1. **INTRODUCTION**

Heart disease records to be the main source of death around the world. It is difficult for medical practitioners to predict the heart attack as it is a complex task that requires experience and knowledge. The wellbeing area today contains concealed data that can be significant in deciding. Data mining here comes into the image. It is the way toward finding concealed examples in enormous data sets as indicated by alternate points of view for order into valuable data.

Data mining enable the health sector to predict patterns in the dataset.

In this project we will be applying one of the supervised technique that is classification on our dataset to find relationships between the available features and will classify the labels accordingly. Talking about classification when the true goal of our analysis is to predict to which class or group an observation belongs, the techniques we use are termed as classification techniques.

Classification is a classic data mining technique based on machine learning. Basically, classification is used to classify each item in a set of data into one of a predefined set of classes or groups.

Classification method makes use of mathematical techniques such as decision trees, KNN, neural network, and statistics.

**1.1 Symptoms of Disease**

**1.1.1** Fluttering in your chest

**1.1.2** Racing heartbeat (tachycardia)

**1.1.3** Slow heartbeat (bradycardia)

**1.1.4** Chest pain or discomfort

**1.1.5** Shortness of breath

**1.1.6** Lightheadedness

1. **PROBLEM STATEMENT**

The heart disease accounts to be the leading cause of death worldwide. It is difficult for medical practitioners to predict the heart attack as it is a complex task that requires experience and knowledge. The health sector today contains hidden information that can be important in making decisions. Data mining can solve the problem by discovering hidden patterns in large data sets according to different perspectives for categorization into useful information.

1. **OBJECTIVE**

Our objective being effectively using the important data that is generated from the hospitals to find patterns using a well-known Data mining algorithm known as KNN . Heart disease is no doubt has the largest proportions of deaths in the world and researchers are continuously working in this field to find a way out to solve this problem. We discovered some new methods to load the data sets, pre-process the data and build the classifier in C language and also tried to minimize the time complexity of our code.

1. **Design**
   1. **Methodology**

**4.1.1 Load the dataset:** Before any calculations it is necessary to load the dataset in our workspace. We used comma-separated values(.csv) to load the dataset. After that we used arrays for this purpose as these provide a good way for doing necessary calculations in the data.

**4.1.2 Data Preprocessing:** Data preprocessing or data cleaning steps involves handling the along with removal of unnecessary features. In the data preprocessing we used the modular programming approach.

* Data Cleaning: In data cleaning we used to separate the NULL values by the mean and median of that value.
* Data Reduction: In this process we have to reduce the dimension of our dataset so the non-relevant attributes can be removed.
* Feature Scaling: The feature scaling is used to reduce the range of any attributes. For feature scaling we used the standardization to reduce the value.

**New = Xi-Mean/standard deviation**

**4.1.3 Splitting the data into train and test sets:** once we have trained our classifier, we can’t test it on the same data set we used for training, thus before training a classifier we split our dataset into train and test sets.

**4.1.4 Build the classifier:** we will use K Nearest Neighbors algorithm for classifying the observations. KNN works by selecting majority value from the k nearest data points and classify the new observation. K Nearest Neighbours is one of the most essential classification algorithms and is widely used with small data sets to classify the observations. The algorithm belongs to the supervised learning domain and finds intense application in pattern recognition, data mining and intrusion detection. After splitting the dataset into train and test sets build KNN classifier using the algorithm below

* 1. **Algorithum**

**4.2.1** Load the data.

**4.2.2** Initialize K to your chosen number of neighbors.

**4.2.3** For each data sample.

**4.2.3.1** Calculate the distance between the new data point and every other data point in the data set.

**4.2.3.2** Append the distance and the index of the new data point to a new array.

**4.2.4** Sort the ordered collection of distances and indices from in ascending order of distance

**4.2.5** Pick the first K values from the sorted collection.

**4.2.6** Get the labels of the selected K values.

**4.2.7** If regression, return the mean of the K labels.

**4.2.8** If classification, return the mode of the K labels.

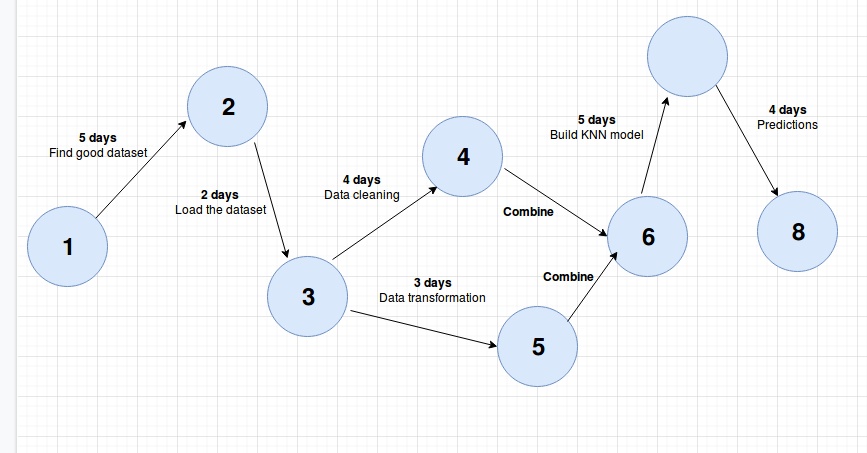
**Pert Chart**

Fig 4.1 “Pert Chart”

