ACOUSTIC ENVIRONMENT DETECTION

by

Name	Roll No.	Registration No:
Avranil Roy	*11700314027*	141170110209 of 2014-2015
Trinanjana Chaudhuri	*11700314118*	141170110300 of 2014-2015
Ankita Das	*11700314020*	141170110202 of 2014-2015
Abhishek Sarkar	*11700314004*	141170110186 of 2014-2015

A comprehensive project report has been submitted in partial fulfillment of the requirements for the degree of

Bachelor of Technology
in
ELECTRONICS & COMMUNICATION ENGINEERING

Under the supervision of

Mr. Sujoy Mondal

Assistant Professor

श्रमम् बिना न किमपि साध्यम्

Department of Electronics & Communication Engineering
RCC INSTITUTE OF INFORMATION TECHNOLOGY
Affiliated to Maulana Abul Kalam Azad University of Technology, WestBengal
CANAL SOUTH ROAD, BELIAGHATA, KOLKATA – 700015

May,2018

CERTIFICATE OF APPROVAL

This is to certify that the project titled "**ACOUSTIC ENVIRONMENT DETECTION**" carried out by

Name	Roll No.	Registration No:
Avranil Roy	*11700314027*	141170110209 of 2014-2015
Trinanjana Chaudhuri	*11700314118*	141170110300 of 2014-2015
Ankita Das	*11700314020*	141170110202 of 2014-2015
Abhishek Sarkar	*11700314004*	141170110186 of 2014-2015

for the partial fulfillment of the requirements for B.Tech degree in **Electronics and Communication Engineering** from **Maulana Abul Kalam Azad University of Technology, West Bengal** is absolutely based on his own work under the supervision of Mr. . The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Optional in case of External Supervisor

....

Designation and Department
Institute

....

Mr. Sujoy Mondal
Asst. Professor , Dept. of ECE
RCC Institute of Information Technology

....

Dr. Abhishek Basu
Head of the Department (ECE)
RCC Institute of Information Technology

DECLARATION

"We Do hereby declare that this submission is our own work conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute and that, to the best of our knowledge and belief, it contains no material previously written by another neither person nor material (data, theoretical analysis, figures, and text) which has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgement has been made in the text."

....

ABHISHEK SARKAR

Registration No: 141170110186 of 2014-2015

Roll No: 11700314004

....

ANKITA DAS

Registration No: 141170110202 of 2014-2015

Roll No: 11700314020

....

TRINANJANA CHAUDHURI

Registration No: 141170110300 of 2014-2015

Roll No: 11700314118

....

AVRANIL ROY

Registration No: 141170110209 of 2014-2015

Roll No: 11700314027

Date: 10.09.2018

Place: KOLKATA

CERTIFICATE of ACCEPTANCE

This is to certify that the project titled "**ACOUSTIC ENVIRONMENT DETECTION**" carried out by

Name	Roll No.	Registration No:
ABHISHEK SARKAR	*11700314004*	141170110186 of 2014-2015
TRINANJANA CHAUDHURI	*11700314118*	141170110300 of 2014-2015
ANKITA DAS	*11700314020*	141170110202 of 2014-2015
AVRANIL ROY	*11700314027*	141170110209 of 2014-2015

is hereby recommended to be accepted for the partial fulfillment of the requirements for B.Tech degree in **Electronics and Communication Engineering** from **Maulana Abul Kalam Azad University of Technology, West Bengal**

Name of the Examiner Signature with Date

1. ……………………………………………………………………

2.…………………………………..……………………………..

3.……………………………………………………………………

4. ……………………………………..…………………………

ABSTRACT

Now a days with the unprecedented growth of science, we can recognize the environment someone is belonging to without the physical presence, even without the image of the surroundings. It is being possible with the help of sound.

That is why Acoustic Environment Detection is going to be a great matter of concern in case of determination of an environment by means of surrounding sounds or noise.

Here in this project we have the problem to detect the environment of the test sound such that we have to determine in which environment the sound has been captured.

We have collected twelve sample sounds of twelve different environments and determined so by matching the test sound with the previously stored ones. We have firstly used Fast Fourier Transform to divide the test signal into several small parts so that the matching can be done easily. Then using Multilayer Perceptron Concept we did the matching by creating models for each type of sound. Then with the help of Confusion Matrix we have determined whether we are getting the desired output or not.

Thus doing so we have determined the environment using sound. With the use of several parameters we have calculated the accuracy, hit rate, misdetection and false alarms of the detection. From the Receiver Operating Characteristics curve we have determined these.

With the help of this project without consuming a huge data the environment can be detected easily by using the surrounding sounds. It has a great future prospect in case of criminal activity detection and with help of this project we get to know sometimes noise also can be a very useful tool for detection.

ACKNOWLEDGEMENT

The success and final outcome of this project i.e **"ACOUSTIC ENVIRONMENT DETECTION**" required a lot of guidance and assistance from many people and I am extremely privileged to have got this all along the completion of our project. All that we have done is only due to such supervision and assistance and I would not forget to thank them.

