FICHE DE TRAVAIL

NUMERIQUE ET SCIENCES INFORMATIQUES

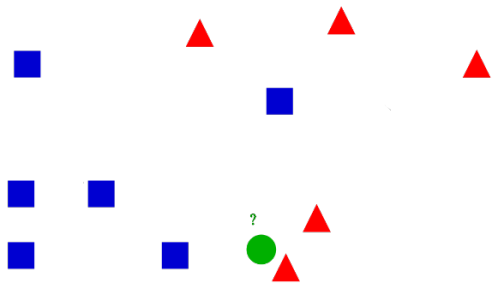
1ere – Algorithme des k plus proches voisins

Durée : 4 séances de 1h

Nom Prénom :

ACTIVITE N°1 ≈ 10 min

Ci-dessous, on donne un échantillon constitué de 11 éléments réparti selon 2 classes (les carrés en bleu et les triangles en rouge)



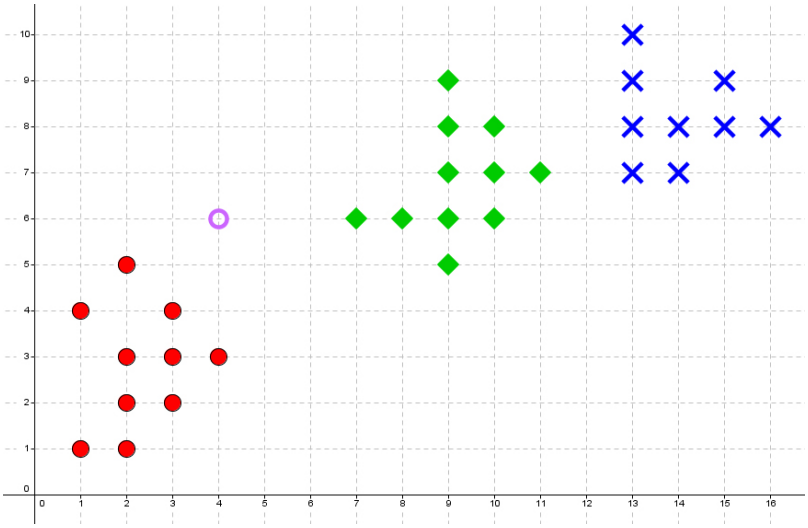
Appliquer l'algorithme KNN pour le nouvel élément représenté par un cercle vert
1/ pour k = 3
2/ pour k = 5
3/ pour k = 9

ACTIVITE N°2 ≈ 40 min

Implémenter en Python les fonctions du cours permettant d'exécuter l'algorithme des k plus proches voisins sur l'analyse d'un seul critère. (1 seule dimension x). On prendra l'échantillon du cours.

ACTIVITE N°3 ≈ 60 min

Ci-dessous, on donne un échantillon constitué de 30 éléments réparti en 3 classes (les cercles en rouge, les losanges en vert et les croix en bleu) selon 2 critères (2 dimensions : x et y). On cherche à classifier l'élément représenté en violet.



1/ Rappeler la formule mathématique correspondant à la distance euclidienne d entre 2 points A et B de coordonnées respectives (X_1, X_2) et (Y_1, Y_2)

Note : Cette formule se généralise à n dimensions, on a donc pour 2 points $A(X_1, X_2, \ldots, X_n)$ et $B(Y_1, Y_2, \ldots, Y_n)$,

$$D = \sqrt{(X_1 - Y_1)^2 + (X_2 - Y_2)^2 + \cdots + (X_n - Y_n)^2} = \sqrt{\sum_{i=1}^{i=n} (X_i - Y_i)^2}$$

2/ Implémenter en Python la « nouvelle » fonction permettant de calculer la distance euclidienne entre 2 points possédant *n* critères. On pourra prendre ces critères (réels) comme une liste possédant *n* éléments de type réel. Cette fonction prendra donc en paramètre 2 listes : les n critères du 1er élément et les n critères du 2eme élément.

3/ Que doit-on changer dans les fonctions ***Kvoisins*** et ***PredireLaClasse*** (implémentées dans l'activité n°2) pour s'adapter à une liste d'éléments possédant plusieurs critères ?

Note : On prendra par exemple pour les éléments de coordonnées (1.0, 1.0), (2.0, 1.0) et (8.0, 6.0) de classe respective 'Cercle', 'Cercle' et 'Croix', l'implémentation suivante :

> *Element = [[1.0, 1.0], [2.0, 1.0], [8.0, 6.0]]*
> *Classe = ['Cercle', 'Cercle', 'Croix']*

4/ Implémenter la solution permettant de résoudre notre problème initial.

ACTIVITE N°4 ≈ 120 min

En 1936, Edgar Anderson a collecté des données sur 3 espèces d'iris : "*iris setosa*", "*iris virginica*" et "*iris versicolor*".

		
iris setosa	*iris virginica*	*iris versicolor*

Pour chaque iris étudié, Anderson a mesuré (en cm), entre autres :

- La largeur des pétales
- La longueur des pétales
- L'espèce

On a récupéré un échantillon de 150 éléments de son travail mis en forme dans un fichier .CSV (voir dans les ressources).

1/ En s'aidant des activités précédentes et en important les données du fichier .CSV. « Prédire » avec l'algorithme 5NN l'espèce de l'iris que vous avez trouvé en vous promenant et dont la longueur et la largeur de ses pétales sont respectivement de 2.5 cm et 0.75 cm.

2/ Proposer une illustration de ces éléments avec un repère dont l'abscisse correspondra à la longueur des pétales (en cm) et l'ordonnée à la largeur des pétales (en cm)

- 1 : avec Excel
- 2 : Faire une interface graphique avec tkinter dans Python permettant non seulement de visualiser l'échantillon mais aussi de faire varier le nombre de voisins (k) de 1 à 10 et de saisir les critères (coordonnées) du nouvel élément à classifier.

Note : On pourra s'aider de l'exemple proposé dans les ressources

Aides pour l'utilisation du module *tkinter* :

- http://tkinter.fdex.eu/index.html
- http://fsincere.free.fr/isn/python/cours_python_tkinter.php