

ABIJEN

18/06/2020

Guël

MPSE 1)

DS Informatique

Question 1: Select id From MedicalP
where etat = "hernie discale" ;

Question 2: Select nom, prenom
From medicalP Patient
Join medicalP on ~~id~~ patient . id = medicalP . idpatient
where etat = "spondylolisthésis" ;

Question 3: Select Count(idpatient), etat
From medicalP
Group by etat ;

Question 4: L'utilisation de la bibliothèque numérique Numpy
est plus pratique car on a accès à plusieurs fonctions de
Numpy qui nous permettent de manipuler les tableaux de
grande taille avec plus de facilité. que

Question 5 : Tout d'abord le tableau est formé de 100 000 lignes et de 6 colonnes. chaque réel est codé sur 32 bits soit 4 octets. on obtient donc en tout 100 000 $\times 6 \times 4$ octets = 2 400 000 octets donc 2,4 Mo

- Pour le vecteur de données, il est de taille N et chaque entier est codé sur 8 bits soit 1 octet. Donc il y a au total 100 000 octets soit 0,1 Mo

Enfin, pour stocker les tableaux data et état lorsqu'il y a 100 000 patients, il faut 2,5 Mo

Question 6:

```
def separationpar groupe (data, état)
    sous_tableaux = [[] * 3]
    for k in range len(data):
        sous_tableaux[état[k]].append(data[k])
    return sous_tableaux
```

Question 7: $ArgS1 = (n, n, \ln) + j + 1$

Test = if $i \neq j$:

$ArgS3 = \text{label_attributs}[i]$

$ArgS2 = ($

flu

Question 8: Les diagrammes sur la diagonale permettent d'avoir pour chaque attribut des statistiques sur le patient et ainsi de leur attribuer un symbole. Les tableaux hors de la diagonale permettent de voir si les attributs sont reliés entre eux (si un attribut a une incidence sur un autre)

Question 9: $norm_j = \frac{x_j + \min(x) + \max(x)}{3}$

Question 10: ~~def~~ def min_max(x)

n = len(x)

min = x[0]

max = x[n-1]

~~for~~

for k in range(n):

ind = 0 if x[k] > max:

max = x[k]

ind = 1 if x[k] < min:

min = x[k]

return min, max

Question 11: def distance(z, data)

n = len(data[0])

L = list()

from math import sqrt

S = 0

for i in range(N+1):

for j in range(n):

a = z[j]**2 - data[i][j]**2

S += a

L.append(sqrt(S))

Inductible

12) Dans la partie 2: On crée une Liste T dans laquelle on place toutes les valeurs de distances euclidiennes calculées dans la question précédente. on trie ensuite cette liste par ordre croissant

la variable locale T est une Liste de Listes à 2 éléments contenant:

- la distance entre le n -uplet à l'état n et le n -uplet connu

- la valeur de l'état avant le n -uplet connu

la variable locale doit correspondre à la liste de taille N contenant les distances entre chaque n -uplet x et le n -uplet z

Pas non

Question 14 : on peut dire que plus le nombre de voisins est grand, plus le taux de réussite de l'algorithme est élevé (même si ce n'est pas évident vers la fin). Si P , inverse, lorsque le nombre de proches voisins est petit (moins de 5 personnes) le taux de réussite de l'algorithme est inférieur à 70%

Question 15: def moyenne(x):

$n = \text{len}(x)$

somme = 0

for k in range(n):

 somme = somme + x[k]

return (somme/n)

def Variance(x)

$n = \text{len}(x)$

moy = 0

var = 0

for k in range(n):

 moy = moy + x[k]

 var = var + x[k]²

return (var/n - (moy²))

4