Breast Cancer Classification using Logical Analysis of Data

A DISSERTATION

Submitted in the partial fulfilment of

the requirements for the award of the degree

of

Master of Technology

in

COMPUTER SCIENCE AND ENGINEERING

by

Nagendra Sunil Nigade

Department of Computer Science and Engineering

Indian Institute of Technology, Roorkee

Roorkee – 247667

May, 2019

CANDIDATE’S DECLARATION

I declare that the work presented in this dissertation with title “**Breast Cancer Classification using Logical Analysis of Data**” towards fulfilment of the requirement for the award of the degree of **Master of Technology** in **Computer Science and Engineering, Indian Institute of Technology Roorkee , India** is an authentic record of my own work carried out during the period of **June 2018 to May 2019** under the supervision of **Dr**. **Sugata Gangopadhyay** , Professor, Department of Computer Science and Engineering, Indian Institute of Technology Roorkee, India. The content of this dissertation has not been submitted by me for the award of any other degree of this or any other institute.

Date: Nagendra Sunil Nigade
Place: Roorkee (17575017)

CERTIFICATE

This is to certify that the above statement made by candidate is correct to the best of my knowledge and belief.

Date: **Dr. Sugata Gangopadhyay**
Place: Roorkee Dissertation Supervisor

ACKNOWLEDGEMENTS

I would never have been able to complete my dissertation without the guidance of my supervisor, help from friends, and support from my family and loved ones.

First and foremost, I would like to extend my heartfelt gratitude to my guide and mentor **Dr. Sugata Gangopadhyay**, Professor, Department of Computer Science and Engineering, Indian Institute of Technology Roorkee, for his invaluable guidance, and encouragement and for sharing his broad knowledge. His wisdom, knowledge and commitment to the highest standards inspired and motivated me. He has been very generous in providing the necessary resources to carry out my research. He is an inspiring teacher, a great adviser, and most importantly a nice person.

I am also grateful to the Dept. of Computer Science and Engineering, IIT-Roorkee for providing valuable resources to aid my research. Finally, hearty thanks to my parents, wife and siblings, who encouraged me in good times, and motivated me in the bad times, without which this dissertation would not have been possible.

Abstract

In current situation, each user is generating gigabytes of data each day. The quantity of data is so huge that we have dependent on machine learning models to capture import data / to capture patterns from data that can be useful for future prediction.

To make use of machine learning models, we required continuous computational power. The system will breakdown if connection to computation breaks. When we try to reduce the computational power then we have to compromise with accuracy.

This thesis represents the idea of “Breast Cancer Classification using Logical Analysis of Data”

The thesis presents a review of the basic concepts of the Logical Analysis of Data & put the focus on the various methods those can be used in different components of LAD. Binarization methods includes different methodologies to covert complex attributes into binary.

The main feature of the Logical Analysis of Data (LAD) is to find minimum set of features those can cover all the observation with approaches like coefficient correlation, threshold count, set covering. The decision tree classifier has been used to find the patterns in the observations.

This thesis also looks for a hardware implementation of classifier so that continuous connectivity is no longer needed.

Table of Contents

Table of Figures

Chapter 1

1.Introduction

1.1 Logical Analysis of Data (LAD)

Logical Analysis of Data (LAD) is a method to analyze data based on combinatorial and optimization approach. Logically analyzing of data has been used in many fields, such as economics, health care, business etc. The basic concept behind LAD is to use combinatorial approach with differentiation and integration combined together on a data set D containing positive as well as negative observations, and the "unclassified new" ones i.e. which have not been seen till now, where differentiation means finding out a collection of subsets of attributes of D which are either strongly positive or strongly negative characterized and for the integration process, union of these collected subsets is taken as the approximation of data points of D representing positive or negative observations.

Following are the main components of LAD:

a. The LAD tool is designed to logically analyze the real world data using binary attributes. The tool learns from the binary attributes. The data can be Boolean type as well as real. If the input data is not of binary form, it is expressed with the help of k new binary attributes. The input data is converted to binary values and this process of conversion is called binarization.

b. Above process can lead to a lot of new attributes. The new set of attributes may have a few attributes which may not have significant effect on the observation's result value. To remove those such redundant attributes, support set minimization is carried out.

c. We find out positive and negative patterns separately from the positive and negative observation data set respectively in such a way that they cover significant number of observations in their respective classes.

d. Two new sets are formed, one having all positive patterns and other having all the negative patterns. Union of these two sets is defined as a "model".
e. This model is used to classify data directly into either positive or negative class for the observations covered by this union and to predict a class label for the observations uncovered by this set.
f. The results of the classification system discussed above are tested to verify the accuracy of the model.

The above specified techniques of LAD such as binarization of real world data, eliminating redundant attributes, finding the frequent patterns, forming a model along with testing and analyzing it are discussed in details in the following chapters.

1.2 Problem Statement

Implement modules for the components of LAD, namely, binarization, support set minimization and pattern generation and integrate them to produce a working logical data analysis tool.
We are also going to develop hardware device that will help to classify the real time data. This device will work without constant internet connectivity and low power supply of 5V.

1.3 Report Structure

The flow of the complete report is as follows. In chapter 2, we focused on literature review and research gaps. Chapter 3 presents insight about binarization and how it is implemented in this project. Chapter 4 provides the detailed discussion about different techniques to achieve support set minimization. Chapter 5 presents different approaches to perform the pattern generation. Chapter 6 gives the brief idea about hardware implementation. Chapter 7 gives focus on the Arduino code which is essential for successful execution of microcontroller. Chapter 8 shows entire flow of the proposed work with details. Chapter 9 presents the results of the project. Chapter 10 provides the conclusion of this report.

Chapter 2

2.Literature Review

Over the period of time, the incredible amount of data is being generated in every minute. To process this data efficiently we take the help of machine learning / Deep learning models.

Neural network models like Alex Net required really high computation power & takes days in training. Some real time application requires quick results such as stock classification, Intrusion detection. Most of the machine learning models provides either accuracy or efficiency. If your application requires higher accuracy, then you have to compromise with response time or vice versa.

2.1 Comparison with other module

Accuracy vs Computation

It's very difficult and tricky to choose one over another. Most of the machine learning algorithms are either computation efficient or accuracy effective.

Throughout internet connectivity

Even though internet connectivity / computational power is easily available. Many parts of world are still lacking in basic needs like electricity, Internet connectivity, computation devices. So we should look for models that are energy efficient (runs on battery) & doesn't require constant connectivity to internet to perform prediction. As machine learning models requires more computation, it is mandatory to have constant connection to computation. Disconnection to computation device will lead to failure of system.

Speed

How long does it take to build the model & how long does it take to predict outcome is really important. Speed always plays important role in real time application. Most of the current model lacks in speed when we demand for acceptable accuracy. Highest speed can be achieved if we build the model at hardware level like electric circuit.

Demand for data

Machine learning & deep learning models requires large amount of data. Deep learning models have higher accuracy in abundance of data but they perform poorly than traditional models when data is limited. Deep learning models work best when the amount of training data available for each class is roughly equal.

2.2 Research Gaps

One major drawback of neural network architectures is that the reasoning behind the decisions made by them is not interpretable by humans. Although deep learning models have high accuracy on most of the task, they are only suitable for research purposes. Applications of neural network models in real world cannot be used until we make sure that no decision is made that can severely affect a human’s life adversely. Neural networks do not have the concept of support sets which can be tracked to identify the frequency of occurrence of a particular pattern. In the real world, where actions taken based on the predictions of AI systems affect everyone, cannot and should not be taken without knowing the reason behind the prediction. The lack of justifiability of decisions is an important drawback of the current deep learning models. Also, due to lack of interpretability by humans, neural networks can be difficult to debug. A source of bug may never be understood and the whole model may have to be re-implemented / redesigned because of this. A lack of interpretability hinders development process for building solutions.

One aspect in which deep learning models lack is the cost to develop a solution. Deep learning models are computationally very expensive. The time required for training a neural network model is tremendous. It is not unheard of to train a neural network for several weeks to get a good accuracy rate. This type of process is only viable for specific circumstances. Faster and cheaper models need to be used to get a quick prediction while waiting for the deep learning model to finish training.

The demand for data in deep learning models is huge. Millions of images have to be provided to the model in order to achieve good results. Although sometimes deep learning models work with limited data, most of the times they do not. The demand for such huge amounts of data is tremendous and cannot always be met. For smaller datasets, traditional machine learning models work much better than deep learning models.

A few of the above discussed problems are addressed in the proposed solution.

Chapter 3

3.Binarization

3.1 Introduction

The initial version of LAD was developed to analyze the dataset which consist only binary (0, 1) attributes. But most of the real life applications have real values & to perform the conversion from real values to binary values, a Binarizaion method was proposed in.

