

Département Mathématique et Informatique

Filière :

‘ Ingénierie Informatique : Génie du logiciel et des Systèmes
Informatiques Distribués ’

Rapport :

Réalisation d'un projet
Prédiction d'insuffisance cardiaque
utilisant l'algorithme KNN

Réalisé par :

ENNANI Abdelmalek

DOUHI Chaimae

Encadré par :

Mr. QBADOU Mohamed

Année universitaire : 2021/2022

Sommaire :

Introduction -----	3
Contexte Général du projet -----	4
L'algorithme KNN -----	5
Étapes de fonction de KNN-----	6
Data-set -----	7
Format de data-set -----	8
Thème de projet -----	9
Outils utilisés -----	9
Les composants de programme -----	10
Informations requises par le programme -----	11
Le code : -----	12
Disease.h -----	12
Knn-funcs.c -----	13
Main.C -----	19
Exécution de code -----	20
Conclusion -----	21

Introduction

Les algorithmes sont bien plus présents dans nos vies qu'on ne le pense. En effet, ces calculs informatiques sont la base de la majorité des logiciels, de l'intelligence artificielle et de la plupart de nos programmes et services. Aujourd'hui, ces calculs d'entiers et les structures de données engendrées sont la base et l'essence même de notre société.

De nos jours, écrire un algorithme se fait désormais sur ordinateur, pour effectuer une tâche ou un programme informatique. Or, pour être fonctionnel, il est nécessaire « d'indiquer » à l'ordinateur ce qu'il est censé faire et ce, avec précision et étape par étape.

L'un parmi ces algorithmes est l'algorithme KNN.

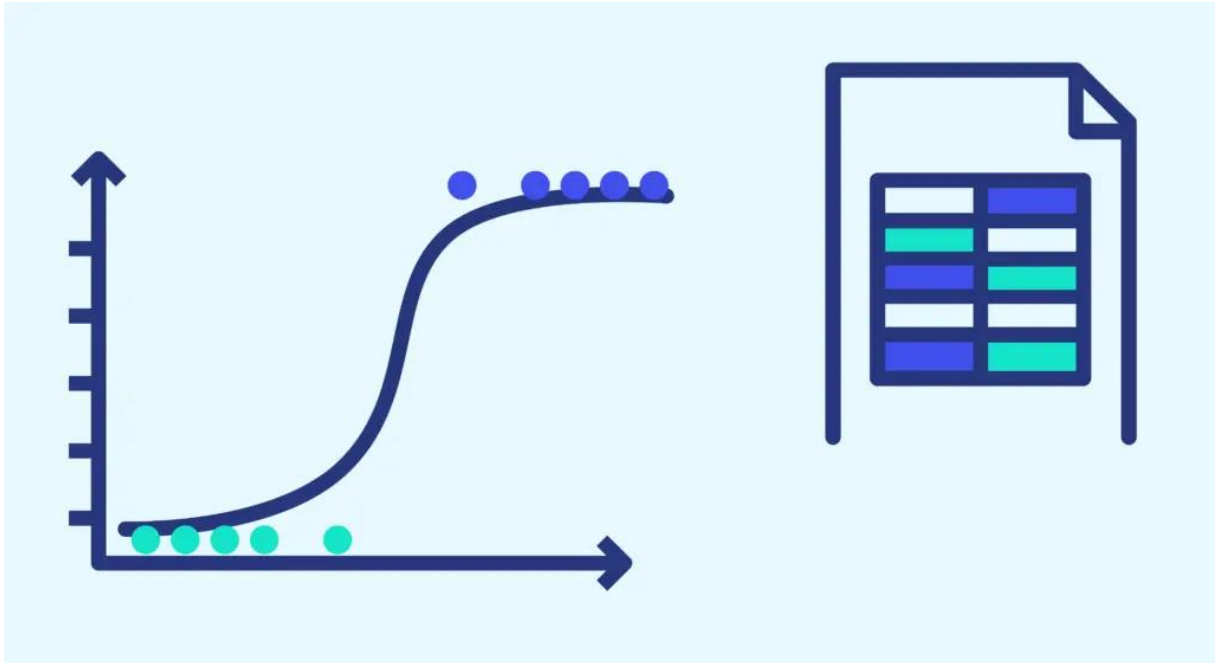
Contexte Général du projet :

