ROUX
Anthony

DS Informatique

TPSI 2.

Q1 SELECT idpatient FROM MEDICAL WHERE etat='lernie discale';

Q2 SELECT nom, prenom FROM PATIENT
JOIN MEDICAL ON PATIENT.id = idpatient
WHERE etat = 'spondylolisthesis';

Q3 SELECT etat, count(*) FROM MEDICAL
GROUP BY etat ;

Distinct

Q4 : Tableau N lignes, n colonnes sur 32 bits.

> Pour de tableaux de grande taille, la bibliothèque numpy permet de faire du calcul matriciel et ainsi d'accéder rapidement aux données : par ligne ou colonne.

Q5 32 Bits correspondent à 4 octets.

data Tableau : $N \times n \times 4 = 100\,000 \times 6 \times 4 = 240\,000$ octets
$= 2{,}4$ Mo

stat Vecteur : $N \times 4 = 100\,000 \times 4 = 400\,000 = 0{,}4$ Mo.

Q6 def separationParGroupe(data, etat) :

Q7

(ARGS 1) : (6, 6, i)

(ARGS 2) : ('incidence_bassin', 'orientation_bassin', marker = 'o')

(ARGS 3) : (data)

TEST : i != j

Q8 : Les diagrammes de la diagonale donnent une représentation globale de la proportion de patients en fonction du degré d'incidence du bassin. ✓

Ceux hors-diagonale permettent de relier incidence et l'orientation du bassin ainsi que l'angle de lordose lombaire selon les états des patient cette fois-ci : normal, hernie discale ou spondylolisthesis ✓

Q9 X_i vecteur colonne : de data.

$0 \leq x_{norm\,j} \leq 1$.

Si $x_j = min(X)$ alors $x_{norm\,j} = 0$.
Si $x_j = max(X)$ alors $x_{norm\,j} = 1$.
Si $x_j \neq min(X)$ et $x_j \neq max(X)$ alors $x_{norm\,j} \in]0;1[$.

Il faut donner une relation

Q10
```
def min_max(X):
    max, min = X[0], X[0]
    for i in range(1, len(X)):
        if X[i] > max:
            max = X[i]
        if X[i] < min:
            min = X[i]
    return max, min
```

Et la complexité ?

Q11
```
def distance(z, data):
    for i in range(len(data)):
    return abs(z - i).
```

A revoir.

Q12 : T liste de : listes à 2 éléments ✓

dist : représente les distances euclidiennes entre le n-uplet z et chaque n-uplet x du tableau data. ✓ ✓

de partie 1 ajoute à la liste T les distances euclidiennes à leur position respective (position i) : ÉTAPE de création de T.

Partie 2 : on ajoute 1 à la valeur de chaque état, après multiplication par 3 si n'y a donc plus d'état portant le numéro 0.

Partie 3 : Permet de renvoyer pour chaque élément de T, le nombre de plus proches voisins k.

Q14 La plupart du temps l'algorithme fonctionne (donc plus de 70% des cas).

Cependant, le taux de réussi reste toujours inférieur à 75%. 1 fois 4 au moins, l'algorithme échoue quelque que soit la valeur de K. Efficacité moyenne.

Q15

```
def moyenne(x):
    moy = 0
    for i in range(len(x)):
        moy = moy + x[i]
    return moy/len(x)
```

```
def variance(x):
    var = 0
    for i in range(len(x)):
        var = var + (x[i] - (moyenne(x))**2)
    return var/len(x)
```

ROUX
Anthony

Q16
```
def synthese(data, etat):
    for i in range(len(data)):
        for j in range(len(etat)):
            L = [moyenne(x[j]), variance(x[j])]
    return L
```

A indexer et initialiser

Q17
```
def fonction_gaussienne(a, moy, v):
    return (1/(2*pi*v)**0.5)
        * exp(-((a-moy)**2)/(2*v))
```

Q18
```
def probabiliteGroupe(z, data, etat):
```

Q19
```
def prediction(p0, p1, p2):
    if (p0 >= p1) and p0 >= p2):
        return p0
    elif p1 >= p2:
        return p1
    else:
        return p2.
```

Consignes pas respectées

Q20 . Comme $P(X_i = z_i \mid Y = y_j)$ s'écrit selon une exponentielle, l'utilisation du logarithme permet la simplification des calculs dans l'algorithme.

(En effet le logarithme d'exponentielle x simplifie l'expression à x par ex).

Q21 Méthode KNN pour $K = 8$.

Méthode naïve bayésienne.

$\begin{pmatrix} 23 & 9 & 8 \\ 9 & 10 & 1 \\ 10 & 1 & 49 \end{pmatrix}$, matrice de confusion.