

Compte rendu de projet

L2, Semestre 3 (année 2022/2023)



Contents

1	Introduction	3
2	Description du projet	3
3	Description des fonctions	4
3.1	Fonctions qui récupèrent les infos du fichier csv	4
3.2	Fonctions qui composent l'algorithme des KNN	4
4	Informations à prendre en compte	5
5	Problèmes rencontrés	5
6	ANNEXE	6
6.1	Exemple d'exécution du programme	6

1 Introduction

Le principe de notre projet est le suivant:

Utiliser l'algorithme des K Plus Proches Voisins (K Nearest Neighbors ou KNN en anglais), et l'appliquer afin de déterminer si une personne risque de mourir ou non, après avoir été victime d'une crise cardiaque.

Le principe de l'algorithme des KNN est relativement simple. On dispose d'une liste de vecteurs dont les catégories sont connues (exemple: chaque vecteur est de catégorie chien ou chat).

On va comparer un vecteur dont on ne connaît pas la catégorie (on ne sait pas si c'est un chien ou un chat) à cette liste de vecteurs.

Pour ce faire, on choisit un nombre K, qui correspond au rayon du cercle dans lequel on va étudier les vecteurs. Le centre de ce cercle correspond aux coordonnées de notre vecteur de catégorie inconnue. Les vecteurs en dehors du cercle ne nous intéressent pas.

On va ensuite compter le nombre de vecteurs par catégories (combien de chats et combien de chiens) parmi ceux à l'intérieur du cercle. La catégorie qui sera la plus présente sera celle qui sera affectée à notre vecteur (exemple: s'il y a 45 catégories "chat" et 32 catégories "chien", notre vecteur se verra affecter la catégorie "chat").

2 Description du projet

Nous avons un fichier CSV nommé "heart failure clinical records dataset.csv", dans lequel on trouve des informations de santé sur environ 300 personnes ayant été victimes de crises cardiaques.

On récupère les infos du csv et on les stocke dans un tableau de vecteurs. Chaque vecteur contient les infos de santé d'une personne ayant eu une crise cardiaque. On entre les infos d'un vecteur qu'on souhaite évaluer (qui correspond à une personne).

On choisit un "K", qui va correspondre au rayon à l'intérieur duquel on va regarder les vecteurs présents

On applique ensuite l'algorithme des KNN.

Cela nous permet de déterminer si, après une crise cardiaque, cette personne va survivre ou non

L'algorithme des KNN n'est pas entièrement fiable car, pour renvoyer la catégorie du vecteur testé, il se base sur les catégories des vecteurs dans un certain rayon autour de lui.

C'est pourquoi le résultat renvoyé n'est pas une vérité absolue, d'autant plus que dans ce cas-là, on prend en compte seulement quelques paramètres de santé sur les dizaines voire centaines qui peuvent influencer.

3 Description des fonctions

3.1 Fonctions qui récupèrent les infos du fichier csv

nb lignes fichier csv:

Fonction qui renvoie le nombre de lignes dans le fichier csv qu'on utilise

nb colonnes fichier csv:

Fonction qui renvoie le nombre de colonnes dans le fichier csv qu'on utilise

remplir tableau:

Fonction qui remplit un tableau de vecteurs avec les informations contenues dans le fichier csv

Chaque case du tableau est un vecteur qui correspond à une ligne du fichier csv

3.2 Fonctions qui composent l'algorithme des KNN

struct vecteur:

Structure de données appelée "vecteur", qui va nous servir à stocker les infos du fichier csv

***creation vecteur:**

Fonction qui crée un vecteur et le renvoie

***creation tab vec:**

Fonction qui remplit un tableau de vecteurs passé en argument

distance:

Fonction qui renvoie la distance euclidienne entre deux vecteurs

dans cercle:

Fonction qui renvoie 1 si un vecteur est dans le cercle de rayon K, et 0 sinon

voisins:

Fonction qui remplit le tableau des vecteurs voisins passé en argument et nous donne sa nouvelle taille

compteur cate:

