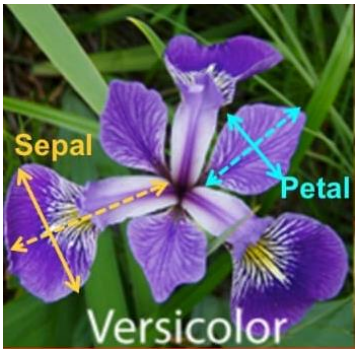
Chapitre 7 - Algorithme des k plus proches voisins

L'algorithme des k plus proches voisins est l'un des algorithmes utilisés dans le domaine de l'intelligence artificielle, notamment pour la reconnaissance de formes mais aussi pour réaliser certains diagnostics médicaux. Cette méthode permet de prédire à quelle famille une nouvelle entrée (une nouvelle donnée) va appartenir. Il calcule pour cela « des distances » entre la nouvelle donnée et toutes celles déjà présentes dans un jeu de données existant. C'est un algorithme d'apprentissage automatique car son jeu de données s'enrichit progressivement au fil des exécutions.

1- Exemple : Iris de Fisher



En 1936, M. Fisher a étudié les iris de Gaspesie, au Québec. Ces fleurs comportent trois familles : Setosa, Versicolore et Verginica. Il a étudié la longueur des sépales et pétales pour 150 iris, ce qui a donné naissance au jeu de données Iris, aussi appelé « Iris de Fisher ».

Iris Versicolor **Iris Setosa** **Iris Virginica**

Ces données, dont un extrait est donné ci-contre, sont contenues dans fichier .txt de 150 lignes, indiquant en cm, les longueur et largeur des sépales puis des pétales et le nom de leur famille.

```
sepal_length,sepal_width,petal_length,petal_width,species
5.1,3.5,1.4,0.2,setosa
4.9,3.0,1.4,0.2,setosa
4.7,3.2,1.3,0.2,setosa
4.6,3.1,1.5,0.2,setosa
5.0,3.6,1.4,0.2,setosa
```

Problème que l'on se pose :

Pour une personne non experte, il est difficile de déterminer visuellement à quelle famille appartient une iris de Gaspesie. L'algorithme des k plus proches voisins permet de le faire pour une nouvelle fleur identifiée, à partir de la mesure des dimensions de ses sépales et pétales. Cet algorithme compare ces mesures aux dimensions des 150 fleurs répertoriées dans le jeu de données. En recherchant celles qui ont les

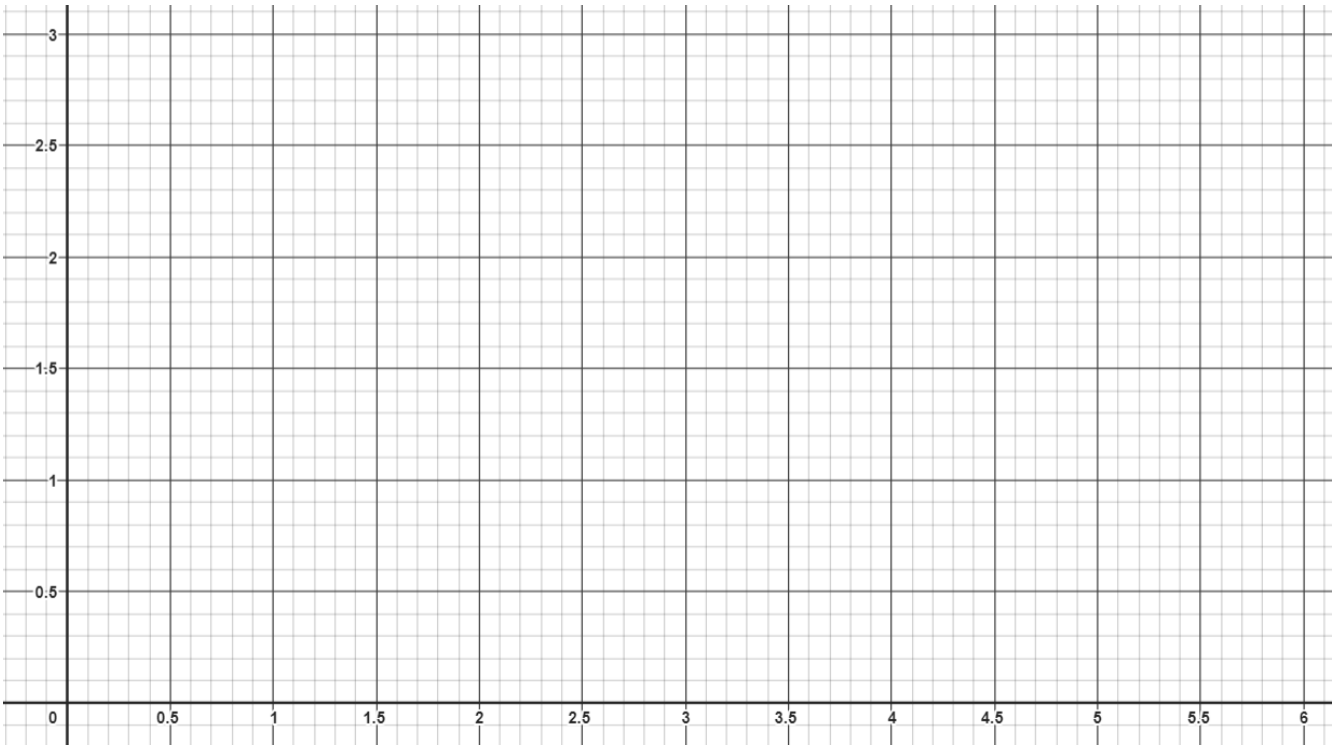
caractéristiques les plus proches et pour lesquelles on connait leur famille, il peut prédire la famille de la nouvelle fleur identifiée.