**Flow Chart**

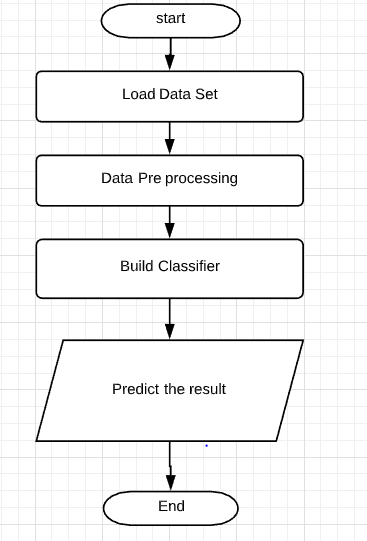


Fig 4.2 “Flow Chart of Work”

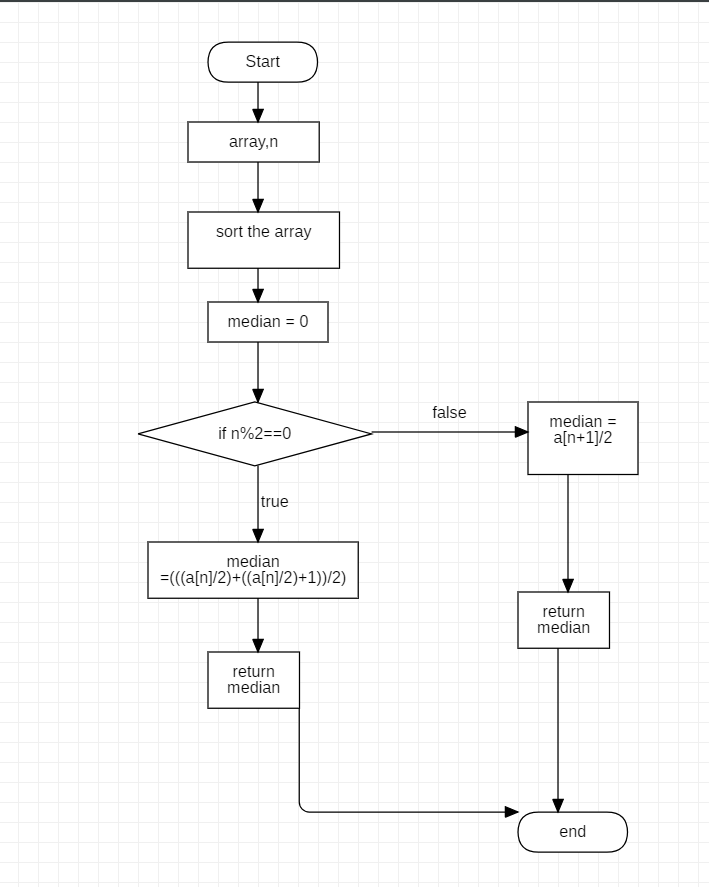


Fig 4.3 “Flow Chart Median”

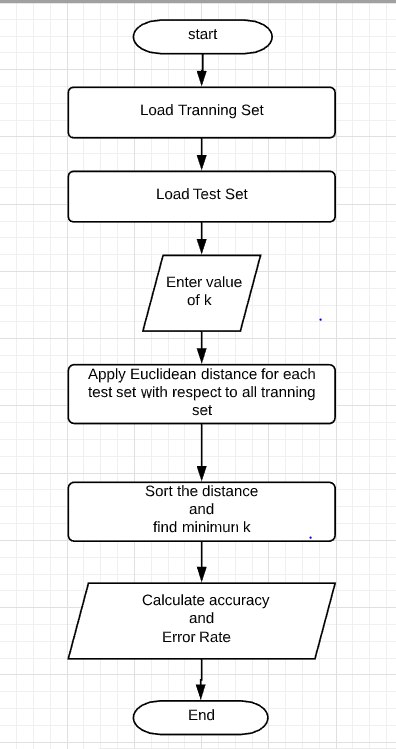


Fig 4.4 “Flow Chart KNN”

1. **Implementation**

**5.1 Project Code**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include<math.h>

#include"median.h"

#include"mean.h"

#define BSIZE 304

#define B1SIZE 14

#define Train 242

#define Test 61

int u=0,w=0,c=Train;

//Distance Calculation

void distance(float gl2[Test][B1SIZE],float gl1[Train][B1SIZE],int x,int t,int c)

{

int k = 5,temp,di[Train],count=0,count1=0;

int i,j=0,m,a[Test],a2[Test];

float val=0;

float dis[Train],z;

float res[Train],res1[Test];

for(i=1; i<Train ;i++)

{

for(j=0;j<13;j++)

{

val += pow((gl2[c][j]-gl1[i][j]),2);

}

dis[i] = sqrt(val);

res[i] = gl1[i][13];

di[i]=i;

}

for(i=Train;i<BSIZE;i++)