Fonction qui remplit le tableau contenant le nombre de vecteurs de chaque catégorie. A chaque tour de boucle, elle prend la catégorie du vecteur sélectionné et ajoute +1 à la case correspondante dans le tableau des catégories (exemple:

Si la catégorie du vecteur sélectionné est 1, le programme ajoutera +1 à la case d'indice 1 du tableau des catégories

maxi:

Fonction qui renvoie la catégorie ayant le plus de vecteurs dans le cercle (=catégorie la plus représentée)

4 Informations à prendre en compte

Il y a plusieurs choses à savoir et à prendre en compte pour la correction de ce projet:

Il y a un problème qui n'est pas lié au code mais au contenu du fichier csv. Les infos qu'il contient ne sont pas toutes du même ordre de grandeur (exemple: la colonne "platelets" est de l'ordre de la centaine de milliers, tandis que certaines colonnes sont de l'ordre de l'unité).

Le code fonctionne sans problème avec ces différences d'ordre de grandeur, mais cela pose un problème au niveau de la cohérence du résultat renvoyé par le programme.

En effet, l'algorithme des KNN utilise la distance euclidienne. Cela va donc donner beaucoup plus de poids aux colonnes ayant des valeurs élevées (comme "platelets") qu'à la colonne "âge" par exemple, alors que l'âge est aussi un facteur très important lors d'une crise cardiaque.

Afin de régler ça, il faudrait mettre toutes les valeurs du fichier csv au même ordre de grandeur, mais nous ne savions pas si c'était correct scientifiquement parlant donc nous avons laissé le csv tel quel.

A cause de ça, il faut mettre un K très grand (250 000 par exemple) afin d'avoir un bon nombre de vecteurs présents dans le cercle.

5 Problèmes rencontrés

Nous avons rencontrés un bon nombre de problèmes, notamment:

- Hésitations sur la façon de structurer le type vecteur que nous avons créé
- Problèmes lors de la récupération des valeurs du fichier csv
- Beaucoup de problèmes liés aux pointeurs
- Parfois des difficultés à respectivement comprendre le code l'un de l'autre
- Quelques problèmes de communication et compréhension entre les membres du binôme

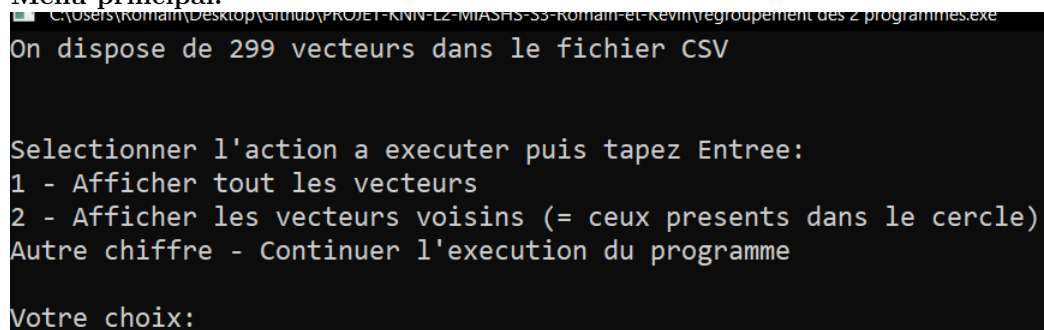
6 ANNEXE

6.1 Exemple d'exécution du programme

Attention: Les valeurs du résultat présentes dans les captures d'écran varient selon le choix du nombre K et selon les coordonnées du vecteur à tester, il s'agit ici seulement d'un exemple (si vous changez l'un des deux, il est probable qu'il n'y ait plus 89 vecteurs de catégorie 0 par exemple)

Si vous souhaitez appliquer le programme à un autre fichier, c'est possible, il faudra simplement changer le nombre de catégories (variable "nb cate" dans le main), ainsi que le nom du fichier et adapter les valeurs du vecteur qu'on teste

Menu principal:



```
C:\users\romain\Desktop\Gitlab\PROJET-KVIN-L2-MIASHS-SS-Romain-et-Kevin\regroupement des 2 programmes.exe
On dispose de 299 vecteurs dans le fichier CSV

Selectionner l'action a executer puis tapez Entree:
1 - Afficher tout les vecteurs
2 - Afficher les vecteurs voisins (= ceux presents dans le cercle)
Autre chiffre - Continuer l'execution du programme

Votre choix:
```

Choix 1 dans le menu:

Votre choix: 1

0	75.000000	0.000000	582.000000	0.000000	20.000000	1.000000	265000.000000	1.900000	130.000000	1.000000	0.000000	4.000000	category:1
1	55.000000	0.000000	7861.000000	0.000000	38.000000	0.000000	263358.031250	1.100000	136.000000	1.000000	0.000000	6.000000	category:1
2	65.000000	0.000000	146.000000	0.000000	20.000000	0.000000	162000.000000	1.300000	129.000000	1.000000	1.000000	7.000000	category:1
3	50.000000	1.000000	111.000000	0.000000	20.000000	0.000000	210000.000000	1.900000	137.000000	1.000000	0.000000	7.000000	category:1
4	65.000000	1.000000	160.000000	1.000000	20.000000	0.000000	327000.000000	2.700000	116.000000	0.000000	0.000000	8.000000	category:1
5	90.000000	1.000000	47.000000	0.000000	40.000000	1.000000	204000.000000	2.100000	132.000000	1.000000	1.000000	8.000000	category:1
6	75.000000	1.000000	246.000000	0.000000	15.000000	0.000000	127000.000000	1.200000	137.000000	1.000000	0.000000	10.000000	category:1
7	60.000000	1.000000	315.000000	1.000000	60.000000	0.000000	454000.000000	1.100000	131.000000	1.000000	1.000000	10.000000	category:1
8	65.000000	0.000000	157.000000	0.000000	65.000000	0.000000	263358.031250	1.500000	138.000000	0.000000	0.000000	10.000000	category:1
9	80.000000	1.000000	123.000000	0.000000	35.000000	1.000000	388000.000000	9.400000	133.000000	1.000000	1.000000	10.000000	category:1
10	75.000000	1.000000	81.000000	0.000000	38.000000	1.000000	368000.000000	4.000000	131.000000	1.000000	1.000000	10.000000	category:1
11	62.000000	0.000000	231.000000	0.000000	25.000000	1.000000	253000.000000	0.900000	140.000000	1.000000	1.000000	10.000000	category:1
12	45.000000	1.000000	981.000000	0.000000	30.000000	0.000000	156000.000000	1.100000	137.000000	1.000000	0.000000	11.000000	category:1
13	50.000000	1.000000	168.000000	0.000000	38.000000	1.000000	276000.000000	1.100000	137.000000	1.000000	0.000000	11.000000	category:1
14	49.000000	1.000000	89.000000	0.000000	30.000000	1.000000	427000.000000	1.000000	138.000000	0.000000	0.000000	12.000000	category:1
15	82.000000	1.000000	379.000000	0.000000	50.000000	0.000000	47000.000000	1.300000	136.000000	1.000000	0.000000	13.000000	category:1
16	87.000000	1.000000	149.000000	0.000000	38.000000	0.000000	262000.000000	0.900000	140.000000	1.000000	0.000000	14.000000	category:1
17	45.000000	0.000000	582.000000	0.000000	14.000000	0.000000	166000.000000	0.800000	127.000000	1.000000	0.000000	14.000000	category:1
18	70.000000	1.000000	125.000000	0.000000	25.000000	1.000000	237000.000000	1.000000	140.000000	0.000000	0.000000	15.000000	category:1
19	48.000000	1.000000	582.000000	1.000000	55.000000	0.000000	87000.000000	1.900000	121.000000	0.000000	0.000000	15.000000	category:1
20	65.000000	1.000000	92.000000	0.000000	25.000000	1.000000	276000.000000	1.300000	137.000000	1.000000	0.000000	15.000000	category:1
21	65.000000	1.000000	128.000000	1.000000	30.000000	1.000000	297000.000000	1.600000	136.000000	0.000000	0.000000	20.000000	category:1
22	68.000000	1.000000	220.000000	0.000000	35.000000	1.000000	289000.000000	0.900000	140.000000	1.000000	1.000000	20.000000	category:1
23	53.000000	0.000000	63.000000	1.000000	60.000000	0.000000	368000.000000	0.800000	135.000000	1.000000	0.000000	22.000000	category:1
24	75.000000	0.000000	582.000000	1.000000	30.000000	1.000000	263358.031250	1.830000	134.000000	0.000000	0.000000	23.000000	category:1
25	80.000000	0.000000	148.000000	1.000000	38.000000	0.000000	149000.000000	1.900000	144.000000	1.000000	1.000000	23.000000	category:1
26	95.000000	1.000000	112.000000	0.000000	40.000000	1.000000	156000.000000	1.000000	138.000000	0.000000	0.000000	24.000000	category:1
27	70.000000	1.000000	122.000000	1.000000	45.000000	1.000000	185000.000000	1.200000	140.000000	1.000000	0.000000	24.000000	category:1
28	58.000000	1.000000	60.000000	0.000000	38.000000	0.000000	153000.000000	5.800000	134.000000	1.000000	0.000000	26.000000	category:1
29	82.000000	1.000000	30.000000	1.000000	30.000000	0.000000	200000.000000	1.200000	132.000000	1.000000	1.000000	26.000000	category:1
30	94.000000	0.000000	582.000000	1.000000	38.000000	1.000000	263358.031250	1.830000	134.000000	1.000000	0.000000	27.000000	category:1
31	85.000000	0.000000	23.000000	0.000000	45.000000	0.000000	360000.000000	3.000000	132.000000	1.000000	0.000000	28.000000	category:1
32	50.000000	1.000000	249.000000	1.000000	35.000000	1.000000	319000.000000	1.000000	128.000000	0.000000	0.000000	28.000000	category:1
33	50.000000	1.000000	159.000000	1.000000	30.000000	0.000000	302000.000000	1.200000	138.000000	0.000000	0.000000	29.000000	category:1
34	65.000000	1.000000	92.000000	0.000000	50.000000	1.000000	185000.000000	1.300000	140.000000	1.000000	0.000000	29.000000	category:1
35	69.000000	1.000000	582.000000	1.000000	35.000000	0.000000	218000.000000	3.500000	134.000000	1.000000	0.000000	30.000000	category:1
36	90.000000	1.000000	60.000000	1.000000	50.000000	0.000000	226000.000000	1.000000	134.000000	1.000000	0.000000	30.000000	category:1
37	82.000000	1.000000	855.000000	1.000000	50.000000	1.000000	321000.000000	1.000000	145.000000	0.000000	0.000000	30.000000	category:1
38	60.000000	0.000000	2656.000000	1.000000	30.000000	0.000000	305000.000000	2.300000	137.000000	1.000000	0.000000	30.000000	category:1
39	60.000000	0.000000	335.000000	1.000000	35.000000	0.000000	320000.000000	2.000000	143.000000	0.000000	0.000000	30.000000	category:1