We respect and heartily thank **Mr. Sujoy Mondal** , **Asst.Professor, ECE, RCCIIT** for providing us an opportunity to do the project work in college premises and giving us all support and guidance which made us complete the project duly. We are extremely thankful to him for providing such a nice support and guidance, although he had busy schedule managing the corporate affairs.

We owe our deep gratitude to our project guide, who took keen interest on our project work and guided us all along, till the completion of our project work by providing all the necessary information for developing a good system.

We are thankful to and fortunate enough to get constant encouragement, support and guidance from all our friends, family who helped us in successfully completing our project work. Also, we would like to extend our sincere esteems to all staff in laboratory for their timely support.

CONTENTS

LIST OF ABBREVIATIONS

MLP	**Multi Layer Perceptron**
FA	**False Alarm**
MD	**Mis Detection**
TP	**True Positive**
TN	**True Negative**
FP	**False Positive**
FN	**False Negative**
Acc	**Accuracy**
ROC	**Receiver_Operating Characteristic**

LIST OF FIGURES

LIST OF TABLES

Introduction

1.1. Acoustic Environment Detection

1.1.1. Acoustic Environment

The acoustic environment is an important aspect of quality in the experience of the natural and cultural environment.
The emission of sound waves from natural and manmade sources, their propagation through the atmosphere, and their detection through auditory or other means at a noise sensitive receptor in the ambient environment, characterize sound quality.

Acoustic Environment is a valued environmental component (VEC) for the environmental impact assessment (EIA). There is also potential for Project activities to generate vibration in the immediate vicinity of the Project that, if excessive, could be objectionable or cause property damage—thus for the purpose of this VEC, the Acoustic Environment also includes Project related vibration that could affect nearby human receptors.

3.3.3. Principles of noise

Sound travels at around 344 m/s at 20Âº and is a series of mechanical vibrations in a solid, liquid or gas medium Noise is defined as any sound that a listener does not wish to hear.
The concept of noise is difficult as each individual has a different concept of unwanted sound and the acceptable level of that sound.
Noise levels are generally measured as sound pressure levels and reported as a decibel level. The decibel (dB) scale is logarithmic.
All noise that can be measured in an area or neighborhood is the ambient noise of that area.
Ambient noise includes all noise from traffic, people, animals and machinery.
A variety of noise measurements are made depending on the circumstances:
Ambient noise, background noise, statistical levels and maximum levels.

The level of a noise source is sometimes compared with the background noise level.
The background noise with respect to a particular noise source is the ambient noise that is exceeded 90% of the time when noise from the source is excluded.
A general rule is that if any source of noise exceeds the background by 5dB (A) then a nuisance may be created.

1.1.3. The acoustic environment affects our experience

Noise – unwanted sound – is becoming an increasingly widespread problem.
We seek quiet in our leisure time, away from our noisy everyday environment.
Walks in the forests, picking berries and mushrooms, hunting and fishing, skiing, swimming and boating – silence is an important part of the experience.
Many sounds "form part" of the experience.
Birdsong, the rustle of leaves, skis cutting through the crust of snow and the many different sounds of water enrich our stay in the countryside.
The acoustic environment can affect us without us being aware of it.
We can become stressed and suffer high blood pressure because of noise, without the noise bothering us.

1.1.4. Silence –Also an acoustic environment

Absolute silence is not always the best acoustic environment.
In an acoustic laboratory we may experience a very low level of sound, but then the body's own sounds become almost terrifying.
We hear our heart beat and our blood coursing through our arteries and veins.
The quietest situation we can experience in nature is a winter's day with no wind, far away from buildings and roads.
There the level of sound can approach the situation in a laboratory.
But we like to hear nature's own sounds, the babbling of streams, the rustling of leaves or the dripping of melting snow.
Nor is silence the only thing we look for in nature.
If an area is to be attractive, it has to be able to offer something more than just silence. This "something more" is often associated with sound.
The ski lift, the snow guns, the skiers mean that we hear that we are on a ski slope and can enhance the experience for the downhill skier.
When we go walking in the mountains, we like to pitch camp alongside water, so that we can hear the quiet murmur of the mountain stream or the splashing of the waves on the shore of the lake.

1.1.5. Our Purpose

The emission of sound waves from natural and manmade sources, their propagation through the atmosphere, and their detection through auditory or other means at a noise sensitive receptor in the ambient environment, characterize sound quality.
Here we have selected twelve types of circumstances and classified the test signal with help of those.
The process is described below via the block diagram-

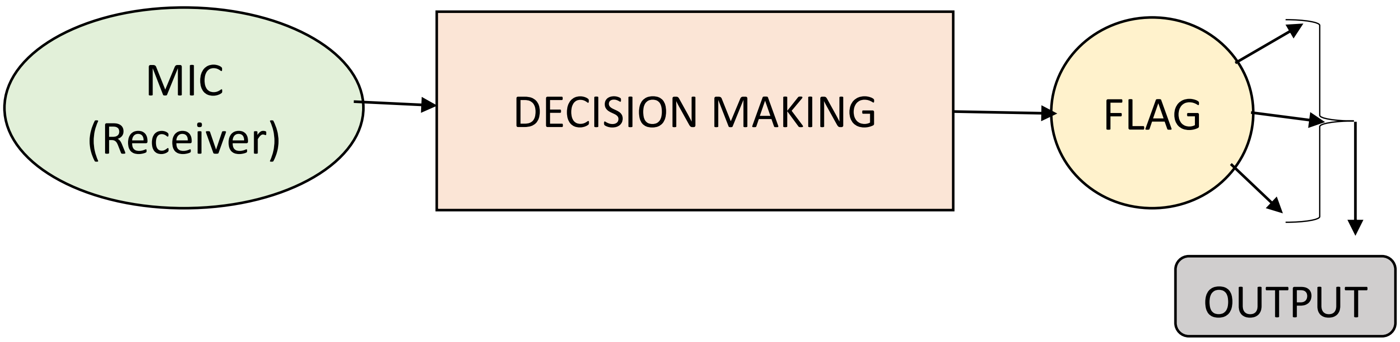


FIG : 1.1

The mic is used as the receiver here. The hardware part is the computer.
MATLAB is used as the software which generates a flag.
Based upon the flag the desired output occurs.

MULTILAYER PERCEPTRON

2.1. MULTILAYER PERCEPTRON

A multilayer perceptron (MLP) is a class of feed forward artificial neural network.
An MLP consists of at least three layers of nodes.
Except for the input nodes, each node is a neuron that uses a nonlinear activation function.
MLP utilizes a supervised learning technique called back propagation for training.
Its multiple layers and non-linear activation distinguish MLP from a linear perceptron.
It can distinguish data that is not linearly separable.

The perceptron, that neural network whose name evokes how the future looked from the perspective of the 1950s, is a simple algorithm intended to perform binary classification; i.e. it predicts whether input belongs to a certain category of interest or not.

Multilayer perceptron is sometimes colloquially referred to as “vanilla” neural networks, especially when they have a single hidden layer.

Machine learning and Data mining

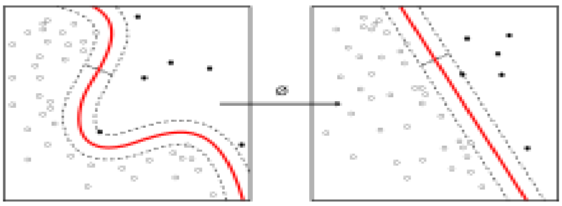


FIG : 2.1(A)

The perceptron holds a special place in the history of neural networks and artificial intelligence, because the initial hype about its performance led to a rebuttal by Minsky and Papert.

The wider spread backlash that cast a pall on neural network research for decades, a neural net winter that wholly thawed only with Geoff Hinton's research in the 2000s, the results of which have since swept the machine-learning community.

Frank Rosenblatt, godfather of the perceptron, popularized it as a device rather than an algorithm.

The perceptron first entered the world as hardware.

Rosenblatt, a psychologist who studied and later lectured at Cornell University, received funding from the U.S. Office of Naval Research to build a machine that could learn. His machine, the Mark I perceptron, looked like this.

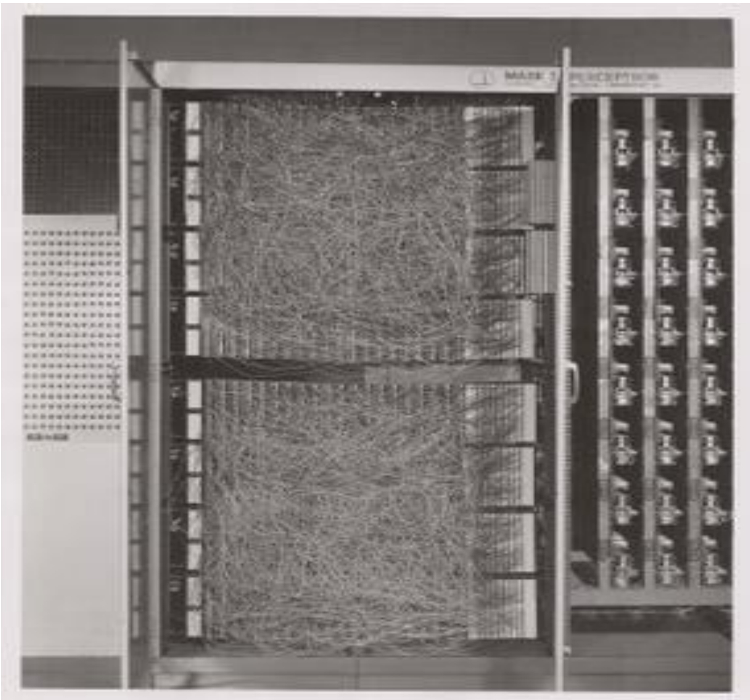


FIG : 2.1(B)

Subsequent work with multilayer perceptron has shown that they are capable of approximating an XOR operator as well as many other non-linear functions.