Les maladies cardiovasculaires (MCV) sont la première cause de décès dans le monde, prenant environ 17,9 millions de vies chaque année, ce qui représente 31 % de tous les décès dans le monde. Quatre décès par 5CVD sont dus à des crises cardiaques et à des accidents vasculaires cérébraux, et un tiers de ces décès surviennent prématurément chez des personnes de moins de 70 ans. L'insuffisance cardiaque est un événement courant causé par les maladies cardiovasculaires et cet ensemble de données contient 11 caractéristiques qui peuvent être utilisées pour prédire une éventuelle maladie cardiaque.

Les personnes atteintes de maladies cardiovasculaires ou à haut risque cardiovasculaire (en raison de la présence d'un ou plusieurs facteurs de risque tels que l'hypertension, le diabète, l'hyperlipidémie ou une maladie déjà établie) ont besoin d'une détection et d'une prise en charge précoces dans lesquelles un modèle d'apprentissage automatique peut être d'une grande aide.

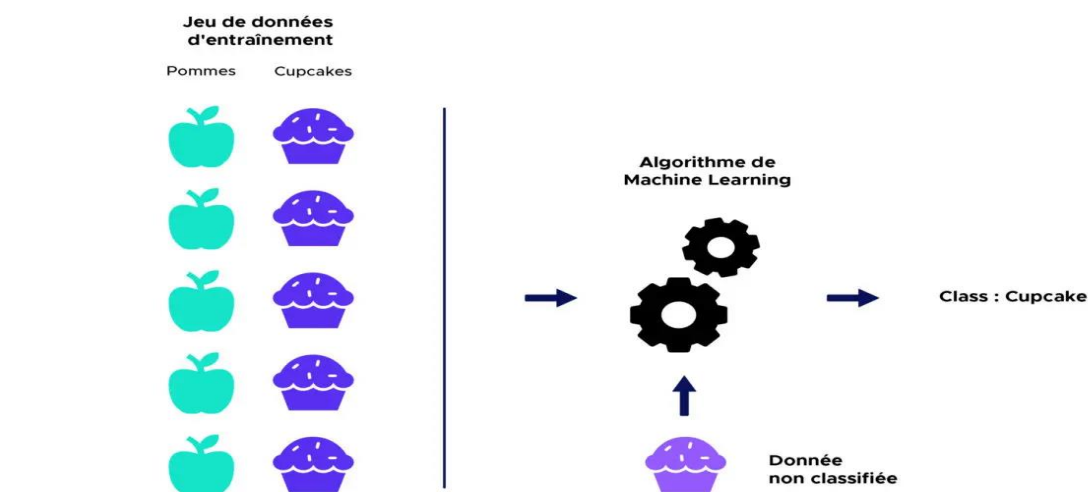
C'est pourquoi nous avons décidé de développer un programme utilisant le langage C, un programme qui nous permet de prédire l'insuffisance cardiaque sur la base d'un ensemble de données qui contient beaucoup d'informations sur cette maladie. Pour cela, nous avons utilisé l'algorithme **KNN (K plus proches voisins ou K-nearest neighbors)**

L'algorithme KNN :



L'algorithme des K plus proches voisins ou K-nearest neighbors (kNN) est un algorithme de Machine Learning qui appartient à la classe des algorithmes d'apprentissage supervisé simple et facile à mettre en œuvre qui peut être utilisé pour résoudre les problèmes de classification et de régression.

Cette algorithme reçoit un ensemble de données qui est étiqueté avec des valeurs de sorties correspondantes sur lequel il va pouvoir s'entraîner et définir un modèle de prédiction. Cet algorithme pourra par la suite être utilisé sur de nouvelles données afin de prédire leurs valeurs de sorties correspondantes.



Ex :

L'intuition derrière l'algorithme des K plus proches voisins est l'une des plus simples de tous les algorithmes supervisés :

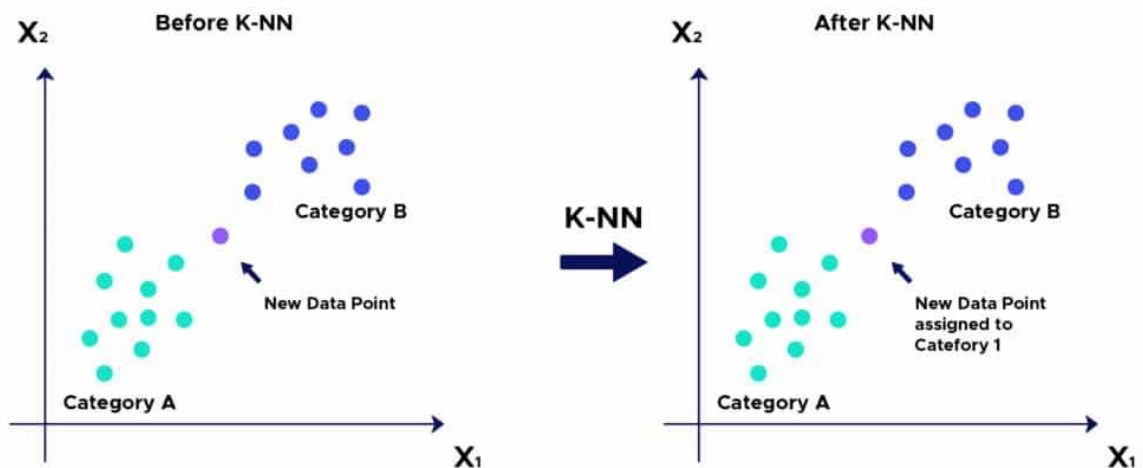
- **Étape 1** : Sélectionnez le nombre K de voisins
- **Étape 2** : Calculez la distance Du point non classifié aux autres points.

On utilise l'EUCLIDIENNE pour calculer la distance :

$$\sum_{i=1}^n |x_i - y_i|$$

- **Étape 3** : Prenez les K voisins les plus proches selon la distance calculée.
- **Étape 4** : Parmi ces K voisins, comptez le nombre de points appartenant à chaque catégorie.
- **Étape 5** : Attribuez le nouveau point à la catégorie les plus présents parmi ces K voisins.

Exemple :



Data-set Utilisé :

Ce jeu de données a été créé en combinant différents jeux de données déjà disponibles indépendamment mais non combinés auparavant. Dans cet ensemble de données, 5 ensembles de données cardiaques sont combinés sur 11 caractéristiques communes, ce qui en fait le plus grand ensemble de données sur les maladies cardiaques disponible à ce jour à des fins de recherche. Les cinq ensembles de données utilisés pour sa conservation sont :

- Cleveland : 303 observations
 - Hongrois : 294 observations
 - Suisse : 123 observations
 - Long Beach Virginie : 200 observations
 - Ensemble de données Stalog (cœur): 270 observations
-
- Total : 1190 observations
 - Dupliqué : 272 observations

Informations sur les attributs :

1. Âge : âge du patient [années]
2. Sexe : sexe du patient [H : Masculin, F : Féminin]
3. ChestPainType : type de douleur thoracique [TA : Angine typique, ATA : Angine atypique, NAP : Douleur non angineuse, IS : Asymptomatique]
4. RestingBP: tension artérielle au repos [mm Hg]
5. Cholestérol : cholestérol sérique [mm/dl]
6. FastingBS : glycémie à jeun [1 : si FastingBS > 120 mg/dl, 0 : sinon]
7. RestingECG: résultats de l'électrocardiogramme au repos [Normal : normal, ST : présentant une anomalie de l'onde ST-T (inversions de l'onde T et/ou élévation ou dépression du segment ST > 0,05 mV), HVG : montrant une hypertrophie ventriculaire gauche probable ou certaine selon les critères d'Estes]
8. MaxHR: fréquence cardiaque maximale atteinte [Valeur numérique entre 60 et 202]
9. ExerciseAngina: exercise-induced angina [Y: Oui, N: Non]
10. Oldpeak: oldpeak = ST Valeur numérique mesurée en dépression]
11. ST_Slope: the slope of the peak exercise ST segment [Up: upsloping, Flat: flat, Down: downsloping]
12. HeartDisease: output class [1: heart disease, 0: Normal]

Format de data-set :

Age	Sex	ChestPainType	RestingBP	Cholesterol	FastingBS	RestingECG	MaxHR	ExerciseAngina	Oldpeak	ST_Slope	HeartDisease
40	M	ATA	140	289	0	Normal	172	N	0	Up	0
49	F	NAP	160	180	0	Normal	156	N	1	Flat	1
37	M	ATA	130	283	0	ST	98	N	0	Up	0
48	F	ASY	138	214	0	Normal	108	Y	1.5	Flat	1
54	M	NAP	150	195	0	Normal	122	N	0	Up	0
39	M	NAP	120	339	0	Normal	170	N	0	Up	0
45	F	ATA	130	237	0	Normal	170	N	0	Up	0
54	M	ATA	110	208	0	Normal	142	N	0	Up	0
37	M	ASY	140	207	0	Normal	130	Y	1.5	Flat	1
48	F	ATA	120	284	0	Normal	120	N	0	Up	0
37	F	NAP	130	211	0	Normal	142	N	0	Up	0
58	M	ATA	136	164	0	ST	99	Y	2	Flat	1
39	M	ATA	120	204	0	Normal	145	N	0	Up	0
49	M	ASY	140	234	0	Normal	140	Y	1	Flat	1
42	F	NAP	115	211	0	ST	137	N	0	Up	0
54	F	ATA	120	273	0	Normal	150	N	1.5	Flat	0
38	M	ASY	110	196	0	Normal	166	N	0	Flat	1
43	F	ATA	120	201	0	Normal	165	N	0	Up	0
60	M	ASY	100	248	0	Normal	125	N	1	Flat	1
36	M	ATA	120	267	0	Normal	160	N	3	Flat	1
43	F	TA	100	223	0	Normal	142	N	0	Up	0
44	M	ATA	120	184	0	Normal	142	N	1	Flat	0
49	F	ATA	124	201	0	Normal	164	N	0	Up	0
44	M	ATA	150	288	0	Normal	150	Y	3	Flat	1

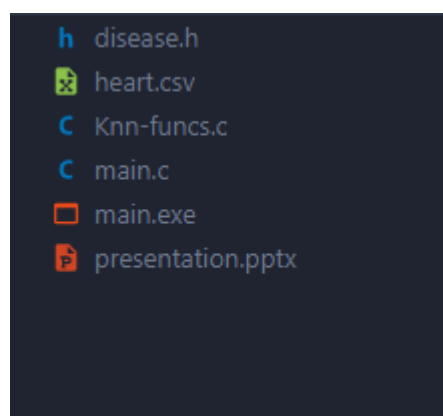
Thème de projet :

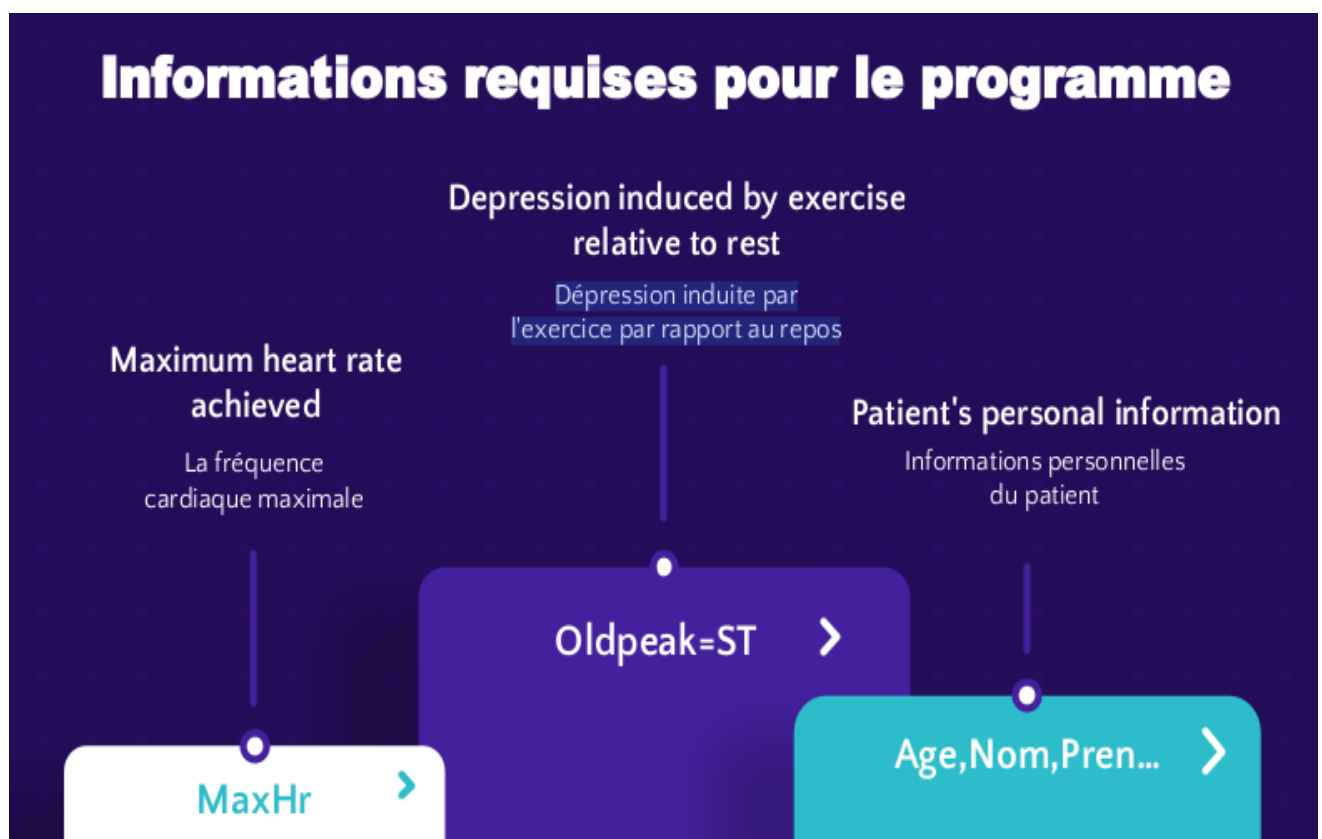
Un projet programmé dans le langage de programmation C qui nous permet de savoir si une personne a une maladie cardiaque ou non, en utilisant l'algorithme kNN et le DATASET qui contiennent des informations de personnes atteintes de maladie cardiaque et de personnes non cardiaques

Outils Utilisé :

- **Langage C** : est un langage de programmation informatique procédural à usage général prenant en charge la programmation structurée, la portée des variables lexicales et la récursivité, avec un système de type statique. De par sa conception, C fournit des constructions qui correspondent efficacement aux instructions machine typiques
- **Heart.csv** : data-set qui contient les informations sur la maladie cardiaque
- **Vs code** : Visual studio code ou VS Code est un éditeur de code développé par Microsoft en 2015. Contrairement à ce à quoi Microsoft a eu l'habitude de nous habituer durant des années, il est l'un de ces premiers produits open source et gratuit, et surtout disponible sur les systèmes d'exploitation Windows, Linux et Mac.