{

res1[i]=gl2[i][13];

a2[i]=res1[i];

}

//Sorting the Distance and Corresponding Class Value

for(i=1;i<Train;i++)

{

for(m=1;m<Train-i;m++)

{

if(dis[m]>dis[m+1])

{

temp=dis[m];

dis[m]=dis[m+1];

dis[m+1]=temp;

temp=res[m];

res[m]=res[m+1];

res[m+1]=temp;

temp=di[m];

di[m]=di[m+1];

di[m+1]=temp;

}

}

}

//Printing value of K-nearest Distance

printf("\n------------------------\n");

printf("\nTarget values of %d nearest neighbors are:\n",k);

printf("\n------------------------\n");

for(i=1;i<=k;i++)

{

printf("%f\t %0.0f\n",dis[i],res[i]);

}

for(i=1;i<=k;i++)

{

if(res[i]==0)

{

count++;

}

else

{

count1++;

}

}

if(count>count1)

{

a[x]=0;

}

else

{

a[x]=1;

}

printf("Calculated Value: %d\n",a[x]);

printf("Observed Value:%d\n",a2[t]);

if(a[x] == a2[t])

{

u++;

}

else

{

w++;

}

//Checking for Correct Value

if(x==60)

{

printf("\n------------------------\n");

printf("Correct Value: %d\n",u);

printf("InCorrect Value: %d\n",w);

z=((float)u/(float)(Test-1))\*100;

//Printing Accuracy and Error

printf("\n------------------------\n");

printf("Accuracy: %f\n",z);

printf("Error: %f\n",100-z);

}

}

//Calculating Standard Deviation

float calculateSD(float data[],int n,float mean)

{

float standardDeviation = 0.0;

int i;

for(i=0;i<n;i++)

standardDeviation += pow((data[i] - mean), 2);

return sqrt(standardDeviation/n);

}

//Main Function

int main()

{

int n = 13;

// specify filename

float median=0,\*ch,sum=0.0,mean,mean2,mean3,sd,sd1,sd2;

char filename[] = "heart.csv";

char buffer[BSIZE];

FILE \*f;

// we need to specify the column names

char \*field;

float age[BSIZE],sex[BSIZE],cp[BSIZE],trestbps[BSIZE], chol[BSIZE],fbs[BSIZE],restecg[BSIZE],thalach[BSIZE],exang[BSIZE],oldpeak[BSIZE],slope[BSIZE],ca[BSIZE],thal[BSIZE],target[BSIZE];

char gl[BSIZE][B1SIZE];

int i=0,j=0,x=1,t=Train;

float glu[BSIZE][B1SIZE],gl1[Train][B1SIZE],gl2[Test][B1SIZE];

/\* open the CSV file \*/

f = fopen(filename,"r");

if( f == NULL)

{

printf("Unable to open file '%s'\n",filename);

exit(1);

}

while(fgets(buffer,BSIZE,f))

{

/\* get age \*/

field=strtok(buffer,",");// strtok() is used for splitting a string by some delimiter.

age[i]=atof(field);//atof is a function that converts a string into an float representation.

/\* get sex \*/

field=strtok(NULL,",");

sex[i]=atof(field);

/\* chest pain type \*/

field=strtok(NULL,",");

cp[i]=atof(field);

/\* resting blood pressure (in mm Hg on admission to the hospital) \*/

field=strtok(NULL,",");

trestbps[i]=atof(field);

/\* serum cholestoral in mg/dl \*/

field=strtok(NULL,",");

chol[i]=atof(field);

/\* (fasting blood sugar > 120 mg/dl) (1 = true; 0 = false) \*/

field=strtok(NULL,",");

fbs[i]=atof(field);

/\* restecgresting electrocardiographic results \*/

field=strtok(NULL,",");

restecg[i]=atof(field);

/\* maximum heart rate achieved \*/

field=strtok(NULL,",");

thalach[i]=atof(field);

/\* exercise induced angina (1 = yes; 0 = no) \*/

field=strtok(NULL,",");

exang[i]=atof(field);

/\* ST depression induced by exercise relative to rest \*/

field=strtok(NULL,",");

oldpeak[i]=atof(field);

/\* the slope of the peak exercise ST segment \*/

field=strtok(NULL,",");

slope[i]=atof(field);

/\* number of major vessels (0-3) colored by flourosopy \*/

field=strtok(NULL,",");

ca[i]=atof(field);

/\* 3 = normal; 6 = fixed defect; 7 = reversable defect \*/

field=strtok(NULL,",");

thal[i]=atof(field);

/\* 1 or 0 \*/

field=strtok(NULL,",");

target[i]=atof(field);

i++;

}

//storing value row-wise

for(i=1;i<BSIZE;i++)

{

glu[i][j]=age[i];

glu[i][j+1]=sex[i];

glu[i][j+2]=cp[i];

glu[i][j+3]=trestbps[i];

glu[i][j+4]=chol[i];

glu[i][j+5]=fbs[i];

glu[i][j+6]=restecg[i];

glu[i][j+7]=thalach[i];

glu[i][j+8]=exang[i];

glu[i][j+9]=oldpeak[i];

glu[i][j+10]=slope[i];

glu[i][j+11]=ca[i];

glu[i][j+12]=thal[i];

glu[i][j+13]=target[i];

j=0;