In many problems, it is not necessary that all the attributes will be in binary form. Most of the attributes are complex in structure means they can real number or nominal attribute. Example of nominal attributes consist color, type, Citizenship, Employment Type, Etc.
To fit such problem in concept of LAD, the problems needs to be converted into binary format.

Binarizaion is process of converting complex attribute into binary form. The real value variables are transformed into some k binary variables based on thresholds. Binarization module manages the transformation of real value data to binary values.

The binarization technique have huge impact on the accuracy of the LAD module. So we have to try multiple techniques of binarization, and best one will be selected based on the accuracy.

3.2 Algorithm used for binarization

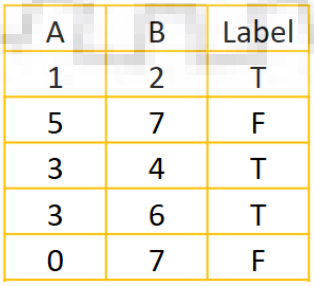
We order the dataset in non-descending way on the basis of attribute X, and the attribute has values $X_1 >= X_2 >= . . . >= X_n$. [1]

(i) We consider the threshold between the interval X_i, X_{i+1} where X_i & X_{i+1} are the observation belongs to different class. (Positive and Negative class)

(ii) At most one threshold should be captured from each of the above intervals X_i, X_{i+1}

```
1. For each Numerical attribute in dataset:
    a. Sort the dataset in descending order with respect to attribute selected above.
    b. For each row i in dataset:
        i. If data.result(i) != data.result(i+1) and data.attribute(i) != data.attribute(i+1)
        ii. Then place the cut point, threshold is calculated using mean value of both the attributes.
        iii. Put the threshold value in array
    c. For each value in threshold array:
        i. Introduce new binary variable
        ii. The value of binary variable is 0 if it's less than threshold value and 1 if it's greater than threshold value.
```

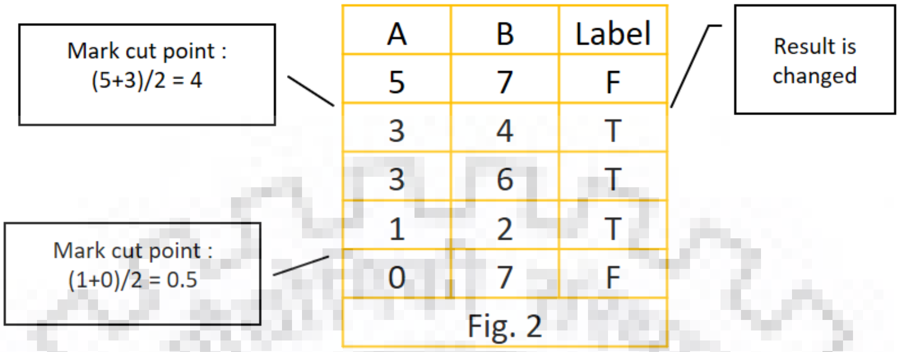
3.3 Binarization – Step by Step



A	B	Label
1	2	T
5	7	F
3	4	T
3	6	T
0	7	F

Figure 1 Binarization-1 Dataset before binarization

Sort the database in descending order on the basis of attribute A, to find the threshold of attribute A.

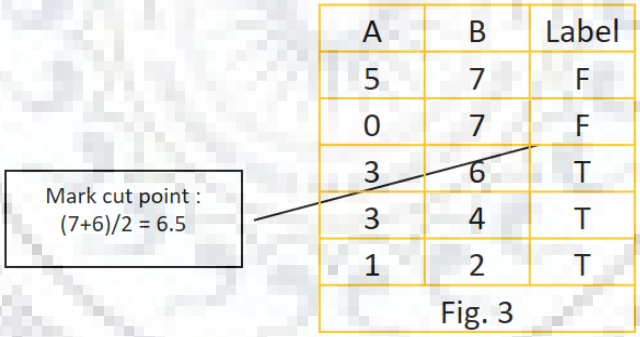


A	B	Label
5	7	F
3	4	T
3	6	T
1	2	T
0	7	F

Figure 2 Binarization-2

Here, we can see that the results have been flipped for first and last two rows of dataset. At the same time the values associated with this rows are also not equal.

Threshold vector for attribute A consist 2 values, (5+3)/2=4 and (0+1)/2=0.5



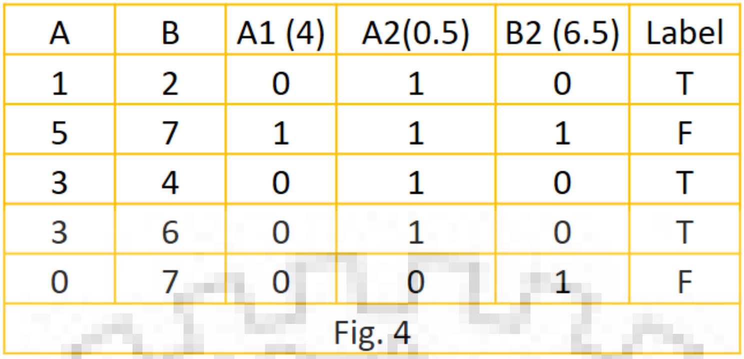
A	B	Label
5	7	F
0	7	F
3	6	T
3	4	T
1	2	T

Figure 3 Binarization-3

Similarly, threshold for the attribute B calculated as 6.5.

So binarization process will have 2 threshold matrix. Matrix 1 for the attribute A = {4,0.5} and matrix 2 for the attribute B = {6.5}

For binarization process each of the attribute has to pass through its threshold matrix. This will covert original attribute to K interval attributes.



A	B	A1 (4)	A2(0.5)	B2 (6.5)	Label
1	2	0	1	0	T
5	7	1	1	1	F
3	4	0	1	0	T
3	6	0	1	0	T
0	7	0	0	1	F
Fig. 4					

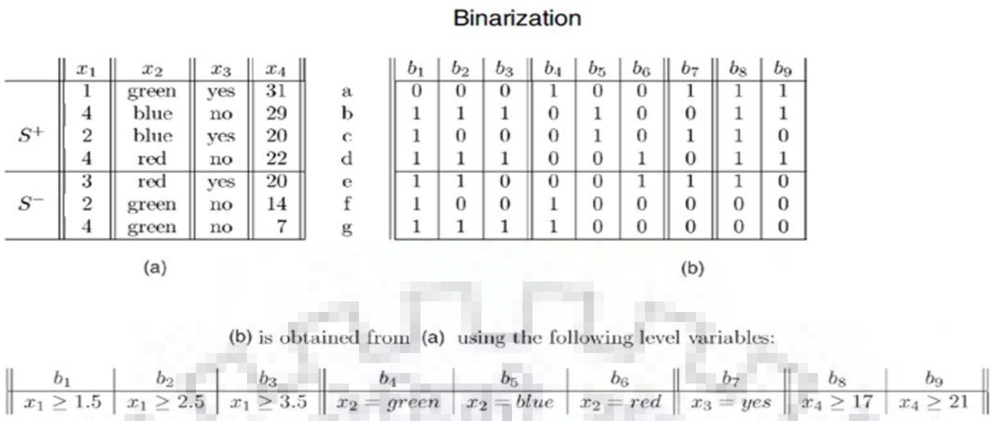
Figure 4 Table after binarization process.

3.4 What if attribute is Non-Numerical / nominal

The non-binary attribute is called as descriptive or nominal attributes. Nominal attributes are the one which consist information like Employment type and the values can be Government Employee, Private Sector Employee, Freelancer, Businessman and so on. Straightforward way to convert this attribute into K binary attributes by associating with each value V_s of the attribute X to Boolean variable b(x, V_s)[2] such that

$$b(x, v_s) = \begin{cases} 1 & \text{if } x = v_s \\ 0 & \text{otherwise} \end{cases}.$$

Figure 5 Binarization for Nominal Attributes



Binarization

	x_1	x_2	x_3	x_4
S^+	1	green	yes	31
	4	blue	no	29
	2	blue	yes	20
	4	red	no	22
S^-	3	red	yes	20
	2	green	no	14
	4	green	no	7

(a)

	b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8	b_9
a	0	0	0	1	0	0	1	1	1
b	1	1	1	0	1	0	0	1	1
c	1	0	0	0	1	0	1	1	0
d	1	1	1	0	0	1	0	1	1
e	1	1	0	0	0	1	1	1	0
f	1	0	0	1	0	0	0	0	0
g	1	1	1	1	0	0	0	0	0

(b)

(b) is obtained from (a) using the following level variables:

b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8	b_9
$x_1 \geq 1.5$	$x_1 \geq 2.5$	$x_1 \geq 3.5$	$x_2 = green$	$x_2 = blue$	$x_2 = red$	$x_3 = yes$	$x_4 \geq 17$	$x_4 \geq 21$

