

```

# -*- coding: utf-8 -*-
"""
Éditeur de Spyder

"""

# Import de la bibliothèque numpy en tant que np
import numpy as np

# Chemin d'accès au fichier
fichier = open('/home/willie/Dropbox/Informatique PTSTI/TP/TP12 Projet instruments de
musique/instrument_features.csv', 'r')

# préparation d'une liste destinée à recueillir les données des lignes du fichier
donnees=[]

#=====
# réponse rapide à base de méthodes du coeur de python
# sinon reprendre le corrigé du TP6
for ligne in fichier: # on parcourt une à une les lignes du fichier
    """ chaque ligne du fichier est scindée en liste d'éléments distincts
    grâce à la méthode split à partir du séparateur virgule split(',')
    après être débarrassée des caractères d'espacement et d'échappement
    (par exemple le caractère de fin de ligne \n) grâce à la méthode strip()"""

    donnees=donnees+[ligne.strip().split(',')]
#=====

# transformation de la liste de lignes en tableau numpy
donnees=np.array(donnees)

# préparation de deux tableaux destinés à recueillir les données numériques
# et textuelles
# partie numérique : on prépare un tableau de zéros flottants destinés
# à recueillir toutes les données sauf les deux dernières colonnes
donneesnumeriques=np.zeros((np.shape(donnees)[0],np.shape(donnees)[1]-2),dtype='float')

# passage en flottant des données numériques
for i in range(donneesnumeriques.shape[0]):
    for j in range(donneesnumeriques.shape[1]):
        donneesnumeriques[i,j]=float(donnees[i,j])

# partie textuelle : on isole les deux dernières colonnes
donneestextuelles=donnees[:,np.shape(donnees)[1]-2:np.shape(donnees)[1]]

#=====
# alternative rapide à base de méthodes du coeur de numpy
#
# donnees=np.array(donnees)
#
# donneesnumeriques=donnees[:,0:np.shape(donnees)[1]-2].astype(float)
# donneestextuelles=donnees[:,np.shape(donnees)[1]-2:np.shape(donnees)[1]]
#=====

# on demande à l'utilisateur le son-candidat dont il veut trouver l'instrument
numero=int(input('Quel numero de son voulez-vous tester (0-59) ? '))

position=numero

# préparation des données de références
donneesnumeriquesreferences=donneesnumeriques
donneestextuellesreferences=donneestextuelles

```

```

# on élimine du tableau des données de références la ligne correspondant
# au son testé en concaténant deux sous tableaux dont la ligne du son testé
# est exclue
donneesnumeriquesreferences=np.concatenate(\
(donneesnumeriquesreferences[0:position,:],\
donneesnumeriquesreferences[position+1:donneesnumeriquesreferences.shape[0],:]),axis=0\
)

donneestextuellesreferences=np.concatenate(\
(donneestextuellesreferences[0:position,:],\
donneestextuellesreferences[position+1:donneestextuellesreferences.shape[0],:]),axis=0\
)

# pour chacune des lignes de référence, et pour toutes les colonnes
# on calcule les carrés des différences terme à terme avec la ligne
# du son testé
carresdesdifferences=(donneesnumeriques[numero,:]-donneesnumeriquesreferences[:,:])**2

# pour chaque ligne on fait la somme de tous les carrés
# (donc on somme les colonnes de carresdesdifferences --> axis=1)
sommecarres=np.sum(carresdesdifferences,axis=1)

# calcul des distances
distance=np.sqrt(sommecarres)

# détermination de la ligne qui réalise le minimum de distance
position=int(np.where(distance==distance.min())[0])

#on affiche la categorie trouvee
print(donneestextuellesreferences[position,-1])

#verification de la categorie
print(donneestextuellesreferences[position,-1]==donnees[numero,-1])

```