Fichiers de programme:





disease.h :

```
#ifndef DESEASE_HEART
#define DESEASE_HEART
typedef struct {
    int desease;
    double distance;
}Desease;

typedef struct {
    int maxHr;
    float oldpeak;
    char nom[20];
    char prenom[20];
    int age;
}Person;

// fonction pour inserer les informations de la personne qu'on va vérifier si il est malade ou non
Person insereUnNouvellePerson();
// fonction pour printer les informations sur le projet (header)
void printhead();
// fonction pour printer les information de la personne
void printPerson(Person);
// fonction pour Calculer la distance euclidienne
double Euclidean(Person,int,float);
//Fonction pour classer le patient et retourner si malade ou non
int classierPoint(FILE* , int , Person);
// fonction pour supprimer les espaces ou autres caractères prédéfinis du côté gauche d'une chaîne
char *ltrim(char*);
// fonction pour supprimer les espaces ou autres caractères prédéfinis du côté droite d'une chaîne
char *rtrim(char*);
//fonction pour supprimer les espaces et autres caractères prédéfinis des deux côtés d'une chaîne
char *trim(char*);
#endif
```

Knn-funcs.c :

```
void printHeader() {
    system("COLOR 0A");
    printf("\t\t\t\t\t ----- \n");
    Sleep(600);
    system("COLOR 3");
    printf("\t\t\t\t\t | Ecole normale d'enseignement technique ENSET | \n");
    Sleep(600);
    system("COLOR 5");
    printf("\t\t\t\t\t |                Mohammedia                | \n");
    Sleep(600);
    system("COLOR 6");
    printf("\t\t\t\t\t |                Annee Scolaire : 2021/2022                | \n");
    Sleep(600);
    system("COLOR 7");
    printf("\t\t\t\t\t |                Langage C : KNN project                | \n");
    Sleep(600);
    system("COLOR 8");
    printf("\t\t\t\t\t |                Realise par:                | \n");
    Sleep(600);
    system("COLOR 9");
    printf("\t\t\t\t\t |                ENNANI Abdelmalek                | \n");
    Sleep(600);
    system("COLOR B");
    printf("\t\t\t\t\t |                DOUHI Chaïmae                | \n");
    Sleep(600);
    system("COLOR E");
    printf("\t\t\t\t\t |                GLSID-1                | \n");
    Sleep(600);
    system("COLOR 0A");
    printf("\t\t\t\t\t ----- \n");
    Sleep(600);
    printf("\n\n");
    printf("\t\t\t-----\n");
    Sleep(600);
    printf("\t\t\tEnsemble de donnees de prediction d'insuffisance cardiaque utilisant\n");
    printf("\t\t\tl'algorithme KNN\n");
    Sleep(600);
    printf("\t\t\t-----\n\n");
    Sleep(600);
    printf("\t\t\t----- Informations ----- \n");
    Sleep(600);
    printf("\t\t\tLe programme vous demandera des informations sur la personne, pour\n");
    printf("\t\t\tv❖rifier si il est \n\n");
    printf("\t\t\t\tmalade ou non, y compris ces informations suivants : \n\n");
    Sleep(600);
    printf("\t\t\t\t->maxHr : La frequence cardiaque maximale\n");
    Sleep(600);
    printf("\t\t\t\t->oldPeak : depression induced by exercise relative to rest (must be\n");
    printf("\t\t\t\tbetween -2.6 and 6.2)\n");
    Sleep(600);
    printf("\t\t\t\t-----\n\n");
    Sleep(1000);
}
```

Pour pinter des informations sur le programme (header informations)


```
// print newPers information
void printPerson(Person newPerson) {
    printf("\t\tLa personne avec les informations suivant : \n");
    printf("\t\t\tNom : %s\n", newPerson.nom);
    printf("\t\t\tPrenom : %s\n", newPerson.prenom);
    printf("\t\t\tAge : %d\n", newPerson.age);
    printf("\t\t\tmaxHr : %d\n", newPerson.maxHr);
    printf("\t\t\toldPeak : %.1f\n", newPerson.oldpeak);
}
```