3.5 Interval Variable

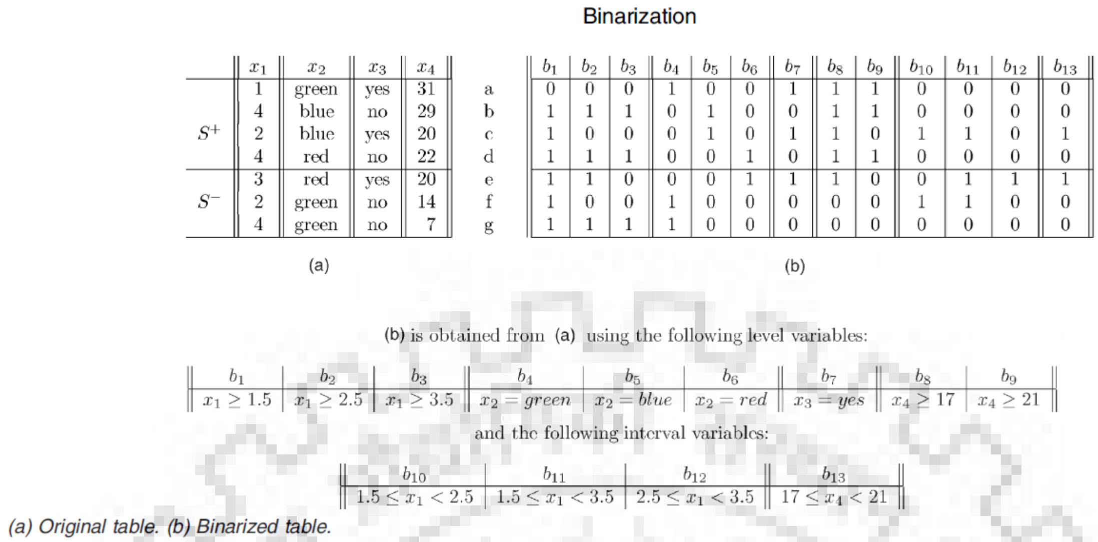
“Boros & Hammer (An implementation of logical analysis of data, 2000) introduced additional binary variable to make binarization process more powerful.”

The interval variables have been introduced in binarization process which are associated with every pair of cut-points. The value of this variable is calculated depends on whether it is inside or outside of the interval of two cut points.

For every attribute X and pair of cut points t` & t``, the value of introduced Boolean attribute b(x, t`, t``) such that

$$b(x, t', t'') = \begin{cases} 1 & \text{if } t' \leq x < t'' \\ 0 & \text{otherwise} \end{cases}.$$

For example, if attribute A has 3 cut-points {1.5, 2.5, 3.5} then 3 binary variables will be introduced by binarization process. The value A_1 will be 1 if it’s in range of 1.5-2.5, 0 otherwise. For A_2 it should be in range of 1.5-3.5 and so on.



Binarization

	x_1	x_2	x_3	x_4
S^+	1	green	yes	31
	4	blue	no	29
	2	blue	yes	20
	4	red	no	22
S^-	3	red	yes	20
	2	green	no	14
	4	green	no	7

(a)

	b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8	b_9	b_{10}	b_{11}	b_{12}	b_{13}
a	0	0	0	1	0	0	1	1	1	0	0	0	0
b	1	1	1	0	1	0	0	1	1	0	0	0	0
c	1	0	0	0	1	0	1	1	0	1	1	0	1
d	1	1	1	0	0	1	0	1	1	0	0	0	0
e	1	1	0	0	0	1	1	1	0	0	1	1	1
f	1	0	0	1	0	0	0	0	0	1	1	0	0
g	1	1	1	1	0	0	0	0	0	0	0	0	0

(b)

(b) is obtained from (a) using the following level variables:

b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8	b_9
$x_1 \geq 1.5$	$x_1 \geq 2.5$	$x_1 \geq 3.5$	$x_2 = green$	$x_2 = blue$	$x_2 = red$	$x_3 = yes$	$x_4 \geq 17$	$x_4 \geq 21$

and the following interval variables:

b_{10}	b_{11}	b_{12}	b_{13}
$1.5 \leq x_1 < 2.5$	$1.5 \leq x_1 < 3.5$	$2.5 \leq x_1 < 3.5$	$17 \leq x_4 < 21$

(a) Original table. (b) Binarized table.

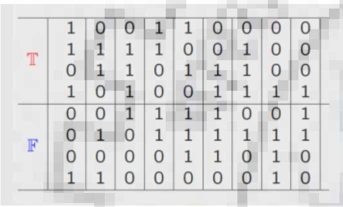
Figure 6 Interval Variable in Binarization

Chapter 4

4.Support Set Minimization

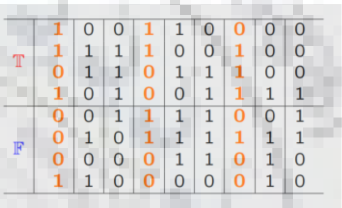
4.1 Introduction

Find a small (smallest, if possible) subset of the attributes which distinguishes the sets T and F. Such a subset is called a support set. (**Crama, Hammer and Ibaraki (1988)**



T	1	0	0	1	1	0	0	0	0
	1	1	1	1	0	0	1	0	0
	0	1	1	0	1	1	1	0	0
	1	0	1	0	0	1	1	1	1
F	0	0	1	1	1	1	0	0	1
	0	1	0	1	1	1	1	1	1
	0	0	0	0	1	1	0	1	0
	1	1	0	0	0	0	0	1	0

⟹



T	1	0	0	1	1	0	0	0	0
	1	1	1	1	0	0	1	0	0
	0	1	1	0	1	1	1	0	0
	1	0	1	0	0	1	1	1	1
F	0	0	1	1	1	1	0	0	1
	0	1	0	1	1	1	1	1	1
	0	0	0	0	1	1	0	1	0
	1	1	0	0	0	0	0	1	0

Our dataset is partitioned into two sets. Set of positive observations and the set of negative observation. It is basic property that the union of this two partition is an empty set means their wont be any observation that is positive and negative at the same time. This property needs to be preserved & binarization process preservers the above property.

The aim of this component is to eliminate the redudant variables as well as the varibles who doesn’t have any impact on the results. Goals to find and reduce most possible variables while preserving the basic property mentioned above.

4.2 Correlation

To find the redundant attributes, we can use correlation analysis. [3]

Correlation is defined as relationship or connection between two or more things. Correlation coefficient can be applied on numerical attributes.

If correlation coefficient of two attributes is close to +1 or -1 means both attributes are linearly dependent & if it is close to 0 then we can that the particular attribute is independent.

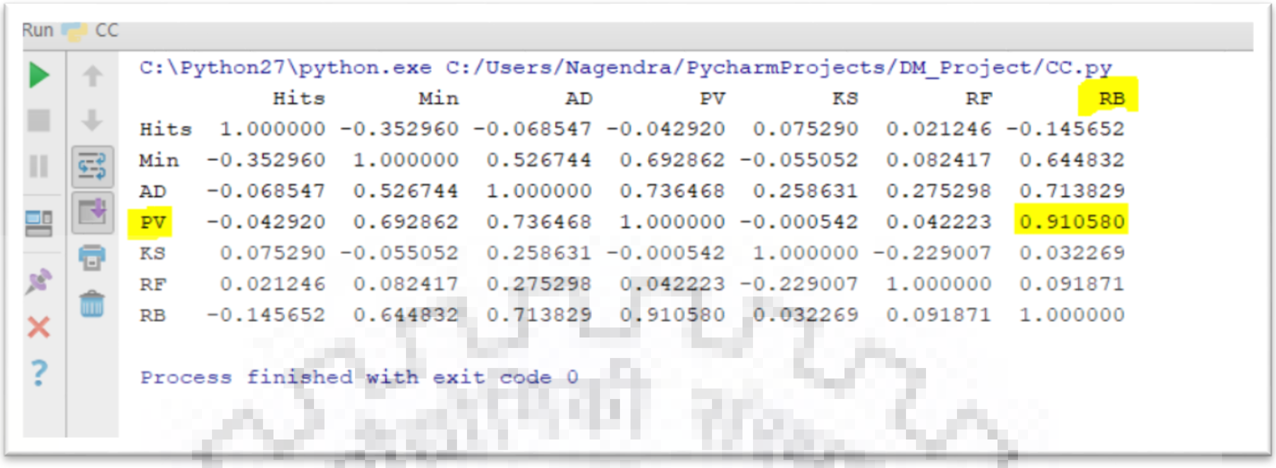
Correlation coefficient analysis can be applied on –

- Attribute to attribute
- Attribute to result

Attribute to attribute correlation – In this method, attribute to attribute correlation is calculated and if it is close to 1 or -1 then we can say that this two attribute are linearly dependent on each other. This is also called as redundant attribute. So we can go ahead with removing any of the two attribute.

