

- 1) SELECT idpatient FROM MEDICAL WHERE état = 'hernie discale'
- 2) SELECT nom, prenom FROM PATIENT JOIN MEDICAL ON PATIENT.id = idpatient WHERE état = 'spondylolisthésis'
- 3) SELECT état, COUNT(patient) as nb FROM MEDICAL GROUP BY état
- 4) Le module Array de numpy et sa représentation matricielle rendent les tableaux de grande taille plus lisibles
- 5) On a 6 colonnes et 100000 lignes pour un total de 600000 valeurs car chacune est un mot de 32 bits soit  $32 \times 600000$  bits au total.  
Dans le vecteur on a 100000 entiers codés sur 8 bits  
Pour tout stocker il faut donc  $600000 \times 32 \times 8 \times 100000 = 1000000000$  bits  
Soit 2,5 Mo.

6) def separation Par Groupe (data, etat):

liste = [[], [], []]

for i in range (len(data)):

if etat[i][0] == 0:

liste[0].append (data[i])

elif etat[i][0] == 1:

liste[1].append (data[i])

elif etat[i][0] == 2:

liste[2].append (data[i])

returne liste

7) ARGS1 = m, n, i \* m + j + 1

$$9) x_{\min} = \frac{x_j - \min(x)}{\max(x) - \min(x)}$$

10) def min\_max (X):

min = X[0]

max = 0

for i in range (len(X)):

if X[i] < min:

min = X[i]

elif X[i] > max:

max = X[i]

return (min, max)

```

11) def distance(g, data):
    tab = data.copy()
    for i in range(len(data[0])):
        col = []
        for j in range(len(data)):
            col.append(data[j][i])
        maxc, minc = max_min(col)
        for j in range(len(data)):
            data[j][i] = (data[j][i] - minc) / (maxc - minc)
        g[i] = (g[i] - min) / (maxc - min)
        for j in range(len(data)):
            tab[j][i] = abs(data[j][i] - g[i])
    return tab

```

12) La partie 1 crée la liste T à partir des distances puis la trie

14) On remarque que la courbe est toujours au dessus des 70%  
donc l'algorithme est assez efficace mais peut quand même faire des erreurs.