}

j=0;

//Data Preprocessing

//finding NULL value in the data set

for(i=1;i<BSIZE;i++)

{

if(glu[i][j]==0.0)

{

age[i]=Find\_median(age , BSIZE);// if NULL value the put median in place of NULL value

glu[i][j]=age[i];

}

}

j=0;

//feature scaling using Standardization

mean = mean1(chol,BSIZE);//finding mean

mean2 = mean1(trestbps,BSIZE);//finding mean

mean3 = mean1(thalach,BSIZE);//finding mean

sd=calculateSD(chol,BSIZE,mean);//calculating Standard deviation from calculateSD function

sd1=calculateSD(trestbps,BSIZE,mean2);

sd2=calculateSD(thalach,BSIZE,mean3);

for(i=1;i<BSIZE;i++)

{

chol[i]=((chol[i]-mean)/sd);//using Standardization formula xnew=((x[i] -xmean)/standard deviation)

trestbps[i]=((trestbps[i]-mean2)/sd1);

thalach[i]=((thalach[i]-mean3)/sd2);

glu[i][j+3]=trestbps[i];

glu[i][j+4]=chol[i];

glu[i][j+7]=thalach[i];

}

//printing the data set

printf("age \t sex \t cp \t trestbps \t chol \t fbs \t restecg \t thalach \t exang \t oldpeak \t slope \t ca \t thal \t target\n\n");

for(i=1;i<BSIZE;i++)

{

for(j=0;j<B1SIZE;j++)

{

printf("%0.2f\t",glu[i][j]);

}

printf("\n");

}

j=0;

//Divide the data into training and test

//Training Dataset

for(i=1;i<=Train;i++)

{

gl1[i][j]=glu[i][j];

gl1[i][j+1]=glu[i][j+1];

gl1[i][j+2]=glu[i][j+2];

gl1[i][j+3]=glu[i][j+3];

gl1[i][j+4]=glu[i][j+4];

gl1[i][j+5]=glu[i][j+5];

gl1[i][j+6]=glu[i][j+6];

gl1[i][j+7]=glu[i][j+7];

gl1[i][j+8]=glu[i][j+8];

gl1[i][j+9]=glu[i][j+9];

gl1[i][j+10]=glu[i][j+10];

gl1[i][j+11]=glu[i][j+11];

gl1[i][j+12]=glu[i][j+12];

gl1[i][j+13]=glu[i][j+13];

j=0;

}

// Printing Training Data

printf("age \t sex \t cp \t trestbps \t chol \t fbs \t restecg \t thalach \t exang \t oldpeak \t slope \t ca \t thal \t target\n\n");

for(i=1;i<Train;i++)

{

for(j=0;j<B1SIZE;j++)

{

printf("%0.2f\t",gl1[i][j]);

}

printf("\n");

}

j=0;

//Testing Dataset

for(i=Train;i<BSIZE;i++)

{

gl2[i][j]=glu[i][j];

gl2[i][j+1]=glu[i][j+1];

gl2[i][j+2]=glu[i][j+2];

gl2[i][j+3]=glu[i][j+3];

gl2[i][j+4]=glu[i][j+4];

gl2[i][j+5]=glu[i][j+5];

gl2[i][j+6]=glu[i][j+6];

gl2[i][j+7]=glu[i][j+7];

gl2[i][j+8]=glu[i][j+8];

gl2[i][j+9]=glu[i][j+9];

gl2[i][j+10]=glu[i][j+10];

gl2[i][j+11]=glu[i][j+11];

gl2[i][j+12]=glu[i][j+12];

gl2[i][j+13]=glu[i][j+13];

j=0;

}

//Printing Test Data

printf("age \t sex \t cp \t trestbps \t chol \t fbs \t restecg \t thalach \t exang \t oldpeak \t slope \t ca \t thal \t target\n\n");

for(i=Train;i<BSIZE;i++)

{

for(j=0;j<B1SIZE;j++)

{

printf("%0.2f\t",gl2[i][j]);

}

printf("\n");

}

j=0;

printf("\n----Applying KNN----\n");

printf("\nDistances calculated");

printf("\nDistances sorted");

for(i=1;i<Test;i++)

{

distance(gl2,gl1,x,t,c);

x++;

t++;

c++;

}

fclose(f);//closing the file

return(0);

}

**MEAN**

#include<stdio.h>

float mean1(float data[304],int n)

{

int i;

float mean,sum=0.0;

for(i=0 ; i<n ; i++)

{

sum=sum+data[i];

}

mean=sum/n;

return mean;

}

**MEDIAN**

#include<stdio.h>

void merge(float a[],int i1,int j1,int i2,int j2);

void mergesort(float a[],int i,int j)

{

int mid;

if(i<j)

{

mid=(i+j)/2;

mergesort(a,i,mid); //left recursion

mergesort(a,mid+1,j); //right recursion

merge(a,i,mid,mid+1,j); //merging of two sorted sub-arrays

}

}

void merge(float a[],int i1,int j1,int i2,int j2)