In this method, we calculate correlation coefficient matrix. All the attributes whose correlation is greater than threshold will be considered as redundant and can be removed safely from dataset.

Attribute to Result correlation – In this method, we are looking for the attribute which is independent with result. If the attribute correlation with result is close to 0 we can say that it’s an independent attribute and can be removed from dataset as it doesn’t have an effect on outcome.



```
C:\Python27\python.exe C:/Users/Nagendra/PycharmProjects/DM_Project/CC.py
          Hits       Min        AD        PV        KS        RF        RB
Hits  1.000000 -0.352960 -0.068547 -0.042920  0.075290  0.021246 -0.145652
Min  -0.352960  1.000000  0.526744  0.692862 -0.055052  0.082417  0.644832
AD   -0.068547  0.526744  1.000000  0.736468  0.258631  0.275298  0.713829
PV   -0.042920  0.692862  0.736468  1.000000 -0.000542  0.042223  0.910580
KS    0.075290 -0.055052  0.258631 -0.000542  1.000000 -0.229007  0.032269
RF    0.021246  0.082417  0.275298  0.042223 -0.229007  1.000000  0.091871
RB   -0.145652  0.644832  0.713829  0.910580  0.032269  0.091871  1.000000

Process finished with exit code 0
```

Figure 7 Correlation Matrix

Above figure shows the correlation matrix for attribute to attribute. If we considered the threshold as 0.90 we can say that attribute RB can be redundant and can be removed safely from dataset.

4.3 Set Covering Model

A set covering model can be used to find smallest support set. In LAD, set cover problem is used to find the minimum number of attributes that can cover the entire observations. This model can be solved either exactly or heuristically.

For every pair of false example X and true example Y, express that at least one of the attributes differentiating X from Y must be chosen:

$$\forall X \text{ € } F , Y \text{ € } T , \sum_{i:Xi \neq Yi} ai \geq 1$$

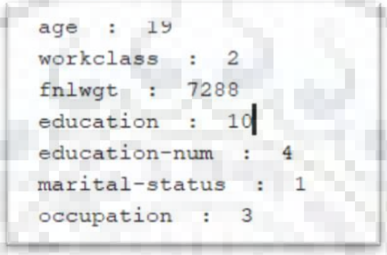
Where α_i is a variable associated with each attribute A_i

Minimize $\sum_i \alpha_i$

4.4 Other ways

Apart from mathematical methods, we can use the threshold values we got from binarization process.

In binarization process, we get the threshold matrix for each attribute. If the size of particular threshold matrix is large compared to other attribute threshold matrix, then we can say that the particular attribute is not affecting on result.



```
age  :  19
workclass  :  2
fnlwgt  :  7288
education  :  10
education-num  :  4
marital-status  :  1
occupation  :  3
```

Looking into above stats, we can say that attribute fnlwgt can be removed as it’s flipping very frequently between classes irrespective of attribute value.

Chapter 5

5.Pattern Generation

It is the process of applying different methods to find interesting and previously-unknown patterns within said set of data. A positive pattern is simply a sub cube of the unit cube which intersects Ω+ 1s and is disjoint from Ω−1s. A negative pattern is simply a sub cube of the unit cube which intersects Ω- 1s and is disjoint from Ω+1s.

The main concept in LAD is that of pattern which plays critical role in detection of subclasses, selection of the feature, classification and other problems.

The basic and effective may for generating the pattern is the use of combinatorial enumeration techniques. Pattern generation will result to many patterns it is important for pattern generation procedure no to miss the best patterns.

Pattern generation technique can be performed in two ways-

- Top-Down approach
- Bottom-Up approach

5.1 Top-Down approach-

Top down approach associates the characteristic term of every positive observation to generate a pattern. But even after removing several literals we can cover the positive observations and the resulting term will remain pattern. This approach will remove literals one by one, systematically, until we get the prime pattern.

5.2 Bottom-Up approach

This process start with terms of degree one, if those terms covered some of the positive observations without having any negative observation in them then it is a pattern. If it has some negative observation in them then the literals are added to term one by one till the generation of pattern.

Decision Tree Classification

A decision tree classifier repeated divides the observations into sub parts by identifying the characteristics. It is a tree like graph. Decision tree is used to identify the strategy that help particular to reach the goal means it helps to get the patterns that lead to particular outcome.

We will terminate the division into sub tree's if it has divided into classes that are pure or some criteria of classifier attributes are met.

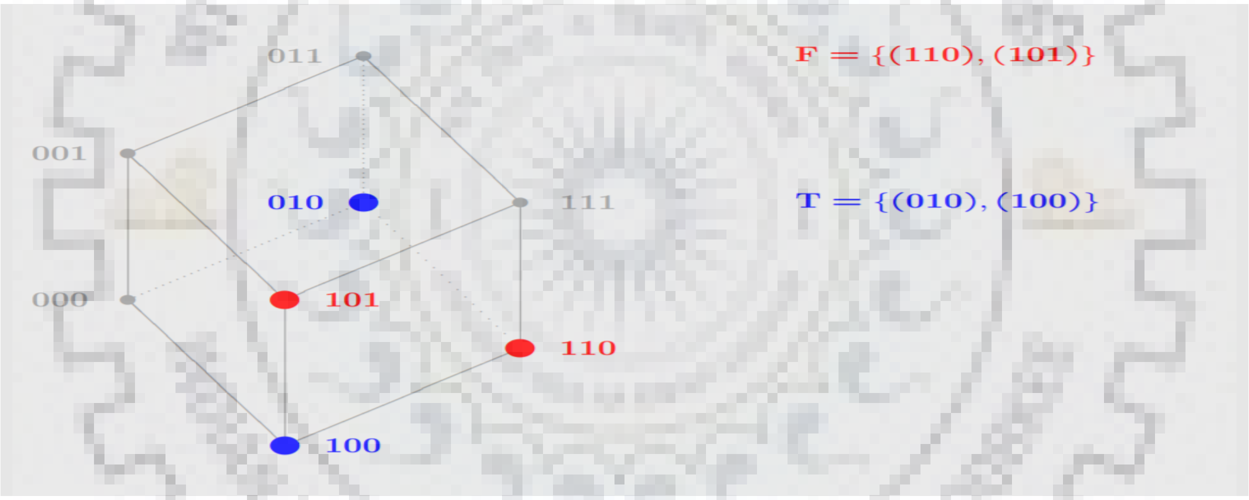


Figure 8 Decision Tree Step by Step-1

Given the dataset, we have set of positive observations and negative observations.

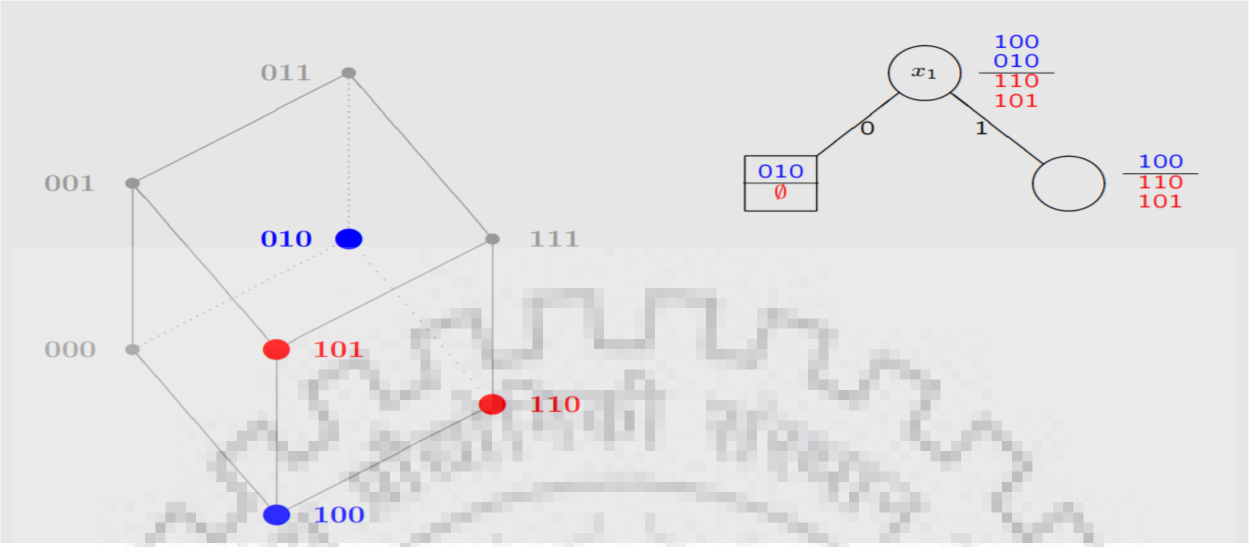


Figure 9 Decision Tree Step by Step-2

After applying the binary x1 variable in decision tree, the left side consist all the positive observations so we can stop the exploring left side.