Just as Rosenblatt based the perceptron on a McCulloch-Pitts neuron, conceived in 1943, so too, perceptron themselves are building blocks that only prove to be useful in such larger functions as multilayer perceptron**.**

2.1.1. Activation function

If a multilayer perceptron has a linear activation function in all neurons, that is, a linear function that maps the weighted inputs to the output of each neuron, then linear algebra shows that any number of layers can be reduced to a two-layer input-output model.

In MLPs some neurons use a nonlinear activation function that was developed to model the frequency of action potentials, or firing, of biological neurons.

The two common activation functions are both sigmoid, and are described by-

$$y(v_i) = \tanh(v_i) \text{ and } y(v_i) = (1 + e^{-v_i})^{-1}.$$

The first is a hyperbolic tangent that ranges from -1 to 1, while the other is the logistic function, which is similar in shape but ranges from 0 to 1.

Here is the output of the nth node (neuron) and is the weighted sum of the input connections.

Alternative activation functions have been proposed, including the rectifier and soft plus functions.

More specialized activation functions include radial basis functions (used in radial basis networks, another class of supervised neural network models).

2.1.2. Layers

The MLP consists of three or more layers (an input and an output layer with one or more hidden layers) of nonlinearly-activating nodes making it a deep neural network.

Since MLPs are fully connected, each node in one layer connects with a certain weight to every node in the following layer.

2.1.3. Learning

Learning occurs in the perceptron by changing connection weights after each piece of data is processed, based on the amount of error in the output compared to the expected result.

This is an example of supervised learning, and is carried out through back propagation, a generalization of the least mean squares algorithm in the linear perceptron.

We represent the error in output node j in the nth data point (training example) by

$$e_j(n) = d_j(n) - y_j(n),$$

Where d is the target value and y is the value produced by the perceptron.

The node weights are adjusted based on corrections that minimize the error in the entire output, given by

$$\mathcal{E}(n) = \frac{1}{2} \sum_j e_j^2(n).$$

Using gradient descent, the change in each weight is

$$\Delta w_{ji}(n) = -\eta \frac{\partial \mathcal{E}(n)}{\partial v_j(n)} y_i(n)$$

Where y_i the output of the previous neuron and η is the learning rate, which is selected to ensure that the weights quickly converge to a response, without oscillations.

The derivative to be calculated depends on the induced local field v_j, which itself varies. It is easy to prove that for an output node this derivative can be simplified to

$$-\frac{\partial \mathcal{E}(n)}{\partial v_j(n)} = e_j(n)\phi'(v_j(n))$$

Where ϕ' is the derivative of the activation function described above, which itself does not vary.

The analysis is more difficult for the change in weights to a hidden node, but it can be shown that the relevant derivative is

$$-\frac{\partial \mathcal{E}(n)}{\partial v_j(n)} = \phi'(v_j(n)) \sum_k -\frac{\partial \mathcal{E}(n)}{\partial v_k(n)} w_{kj}(n).$$

This depends on the change in weights of the kth nodes, which represent the output layer.

So to change the hidden layer weights, the output layer weights change according to the derivative of the activation function, and so this algorithm represents a back propagation of the activation function.

2.2. Terminology

The term "multilayer perceptron" does not refer to a single perceptron that has multiple layers. Rather, it contains many perceptron that are organized into layers.

An alternative is "multilayer perceptron network". Moreover, MLP "perceptron" are not perceptron in the strictest possible sense.

True perceptron are formally a special case of artificial neurons that use a threshold activation function such as the Heaviside step function.

MLP perceptron can employ arbitrary activation functions.

A true perceptron performs binary classification (either this or that), an MLP neuron is free to either perform classification or regression, depending upon its activation function.

The term "multilayer perceptron" later was applied without respect to nature of the nodes/layers, which can be composed of arbitrarily defined artificial neurons, and not perceptron specifically.

This interpretation avoids the loosening of the definition of "perceptron" to mean an artificial neuron in general.

2.3. Applications

MLPs are useful in research for their ability to solve problems stochastically, which often allows approximate solutions for extremely complex problems like fitness approximation.

MLPs are universal function approximates as showed by Cybenko's theorem, so they can be used to create mathematical models by regression analysis.
As classification is a particular case of regression when the response variable is categorical, MLPs make good classifier algorithms.

MLPs were a popular machine learning solution in the 1980s, finding applications in diverse fields such as speech recognition, image recognition, and machine translation software, but thereafter faced strong competition from much simpler support vector machines.
Interest in back propagation networks returned due to the successes of deep learning.