La fonction printPerson vous permet de printer les informations du patient

```
// Euclidean function pour calculer la distance
double Euclidean(Person newPers ,int maxHr , float oldpeak) {
    return sqrt(
        (maxHr - newPers.maxHr) * (maxHr - newPers.maxHr) +
        (oldpeak - newPers.oldpeak) * (oldpeak - newPers.oldpeak)
    );
}
```

La fonction Euclidean vous permet de calculer la distance euclidienne entre la personne saisie et chaque personne dans l'ensemble de données



```
// function for ltrim
char *ltrim(char *s)
{
    while(isspace(*s)) s++;
    return s;
}

// fnction for rtrim
char *rtrim(char *s)
{
    char* back = s + strlen(s);
    while(isspace(*--back));
    *(back+1) = '\0';
    return s;
}

// function trim
char *trim(char *s)
{
    return rtrim(ltrim(s));
}
```

Les fonctions trim, ltrim et rtrim permettent de supprimer les espaces avant et après une chaîne


```

// classifier le patient et retourner si il est malade ou non
int classifieurPoint(FILE *data , int k, Person newPers) {
    char test[255]; //pour stocker le contenu de chaque ligne de fichier
    Disease arr[MAX]; //Pour stocker chaque personne et la distance entre lui et la
    //personne que nous étudions
    Disease Point;
    fgets(test , 255 , data); // sauter la première ligne du fichier
    int j=0;
    while(fgets(test , 255 , data)) {
        char *piece = strtok(test , ","); // divise la chaîne de caractères en une série
        // de jetons à l'aide du délimiteur delim ","
        int i=0;
        int maxHr , yesNo; // yesNo : 1 => malade || 0 => n'est pas malade
        float oldpeak;
        while (piece = strtok(NULL , ",")) {
            i++;
            if( i == 7 ) {
                maxHr = atoi(piece);
            } else if ( i == 9 ) {
                oldpeak = (float)atof(piece);
            } else if ( i == 11) {
                yesNo = atoi(trim(piece));
                // printf("la distance est : %f\n" , Euclidean(newPers , maxHr , oldpeak));
                Point.disease = yesNo;
                Point.distance = Euclidean(newPers , maxHr , oldpeak);
                arr[j] = Point;
                j++;
            }
        }
    }

    // trier le tableaux
    int h , c;
    Disease valTemp;
    for ( h=0 ; h<MAX ; h++ ) {
        for( c=1 ; c<MAX ; c++ ) {
            if ( arr[h].distance < arr[c].distance ) {
                valTemp = arr[c];
                arr[c] = arr[h];
                arr[h] = valTemp;
            }
        }
    }

    // get le nombre dominant
    int fre1= 0 , fre2 = 0 , i;
    for ( i=0 ; i<k ; i++ ) {
        if(arr[i].disease == 0) {
            fre1++;
        }
    }

    // return a number (0: n'est pas un patient cardiaque 1:patient cardiaque)
    return fre1 >= 3 ? 0 : 1;
}

```


La fonction `classifierPoint` permet de :

- Recevoir le fichier, la valeur `k` et les informations personnelles comme paramètres
- lire chaque ligne de la fichier `heart.csv`
- extraire les valeurs nécessaires de chaque ligne (`max` et `oldPeak`)
- Mémorisation de la valeur de la maladie (0 ou 1) et de la distance euclidienne entre chaque personne du fichier et la personne dont on veut savoir si elle est malade ou non dans un tableau
- Trier le tableau
- Retourner le nombre dominant dans le premier `K` éléments dans le tableau
- Si 1 est le nombre dominant donc la personne est malade
- Si 0 est le nombre dominant donc la personne n'est pas malade

Main.C fichier :