On the right hand side, we have both positive as well as negative observations so we have to divide right node into sub tree.

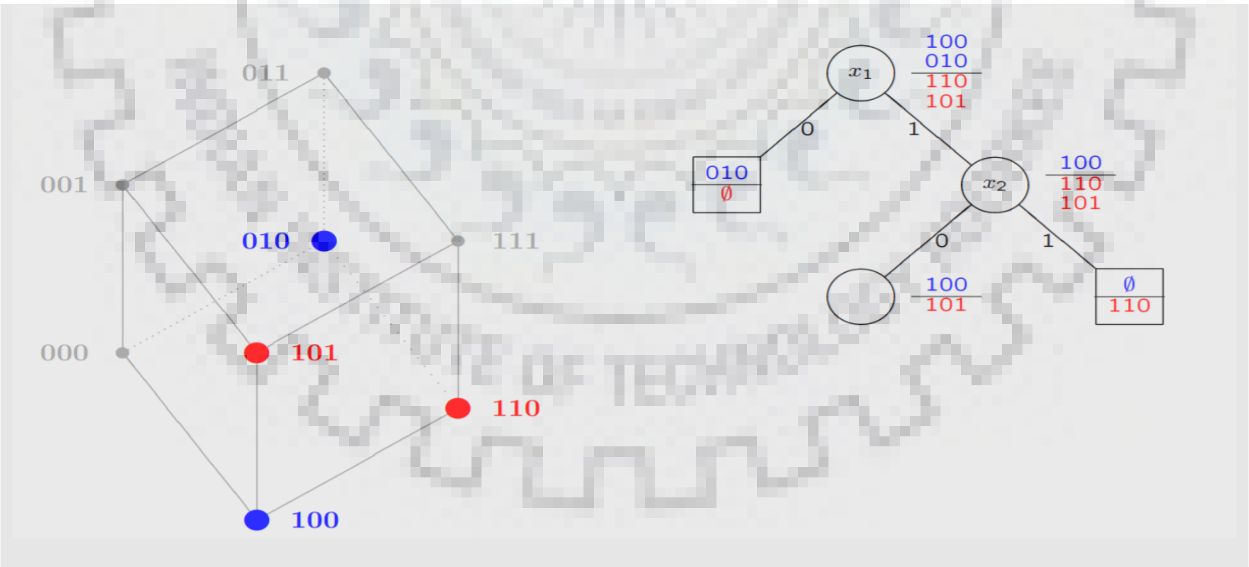


Figure 10 Decision Tree Step by Step-3

After applying 2nd level attribute, we still unable to distinguish some of the observations. We will go in depth till each leaf node in tree belong to either positive or negative class.

Once the tree has been generated, we can derive the function for each class.

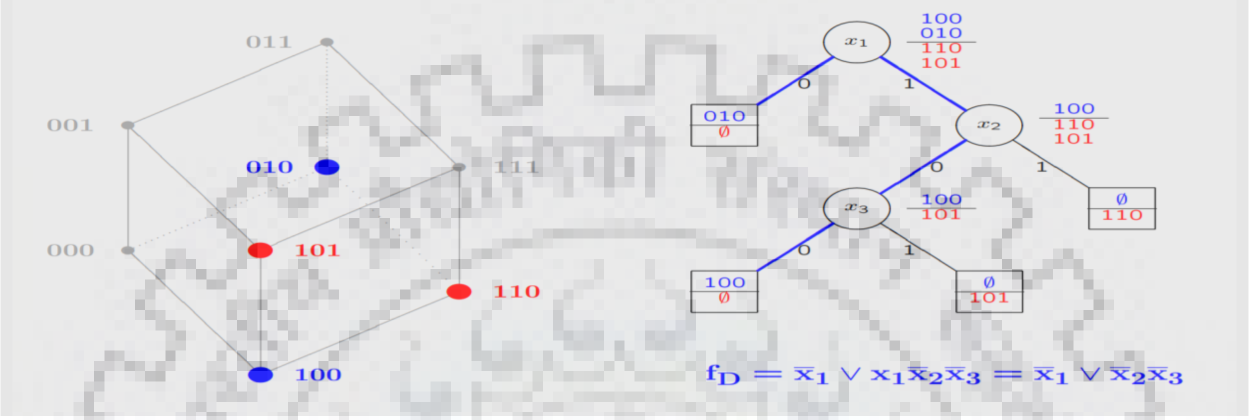


Figure 11 Shows the positive function for the given dataset

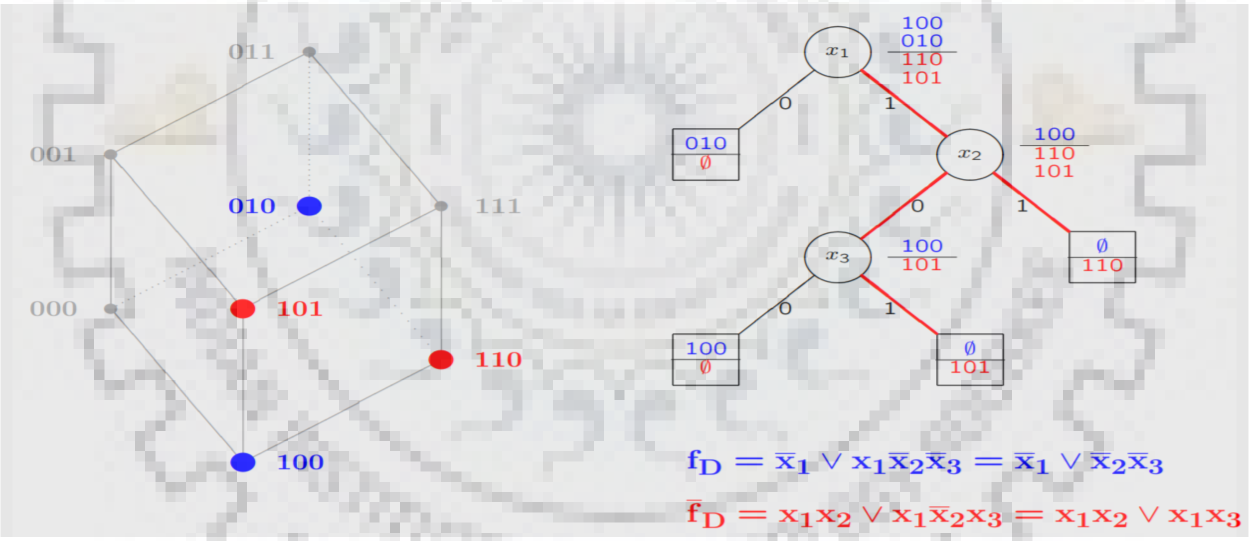

011
001
010
111
000
101
110
100
x_1
100
010
110
101
0
1
010
∅
x_2
100
110
101
0
1
x_3
100
101
∅
110
0
1
100
∅
∅
101
$f_D = \bar{x}_1 \vee x_1\bar{x}_2\bar{x}_3 = \bar{x}_1 \vee \bar{x}_2\bar{x}_3$
$f_D = x_1x_2 \vee x_1\bar{x}_2x_3 = x_1x_2 \vee x_1x_3$

Figure 12 Shows the negative function for the given dataset.

Using this positive and negative functions, we can find the result for the remaining observations.

5.3 Feature Sets

This technique works on grouping of different features of dataset and then check whether this group of feature, together, is sufficient to classify all the rows of dataset into two classes.

The quality of patterns can be improved by considering coverage of each pattern and having a threshold cut-off. Coverage is the number of times it occurs in that particular class.

Different sets can be generated using bit manipulation method and for each set of features our model will find the coverage of that set.

Chapter 6

6.Hardware Implementation

The extension to LAD is to develop hardware circuit that will work without throughout internet connectivity and with help of minimum power supply.