Choix 2 dans le menu:

votre choix: 2														
0	65.000000	0.000000	146.000000	0.000000	20.000000	0.000000	162000.000000	1.300000	129.000000	1.000000	1.000000	7.000000	category: 1	distance: 162000.125000
1	50.000000	1.000000	111.000000	0.000000	20.000000	0.000000	210000.000000	1.900000	137.000000	1.000000	0.000000	7.000000	category: 1	distance: 210000.093750
2	90.000000	1.000000	47.000000	0.000000	40.000000	1.000000	204000.000000	2.100000	132.000000	1.000000	1.000000	8.000000	category: 1	distance: 204000.062500
3	75.000000	1.000000	246.000000	0.000000	15.000000	0.000000	127000.000000	1.200000	137.000000	1.000000	0.000000	10.000000	category: 1	distance: 127000.335938
4	45.000000	1.000000	981.000000	0.000000	30.000000	0.000000	136000.000000	1.100000	137.000000	1.000000	0.000000	11.000000	category: 1	distance: 136000.609375
5	82.000000	1.000000	379.000000	0.000000	50.000000	0.000000	47000.000000	1.300000	136.000000	1.000000	0.000000	13.000000	category: 1	distance: 47000.820313
6	45.000000	0.000000	582.000000	0.000000	14.000000	0.000000	166000.000000	0.800000	127.000000	1.000000	0.000000	14.000000	category: 1	distance: 166000.078125
7	70.000000	1.000000	125.000000	0.000000	25.000000	1.000000	237000.000000	1.000000	140.000000	0.000000	0.000000	15.000000	category: 1	distance: 237000.093750
8	48.000000	1.000000	582.000000	1.000000	55.000000	0.000000	87000.000000	1.900000	121.000000	0.000000	0.000000	15.000000	category: 1	distance: 87000.062500
9	80.000000	0.000000	148.000000	1.000000	38.000000	0.000000	149000.000000	1.900000	144.000000	1.000000	1.000000	23.000000	category: 1	distance: 149000.171875
10	95.000000	1.000000	112.000000	0.000000	40.000000	1.000000	196000.000000	1.000000	138.000000	0.000000	0.000000	24.000000	category: 1	distance: 196000.109375
11	58.000000	1.000000	60.000000	0.000000	38.000000	0.000000	153000.000000	5.800000	134.000000	1.000000	0.000000	26.000000	category: 1	distance: 153000.093750
12	82.000000	0.000000	70.000000	1.000000	30.000000	0.000000	200000.000000	1.200000	132.000000	1.000000	1.000000	26.000000	category: 1	distance: 200000.078125
13	65.000000	0.000000	94.000000	1.000000	50.000000	1.000000	188000.000000	1.900000	140.000000	1.000000	0.000000	29.000000	category: 1	distance: 188000.093750
14	69.000000	0.000000	582.000000	1.000000	35.000000	0.000000	228000.000000	3.500000	134.000000	1.000000	0.000000	30.000000	category: 1	distance: 228000.781250
15	90.000000	1.000000	60.000000	1.000000	50.000000	0.000000	226000.000000	1.000000	134.000000	1.000000	0.000000	30.000000	category: 1	distance: 226000.062500
16	50.000000	0.000000	124.000000	1.000000	30.000000	1.000000	153000.000000	1.200000	136.000000	0.000000	1.000000	32.000000	category: 1	distance: 153000.125000
17	70.000000	0.000000	571.000000	1.000000	45.000000	1.000000	185000.000000	1.200000	139.000000	1.000000	1.000000	33.000000	category: 1	distance: 185000.953125
18	72.000000	0.000000	127.000000	1.000000	50.000000	1.000000	218000.000000	1.000000	134.000000	1.000000	0.000000	33.000000	category: 0	distance: 218000.093750
19	60.000000	1.000000	589.000000	1.000000	60.000000	0.000000	194000.000000	1.100000	142.000000	0.000000	0.000000	33.000000	category: 1	distance: 194000.969750
20	80.000000	1.000000	553.000000	0.000000	20.000000	1.000000	140000.000000	4.400000	133.000000	1.000000	0.000000	41.000000	category: 1	distance: 140000.187500
21	68.000000	1.000000	577.000000	0.000000	25.000000	1.000000	166000.000000	1.000000	138.000000	1.000000	0.000000	43.000000	category: 1	distance: 166000.078125
