## Algorithme des k plus proches voisins

**Correction des exercices**

**Exercice 1**On considère le fichier csv dont le contenu est représenté ci-contre. Il contient les données de 10 points de l'espace colorés selon une certaine logique spatiale.

x,y,z,couleur  
3,7,5,noir

4,6,2,noir

3,7,8,blanc

0,1,2,noir

1,0,7,blanc

5,4,4,blanc

9,1,2,noir

5,3,3,noir

1,1,4,blanc

3,3,7,blanc

On considère un onzième point A de coordonnées x=4, y=4 et z=5 dont on cherche à prédire la couleur en appliquant l'algorithme des k plus proches voisins.

1. Combien de descripteurs vont être utilisés pour effectuer la prédiction ? Quels sont-ils ? 3 descripteurs : x, y et z
2. Combien y a-t-il d'étiquettes différentes ? Quelles sont-elles ? il y a deux étiquettes différentes : blanc et noir

|  |  |
| --- | --- |
| point de la table | distance au point A |
| 3,7,5,noir  4,6,2,noir  3,7,8,blanc  0,1,2,noir  1,0,7,blanc  5,4,4,blanc  9,1,2,noir  5,3,3,noir  1,1,4,blanc  3,3,7,blanc | 3.16 3.60 4.36 5.83 5.39 1.41 6.56 2.45 4.36 2.45 |

1. Après un import de ce fichier dans Python sous forme de table, quel serait le dictionnaire associé à la ligne 4 du fichier ( 3,7,8,blanc) ?

{'x':3, 'y':7, 'z':8, 'couleur':'blanc'} ou  
{'x':'3', 'y':'7', 'z':'7', 'couleur':'blanc'}

1. On a obtenu le tableau incomplet des distances ci-contre qui donne les distances entre le point A(4;4;5) et les points de la table. Calculer les deux distances manquantes.
2. On applique l'algorithme des k plus proches voisins pour prédire la couleur au point A.

- Si k = 1, quelle prédiction obtient-on ? blanc (distance 1.41)  
 - Si k = 3, quelle prédiction obtient-on ? blanc (distances 1.41, 2.45 et 2.45)  
 - Si k = 5, quelle prédiction obtient-on ? noir (distances 1.41, 2.45, 2.45, 3.16 et 3.60)

1. Pourquoi ne prend-on pas une valeur paire pour k ? pour éviter les ex-aequo

**Exercice 2**On considère trois jeux de données différents pour lesquels on a deux descripteurs sur lesquels baser la prédiction. La prédiction porte sur une couleur (étiquette rouge ou étiquette bleue). On a représenté les 3 jeux de données ci-dessous.   
Lequel ou lesquels de ces 3 jeux de données ne vont pas permettre de réaliser des prédictions fiables ?  
Celui le plus à gauche car il ne semble pas y avoir de zone 'réservée aux rouges' et de zone 'réservée aux bleus'. Tout est mélangé. Il n'y a donc pas l'air d'y avoir de lien entre l'étiquette d'un point et l'étiquette de ses proches voisins. Il parait donc peu pertinent d'appliquer l'algorithme des k plus proches voisins sur un tel jeu de données (pour en être certain il faudrait 10 ou 20 fois plus de points dans le jeu de données : peut-être verrait on des zones se former ?).

|  |  |  |
| --- | --- | --- |
| cas1.png | cas2.png | cas3.png |

**Exercice 3**On applique l'algorithme des k-plus proches voisins sur une table table\_donnees dont deux enregistrements sont par exemple :

#1 {'nb\_clics': 0.489, 'duree': 0.517, 'panier': 0.854, 'satisfait': 'O'}  
#2 {'nb\_clics': 0.828, 'duree': 0.865, 'panier': 0.142, 'satisfait': 'N'}

Les valeurs associées aux différents descripteurs ont toutes été ramenées entre 0 et 1 afin de leur donner la même importance lors du calcul de distance.

1. D'après vous, de quel type d'étude provient cette table de données :

* ~~Etude de satisfaction sur les clients d'un fabricant de paniers en plastique,~~
* Etude de satisfaction sur les clients d'un site web commercial.

1. Pour un nouveau client, on dispose de valeurs associées aux trois descripteurs dans un dictionnaire : {'nb\_clics' : 0.632, 'duree':0.321, 'panier': 0.242}.  
   On souhaite prédire la valeur d'un champ 'satisfait' de ce client.
2. En termes de vocabulaire, quelles sont les deux affirmations correctes ?

* On cherche à effectuer une classification dans deux classes : 'O' et 'N'
* On cherche à étiqueter le client en utilisant deux étiquettes : 'O' et 'N'
* ~~On cherche à distancer le client en utilisant trois descripteurs.~~

1. Calculer la distance entre ce nouveau client et l'enregistrement #1.

1. Ecrire le code d'une fonction python qui prend en paramètres :   
   - un enregistrement de la table client\_table,  
   - un enregistrement client\_nouveau,

et renvoie la distance entre ces deux enregistrements (on utilisera math.sqrt du module math pour calculer la racine carrée).  
def distance\_clients(client\_a, client\_b) :  
 d = ( ( client\_b['nb\_clics'] – client\_a['nb\_clics'] )\*\*2 +   
 ( client\_b['duree'] – client\_a['duree'] )\*\*2 +  
 ( client\_b['panier'] – client\_a['panier'] )\*\*2 )  
 return math.sqrt(d)