{

int temp[304]; //array used for merging

int i,j,k;

i=i1; //beginning of the first list

j=i2; //beginning of the second list

k=0;

while(i<=j1 && j<=j2) //while elements in both lists

{

if(a[i]<a[j])

temp[k++]=a[i++];

else

temp[k++]=a[j++];

}

while(i<=j1) //copy remaining elements of the first list

temp[k++]=a[i++];

while(j<=j2) //copy remaining elements of the second list

temp[k++]=a[j++];

//Transfer elements from temp[] back to a[]

for(i=i1,j=0;i<=j2;i++,j++)

a[i]=temp[j];

}

float Find\_median(float array[] , int n)

{

mergesort(array ,0, n);

float median=0;

// if number of elements are even

if(n%2 == 0)

median = (array[(n-1)/2] + array[n/2])/2.0;

// if number of elements are odd

else

median = array[n/2];

return median;

}

**5.2 Output Screen**

Dataset :-

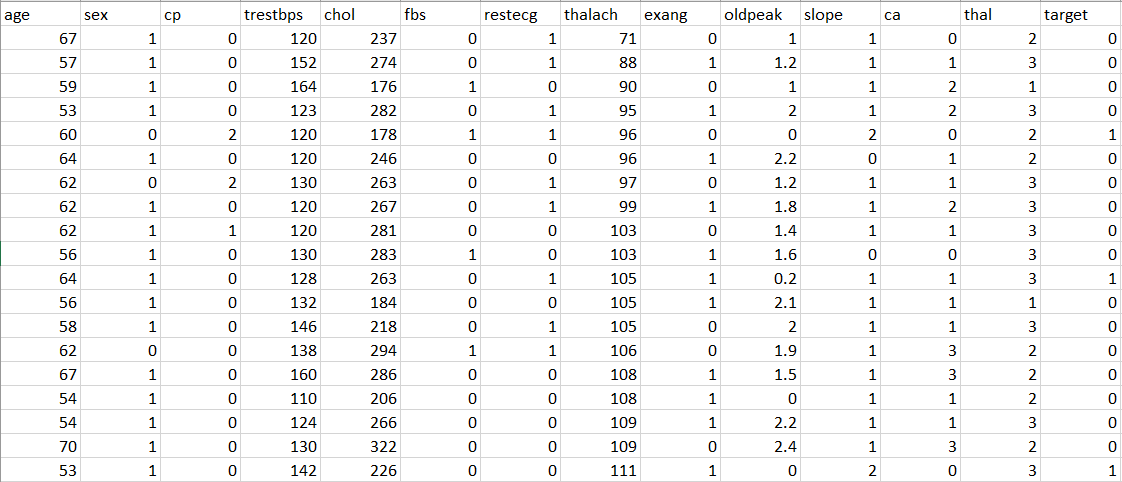


Fig 5.1 “Dataset”

This is the dataset, it has 304 data samples including 13 independent variables and 1 target variable with two labels 0 and 1. 0 specifies that the person is not having a heart disease and vice versa for 1.