2.4. Diagrammatic Representation

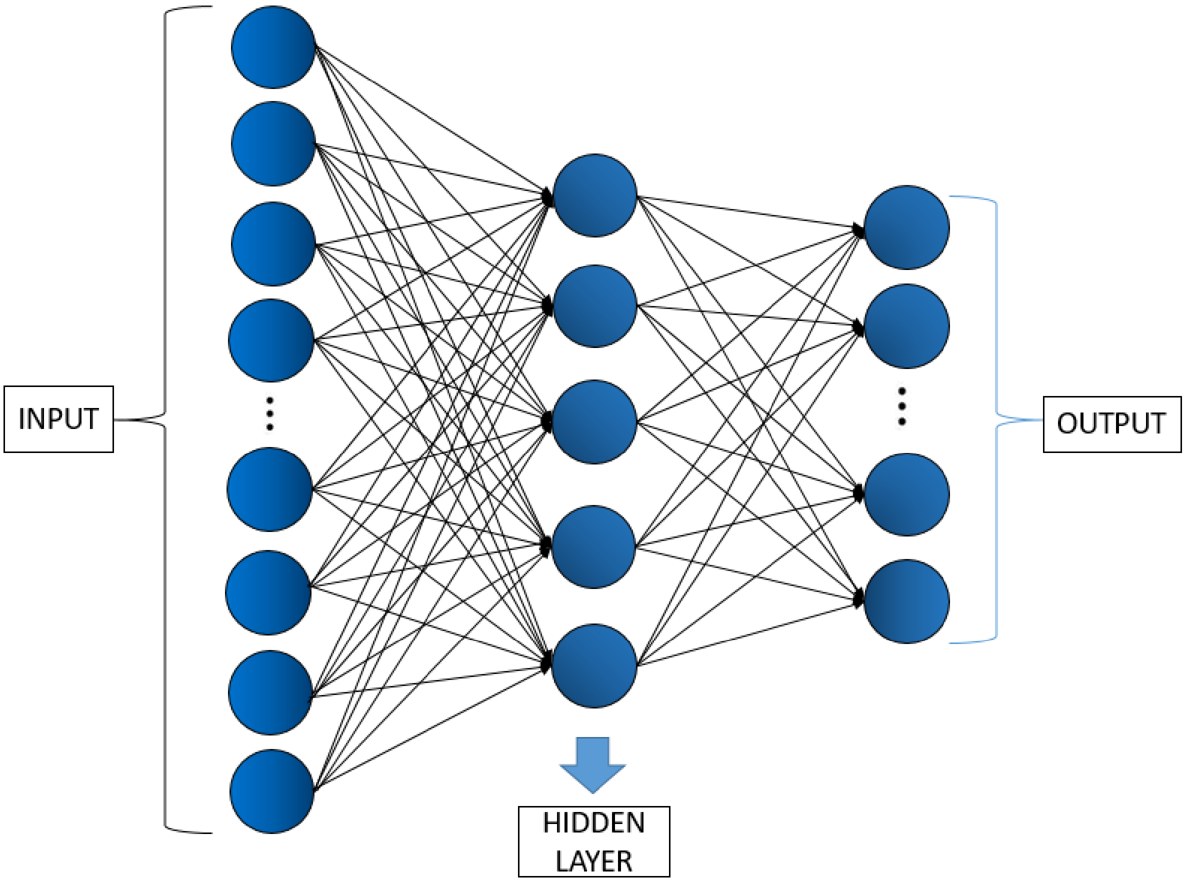


FIG : 2.4

- A multilayer perceptron (MLP) is a class of feed forward artificial neural network.
- An MLP consists of at least three layers of nodes. Except for the input nodes, each node is a neuron that uses a nonlinear activation function.
- Here we have used 129 sample points as input.
- The hidden layer contains 5 neurons.
- In the output layer there are 12 points.

2.5. Feature Matrix

- The matrix used for comparison, consists of 12,000 rows with 129 columns in each as it parts the input in 2min format with 10ms partition.

2.6. Training Matrix

- The Training matrix used for comparison, consists of 144,000 rows with 129 columns in each as it has input in 2min format with 10ms partition.

A multilayer perceptron (MLP) is a deep, artificial neural network. It is composed of more than one perceptron.
They are composed of an input layer to receive the signal, an output layer that makes a decision or prediction about the input, and in between those two, an arbitrary number of hidden layers that are the true computational engine of the MLP.
MLPs with one hidden layer are capable of approximating any continuous function.

Multilayer perceptron are often applied to supervised learning problems, they train on a set of input-output pairs and learn to model the correlation (or dependencies) between those inputs and outputs.
Training involves adjusting the parameters, or the weights and biases, of the model in order to minimize error.
Back propagation is used to make those weigh and bias adjustments relative to the error, and the error itself can be measured in a variety of ways, including by root mean squared error (RMSE).

CONFUSION MATRIX

3. 1 **DEFINITION:** A confusion matrix is a table that is often used to describe the performance of a classification model (or "classifier") on a set of test data for which the true values are known.

In predictive analytics, a table of confusion (sometimes also called a confusion matrix), is a table with two rows and two columns that reports the number of false positives, false negatives, true positives, and true negatives. This allows more detailed analysis than mere proportion of correct classifications (accuracy). Accuracy is not a reliable metric for the real performance of a classifier, because it will yield misleading results if the data set is unbalanced (that is, when the numbers of observations in different classes vary greatly).

