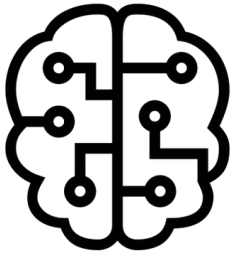


L'intelligence artificielle



Depuis plusieurs années les grandes firmes de l'internet (GAFA) investissent énormément d'argent dans les recherches portant sur l'intelligence artificielle. Des chercheurs français comme Yann Le Cun (très connu pour ces publications sur les neurones profonds) se sont particulièrement illustrés par leurs recherches et sont devenus très sollicités par les médias pour vulgariser leurs travaux auprès du grand public. Mais comment expliquer simplement la modélisation mathématique d'un concept aussi abstrait que l'intelligence ?

Ce terme d'intelligence artificiel est devenu finalement un four-tout très souvent utilisé à défaut. En fait il s'agit de la faculté de reproduire un "raisonnement" par des moyens informatiques. C'est l'ordinateur qui "pense" pour reconnaître, s'adapter à des situations comme le ferait un être humain.

A l'heure actuelle cette IA peut être très performante mais est limitée à une analyse simple des choses (reconnaître un caractère écrit, reconnaître la parole...) et est encore très loin de pouvoir rivaliser avec l'intelligence d'un enfant.



C'est logique ! Où il y a de l'intelligence artificielle, il a forcément de la stupidité artificielle.

1. Technique de création d'intelligence artificielle

Cette faculté de décision peut être obtenue par deux techniques distinctes :

→ l'IA descendante (méthode experte) :

imitation fidèle d'un comportement observé et qui est reproduit à l'identique à l'aide d'un programme informatique. Cette technique a beaucoup évolué dans les années 1980, elle est très performante dans son domaine mais reste confiné dans celui-ci, sans possibilité d'évoluer. Cette méthode est capable à partir de données d'entrée et par application de règles apprises, de déduire une conclusion logique. Elle nécessite que tous les champs de connaissance soient décrits de manière exhaustive, faute de quoi on s'expose à des conclusions erronées.

→ l'IA ascendante (méthode neuronique) : cette fois le comportement humain est mimé par suite d'apprentissage et accumulation de connaissances de plus en plus complexes. Les algorithmes évoluent et échappent progressivement au cadre fixé au départ par leurs auteurs. Ce domaine est en pleine évolution depuis les années 2000 et bénéficie de méthodes d'apprentissage de plus en plus performantes.

Approche basée sur :

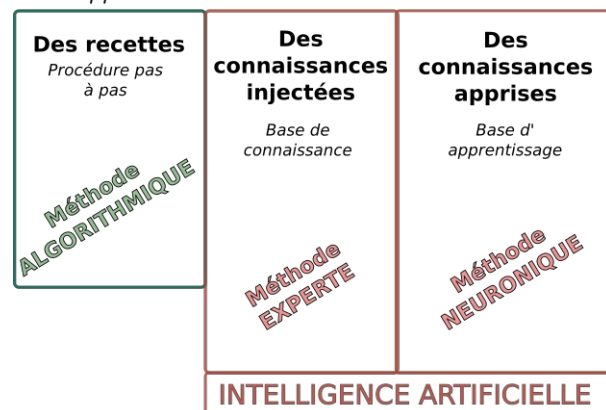


Illustration 1: Méthodes de création d'IA



2. L'algorithme des K plus proches voisins

L'algorithme des K plus proches voisins s'écrit en abrégé K-NN ou KNN, de l'anglais k-nearest neighbors où k est un nombre positif généralement petit.

Il s'agit d'un algorithme de machine learning (apprentissage machine) qui est essentiel dans le milieu de l'intelligence artificielle. Le principe peut être résumé par '*Dis moi qui sont tes amis et je te dirai qui tu es*'.

Ce genre d'algorithme permet par exemple de prédire le comportement d'une personne en s'intéressant à son milieu. Il peut être utilisé par les géants de la vente comme Amazon, Netflix ... afin de prévoir si vous seriez ou non intéressé par un produit. En effet, en disposant de vos données (âge, derniers achats, ...) et en les comparant à celles de clients similaires, un algorithme peut tâcher de prédire si vous seriez intéressé ou non par le produit.

Q1. A partir de la description précédente, **indiquer** la méthode d'analyse utilisée par l'algorithme K-NN (méthode experte ou méthode neuronique). **Justifier** la réponse.

3. Classification de joueurs de rugby

Le rugby est un sport collectif où se côtoient 30 joueurs de morphologies très variées. Cependant quelques soient les équipes, les joueurs évoluant à un poste donné sont tous de même morphotype. Par exemple les ailiers sont très souvent pas très grand et légers alors que les deuxième ligne sont très grands et assez lourds. Il y a donc un lien direct entre la stature du joueur et le poste sur lequel il évolue.

Objectif de l'activité : Déterminer le meilleur poste à attribuer à un nouveau joueur de rugby en fonction de ses caractéristiques morphologiques (poids, taille).

