

Frey
Elsa

MPSI2
18/06

Informatique

2 - Analyse de données.

Question 1 :

Select idpatient from MEDICAL
where etat = 'hernie discale';

✓

Question 2 :

Select nom, prénom from PATIENT
where MEDICAL.idpatient = PATIENT.id
AND MEDICAL.etat = 'spondylolisthésis';

?

✓

Question 3 :

Select etat, calc(etat) as nbpatient
from MEDICAL
Group by etat;

Question 4 :

d'écriture sous forme de tableau numpy et bien plus lisible lorsque les tableaux sont de grande taille, il est plus simple d'effectuer des multiplications sur ces tableaux.

✓

Pas uniquement

Question 5:

On cherche la quantité de mémoire totale nécessaire.

$$\begin{aligned}\text{Pour data: } N \times n \times 32 &= 100\,000 \times 6 \times 32 \\ &= 192\,000\,000 \text{ bits} \\ &\approx 2.4 \text{ Mo}\end{aligned}$$

$$\begin{aligned}\text{Pour état: } N \times 8 &= 100\,000 \times 8 \\ &= 800\,000 \text{ bits} \\ &\approx 0.1 \text{ Mo}\end{aligned}$$

Donc au total il faut

$$\text{Medata} + \text{Métat} = 2.4 + 0.1 = 2.5 \text{ Mo.}$$

Question 6:

def separationParGroupe (data, état):

 t = []

 u = []

 v = []

 n = data.shape

 for i in range (n[0]):

 if état[i] == 0:

 t.append(data[i])

 elif état[i] == 1:

 u.append(data[i])

 else:

 v.append(data[i])

 return [t, u, v].

Question 7:

ARGS1 = data.shape[0], ~~data.shape[1]~~, j.

ARGS2 = label_attributes[i], ~~label_attributes[j]~~,
marker = mark[k]

ARGS3: data [~~j~~]

TEST = ~~i != j~~

Question 8:

L'utilité des diagrammes diagonaux ~~est~~ de comparer le nombre de patient atteint avec un degré de l'attribut donné.

L'utilité des diagrammes hors-diagonale permet de comparer l'état d'un patient en fonction des états des autres pour plusieurs attributs donnés.

3 - Apprentissage et prédiction:

3.1. Méthode KNN

Question 9:

def xnormj(xj):

if xj = max(X):

xnormj = 1

elif xj = min(X):

xnormj = 0

else:

xnormj = (max(X) - xj) / (max(X) - min(X))

return xnormj

Question 10:

def min_max(X):

min = X[0]

max = X[0]

for i in range(X.shape[0]):


```

if X[i] < min:
    min = X[i]
elif X[i] > max:
    max = X[i]
return min, max.

```

Un mot sur la complexité ?

Question 11:

```

def distance(z, data):
    L = []
    for i in range(data.shape[1]):
        for j in range(len(z)):
            t.append(z[j] - data[i][j])
        L.append(t)
    return L.

```

Revoir le calcul de la norme

Question 12:

Partie 1:

Dans cette partie on trie les distances de la plus petites à la plus grande renvoie dans une liste

Partie 2:

Partie 3:

On renvoie l'indice pour lequel select[k] est maximal.

Question 13:

Question 14:

On observe que la courbe obtenue n'est pas stable, de plus le taux de réussite est toujours inférieur à 75%, il est donc peu fiable puisque selon les valeurs de K , on observe aucun pic de réussite très important.



3.2. Question 15:

def moyenne(x):

n = x.shape[0]

s = 0

for i in range n:

s = s + x[i]

return (s / n)



def variance(x):

n = x.shape[0]

s = 0

for i in range n:

s = s + (x[i] - moyenne(x))**2

return s/n.



Question 16:

```
def synthese (data, etat):
```

```
    t = []
```

```
    u = []
```

```
    v = []
```

```
    n = data.shape
```

```
    for i in range (n[0]):
```

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

```
            t.append ([moyenne (data[:,i]), variance(data[:,i])])
```

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

```
            u.append ([moyenne (data[:,i]), variance (data[:,i])])
```

```
        else:
```

```
            v.append ([moyenne (data[:,i]), variance (data[:,i])])
```

```
    return [t, u, v].
```

Question 17:

```
def gaussienne (a, moy, var)
```

```
    return ((1/sqrt(2*pi*var)) * exp(-(a-moy)**2 /
```

```
            (2*var)))
```

Question 18: