Machine Learning
Exercice 2

Dans cet exercice, nous devons définir une mesure de dissimilarité pour un ensemble de données représentant une population.

D'abord nous catégorisons les villes en fonctions de leurs positions géographiques ainsi que les musiques en fonctions de leurs types

```
data = {}

city_categories = [["paris", "lille"], ["marseille", "toulouse"], ["madrid"]]
music_categories = [["trap", "hiphop", "rap"], ["metal", "technical death metal"], ["rock"], ["jazz"], ["classical"]]
```

Pour ensuite mettre les valeurs retournées par le datasets dans une map en 2D

```
with open('dataset.csv') as file:
    for row in file:
        if(escapeFirstLine >= 1):
            x = row.split(',')
            data[x[0]] = x
        escapeFirstLine += 1
```

C'est à partir d'ici, basé sur l'exemple donné sur le GitHub du projet « **code/metrics/hybrid_data/ »** que nous calculons la dissimilarité avec une gestion plus ou moins importante si la valeur est dite « catégorique » ou « quantitave ». Pour ce faire, nous calculons la valeur absolue de l'âge et du poids des deux individues et par la suite déterminer si oui ou non (grâce aux information données dans le CSV), s'il y a une dissimilarité dans ses goûts musicaux, son travail et sa ville.

```
user_1_city_category = findIndex(city_categories, data[user_1_id][4])
user_2_city_category = findIndex(city_categories, data[user_2_id][4])

user_1_music_category = findIndex(music_categories, data[user_1_id][5])
user_2_music_category = findIndex(music_categories, data[user_2_id][5])

age_score = abs(float(data[user_1_id][1]) - float(data[user_2_id][1]))

height_score = abs(float(data[user_1_id][2]) - float(data[user_2_id][2]))

if (data[user_1_id][3] == data[user_2_id][3]):
    job_score = 0
else:
    job_score = 10

if (data[user_1_id][4] == data[user_2_id][4]):
    city_score = 0
elif (user_1_city_category[0] == user_2_city_category[0]) :
    city_score = 5
else:
    city_score = 10

if (data[user_1_id][5] == "other" or data[user_2_id][5] == "other"):
    music_score = 5
elif (data[user_1_id][5] == data[user_2_id][5]):
    music_score = 0
else:
    music_score = abs(user_1_music_category[0] - user_2_music_category[0]) + 5

dissimilarity = age_score ** 2 + job_score + height_score ** 2 + city_score + music_score
```

Tout cela nous donne un score de dissimilarité entre deux personnes que nous affichons de la manière suivante :

```
  print("----")
  print(
      f"user 1 {user_1_id}, user 2 {user_2_id}, dissimilarity score: {dissimilarity}"
  )
  return dissimilarity
```

Par la suite, toujours en se basant sur le fichier présent dans « **code/metrics/hybrid_data/** », nous construisons un graph en ce basant sur la dissimilarité calculée précédemment

Voici un exemple de rendu après avoir lancé l'exercice :

```
----
user 1 198, user 2 191, dissimilarity score: 32.70162112723807
----
user 1 198, user 2 192, dissimilarity score: 140.18077133887556
----
user 1 198, user 2 193, dissimilarity score: 34.395806447352754
----
user 1 198, user 2 194, dissimilarity score: 53.15507792434414
----
user 1 198, user 2 195, dissimilarity score: 129.72572945355375
----
user 1 198, user 2 196, dissimilarity score: 52.52634259685381
----
user 1 198, user 2 197, dissimilarity score: 25.06142426182138
----
user 1 198, user 2 198, dissimilarity score: 0.0
```