6.1 Hardware Components

- Arduino Uno
- LCD 16*2 display
- Keypad 4*4
- breadboard
- Jumper wires
- Power Supply

6.2 Hardware Features

- Microcontroller: ATmega328
- Operating Voltage: 5V
- Input Voltage (recommended): 7-12V
- Input Voltage (limits): 6-20V
- Digital I/O Pins: 14 (of which 6 provide PWM output)
- Analog Input Pins: 6
- Flash Memory: 32 KB of which 0.5 KB used by bootloader
- SRAM: 2 KB (ATmega328)
- EEPROM: 1 KB (ATmega328)
- Clock Speed: 16 MHz

6.3 Circuit Diagram

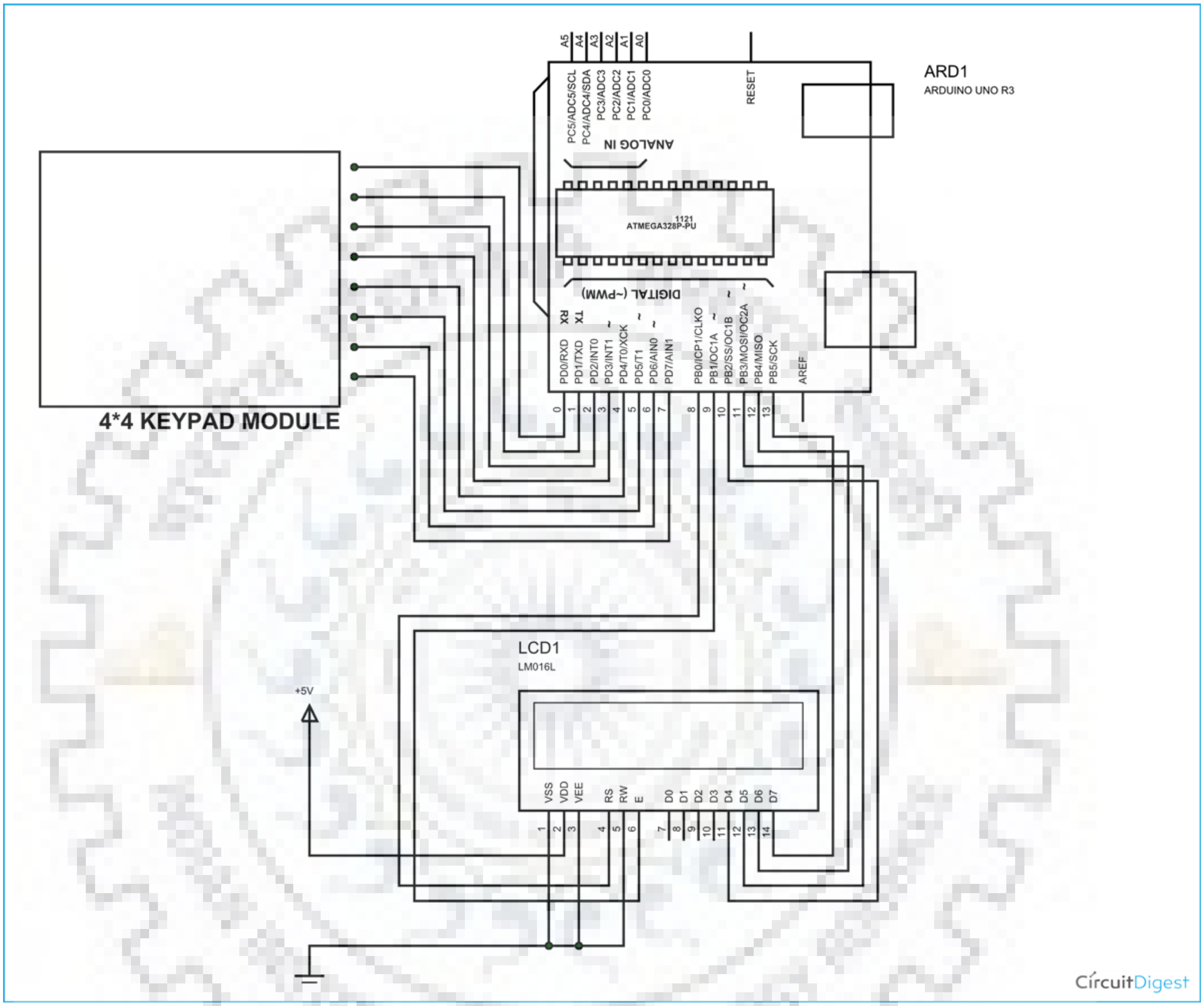
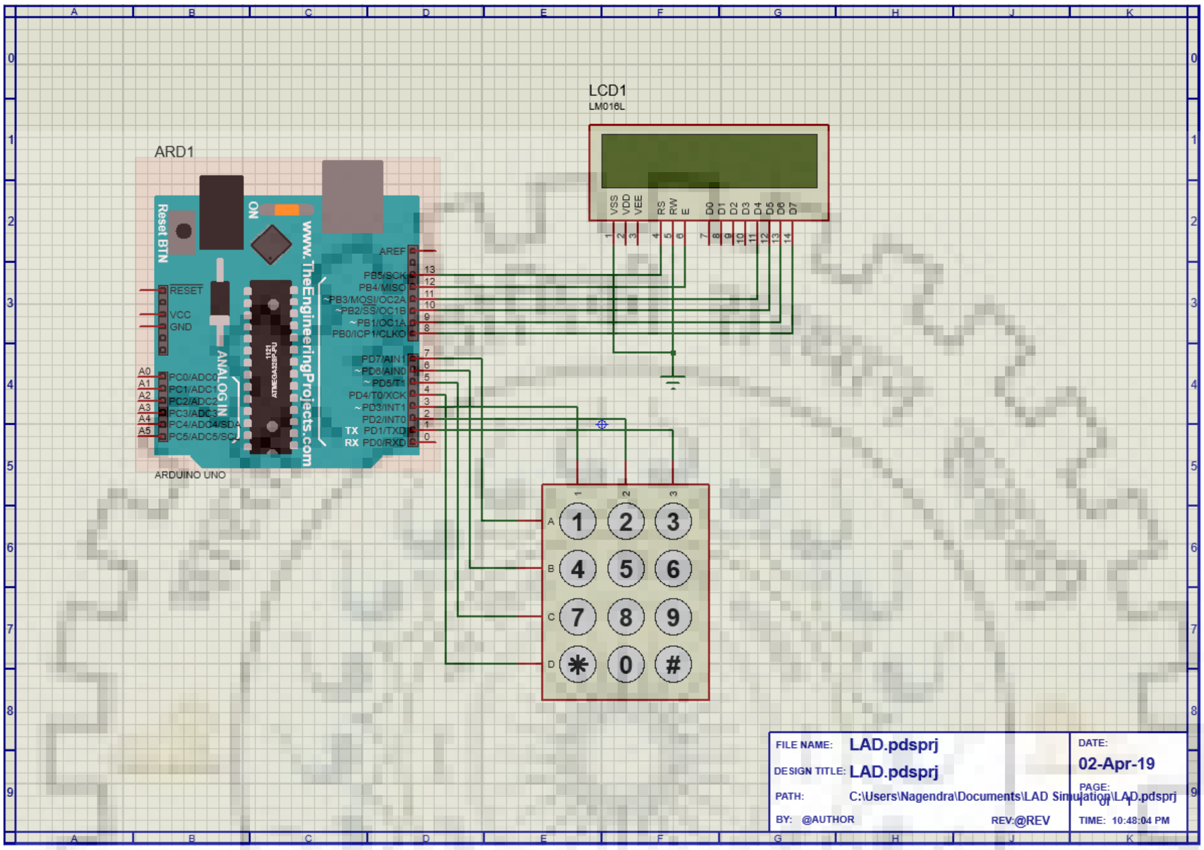


Figure 13 Hardware Circuit Diagram

6.4 Circuit Diagram (Proteus Simulator)



FILE NAME: LAD.pdsprj
DESIGN TITLE: LAD.pdsprj
PATH: C:\Users\Nagendra\Documents\LAD Simulation\LAD.pdsprj
BY: @AUTHOR
REV:@REV
DATE: 02-Apr-19
PAGE: 1 of 1
TIME: 10:48:04 PM

Figure 14 Proteus Simulator

6.5 Actual Circuit:

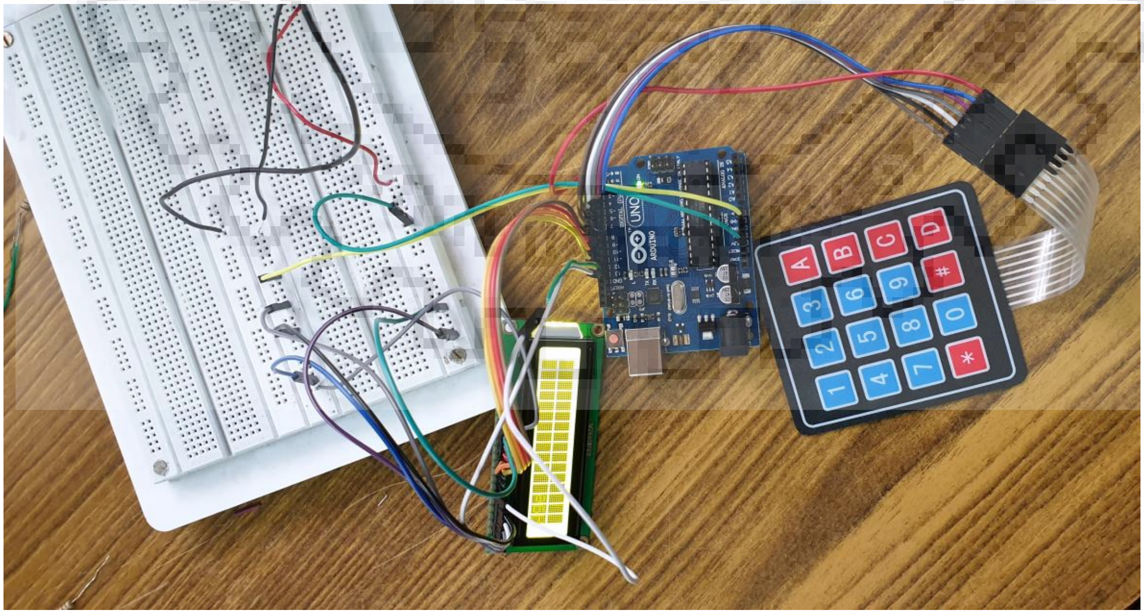


Figure 15 Actual Circuit

In hardware implementation, we connected 4*4 keypad to digital I/O pins which will used to punch the blood parameters into the system. LCD screen is also connected to digital I/O pins of Arduino which will used to provide interactive display.