Exemple :

On présente dans un premier temps la méthode pour l'extrait suivant du jeu de donnée et en prenant uniquement en compte les dimensions des pétales :

4.8,3.0,1.4,0.1,setosa	6.7,3.1,4.4,1.4,versicolor	6.8,3.0,5.5,2.1,virginica
4.3,3.0,1.1,0.1,setosa	5.6,3.0,4.7,1.5,versicolor	5.7,2.5,5.0,2.0,virginica
5.8,4.0,1.2,0.2,setosa	5.8,2.7,4.1,1.0,versicolor	5.8,2.8,5.1,2.4,virginica
5.7,4.4,1.5,0.4,setosa	6.2,2.2,4.7,1.6,versicolor	6.4,3.2,5.3,2.3,virginica
5.4,3.9,1.3,0.4,setosa	5.6,2.5,3.9,1.1,versicolor	6.5,3.0,4.8,1.8,virginica

Comme ici, il n'y a que 2 attributs à analyser, on peut visualiser les données par un nuage de points dans le plan, avec en abscisse la longueur des pétales et en ordonnée, leur largeur.



On suppose que l'on trouve une fleur dont la longueur de pétale est de $\ell = 4.9\ cm$ et la largeur $h = 1.7\ cm$.

On positionne ce nouveau point sur le graphe précédent.

2- FORMULATION D'UN ALGORITHME :

Pour créer un code qui permet de réaliser ce travail automatiquement, mais cette fois-ci avec les 4 attributs (longueur, largeur des sépales **et** des pétales), on peut procéder de la manière suivante :

1- On crée une liste de listes qui contiendra le jeu de donnée. On prévoit un élément supplémentaire égal à ***None*** pour l'instant et qui pourra recevoir la distance calculée ultérieurement. Cela donne par exemple, la liste *donnees* suivante :

```
donnees =   [
             [4.8,3.0,1.4,0.1,"setosa",None],
             [4.3,3.0,1.1,0.1,"setosa",None],
             [6.7,3.1,4.4,1.4,"versicolor",None],
             [6.8,3.0,5.5,2.1,"virginica",None]
             ]
```

2- On crée la liste qui contient les données de la fleur inconnue. Cela donne par exemple :

```
inconnue = [5.6,2.8,4.9,1.7,"inconnue"]
```

3- Pour chaque fleur du jeu de données, on calcule la distance avec la fleur inconnue et on mémorise cette valeur dans le dernier élément des sous-listes. Par exemple la distance entre la la 1ère fleur du jeu de données et la fleur inconnue, sera :

```
donnees =   [
              [4.8,3.0,1.4,0.1,"setosa",None],
inconnue = [5.6,2.8,4.9,1.7,"inconnue"]
```

On mémorise cette valeur dans la sous-liste :

```
donnees =   [
             [4.8,3.0,1.4,0.1,"setosa",
```

4- On utilise un algorithme de tri par sélection (ou autre) avec k itérations seulement, afin de trier partiellement la liste *donnees*, pour placer les k sous-listes pour lesquelles la distance calculée est la plus faible.