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <math.h>
#include <conio.h>
#include "disease.h"
#define MAX 918
#include "Knn-funcs.c"

int main() {
    depart :
    system("cls");
    FILE *data;
    printHeader();
    Person newPerson = insereUnNouvellePerson();
    int k = 5; // nombre of data points
    int res;
    data = fopen("./heart.csv" , "r");
    res = classifierPoint(data , k , newPerson);

    printf("\t\t-----\n");
    if ( res == 0 ) {
        printPerson(newPerson);
        printf("\t\t\tResultat : Il a pas de maladie cardiaque");
    } else {
        printPerson(newPerson);
        printf("\t\t\tResultat : On s'attend 💎 ce qu'il souffre d'une maladie cardiaque");
    }

    int num;
    printf("\n\n\t\t\tentrer 0 pour entrer un autre personne : ");
    scanf("%d", &num);

    if( num == 0 ) goto depart;
    return 0;
}
```

Le fichier main.c permet d'exécuter notre programme de langage C.

Il contient des variable nécessaire pour notre programme, comme la variable data qui a le type FILE , cette variable permet d'ouvrir un fichier à l'aide de la fonction fopen.

- la variable newPerson pour stocker les information de la personne
- la variable k pour les k premiers éléments et seulement dans le tableau
- la variable res pour stocker la valeur retourné par classifierPoint fonction

Comment fonctionne le programme:

```
-----  
Ecole normale d'enseignement technique ENSET  
Mohammedia  
Annee Scolaire : 2021/2022  
Langage C : KNN project  
Realise par:  
ENNANI Abdelmalek  
DOUHI Chaimae  
GLSID-1  
-----  
  
-----  
Ensemble de donnees de prediction d'insuffisance cardiaque utilisant l'algorithme KNN  
-----  
  
----- Informations -----  
Le programme vous demandera des informations sur la personne, pour vérifier si il est  
malade ou non, y compris ces informations suivants :  
  
->maxHr : La frequence cardiaque maximale  
->oldPeak : depression induced by exercise relative to rest (must be between -2.6 and 6.2)  
-----  
  
----- Veuillez entrer les information de la personne :-----  
-----  
Entrer Le nom de la person : _
```

Le programme affichera des informations relatives au programme et à ses objectifs, puis il demandera des informations importantes pour que le programme fonctionne, telles que le nom, le prenom, l'âge et La fréquence cardiaque maximale, ainsi que Dépression induite par l'exercice par rapport au repos (valeur entre 0.2 et 0.6)

```
-----  
----- Veuillez entrer les information de la personne :-----  
-----  
Entrer Le nom de la person : Jhon  
Entrer Le prenom de la person : Doe  
Entrer L'age de la personne : 19  
-----  
Entrer Oldpeak (Dépression induite par l'exercice par rapport au repos (-2.6 - 6.2)) : 0.5  
Entrer maxHr (maximum heart rate achieved) : 200
```

```
La personne avec les informations suivant :  
    Nom : Jhon  
    Prenom : Doe  
    Age : 19  
    maxHr : 200  
    oldPeak : 0.5  
    Resultat : Il a pas de maladie cardiaque
```

Après avoir entré toutes les informations requises, le programme prendra ces informations et classera la personne selon qu'elle appartient au groupe malade ou au groupe sain et en conséquence, il nous dira si la personne est malade ou non

```
trier 0 pour entrer un autre personne :
```

si l'utilisateur souhaite entrer les informations d'une autre personne il lui suffit de taper 0 et d'appuyer sur entrée , le programme se rechargera et lui donnera la possibilité d'entrer les informations d'une autre personne

Conclusion :

Notre Mini Projet a pour but la mise en place une programme de prédiction de l'insuffisance cardiaque. Ce Mini projet a été pour nous l'occasion d'approfondir notre connaissance de langage de programmation C.

Pour conclure, au cours de notre mini projet, nous avons entamé une démarche, à mi-chemin entre la gestion du projet et l'auto-formation, qui a permis de mettre la main sur de nombreuses ressources et composants. Ceci nous a permis de passer toutes les étapes de développement