This circuit will ask to provide the values for 9 parameters and will provide the result either affected or not affected. This system is designed to classify into two classes either positive or negative.

6.6 Steps

- Start the device (Enable the power supply)
- Punch the values for parameters displayed on the screen using keypad. Press # to enter & to move to next parameters.
- Result will be displayed
- Press reset button to check result for another patient.

Chapter 7

7.Arduino Code

Arduino code is written in C++ with an addition of special methods and functions. The Arduino Integrated Development Environment (IDE) is the main text editing program used for Arduino programming. It is where you'll be typing up your code before uploading it to the board you want to program. Arduino code is referred to as sketches.

7.1 Libraries in Arduino

In Arduino, much like other leading programming platforms, there are built-in libraries that provide basic functionality. In addition, it's possible to import other libraries and expand the Arduino board capabilities and features. These libraries are roughly divided into libraries that interact with a specific component or those that implement new functions.

7.2 Libraries Used

- **Keypad.h** – It was created to promote Hardware Abstraction. It improves readability of the code by hiding the pinMode and digitalRead calls for the user.
  Functions:
  - char waitForKey() - This function will wait forever until someone presses a key.
  - Char getKey() - Returns the key that is pressed, if any. This function is non-blocking.
  - void begin(makeKeymap(userKeymap)) - Initializes the internal keymap to be equal to userKeymap

- **LiquidCrystal.h** -This library allows an Arduino board to control LiquidCrystal displays (LCDs).
  Functions:
  - clear() - Clears the LCD screen and positions the cursor in the upper-left corner.

- begin() - Initializes the interface to the LCD screen, and specifies the dimensions (width and height) of the display.
- setCursor() - Position the LCD cursor; that is, set the location at which subsequent text written to the LCD will be displayed.
- print() - Prints text to the LCD.

7.3 Arduino Functions

Setup()

Every Arduino sketch must have a setup function. This function defines the initial state of the arduino upon boot and runs only once.

Here we'll define the following:

- Pin functionality using the pinMode function
- Initial state of pins
- Initialize classes
- Initialize variables
- Code logic

Loop()

The loop function is also a must for every Arduino sketch and executes once setup() is complete. It is the main function and as its name hints, it runs in a loop over and over again. The loop describes the main logic of your circuit.

7.4 Steps to burn a code to hardware

1. First, compile the code using Arduino IDE.
2. Once the compilation done successfully, Choose the serial port your Arduino is currently connected to.
3. Select the board on which you are going to upload the code. (In this case it's Arduino Uno).
4. Upload the sketch by clicking upload button on Arduino IDE. Your Arduino LEDS will flicker once the data is being transferred. Once complete, you'll be greeted with a completion message that tells you Arduino has finished uploading.

Chapter 8

8.Proposed Work

8.1 System Flowchart

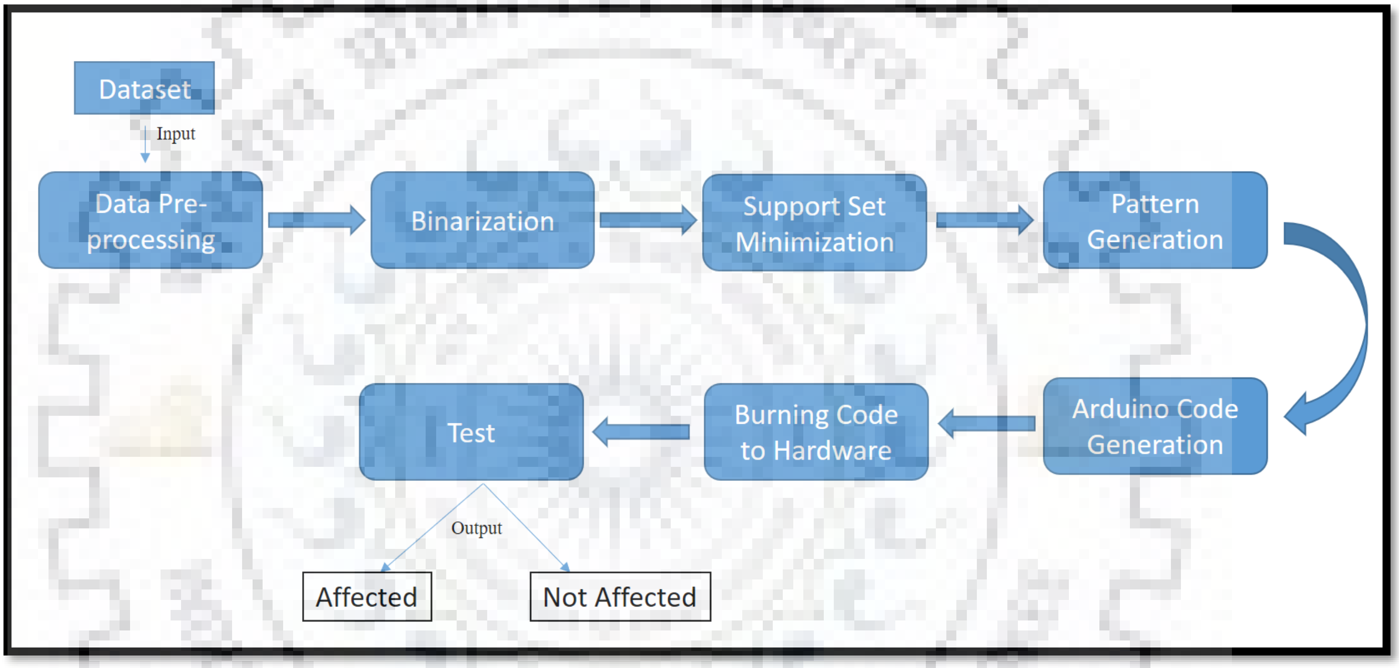


Figure 16 System Flowchart

8.2 Dataset

We are working on UCI dataset which consist the data of patients and classified into two class such as 'Breast Cancer Affected' and 'Not Affected'.

Dataset consist around 1k patients which are almost equally distributed into two classes.

Dataset consist of 9 features out of which two features are 'Age' and 'BMI' while other 7 features are blood factors like 'Glucose', 'Insulin', etc.

8.3 Modules

Step 1: Data Preprocessing

Input dataset will be passed to data preprocessing stage where data is cleansed through processes such as filling in missing values, smoothing the noisy data, or resolving the inconsistencies in the data.

Step 2: Binarization

This stage converts the dataset into binary dataset in which each feature can have either 0 or 1 as their values. Binarization process may introduce additional attributes as described in Binarization chapter in this report.

Step 3: Support Set Minimization

As binarization process introduce many new features in dataset, support set minimization is required to reduce the number of features. Techniques such as number of flips, Correlation Coefficient and set cover problem is used to reduce the less correlated features.

Step 4: Pattern Generation

Decision tree technique is used to extract the pattern for each class.

Step 5: Arduino Code Generation

Arduino microcontroller works on the C++ code, so patterns generated using decision tree technique will be converted to C++ code in this stage.

Step 6: Burning code to hardware

This phase involves burning of code to Arduino Uno microcontroller using Arduino IDE.

Step 7: Test

This device works on 5V power supply. Once powered on, it will ask you to provide the input for 9 features and according to rule result will be displayed on LCD module.

Chapter 9

9.Output/Results

LAD classifier consist of 3 main components such as binarization, support set minimization & pattern generation.

With use of decision tree technique for generation of patterns we are able to achieve the accuracy of 86% & with the use of grouping features technique we are able to achieve the accuracy of 83%

Method	Accuracy
LAD (Decision Tree for pattern generation)	86%
LAD (Grouping of features for pattern generation)	83%

We have also worked on extension of LAD, where we focused on the development of independent hardware that will work on less power supply (5v) and constant internet connectivity is no longer required.

