**Name: Mohammad Zain Abbas**

**Reg #: 6865**

**DE-36 (CE), Syndicate: A**

**LAB 11 JOURNEL**

**Equipment Used:** Notebook Computer, Matlab , Visual Studio(C#)

**Lab Tasks:**

1) Use the given cancer dataset within MATLAB and classify it using KNN classifier:

a) First create a MATLAB script and load ‘cancer’ mat file.

b) Identify features and classes from the loaded dataset.

c) Perform 2-fold cross validation on the dataset by splitting it into testing and

training parts.

d) Implement a KNN classifier using the above algorithm and use training dataset to

classify each of the sample within testing dataset. Plot a graph to display the accuracy

as you vary ‘K’ from 1 to 20. (Use Euclidean distance to m

**SOLUTION CODE:**

load('cancer.mat');

data=dataset;

cutoff=floor(length(data)/2);

trainingFeatures=data(1:cutoff,1:end-1);

testingFeatures=data(cutoff+1:end,1:end-1);

test\_labels=(data(cutoff+1:end,end));

training\_labels=data(1:cutoff,end);

[test\_rows,test\_cols] = size(testingFeatures);

[train\_rows,train\_cols] = size(trainingFeatures);

K\_Nearest\_Values=zeros(1,10);

% K\_Indexes=zeros(1,10);

% K=1:2:20;

closest\_labels=[];

Accuracy=[];

i=1;

predicted\_labels=zeros(test\_rows,1);

E\_Distance = zeros(1,train\_rows);

for K=1:20

for i=1:test\_rows

for j = 1:train\_rows

E\_Distance(j,1)=sqrt(sum((testingFeatures(i,:)-trainingFeatures(j,:)).^2));

end

[Sorted\_Distances,Index]=sort(E\_Distance);

K\_Indexes=Index(1:K);

closest\_labels=training\_labels(K\_Indexes);

index\_class0=0;

index\_class1=0;

for n = 1:length(closest\_labels)

if(closest\_labels(n)==0)

index\_class0= index\_class0+1;

else

index\_class1 = index\_class1+1;

end

end

if(index\_class0>index\_class1)

predicted\_labels(i) = 0;

else

predicted\_labels(i) = 1;

end

count=sum((predicted\_labels-test\_labels)==0);

Accuracy(K)=(count/length(predicted\_labels))\*100;

end

end

[max\_Accuracy,Index\_max]=max(Accuracy);

plot(Index\_max,max\_Accuracy,'rv','MarkerFaceColor','b');

hold on

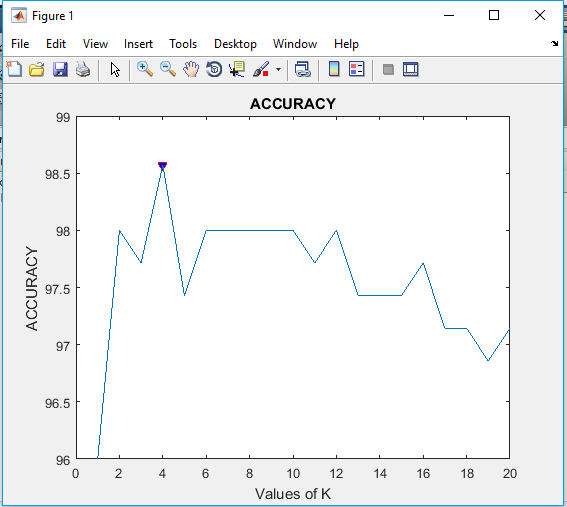
plot(Accuracy);

title('ACCURACY');

xlabel('Values of K');

ylabel('ACCURACY');

**OUTPUT:**



2 2) Develop a C# console application to implement KNN classification algorithm for the

following dataset and classify the given test vector:

**SOLUTION CODE:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Task2\_KNN

{

//class Program

//{

// static void Main(string[] args)

// {

// }

//}

public class Features

{

public int Age;

public int Loan;

public char Defaulter;

public Features(int Age,int Loan,char Defaulter)

{

this.Age = Age;

this.Loan = Loan;

this.Defaulter = Defaulter;

}

}

class KNN\_Algorithm

{

public static void Main()

{

List<Features> Train\_Dataset = new List<Features>();

List<Features> Test\_Dataset = new List<Features>();

Train\_Dataset.Add(new Features(25, 40000, 'N'));

Train\_Dataset.Add(new Features(35, 60000, 'N'));

Train\_Dataset.Add(new Features(45, 80000, 'N'));

Train\_Dataset.Add(new Features(20, 20000, 'N'));

Train\_Dataset.Add(new Features(35, 120000, 'N'));

Train\_Dataset.Add(new Features(52, 18000, 'N'));

Train\_Dataset.Add(new Features(23, 95000, 'Y'));

Train\_Dataset.Add(new Features(40, 62000, 'Y'));

Train\_Dataset.Add(new Features(60, 100000, 'Y'));

Train\_Dataset.Add(new Features(48, 220000, 'Y'));

Train\_Dataset.Add(new Features(33, 150000, 'Y'));

Test\_Dataset.Add(new Features(48, 120000, '?'));

for(int i=0;i<Test\_Dataset.Count();i++)

{

List<Double> E\_Distances = new List<double>();

for (int j=0;j<Train\_Dataset.Count();j++)

{

Double Distance = Math.Sqrt(Math.Pow((Test\_Dataset[i].Age - Train\_Dataset[j].Age), 2) + Math.Pow((Test\_Dataset[i].Loan - Train\_Dataset[j].Loan), 2));

Console.WriteLine("Distance From Age "+Train\_Dataset[j].Age+" and Loan "+Train\_Dataset[j].Loan +" = " + Distance);

E\_Distances.Add(Distance);

}

List<Double> Sorted\_Array = new List<double>(E\_Distances);

Sorted\_Array.Sort();

int Index = E\_Distances.IndexOf(Sorted\_Array[0]);

Console.WriteLine("Minimum Distance = " + Sorted\_Array[0] + ", Index: " + Index + "\n");

Console.WriteLine("Test Element Classify to Class " + Train\_Dataset[Index].Defaulter);

}

}

}

}

**OUTPUT:**

