

```

import numpy as np
import matplotlib.pyplot as plt

f=open('D:/Docs_boulot/PSI_Informatique/Cours_TD_PSI/Cours_TD_2023_2024/11_IA/KNN_Rugby/
liste_lignes=f.readlines()
f.close()

liste_joueurs=[]

for ligne in liste_lignes:
    champs=ligne.split(';')
    d={}
    d['Poste']=champs[3]
    d['Poids']=float(champs[4])
    d['Taille']=float(champs[5])
    liste_joueurs.append(d)

def postes():
    D={} #le dictionnaire que l'on va créer
    for d in liste_joueurs:
        if d['Poste'] not in D: # teste si d['Poste'] est une clef de D
            D[d['Poste']] =0 #on crée le couple (clef, valeur)
    return D

def liste_joueurs_distances(poids, taille):
    liste_distances=[]
    for d in liste_joueurs:
        d1={} #le dictionnaire qu'on va ajouter à liste_distances
        d1['Poste']=d['Poste']
        d1['distance']=np.sqrt((poids-d['Poids'])**2+(taille-d['Taille'])**2)
        liste_distances.append(d1)
    return liste_distances

def KNN(k,poids,taille):
    D=postes() #le dictionnaire des postes à mettre à jour
    liste_distances=liste_joueurs_distances(poids,taille)
    #on va trier cette liste par ordre croissant des distances obtenues:
    liste_distances_triée=sorted(liste_distances,key=lambda d:d['distance'])

    for i in range(k):#on va parcourir les k premiers éléments de la liste triée
        d=liste_distances_triée[i]
        D[d['Poste']] +=1
    return(D)

def prediction(k,poids,taille):
    D=KNN(k,poids,taille)
    #on va chercher quel est le poste majoritaire parmi mes k voisins
    nb_max=-np.inf
    for clef,valeur in D.items():
        if valeur>nb_max: #ce test va être vrai au premier tour de boucle
            poste_prédit=clef
            nb_max=valeur
    return poste_prédit

print(prediction(8,100,190))
#Si je pèse 100kg et que je mesure 1m90, alors je jouerai CENTRE

```