apprentissage : Il s'agit d'un cas où le modèle n'apprend pas suffisamment et se retrouve alors incapable de généraliser efficacement sa prédiction à des entrées inconnues. On peut dans ce cas voir que les performances sur les données d'entraînement et de validation sont médiocres. Ce phénomène arrive généralement lorsque la phase d'apprentissage est trop courte ou lorsque le modèle dispose de trop peu de paramètres, ces deux raisons étant généralement liées aux choix d'hyperparamètres.
- Sur-apprentissage : Dans ce cas, le modèle n'extraie pas les règles implicites mais apprend « par cœur » les données d'entraînement. Il est à nouveau incapable de généraliser efficacement ses prédictions à de nouvelles entrées. Ici on obtient une très bonne précision pour les données d'entraînement et de validation mais les tests donnent des résultats beaucoup moins bons. Ce phénomène arrive généralement quand un modèle contient plus de paramètres que nécessaire à cause d'un mauvais choix d'hyperparamètres. Un ensemble de données d'entraînement trop peu représentatives (ou trop peu nombreuses) conduit également à une forme de sur-apprentissage, le modèle ne se généralisant pas correctement pour des exemples inconnus.

La ressource traitant de l'apprentissage profond [8] présente un exemple plus détaillé de sur-apprentissage et de sous-apprentissage induits par un mauvais choix des hyperparamètres du réseau de neurones.

## 9 – Conclusion

Nous avons essayé de donner dans cette ressource une introduction à la fois large et utile à l'intelligence artificielle. Le domaine est vaste et requiert certains prérequis. Nous allons résumer ici quelques points clés à retenir :

- L'IA tente de reproduire l'intelligence pour aider l'humain dans de multiples domaines ;
- Apprentissage supervisé : le jeu de données d'entraînement est composé d'éléments étiquetés manuellement par des humains et la machine extrait des règles implicites lui permettant d'étiqueter de nouveaux exemples ;
- Apprentissage non-supervisé : le jeu de données d'entraînement n'est pas étiqueté l'algorithme cherche alors de la similarité dans les données pour pouvoir les classer ;
- Apprentissage par renforcement : un agent apprend en interagissant avec son environnement grâce à un système de récompenses favorisant les bonnes actions ;
- L'IA est un paradigme puissant pour modéliser des systèmes complexes, mais pas magique. Voici un rappel de quelques limitations auxquelles ces systèmes font face :
  - Le « No Free Lunch Theorem » indique qu'il faut choisir un algorithme spécifique pour résoudre la tâche voulue, il n'existe pas de méthode universelle ;
  - Les résultats sont toujours probabilistes et il existe toujours une possibilité d'erreur ;
  - Les phénomènes de sous-apprentissage et de sur-apprentissage demande de prendre des précautions lors du choix du modèle et des données d'entraînement ;
  - Le jeu de donnée d'entraînement doit être représentatif et non-biaisé.

Les nombreuses techniques d'apprentissage automatique sont présentes dans un grand nombre de domaines d'application. Il est important de comprendre le principe de fonctionnement des techniques utilisées afin de s'assurer de la pertinence d'une méthode pour un problème donné.

### Références :

[1]: Deep Learning, Ian Goodfellow and Yoshua Bengio and Aaron Courville, MIT Press, 2016

[2]: Artificial Intelligence: A modern approach, 2<sup>nd</sup> Ed., Dr. Afşar Saranlı and Stuart Russel and Peter Norvig, 201

[3]: Pattern Recognition and Machine Learning, Christopher M. Bishop, 2006

[4]: Intelligence artificielle : triomphes et déceptions, Melanie Mitchell, postface de Douglas Hofstadter, traduit de l'anglais (Etats-Unis) par Christian Jeanmougin, Dunod, 2021

[5]: Informatique tronc commun - ITC - PSI, MP, PC, Steeven Janny, Wenqi Shu-Quartier, Théodore Chérière et Jeanne Redaud, Ellipses, 2022

[6]: Personalized Human-Swarm Interaction Through Hand Motion, M. Macchini, L. de Matteis, F. Schiano, D. Floreano, mars 2021, <https://arxiv.org/abs/2103.07731>

[7]: Dossier Intelligence Artificielle, juillet 2022, [https://eduscol.education.fr/sti/si-ens-paris-saclay/ressources\\_pedagogiques/dossier-intelligence-artificielle](https://eduscol.education.fr/sti/si-ens-paris-saclay/ressources_pedagogiques/dossier-intelligence-artificielle)

[8]: Introduction à l'apprentissage profond, S. Janny, L. de Mateis, W. shu-Quartier, juin 2022, [https://eduscol.education.fr/sti/si-ens-paris-saclay/ressources\\_pedagogiques/introduction-lapprentissage-profond](https://eduscol.education.fr/sti/si-ens-paris-saclay/ressources_pedagogiques/introduction-lapprentissage-profond)

Ressource publiée sur Culture Sciences de l'Ingénieur : <https://eduscol.education.fr/sti/si-ens-paris-saclay>