**Name: Mohammad Zain Abbas**

**Reg #: 6865**

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

**LAB 12 JOURNEL**

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

**Lab Tasks:**

1) Use the given cancer dataset within MATLAB and classify it using Bayesian classification

model:

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 Bayesian classifier using the above algorithm and use training dataset

to classify each of the sample within testing dataset.

e) Compute the accuracy from the predicted test samples.

**SOLUTION CODE:**

warning off

clear all;

load ('cancer.mat')

[rows,cols]=size(dataset);

Training\_Dataset=dataset(1:rows/2,1);

Training\_Labels=dataset(1:rows/2,10);

Testing\_Dataset=dataset(rows/2:end-1,1);

Testing\_Labels=dataset(rows/2:end-1,10);

Testing\_Labels=Testing\_Labels';

counter0=0;

counter1=0;

for i=1:length(Training\_Labels)

if(Training\_Labels(i)==0)

counter0=counter0+1;

else

counter1=counter1+1;

end

end

Prior\_Probability0=counter0/length(Training\_Labels);

Prior\_Probability1=counter1/length(Training\_Labels);

sum0=0;

sum1=0;

count0=0;

count1=0;

for i=1:length(Training\_Labels)

if(Training\_Labels(i)==0)

sum0=sum0+Training\_Dataset(i);

count0=count0+1;

else

sum1=sum1+Training\_Dataset(i);

count1=count1+1;

end

end

mean0=sum0/count0;

mean1=sum1/count1;

sum\_var0=0;

sum\_var1=0;

for i=1:length(Training\_Labels)

if(Training\_Labels(i)==0)

sum\_var0=sum\_var0+((Training\_Dataset(i)-mean0)^2);

else

sum\_var1=sum\_var1+((Training\_Dataset(i)-mean1)^2);

end

end

variance0=sum\_var0/count0;

variance1=sum\_var1/count1;

Likelihood\_Probability0=zeros(1,length(Testing\_Dataset));

Likelihood\_Probability1=zeros(1,length(Testing\_Dataset));

for j=1:length(Testing\_Dataset)

Likelihood\_Probability0(j)= (1/sqrt(2\*pi\*variance0))\*(exp(-((Testing\_Dataset(j)-mean0)^2)/(2\*variance0)));

Likelihood\_Probability1(j)= (1/sqrt(2\*pi\*variance1))\*(exp(-((Testing\_Dataset(j)-mean1)^2)/(2\*variance1)));

end

Posterior\_Probability0=Likelihood\_Probability0 .\* Prior\_Probability0;

Posterior\_Probability1=Likelihood\_Probability1 .\* Prior\_Probability1;

Evidance=Posterior\_Probability0+Posterior\_Probability1;

Normalized\_Posterior\_Probability0=Posterior\_Probability0./Evidance;

Normalized\_Posterior\_Probability1=Posterior\_Probability1./Evidance;

Predicted\_Labels=zeros(1,length(Normalized\_Posterior\_Probability1));

for k=1:length(Normalized\_Posterior\_Probability0)

if(Normalized\_Posterior\_Probability0(k)>=Normalized\_Posterior\_Probability1(k))

Predicted\_Labels(k)=0;

else

Predicted\_Labels(k)=1;

end

end

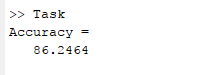
Accuracy=sum(Predicted\_Labels==Testing\_Labels);

Accuracy=(Accuracy/length(Testing\_Labels)\*100);

disp('Accuracy = ');

disp(Accuracy);

**OUTPUT:**



2.Develop a C# console application to implement Bayesian classification model 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 Lab12

{

public class FeatureVector

{

public int Age;

public char Defaulter;

public FeatureVector(int age, char defaulter)

{

this.Age = age;

this.Defaulter = defaulter;

}

}

class Program

{

public static void Main(string[] args)

{

float prior0 = 6 / 11f;

float prior1 = 5 / 11f;

List<FeatureVector> TrainDataset = new List<FeatureVector>();

List<FeatureVector> TestVector = new List<FeatureVector>();

TrainDataset.Add(new FeatureVector(25, '0'));

TrainDataset.Add(new FeatureVector(35, '0'));

TrainDataset.Add(new FeatureVector(45, '0'));

TrainDataset.Add(new FeatureVector(20, '0'));

TrainDataset.Add(new FeatureVector(35, '0'));

TrainDataset.Add(new FeatureVector(52, '0'));

TrainDataset.Add(new FeatureVector(23, '1'));

TrainDataset.Add(new FeatureVector(40, '1'));

TrainDataset.Add(new FeatureVector(60, '1'));

TrainDataset.Add(new FeatureVector(48, '1'));

TrainDataset.Add(new FeatureVector(33, '1'));

TestVector.Add(new FeatureVector(48, 'X'));

float mean0 = mean('0', TrainDataset);

float mean1 = mean('1', TrainDataset);

float var0 = variance('0', mean0, TrainDataset);

float var1 = variance('1', mean1, TrainDataset);

float Likelihood0 = Likelihood\_Probability(TestVector[0].Age,'0', mean0, var0);

float Likelihood1 = Likelihood\_Probability(TestVector[0].Age,'1', mean1, var1);

float Posterior0 = Posterior\_Probability(Likelihood0, '0', prior0);

float Posterior1 = Posterior\_Probability(Likelihood1, '1',prior1);

float evidence = Posterior0 + Posterior1;

float normalized\_Probability0 = Posterior0 / evidence;

float normalized\_Probability1 = Posterior1 / evidence;

Console.WriteLine("Normalized Probability of 0" + ": " + normalized\_Probability0);

Console.WriteLine("Normalized Probability of 1" + ": " + normalized\_Probability1);

if (normalized\_Probability0 > normalized\_Probability1)

{

Console.WriteLine("Sample is assigned to Class 0");

}

else

{

Console.WriteLine("Sample is assigned to Class 1");

}

}

public static float mean(char classVal, List<FeatureVector> TrainData)

{

float mean = 0;

float sum = 0;

int count = 0;

for (int c = 0; c < TrainData.Count; c++)

{

if (TrainData[c].Defaulter == classVal)

{

sum = sum + TrainData[c].Age;

count++;

}

}

mean = sum / count;

Console.WriteLine("Mean " + classVal + ": " + mean);

return mean;

}

public static float variance(char classVal, float meanVal, List<FeatureVector> TrainData)

{

float variance = 0;

float count = 0;

float sum = 0;

for (int c = 0; c < TrainData.Count; c++)

{

if (TrainData[c].Defaulter == classVal)

{

sum = sum + (float)Math.Pow((TrainData[c].Age - meanVal), 2);

count++;

}

}

variance = sum / count;

Console.WriteLine("Variance " + classVal + ": " + variance);

return variance;

}

public static float Likelihood\_Probability(int TestSample, char classVal, float meanVal, float varianceVal)

{

float Probability = 0;

Probability = (float)(1 / Math.Sqrt(2 \* Math.PI \* varianceVal)) \* (float)Math.Exp(-(float)Math.Pow((TestSample - meanVal), 2) / (2 \* varianceVal));

Console.WriteLine("Likelihood Probability " + classVal + ": " + Probability);

return Probability;

}

public static float Posterior\_Probability(float Likelihood, char classVal,float PriorProbability)

{

float Posterior\_Prob = Likelihood \* PriorProbability;

Console.WriteLine("Posterior Probability " + classVal + ": " + Posterior\_Prob);

return Posterior\_Prob;

}

}

}

**OUTPUT:**