3- DIFFERENTES FORMULATION DE LA DISTANCE CALCULEE :

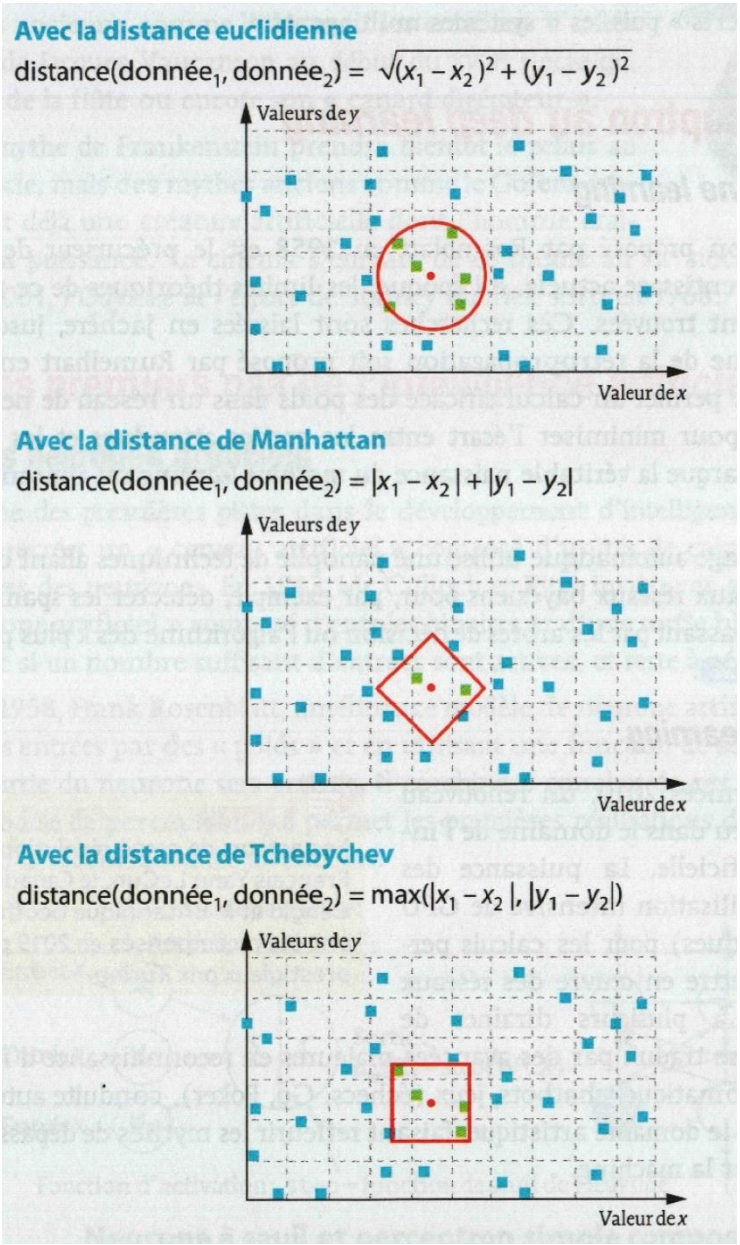
Il existe plusieurs fonctions de calcul de distance, notamment, la distance ***euclidienne***, la distance de ***Manhattan***, la distance de ***Tchebychef***, etc. Le choix est fait en fonction des types de données que l'on manipule. Ainsi pour les données quantitatives (par exemple : dimensions, poids, salaires, taille, montant de

panier électronique etc…) et **du même type**, la distance euclidienne est un bon candidat. Quand les données ne sont pas du même type (exemple : âge, sexe, longueur, poids etc…), la distance de Manhattan est plus appropriée.

Pour un jeu de données d'attributs $\{d_1 , d_2 , d_3 , d_4 , \dots\}$ et une inconnue d'attributs $\{i_1 , i_2 , i_3 , i_4 , \dots\}$, la définition des distances évoquées ci-dessus est la suivante :

- Distance euclidienne : $\sqrt{(d_1 - i_1)^2 + (d_2 - i_2)^2 + (d_3 - i_3)^2 + (d_4 - i_4)^2 + \cdots}$
- Distance de Manhattan : $|d_1 - i_1| + |d_2 - i_2| + |d_3 - i_3| + |d_4 - i_4| + \cdots ..$
- Distance de Tchebychef : $max(|d_1 - i_1| , |d_2 - i_2| , |d_3 - i_3| , |d_4 - i_4|, \dots . .)$

On voit sur les figures ci-contre, pour un cas avec 2 attributs, que selon la distance utilisée, les k voisins trouvés ne sont pas les mêmes :



4- QUELLE VALEUR DE k UTILISEE POUR UNE BONNE PREDICTION :

La valeur de k à utiliser dépend du type de données qui sont manipulées. Ce nombre k ne doit être ni trop petit, ni trop grand. Il est souvent défini de manière empirique.