

TD Informatique TSI2

Chap 5. Machine Learning Part2

Solution

EXERCICE 01

Parmi les assertions suivantes, lesquelles sont vraies ?

A. Il y a 2^n partitions possibles d'un ensemble à n éléments ?

En effet, **l'assertion A is True** d'après le cours sur les dénombrements.

B. Dans un apprentissage supervisé, on dispose d'un ensemble de données classées ?

En effet, **l'assertion B) is True** car c'est dans le cours de ce chapitre.

C. L'ensemble d'apprentissage utilisé pour calculer la matrice de confusion est un ensemble de données classées ?

En effet, **l'assertion C is True**

D. Pour calculer la matrice de confusion, on utilise un ensemble test de données non classées ?

Les données sont classées. C'est ce qui permet de compter les prédictions correctes et les prédictions incorrectes. **L'assertion D is False**

E. Les coefficients $m[i][i]$ d'une matrice de confusion m donnent les seules prédictions correctes ?

En effet, **l'assertion E is True**

F. Un faux positif est un positif qui a été prédit négatif ?

C'est un négatif qui a été prédit positif. Et **l'assertion F is False**

EXERCICE 02

1.

```
In [31]: from random import randint
In [32]: def creer_points(nb, dim, coul):
...:     pts=[]
...:     while len(pts) < nb:
...:         x = randint(0, dim)
...:         y = randint(0, dim)
...:         c = coul[randint(0,2)]
...:         if (x,y,c) not in pts:
...:             pts.append((x,y,c))
...:     return pts
...:
```

```
In [33]: couleurs = ["0.01", "0.5 ", "0.8"]

In [34]: creer_points(50,40,couleurs)
Out[34]:
[(34, 15, '0.8'), (31, 8, '0.5 '), (38, 30, '0.5 '),
 (34, 1, '0.8'), (36, 6, '0.01'), (29, 23, '0.01'),
 (15, 11, '0.5 '), (19, 16, '0.5 '), (38, 24, '0.01'),
 (29, 35, '0.01'), (9, 5, '0.01'), (40, 36, '0.5 '),
 (29, 18, '0.8'), (21, 12, '0.01'), (22, 15, '0.8'),
 (35, 17, '0.8'), (17, 10, '0.01'), (37, 21, '0.8'),
 (12, 14, '0.01'), (34, 11, '0.8'), (33, 5, '0.8'),
 (9, 40, '0.01'), (10, 2, '0.8'), (11, 36, '0.5 '),
 (1, 14, '0.01'), (17, 30, '0.01'),(25, 31, '0.8'),
 (14, 34, '0.01'), (33, 5, '0.5 '), (13, 19, '0.01'),
 (30, 24, '0.8'), (20, 28, '0.5 '), (10, 12, '0.5 '),
 (19, 38, '0.01'),(17, 38, '0.5 '), (11, 3, '0.5 '),
 (17, 20, '0.01'),(26, 8, '0.01'),(4, 3, '0.8'),
 (28, 33, '0.8'),(29, 29, '0.8'),(28, 40, '0.8'),
 (6, 27, '0.01'),(16, 36, '0.01'),(17, 2, '0.5 '),
 (17, 39, '0.5 '), (19, 36, '0.5 '), (20, 10, '0.01'),
 (15, 24, '0.5 '), (12, 15, '0.01')]
```

