

GIL

Pierre-Louis

MPSI 1

DS n°9: Informatique

Note:

Commentaire:

Question n°1:

```
Select idpatient  
from MEDICAL  
where etat = « hernie discale »;
```

Question n°2:

```
Select nom, prenom  
from PATIENT  
join MEDICAL on MEDICAL.idpatient = PATIENT.id  
where etat = « spondylolisthésis ».
```

Question n°4:

Numpy nous donne accès aux objets de type array qui facilite grandement les calculs pour les matrices en réduisant la complexité de nos algorithmes.

Question n° 5:

6 colonnes et 100 000 lignes
de données

Soit 600 000 données codées sur 32 bits

De plus, on a 100 000 données codées sur 8 bits dans le vecteurs

On a donc $19\ 200\ 000 + 800\ 000$ bits soit
20 000 000 bits au total

Donc 2 500 000 octets

Donc 2,5 Mo

Question n° 6:

```
def separationParGroupe(data, etat):  
    N = []  
    H = []  
    S = []  
    for i in range(len(etat)):  
        if etat[i] == 0:  
            N.append(data[i])  
        elif etat[i] == 1:  
            H.append(data[i])  
        else:  
            S.append(data[i])  
    return [N, H, S]
```


Question n° 8: les diagrammes de la diagonale donne un idée du nombre de personnes avec les données dans un intervalle alors que les autres donnent les relations entre les différentes données des patients.

Question n° 9:

$$x_{\text{norm}_j} = \frac{x_j}{\max(x) - \min(x)}$$

Question n° 10:

```
def min_max(X):  
    if X[0] > X[1]:  
        | max = X[0]  
        | min = X[1]  
    | else :  
        | max = X[1]  
        | min = X[0]  
    | for i in range(2, len(X)):  
        | if X[i] > max:  
            | max = X[i]  
        | elif X[i] < min:  
            | min = X[i]  
    | return min, max
```


Question n°12:

partie 1: elle trie les distances obtenues par ordre croissant après les avoir reliées aux lignes de data correspondantes à la distance au n -uplet z .

partie 2: Pour chaque état médical, on donne le nombre de voisins les plus proches de z (i.e. parmi les k plus proche).

partie 3: Elle détermine quel état a le plus de personnes proche de z pour donner l'état de z .

$T \Rightarrow$ liste des distances et des numéros de patient

$dist \Rightarrow$ liste des distances des n -uplets de data avec le n -uplet z .

$select \Rightarrow$ compteur du nombre d'états $select[0] = \text{nbre de patients proches de } z \text{ avec un état} = 0$.

$incl \Rightarrow$ état de l'individu ayant le n -uplet de données z .

Ds n°9 : Info (suite).

Question n° 13:

La diagonale donne le nombre de fois où l'algorithme a eu juste.

On voit que la 1ère ligne nous donne, dans l'ordre, le nombre de cas bien prédit pour l'état normal, le nombre de cas prédit 0

mais étant 1 et le nombre de cas prédit 0 mais étant 2.

La 1ère colonne nous donne, dans l'ordre, le nombre de cas prédit 0 et étant 0, le nombre de cas prédit 1 mais étant 0 et

le nombre de cas prédit 2 mais étant 0.

La matrice nous sert à obtenir les statistiques qu'a la machine et ainsi sa fiabilité.

Question n° 14:

On remarque que le taux de réussite de l'algorithme reste bas (- de 75%),

et n'augmente pas beaucoup en augmentant le nombre de voisins.

1P n'est donc pas très efficace, voir même pas efficace du tout.