3.2 TERMINOLOGIES RELATED TO CONFUSION MATRIX

CONDITION POSITIVE (P) → The number of real positive cases in the data

CONDITION NEGATIVE (N) → The number of real negative cases in the data

TRUE POSITIVE (TP) → Equivalent with hit

TRUE NEGATIVE (TN) → Equivalent with correct rejection

FALSE POSITIVE (FP) → Equivalent with false alarm

FALSE NEGATIVE (FN) → Equivalent with misdetection

3.3 *SAMPLE CONFUSION MATRIX:*

<table>
<tr><td></td><td></td><td colspan="2">True condition</td><td colspan="3"></td></tr>
<tr><td></td><td>Total population</td><td>Condition positive</td><td>Condition negative</td><td>Prevalence = $\frac{\Sigma \text{ Condition positive}}{\Sigma \text{ Total population}}$</td><td colspan="2">Accuracy (ACC) = $\frac{\Sigma \text{ True positive} + \Sigma \text{ True negative}}{\Sigma \text{ Total population}}$</td></tr>
<tr><td rowspan="2">Predicted condition</td><td>Predicted condition positive</td><td>True positive, Power</td><td>False positive, Type I error</td><td>Positive predictive value (PPV), Precision = $\frac{\Sigma \text{ True positive}}{\Sigma \text{ Predicted condition positive}}$</td><td colspan="2">False discovery rate (FDR) = $\frac{\Sigma \text{ False positive}}{\Sigma \text{ Predicted condition positive}}$</td></tr>
<tr><td>Predicted condition negative</td><td>False negative, Type II error</td><td>True negative</td><td>False omission rate (FOR) = $\frac{\Sigma \text{ False negative}}{\Sigma \text{ Predicted condition negative}}$</td><td colspan="2">Negative predictive value (NPV) = $\frac{\Sigma \text{ True negative}}{\Sigma \text{ Predicted condition negative}}$</td></tr>
<tr><td></td><td></td><td>True positive rate (TPR), Recall, Sensitivity, probability of detection = $\frac{\Sigma \text{ True positive}}{\Sigma \text{ Condition positive}}$</td><td>False positive rate (FPR), Fall-out, probability of false alarm = $\frac{\Sigma \text{ False positive}}{\Sigma \text{ Condition negative}}$</td><td>Positive likelihood ratio (LR+) = $\frac{TPR}{FPR}$</td><td rowspan="2">Diagnostic odds ratio (DOR) = $\frac{LR+}{LR-}$</td><td rowspan="2">F_1 score = $\frac{2}{\frac{1}{\text{Recall}} + \frac{1}{\text{Precision}}}$</td></tr>
<tr><td></td><td></td><td>False negative rate (FNR), Miss rate = $\frac{\Sigma \text{ False negative}}{\Sigma \text{ Condition positive}}$</td><td>True negative rate (TNR), Specificity (SPC) = $\frac{\Sigma \text{ True negative}}{\Sigma \text{ Condition negative}}$</td><td>Negative likelihood ratio (LR−) = $\frac{FNR}{TNR}$</td></tr>
</table>

Fig : 3.3

TP → TRUE POSITIVE ; FN → FALSE NEGATIVE

- **SENSITIVITY:** Sensitivity (also called the true positive rate, the recall, or probability of detection in some fields) measures the proportion of positives that are correctly identified as such (e.g. the percentage of sick people who are correctly identified as having the condition).

  EXPRESSION:

SENSITIVITY OR TRUE POSITIVE RATE OR HIT RATE (TPR) = TP/ (TP+FN)

- **SPECIFICITY:** Specificity (also called the true negative rate) measures the proportion of negatives that are correctly identified as such (e.g. the percentage of healthy people who are correctly identified as not having the condition).

  EXPRESSION:

SPECIFICITY OR TRUE NEGATIVE RATE (SPC) = {TN/ (TN+FP)} *100%

- **POSITIVE PREDICTIVE VALUE (PPV) :** The positive predictive value (PPV) is defined as where a "true positive" is the event that the test makes a positive prediction, and the subject has a positive result under the gold standard, and a "false positive" is the event that the test makes a positive prediction, and the subject has a negative result under the gold standard.

EXPRESSION:

PPV=TP/ (TP+FP)

- **NEGATIVE PREDICTIVE VALUE (NPV):** The negative predictive value is defined as where a "true negative" is the event that the test makes a negative prediction, and the subject has a negative result under the gold standard, and a "false negative" is the event that the test makes a negative prediction, and the subject has a positive result under the gold standard.

EXPRESSION:

NPV = TN/(TN+FN)

- **FALL-OUT OR FALSE POSITIVE RATE (FPR) :** A false positive ratio (or false alarm ratio) is the probability of falsely rejecting the null hypothesis for a particular test. The false positive rate is calculated as the ratio between the number of negative events wrongly categorized as positive (false positives) and the total number of actual negative events (regardless of classification).