2.

```
In [36]: import matplotlib.pyplot as plt

In [37]: P = (randint(0,40), randint(0,40),"k")

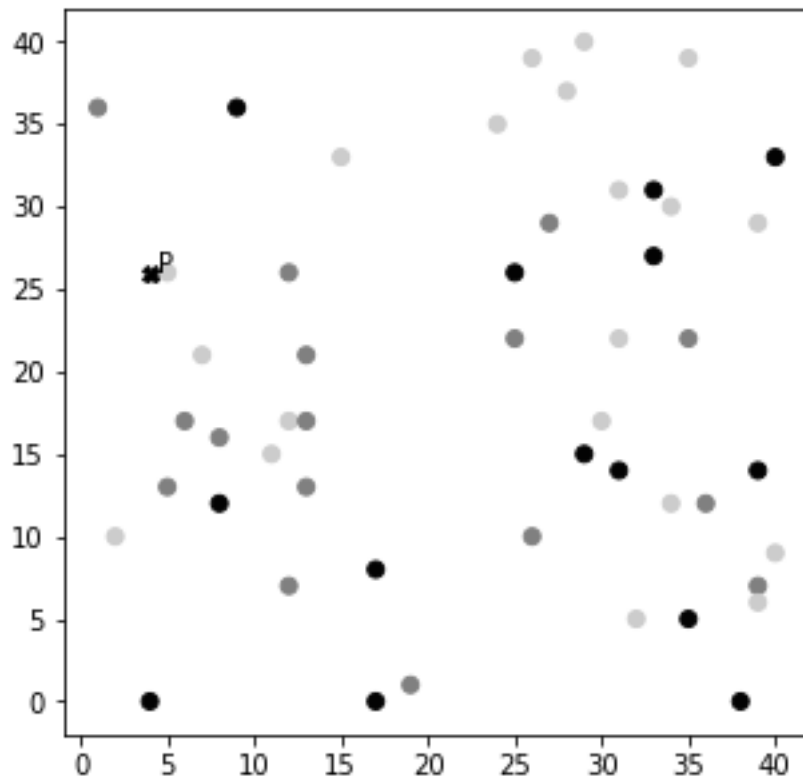
In [39]: points = creer_points(50,40,couleurs)

In [40]: x = [p[0] for p in points]

In [41]: y = [p[1] for p in points]

In [42]: c = [p[2] for p in points]

In [43]: plt.scatter(x,y, linestyle='None', color=c, marker="o");
plt.plot(P[0],P[1], P[2] + "X");
plt.text(P[0]+0.4,P[1],"P"); plt.show()
```



3-a

```
In [45]: def distance(p1,p2):
...:     (x1,y1,c1) = p1
...:     (x2,y2,c2) = p2
...:     return (x1-x2)**2 + (y1-y2)**2
...:

In [46]: distance((1,2,0),(4,5,7))
Out[46]: 18
```

3-b

```
In [51]: def tri(E,P,d):
...:     def choix(elt):
...:         return elt[1]
...:     distances = [(p,d(p,P)) for p in E]
...:     return sorted(distances, key=choix)
...:

In [52]: def knn(E,p,d,k):
...:     pts = tri(E,p,d)
...:     return [elt[0] for elt in pts[0:k]]
...:

In [53]: k = 4

In [54]: vs = knn(points,P,distance,k)
In [55]: vs
Out[55]: [(14, 10, '0.5 '), (11, 3, '0.01'), (9, 3, '0.5 '), (7, 1, '0.01')]
)
```

La couleur '0.8' semble disparue.

```
In [56]: k = 7
In [57]: newvs = knn(points ,P,distance ,k)
In [58]: newvs
Out[58]:
[(14, 10, '0.5 '),
 (11, 3, '0.01'),
 (9, 3, '0.5 '),
 (7, 1, '0.01'),
 (18, 13, '0.8'),
 (20, 11, '0.5 '),
 (5, 13, '0.01')]

# OUF, on a les trois couleurs !
```

3-c

```
In [59]: def couleur_maj(vs,coul):
          cpt = [0] * len(coul)
          for v in vs :
              for i in range(len(coul)):
                  if v[2] == coul[i] :
                      cpt[i] += 1
          ind_maxi = 0
          for i in range(len(cpt)):
              if cpt[i] > cpt[ind_maxi]:
                  ind_maxi = i
          return ind_maxi
```

```
In [60]: choix_couleur = couleurs[couleur_maj(vs,couleurs)]
In [61]: choix_couleur
Out[61]:
'0.5 '
```