Data after preprocessing:-

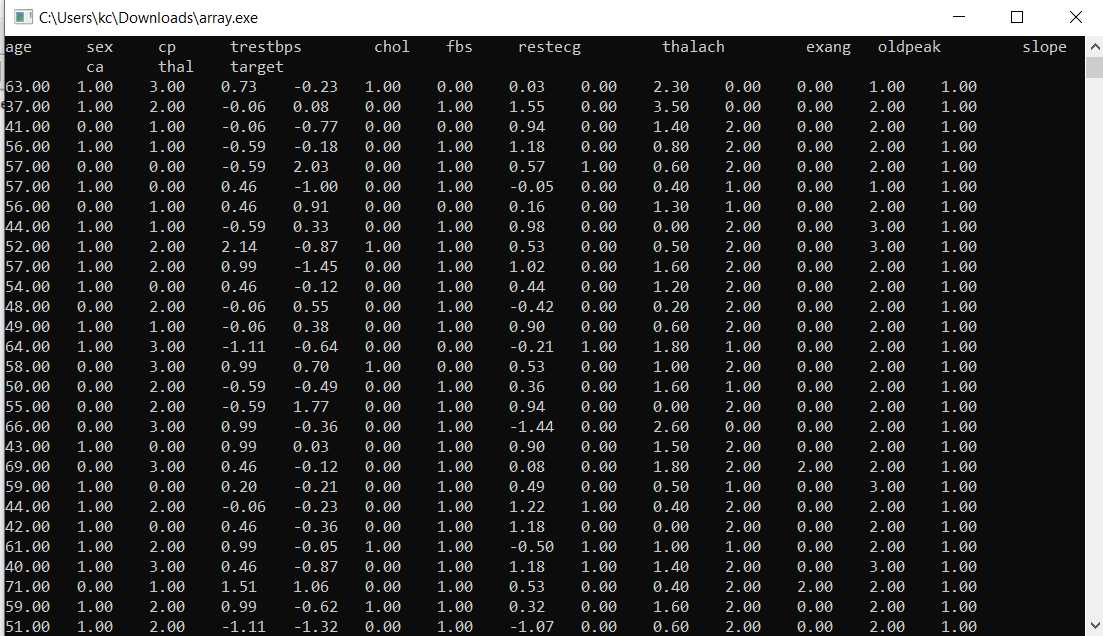
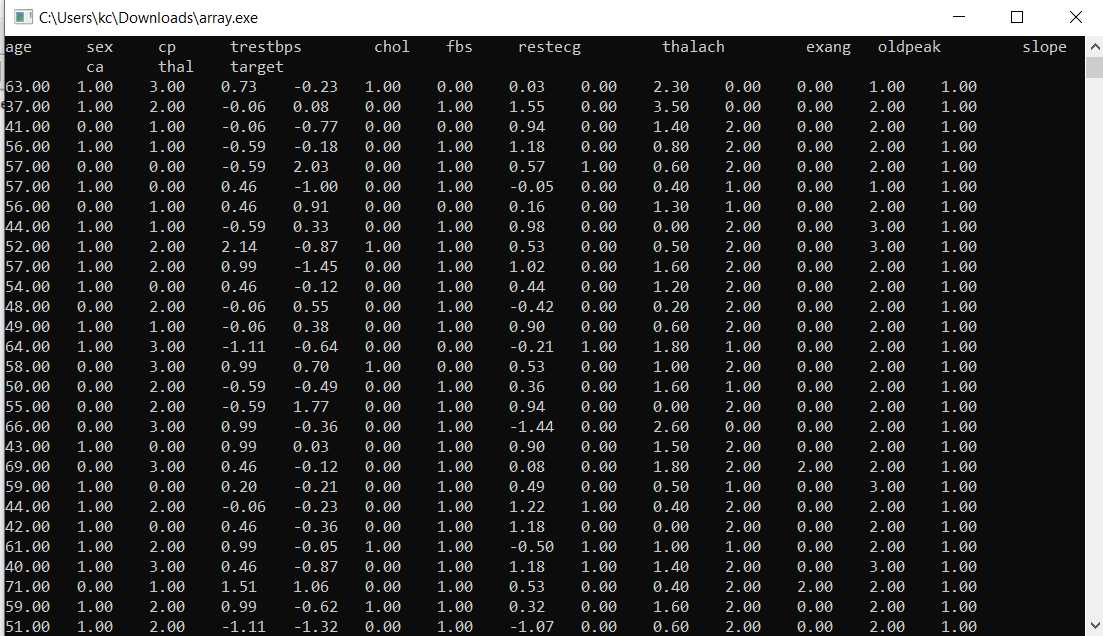


Fig 5.2 “Data after preprocessing”

The dataset later contained some null values so in the data preprocessing step null values were replaced by mean of the variable following which the dataset was transformed using standardization. For feature scaling we used the standardization to reduce the value.

**New\_value = Xi-Mean/standard deviation**

Training Set:-

 Fig 5.3 “Training Set”