EXPRESSION:

FPR=FP/(FP+TN)

- **FALSE NEGATIVE RATE (FNR) :** A false positive error, or in short a false positive, commonly called a "false alarm", is a result that indicates a given condition exists, when it does not.

EXPRESSION:

FNR={FN/(TP+FN)}*100%

3.4 UTILITY OF CONFUSION MATRIX IN OUR PROJECT

A Confusion Matrix is a visual performance assessment of a classification algorithm in the form of a table layout or matrix. Each column of the matrix represents predicted classifications and each row represents actual defined classifications. This representation is a useful way to help evaluate a classifier model. A well behaved model should produce a balanced matrix and have consist percent correctness numbers for accuracy, recall, precision and an F measure. If it does not, there is cause to further evaluate the data used to build the model and the data used to test the model.

Let us assume a situation where we have to determine the possible probabilities that can occur in our practical life. Like a the possible situations can happen while using a fingerprint sensor.

There are 4 possible situations can occur, i.e

1) If the sensor correctly detects the authorized person and allows him to enter the certain room ,this probability is known as TRUE POSITIVE.

2) If the sensor correctly detects the unauthorized person and restrains him to enter the certain room , this probability is known as TRUE NEGATIVE.

3) If the sensor failed to detect the authorized person and restrains him to enter the certain room , this type of probable situation is known as FALSE NEGATIVE OR MISDETECTION.

4) If the sensor incorrectly detects an unauthorized person as an authorized person then this type of probable situation is called FALSE POSITIVE OR FALSE ALARM.

As many cases of true positive or true negative situations better the hit ratio.

P	P
N	N
P	N
N	P

Table : 1

Here,

The first case is known as TRUE POSITIVE

The second case is known as TRUE NEGATIVE

The third case is known as FALSE NEGATIVE

The fourth case is known as FALSE POSITIVE

Let us assume there are 3 classes and the target class is 2

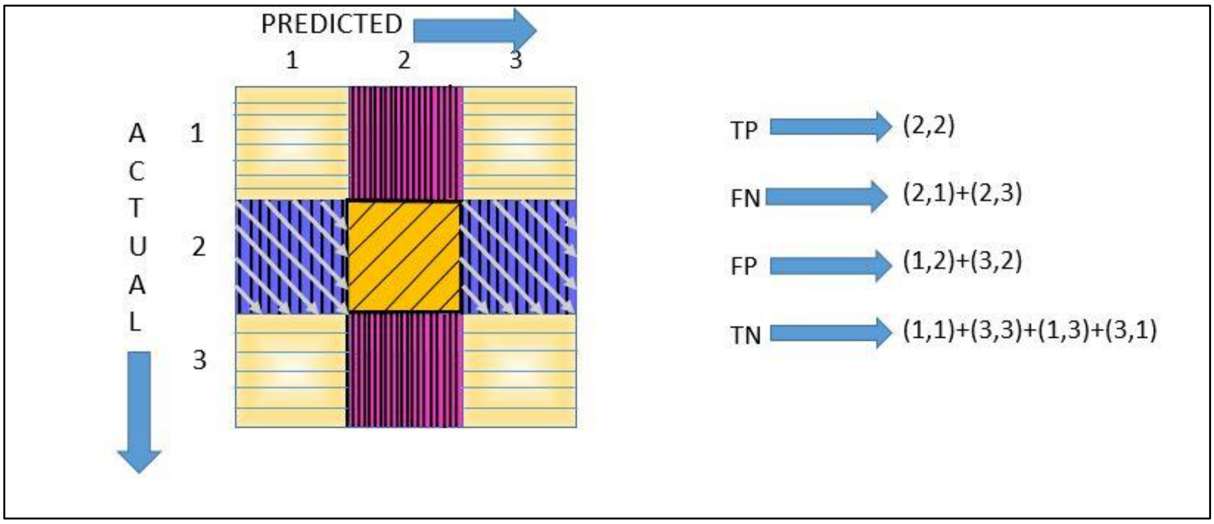


Fig : 3.4

TP → TRUE POSITIVE

FN → FALSE NEGATIVE

FP → FALSE POSITIVE

TN → TRUE NEGATIVE

→ Here the **TRUE POSITIVE RESULT (2,2)** defines that the system detects the signal or input correctly. Better the true positive rate better is the accuracy.

→ Next the **FALSE NEGATIVE RESULTS (2,1),(2,3)** define that the system failed to detect the correct input and treated it as negative. This situation i.e MIS DETECTION is not desired in case of of acquiring a better accuracy.

- ➔ Next the **FALSE POSITIVE RESULT (1,2),(3,2)** define that the system incorrectly detects a wrong input and treated as the desired input. This situation i.e FALSE ALARM is also not desired to achieve a better accuracy.

