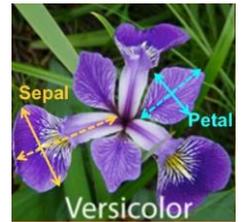


Tp - Algorithme des k plus proches voisins

On se propose de créer un code python qui permette de donner la famille à laquelle appartient une iris de Gaspésie, non encore classifiée.



1- CODE PYTHON :

⇒ Télécharger le zip lié à ce Tp. Décompresser ce dossier dans votre espace de travail.

Le fichier *irisEleve.py* contient la fonction *donneesIris()* qui lit le fichier *iris.txt* contenant le jeu de données des 150 fleurs de Fisher. En exécutant le script de ce fichier, la liste *donnees* est créée. Elle contient 150 sous-listes contenant chacune, les longueur et largeur des sépales, les longueur et largeur des pétales, la famille d'appartenance et un élément supplémentaire dont la valeur est pour l'instant *None*. Pour avoir une idée du contenu de ces sous-listes, par exemple la 99^{ième}, on peut exécuter dans la console :

```
>>> donnees[99]
[5.7, 2.8, 4.1, 1.3, 'versicolor', None]
```

On se propose de compléter ce fichier afin d'obtenir au final, un code du type k plus proches voisins, code qui déterminera la famille d'appartenance d'une fleur inconnue.

Le programme principal

de ce code sera le

suivant :

```
# Main
donnees = donneesIris("iris.txt")
kVoisins([5.6,2.8,4.9,1.7,"inconnue"],20)
```

Le script de la fonction *kVoisins()* sera le suivant :

```
def kVoisins(inconnue,k) :
    """
    Paramètres :inconnue, liste contenant les attributs de la fleur inconnue
                k (int), nombre des plus proches voisins
    Sortie : aucun renvoi
    """
    for i in range(len(donnees)) :
        d = distance(donnees[i],inconnue)
        donnees[i][-1] = d
    triSelection(donnees,k)
    resultat = ecritureConsole(k)
    print(resultat)
    nuage(inconnue,k)
```

Remplace la valeur *None* de chaque sous-liste par la distance avec la fleur inconnue

Trie partiellement la liste *donnees*, pour placer en début de liste, les k sous-listes pour lesquelles la distance est la plus petite

Écrit dans la console les résultats (si analyse pétales seulement):

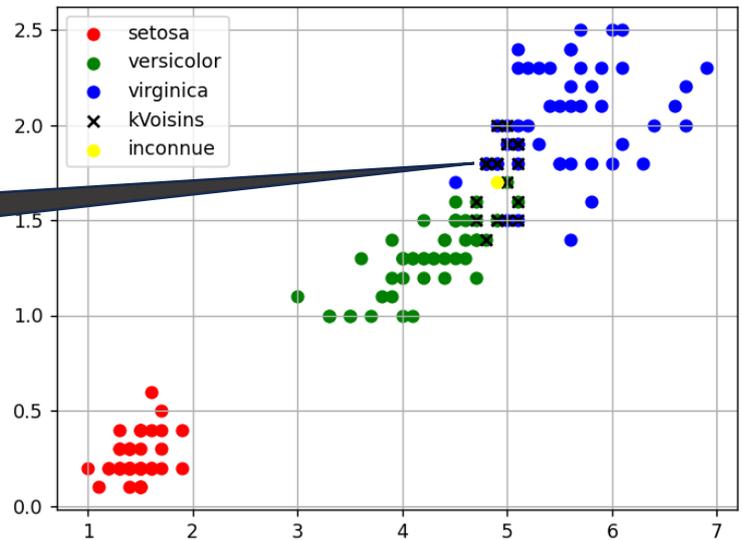
```
Nombres de voisins setosa : 0
Nombres de voisins versicolor : 8
Nombres de voisins virginica : 12
La fleur inconnue est de la famille : virginica
```

Crée le nuage de points de la page suivante :

```
def nuage(inconnue,k):
    graphique("setosa", 'red')
    graphique("versicolor", 'green')
    graphique("virginica", 'blue')
    graphiqueResultat("kVoisins", 'black', k)
    scatter(inconnue[2], inconnue[3], c="yellow", label="inconnue")
    legend()
    grid(True)
    show()
```

Le tracé obtenu après exécution de la fonction *nuage()* est :

Les croix se superposent aux ronds pour les + proches voisins déterminés.



On donne ci-contre le script de la fonction *graphiqueResultat()* :

```
def graphiqueResultat(nom,couleur,k) :  
    X = []  
    Y = []  
    for i in range(k) :  
        X.append(donnees[i][2])  
        Y.append(donnees[i][3])  
    scatter(X,Y,c=couleur,label=nom,marker="x")
```

2- UTILISATION DU CODE MIS AU POINT :

Donner les copies d'écran de la console et du nuage de points, pour les exécutions suivantes :

- a- 10 voisins avec un calcul de distance sur les seuls attributs longueur et largeur des pétales
- b- 50 voisins avec un calcul de distance sur les seuls attributs longueur et largeur des pétales
- c- 10 voisins avec un calcul de distance sur les 4 attributs longueur et largeur des pétales et sépales