Test Set:-

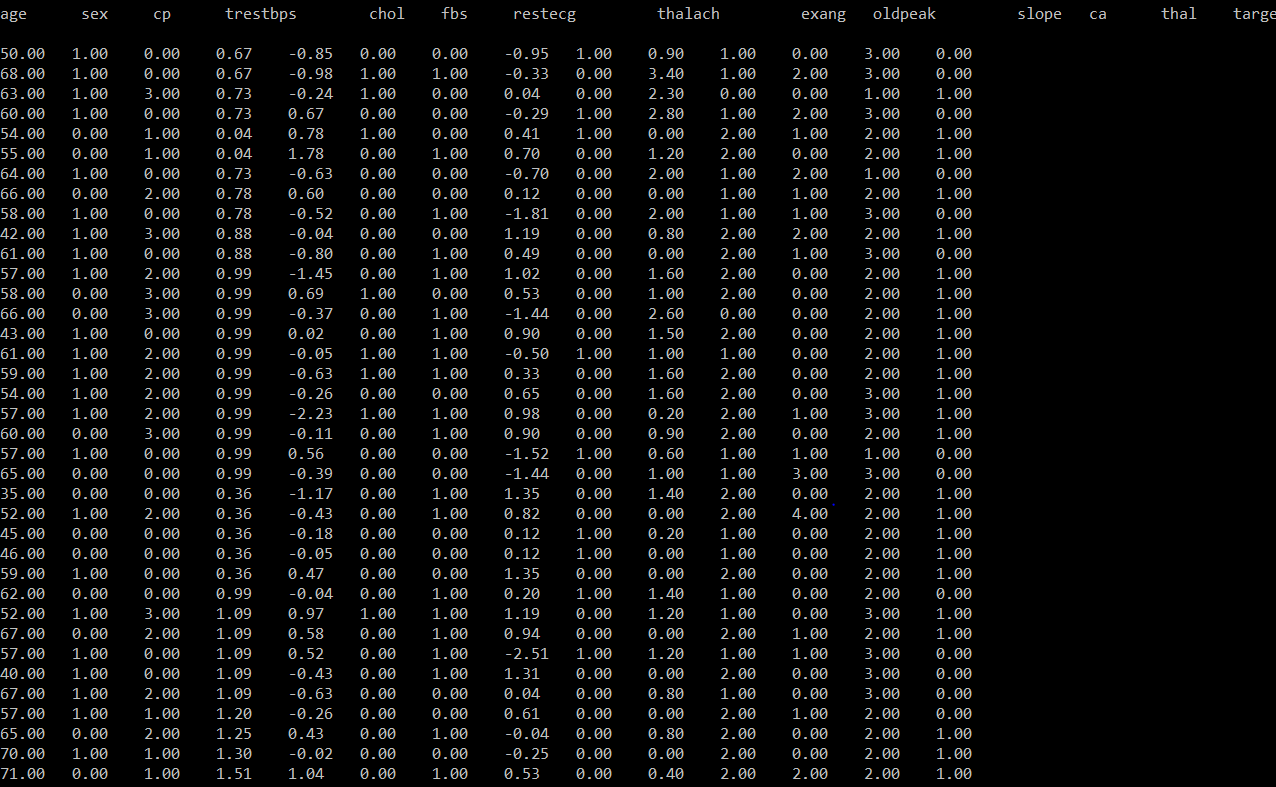


Fig 5.4 “Test Set”

The dataset was split into train and test set in ratio 80-20 with Training set containing 242 observations and Test set with 61 observations. Both the datasets after preprocessing and splitting are shown above.

Apply KNN on Test Set:-

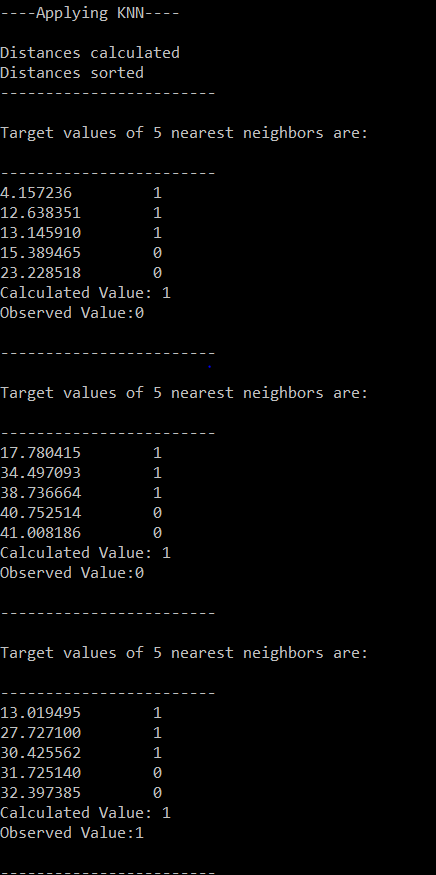


Fig 5.5 “Distance calculated for each test set”

For each test observation Euclidean distance was calculated and shortest K distances were taken into consideration following that the max occurring value from sorted K distance was chosen as the class of the test observation. Above is the snapshot for the distance values calculated for each test data. The code snippet is below. For each test observation the loop iterated over each data sample calculating its distance and storing the distance and the value of its corresponding target variable in res and dis arrays. The values are then finally sorted and K values were taken.

for(i=1; i<Train ;i++)

{

for(j=0;j<13;j++)

{

val += pow((gl2[c][j]-gl1[i][j]),2);

}

dis[i] = sqrt(val);

res[i] = gl1[i][13];

di[i]=i;

}

Result:-

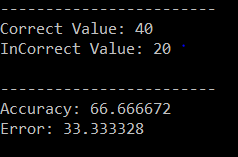


Fig 5.6 “Accuracy of KNN”

After getting the labels for test set accuracy was calculated by dividing correctly predicted observations to total number of observations which came out to be 66% . The accuracy was so because of less size of the dataset(304 observations).

1. **Result analysis**

We trained the KNN model on our test data that gave an accuracy of 66%. The same dataset was tested against python KNN model that gave an accuracy of 67%. The motive being able to build a model in C language that can give results similar to Python and that is what we got. Thus we conclude that even though the accuracy of the model is not much high but it gave an accuracy similar to Python KNN model. That proves the fact that the model is good enough to work well on large datasets similar to Python.

Accuracy with C KNN model:

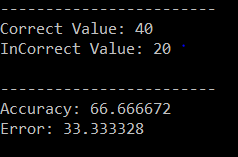


Fig 6.1 “Final result in c”

Accuracy with Python KNN model:

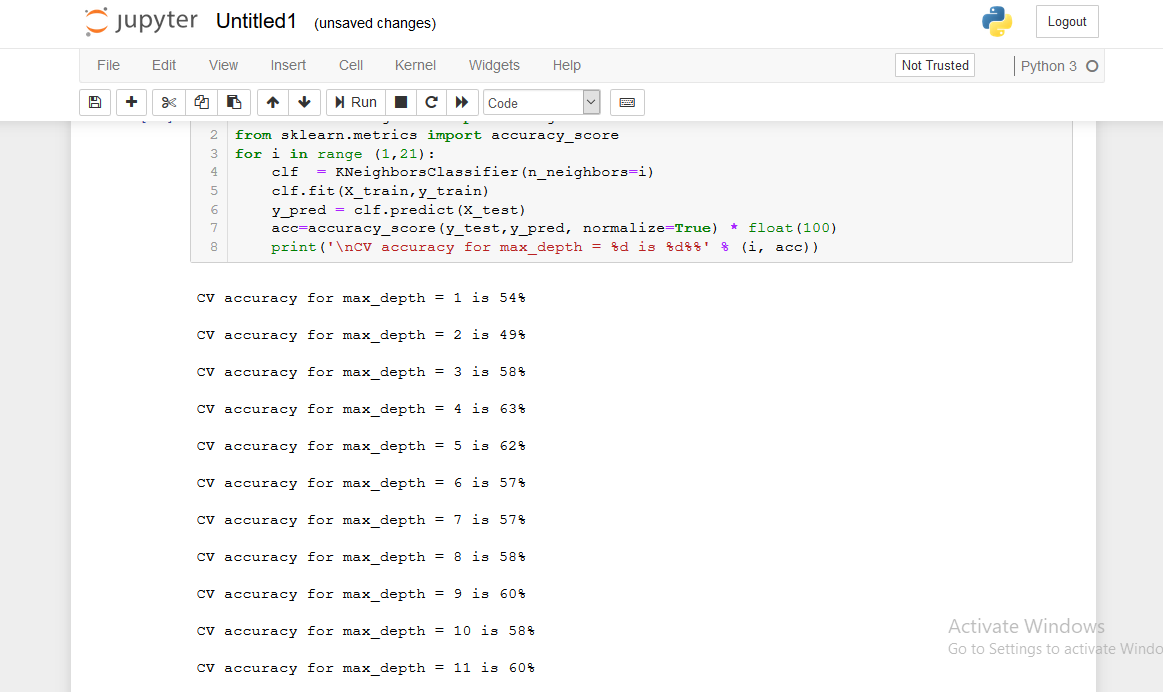


Fig 6.2 “Result in python”

**7. Future Scope**

1. The project gives us a better understanding of how machine learning algorithms work.
2. The project gave an efficient implement of KNN algorithm in C language which was not available till date.
3. The project will be carried ahead and will be transformed into a research paper under the guidance of Dr. Tanupriya Choudhury
4. The project explained how data preprocessing on arrays that turned out to be a new way for handling large amount of data in arrays.

**References**

[1]. K losgen W and Zytkow J M (eds.), Handbook of data mining and knowledge discovery, OUP, Oxford, 2002

[2]. Klosgen W and Zytkow J M (eds.), Handbook of data mining and knowledge discovery, OUP, Oxford, 2002.

[3]. Klosgen W and Zytkow J M (eds.), Handbook of data mining and knowledge discovery, OUP, Oxford, 2002.

[4]. C. Ordonez, “Association rule discovery with the train and test approach for heart disease prediction,” IEEE transactions on Information technology in biomedicine: a publication of the IEEE Engineering in Medicine and Biology Society, vol. 10, no. 2,pp.334– 43,Apr.2006.

[5]. C. Ordonez, “Association rule discovery with the train and test approach for heart disease prediction,” IEEE transactions on Information technology in biomedicine: a publication of the IEEE Engineering in Medicine and Biology Society, vol. 10, no. 2,pp.334–43,Apr.2006.

[6]. [N. Amma, “Cardiovascular disease prediction system using genetic algorithm and neural network,” in International Conference on Computing, Communication and Applications. Dindigul, Tamilnadu, India: IEEE, Feb 2012, pp. 1–5.](https://ieeexplore.ieee.org/abstract/document/5569740)

[7]. Vikas Chaurasia, and Saurabh Pal, Data Mining Approach to Detect Heart Dieses, International Journal of Advanced Computer Science and Information Technology, Vol. 2, No. 4, pp. 56- 66, 2013.

[8]. S. Amin, K. Agarwal, and R. Beg, “Genetic neural network based data mining in prediction of heart disease using risk factors,” in IEEE Conference on Information Communication Technologies. Thuckalay, Tamil Nadu, India, April 2013, pp. 1227–1231. https://doi.org/10.1109/CICT.2013.6558288.

[9]. K. Sudhakar, Dr. M. Manimekalai, “Study of Heart Disease Prediction using Data Mining”, International Journal of Advanced Research in Computer Science and Software Engineering, Volume 4, Issue 1, pp.1157-60, January 2014.

[10]. B. Venkatalakshmi and M. Shivsankar, “Heart disease diagnosis using predictive data mining,” International Journal of Innovative Research in Science, Engineering and Technology, vol. 3, no. 3, pp. 1873–1877, 2014.

[11]. H. Masethe and M. Masethe, “Prediction of heart disease using classification algorithms,” in Proceedings of the world congress on Engineering and Computer Science, San Francisco, USA: Interna-tional Association of Engineers (IAENG), 2014, pp. 22–24.

[12]. H. Almarabeh and E. Amer, “A study of data mining techniques accuracy for healthcare,” International Journal of Computer Appli-cations, vol. 168, no. 3, pp. 12–17, Jun 2017.