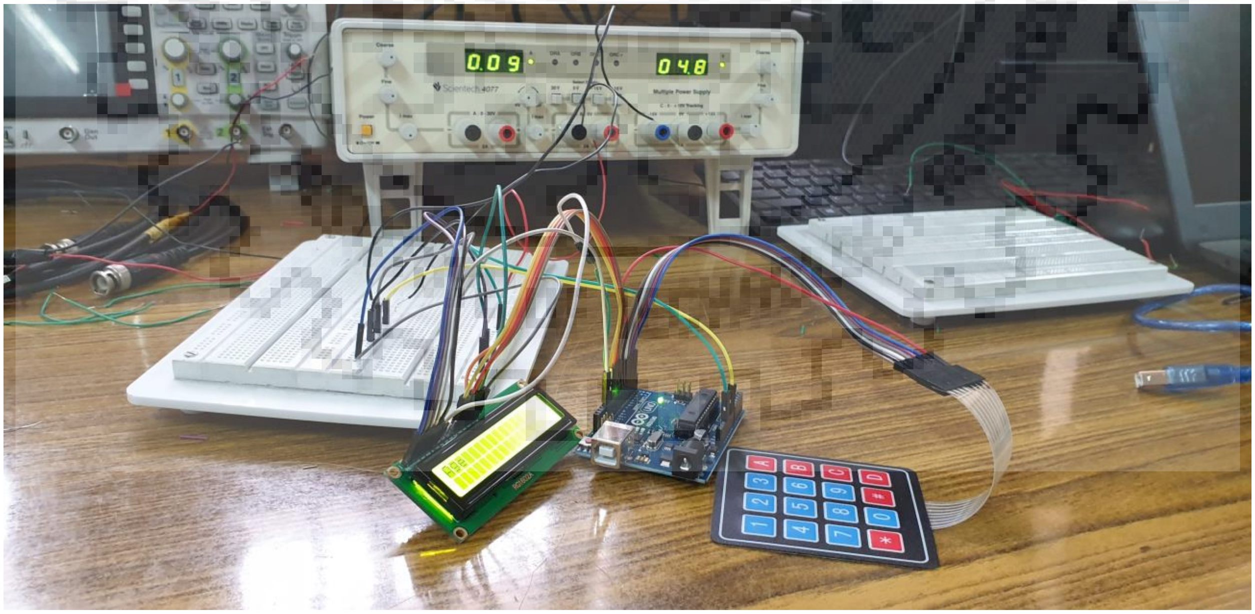


Figure 17 Device Image-1

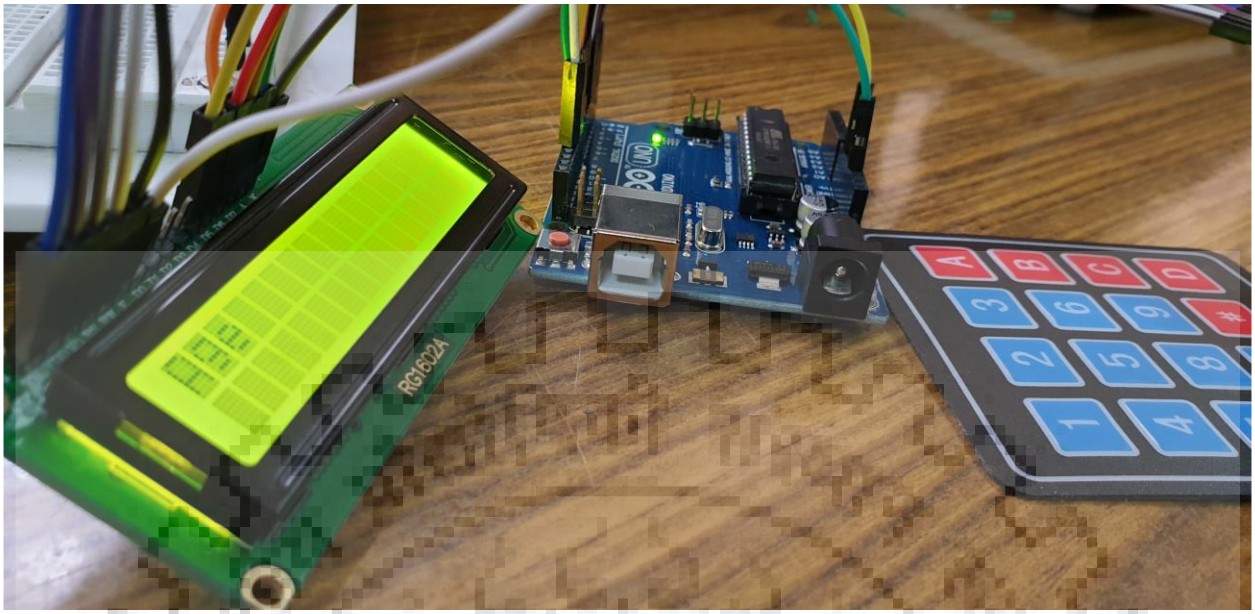


Figure 18 Device Image-2

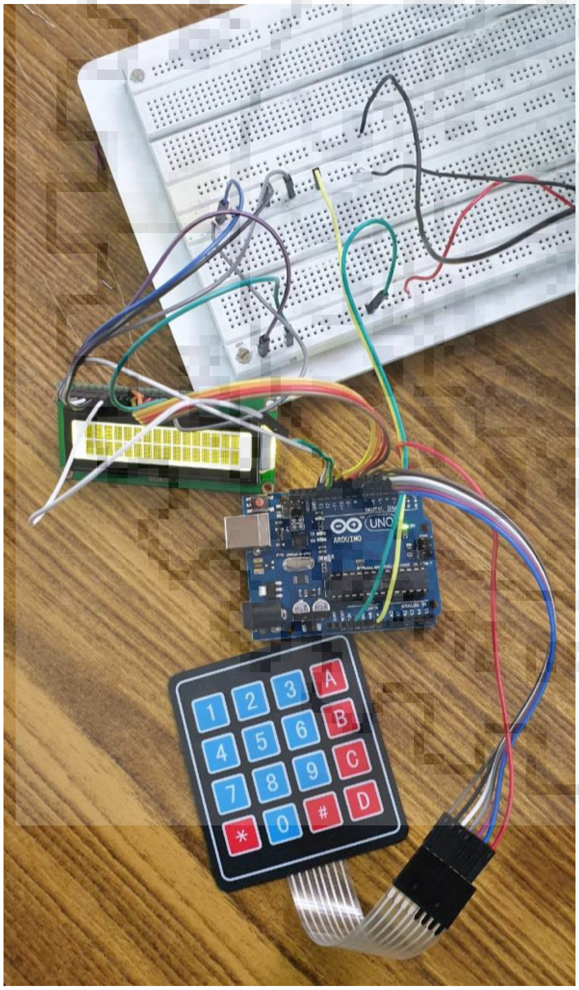


Figure 19 Device Image-3

Chapter 10

10.Conclusion

One common aspect of many problems appearing from security to health care is the need of discovering hidden patterns in the past observations. These patterns establish a causal relationship among observations and their class labels. The LAD is useful for such problems where we need to understand these causal relationships to predict class labels of future observations.

With the help of different techniques to perform binarization, support set minimization, pattern generation, we are able to achieve good accuracy. The additional advantage that we hold here is we have a final dataset which will be in Binarized format. So all the variables in the pattern can hold either 0 or 1 that makes it very suitable to convert into electrical circuit. With the help of such circuit’s, we are no longer required an intensive computational power & 24/7 connectivity to internet.

References

[1] Logical analysis of data—An overview: From combinatorial optimization to medical applications Peter L. Hammer · Tiberius O. Bonates, August 2006

[2] An Implementation of Logical Analysis of Data Endre Boros, Peter L. Hammer, Member, IEEE Computer Society, March 2000,IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 12

[3] Logical analysis of data – the vision of Peter L. Hammer, Gabriela Alexe · Sorin Alexe, 2007

[4] Boolean Functions for Classification:Logical Analysis of Data, Yves Crama ,University of Liege - Belgium and Endre Boros - Rutgers University, USA

[5] Hammer, P. L. and Bonates, T. O. (2006). Logical analysis of data—an overview: from combinatorial optimization to medical applications. Annals of Operations Research,148(1):203–225.

[6] E. Boros, Y. Crama, P.L. Hammer, T. Ibaraki, A. Kogan and K. Makino, Logical Analysis of Data: Classification with justification, Annals of Operations Research 188 (2011) 33-61.

[7] Boros, E., Hammer, P. L., Ibaraki, T., and Kogan, A. (1997). Logical analysis of numerical data. Mathematical Programming, 79(1-3):163–190.

[8] Mohamed Hamdi-Cherif, "Logical Analysis of Data (LAD) - Usage and Experimentation" In proceedings of the International Conference on Artificial Intelligence, 2017, pp. 266-272.

[9] Breast Cancer Data Set, UCI Machine Learning Repository.