22	70.000000	1.000000	75.000000	0.000000	35.000000	0.000000	223000.000000	2.700000	138.000000	1.000000	1.000000	54.000000	category: 0	distance: 223000.078125
23	60.000000	1.000000	607.000000	0.000000	40.000000	0.000000	216000.000000	0.600000	138.000000	1.000000	1.000000	54.000000	category: 0	distance: 216000.962500
24	50.000000	0.000000	318.000000	0.000000	40.000000	1.000000	216000.000000	2.300000	131.000000	0.000000	0.000000	60.000000	category: 1	distance: 216000.281250
25	60.000000	0.000000	68.000000	0.000000	20.000000	0.000000	119000.000000	2.900000	127.000000	1.000000	1.000000	64.000000	category: 1	distance: 119000.101563
26	42.000000	1.000000	250.000000	1.000000	15.000000	0.000000	1213000.000000	1.300000	136.000000	0.000000	0.000000	65.000000	category: 1	distance: 213000.20312
27	70.000000	0.000000	161.000000	0.000000	25.000000	0.000000	244000.000000	1.200000	142.000000	0.000000	0.000000	66.000000	category: 1	distance: 244000.199375
28	40.000000	0.000000	569.000000	1.000000	35.000000	0.000000	139000.000000	1.200000	139.000000	1.000000	0.000000	70.000000	category: 1	distance: 139000.203125
29	85.000000	0.000000	5882.000000	0.000000	35.000000	0.000000	243000.000000	1.000000	132.000000	1.000000	1.000000	72.000000	category: 1	distance: 243071.234375
30	65.000000	0.000000	224.000000	1.000000	50.000000	0.000000	149000.000000	1.300000	137.000000	1.000000	1.000000	72.000000	category: 0	distance: 149000.250000
31	60.000000	1.000000	47.000000	0.000000	20.000000	0.000000	204000.000000	0.700000	139.000000	1.000000	1.000000	73.000000	category: 1	distance: 204000.062500
32	42.000000	0.000000	102.000000	1.000000	40.000000	0.000000	237000.000000	1.200000	140.000000	1.000000	0.000000	74.000000	category: 0	distance: 237000.078125
33	60.000000	1.000000	76.000000	1.000000	25.000000	0.000000	196000.000000	2.500000	132.000000	0.000000	0.000000	77.000000	category: 1	distance: 196000.062500
34	79.000000	1.000000	55.000000	0.000000	50.000000	1.000000	172000.000000	1.800000	133.000000	1.000000	0.000000	78.000000	category: 0	distance: 172000.937500
35	55.000000	0.000000	47.000000	0.000000	35.000000	1.000000	173000.000000	1.100000	137.000000	1.000000	0.000000	79.000000	category: 0	distance: 173000.078125
36	40.000000	0.000000	24.000000	1.000000	20.000000	0.000000	245000.000000	1.200000	140.000000	1.000000	0.000000	79.000000	category: 0	distance: 245000.187500
37	57.000000	1.000000	115.000000	0.000000	25.000000	1.000000	181000.000000	1.100000	144.000000	1.000000	0.000000	79.000000	category: 0	distance: 181000.109375
38	70.000000	0.000000	66.000000	1.000000	45.000000	0.000000	249000.000000	0.800000	136.000000	1.000000	1.000000	80.000000	category: 0	distance: 249000.062500
39	60.000000	1.000000	154.000000	0.000000	25.000000	0.000000	210000.000000	1.700000	135.000000	1.000000	0.000000	82.000000	category: 1	distance: 210000.109375
40	58.000000	1.000000	133.000000	0.000000	60.000000	1.000000	219000.000000	1.000000	141.000000	1.000000	0.000000	83.000000	category: 0	distance: 219000.199375

```
On a 134 vecteurs voisins (a l'interieur du cercle)
```

```
Il y a 89 vecteurs de categorie 0
```

```
Il y a 45 vecteurs de categorie 1
```

```
NOTRE VECTEUR EST DONC DE CATEGORIE 0
```

```
(La categorie 0 correspond a la survie apres arret cardiaque, la categorie 1 correspond au deces apres arret cardiaque)
```

```
Process returned 0 (0x0)   execution time : 2.517 s
```

```
Press any key to continue.
```