Principe de classification

Le site <http://www.allrugby.com> référence la taille et le poids des différents joueurs du Top14 ainsi que leur poste sur le terrain au cours de la saison 2019-2020. A partir de ce jeu de données, il est possible de représenter sous forme graphique les postes des joueurs de Toulouse selon leur morphologie :

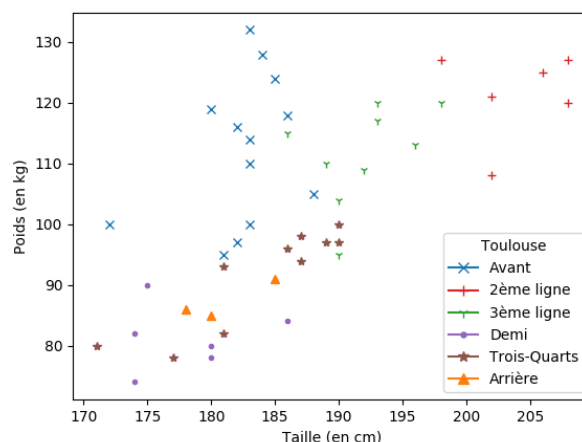


Illustration 2: Poste des joueurs de Toulouse selon leur gabarit



Principe de prédiction de l'algorithme KNN

Pour classifier un nouvel individu, ici attribuer le meilleur poste selon son gabarit, il suffit de rechercher dans le jeu de données le joueur dont la morphologie est la plus proche et d'attribuer au nouveau joueur le même poste que ce joueur.

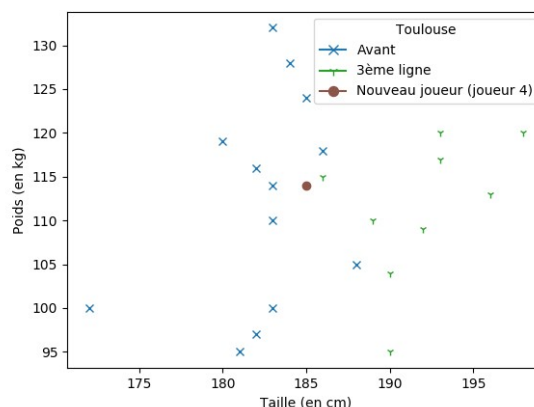
Q2. A partir de cette méthode de prédiction, **déterminer** le meilleur poste à attribuer aux nouveaux joueurs suivants :

	Taille (m)	Masse (kg)
Joueur 1	1,95	112
Joueur 2	1,75	91
Joueur 3	1,80	90
Joueur 4	1,85	114

Q3. Critiquer le résultat obtenu pour le joueur 4.

Influence du facteur K

La figure ci-dessous détaille la zone d'étude du joueur 4



La classification peut être améliorée en tenant compte des k plus proches voisins et en attribuant au nouveau joueur la classe majoritaire de ses voisins.

Q4. Classifier le joueur 4 pour $k = 5$. **Décrire** le rôle de k dans l'algorithme des k plus proches voisins.

4. Programmation de l'algorithme K-NN

Pour prédire la classe d'un nouvel élément il faut donc :

- Calculer la distance qui sépare le nouvel élément de chacun des autres points du jeu de données.
- Stocker ces valeurs de distance d dans une liste du type `[[d, i], [...], ...]`. Avec d la distance qui sépare le nouvel élément du point d'indice i .
- Déterminer l'indice du voisin le plus proche
- Assigner la classe au nouvel élément.



Fonctions mises en œuvre

On mettra en œuvre 4 fonctions :

- Une fonction `charge_jeu2donnees(fichier)` qui charge en mémoire le jeu de données stocké dans un fichier CSV. Cette fonction est déjà réalisée, elle renvoie une liste de t-uplet contenant :

(Nom du joueur, Type de poste, Taille, Masse)

Ex: `charge_jeu2donnees(« joueursToulouse.csv »)` renvoie `[('Charlie FAUMUINA', 'Avant', '184', '128'), ('Clément CASTETS', 'Avant', '183', '110'), ..., ('Selevasio TOLOFUA', '3ème ligne', '186', '115')]`

- Une fonction `distance(p1, p2)` qui calcule et renvoie la distance euclidienne séparant deux points p1 et p2
- Une fonction `distance_voisins(p1, jeu2donnees)` qui renvoie la liste des distances séparant le point p1 (nouveau joueur) de chaque point du jeu de données
- Une fonction `predire_classe(distances, jeu2donnees)` qui renvoie la classe à attribuer au nouveau joueur

Les fichiers utiles pour cette activité sont téléchargeables [ici](#)

Définition de la fonction *distance*

Pour rappel, dans un repère orthonormé du plan, la distance euclidienne séparant deux points $A(x_A; y_A)$ et $B(x_B; y_B)$ se calcule par la formule $\sqrt{(x_B - x_A)^2 + (y_B - y_A)^2}$

Q5. Compléter sur le fichier `knn.py` et en suivant la spécification, la fonction `distance`. **Valider** la fonction avec le test proposé dans la spécification.

Définition de la fonction *distance_voisins*

Q6. Compléter en suivant la spécification, la fonction `distance_voisins`. **Valider** la fonction avec le test proposé dans la spécification.

Définition de la fonction *predire_classe*

Q7. Compléter en suivant la spécification, la fonction `predire_classe`. **Valider** la fonction avec le test proposé dans la spécification.

5. Test de l'algorithme K-NN

On souhaite vérifier le fonctionnement du code développé en l'appliquant sur les joueurs définis à la question Q2 et avec le jeu de données constitué des joueurs de Toulouse (`joueursToulouse.csv`). Ce test consiste à vérifier la classe de chaque nouveau joueur déterminée par le programme ainsi que le nom du plus proche voisin.

Q8. Faire évoluer les programme pour obtenir les informations demandées. **Vérifier** la validité du programme avec les résultats obtenus en question Q2



6. Pour aller plus loin

Le travail réalisé sur cette première partie permet d'appliquer l'algorithme KNN pour une valeur de $K = 1$.

Q9. Modifier la fonction `predire_classe` afin de pouvoir choisir la valeur de k . Il est demandé de définir si besoin de nouvelles fonctions pour améliorer la lisibilité du code.

