

AGR. Algorithmes. k plus proches voisins

L'**algorithme des k plus proches voisins** est l'un des algorithmes utilisés dans le domaine de l'**intelligence artificielle**. Il intervient dans de nombreux domaines de l'apprentissage automatique et de **prédictions**.

I/ Algorithme de prédiction

1/ Un problème de classement

Voici un problème qui peut être résolu en utilisant l'algorithme des k plus proches voisins.

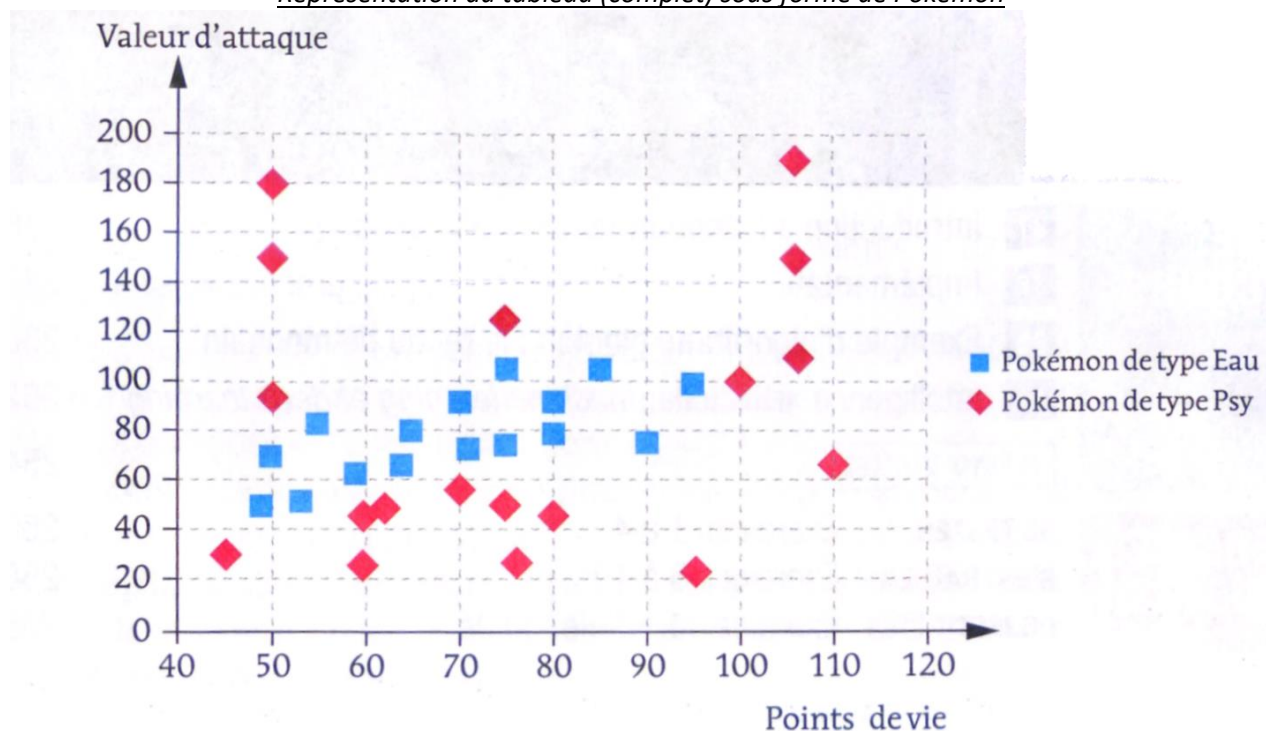
Exemple : Classer des Pokémon.

De façon très simpliste, on admettra ici que les Pokémon ne possèdent que deux caractéristiques : leurs points de vie et leur valeur d'attaque. On suppose également qu'ils se répartissent en deux types seulement : *Eau* et *Psy*.

Quelques exemples de Pokémon

Nom	<i>Ecayon</i>	<i>Deoxys</i>	<i>Eoko</i>	<i>Grotet</i>	<i>Tarpaud</i>
PDV	49	50	80	90	90
Attaque	49	95	45	75	75
Type	Eau	Psy	Psy	Psy	Eau

Représentation du tableau (complet) sous forme de Pokémon



A partir du graphique précédent, on peut **prédire** la classification probable d'un nouveau Pokémon à partir de ses données (PDV, attaque).

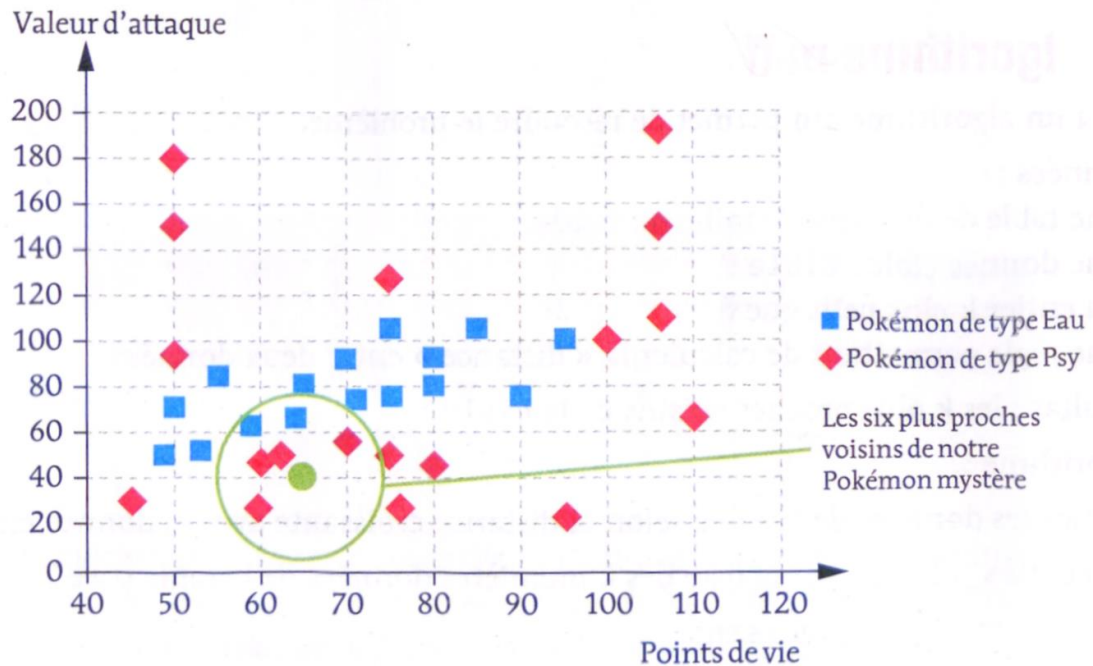
2/ Utilisation d'un diagramme

On considère un Pokémon « mystère » ayant pour caractéristiques :

- 65 points de vies.
- 40 en attaque.

Sur le diagramme ci-dessous, il sera représenté par le point vert.

On cherche les six voisins les plus proches de ce Pokémon : on peut choisir d'évaluer sa distance avec les autres Pokémon connus en traçant un cercle d'un rayon adapté.



On constate qu'il y a quatre Pokémon Psy et deux Pokémon Eau : le **Pokémon mystère** est probablement de type **Psy**.

II/ Implémentation de l'algorithme

1/ Algorithme formel

Voici un algorithme permettant de résoudre le problème.

Données :

- Une table de données de taille n .
- Une donnée cible.
- Un entier k (inférieur à n).
- Une règle établissant la distance entre deux données.

Algorithme :

- Trier les données de la table selon la distance croissante avec la donnée cible.
- Lister les k premiers voisins de la donnée cible.
- Trouver le type de la cible.

2/ En langage Python

Squelette d'un algorithme des k plus proches voisins en Python

```
# Definition du critère de tri
def distance_cible(donnee) :
    # Renvoie la distance en la donnée et la cible
    return distance(donnee,cible)

def k_plus_proches_voisin(table,cible, k) :
    # Tri selon la distance croissante de la table
    table_triee = sorted(table, key = distance_cible)

    # Choix des k plus proches voisins
    proches_voisins = []
    for i in range(k) :
        proches_voisins.append(table_triee[i])

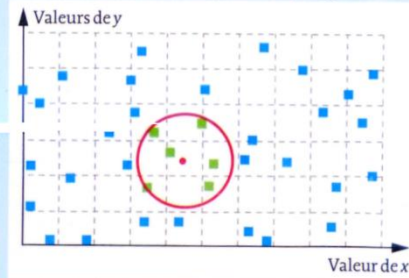
    return proches_voisins
```

A noter : des activités proposeront des exemples d'applications de cet algorithme.

3/ Exemples de distances

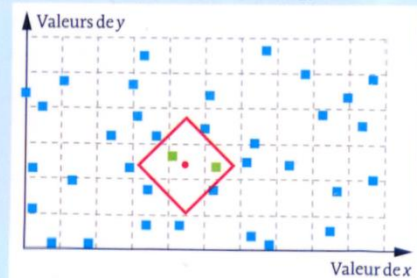
Avec la distance euclidienne

$$\text{distance}(\text{donnée}_1, \text{donnée}_2) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$



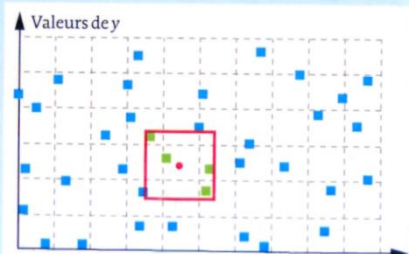
Avec la distance de Manhattan

$$\text{distance}(\text{donnée}_1, \text{donnée}_2) = |x_1 - x_2| + |y_1 - y_2|$$



Avec la distance de Tchebychev

$$\text{distance}(\text{donnée}_1, \text{donnée}_2) = \max(|x_1 - x_2|, |y_1 - y_2|)$$



A noter : $|x - y|$ est la distance entre x et y . Il s'agit de la valeur absolue.

Exemples : $|5 - 3| = 2$; $|4 - 8| = 4$.