- ➔ Last the **TRUE NEGATIVE RESULT (1,1),(1,3),(3,1),(3,3)** are collectively state that the system successfully rejects the undesired inputs. It also helps to achieve a better accuracy.

$$\boxed{\text{ACCURACY} = (\text{TP}+\text{FN})/\text{A}_{\text{TOTAL}}}$$

4. DETAILED ANALYSIS

4.1 CLASSES OF SOUND ENVIRONMENT USED

- COCKPIT 1
- COCKPIT 2
- COCKPIT 3
- ENGINE 1
- ENGINE 2
- FACTORY 1
- FACTORY 2
- MACHINE GUN
- VOLVO BUS
- WHITE NOISE
- CLASSROOM BABEL NOISE
- HF CHANNEL

4.2 DETAILED EXPLANATION

The objective of the Acoustic Analytics: Acoustic Event Detection, Classification and Analysis research project is to develop key signal processing and analytical tools to extract timely, useful, and actionable information from real-world acoustic data on a large scale. Specifically, we are aiming to build acoustic analytic systems for 24/7/365 machine-automated monitoring of human environments on a large scale. The systems should be general enough, fast enough, and robust enough to yield useful information from large volumes of acoustic data.

Sound is second only to vision as a means by which humans sense and understand the world. The severity of deafness as a disability reflects concisely the importance of sound in understanding what's happening in the physical world, including omni-directional and out-of-sight awareness of events and warning of danger. We thus believe that acoustic sensing, sense-making, and analytics of audio data could prove as significant to machine-automated monitoring of human environments as it is to the humans. The massive deployment of mobile phones and microphones in most personal computing devices has raised the quality and driven down the cost of acoustic data acquisition immensely over the past two decades, as well as provided the capacity to form and deploy massive, city-wide networked arrays of acoustic sensors. Smart-phone technologies enable significant computation for real-time point-of-acquisition data analysis. Thus in the past few years, real-time 24/7/365 acoustic monitoring on a very large scale has gone from inconceivably expensive to eminently feasible. In spite of this potential, acoustic data analysis (other than speech recognition) has been much less studied and deployed in comparison with vision.

Today, acoustic data analytics are based on techniques tailored for specific applications (SONAR detection of submarines; automatic speech recognition for specific languages and contexts; detection of whale calls of a specific species; gunshot detection) that do not generalize effectively. These techniques are often very expensive computationally and so are unsuitable for 24/7/365 analysis of large volumes of audio data streams. Therefore, advances in audio signal processing and acoustical analytics are required to process large volumes of audio data to extract timely, useful, and actionable information from the real world.

Audio-based monitoring also has significant impact on surveillance for public safety and security and on urban noise assessment of residential area. Potential safety hazard is raised for crowded public areas like drinking bars and the places for recreation and leisure, especially during nighttime. Crimes like robbery also happen in hidden public areas without awareness from neighbors. It is crucial to detect such emergency cases in a timely manner and alert police for preventing further damage. Audio-based monitoring systems have proven to be very useful tools for detecting such cases. It might normally be the case that there is shouting/screaming/crying sound from these kinds of incidents. Timely acoustic sensing and correctly detecting the anomaly events become very important.

Likewise, urban noise effects are becoming more and more serious to human health. Traffic, business and even recreational activities all contribute to spoiling a city and harming its inhabitants by exposing them to undue levels of noise. Noise issues have to be carefully analyzed and controlled. Noise mapping and prediction is an essential tool to aid the assessment of noise levels over a wide area and to predict the changes in the noise environment due to changes in use. Creating an accurate noise map will be very useful in communicating issues and defining future policy, such as to communicate the noise situation to stakeholders, to inform areas of planning such as construction, traffic & transport and to build a common understanding within the community. Towards our vision of developing the capability to extract timely, useful, and actionable information from real-world acoustic data on a large scale, this project is divided into several major tasks:

- **Acoustic event detection**: performing universal acoustic event detection for quickly discarding the vast majority of uninteresting data in the analysis of massive audio datasets, and improving essential robustness to adapt to different background noises while retaining high sensitivity to detection events;

- **Acoustic event classification**: classifying the detected events into different classes that are precisely defined for human understanding with improved robustness to environmental noise and interference;

- **Anomaly detection**: detecting rare or never-before-observed unusual acoustic events;

- **Acoustic energy density mapping**: monitoring environmental noise levels and mapping over space and time the intensity and other parameters of the noise field;

- **Real-time audio analytics**: Performing high-level audio analysis to exploit and extract the known structure in acoustic data for further data analytics and mining.

- **Sonification**: encoding of non-auditory information into sound for "visualization" for increasing the information-delivery capacity and representing certain types of data more effectively.

4.3 FEATURE MATRIX

In machine learning and pattern recognition, a **feature** is an individual measurable property or characteristic of a phenomenon being observed. Choosing informative, discriminating and independent features is a crucial step for effective algorithms in pattern recognition, classification and regression. Features are usually numeric, but structural features such as strings and graphs are used in syntactic pattern recognition. The concept of "feature" is related to that of explanatory variable used in statistical techniques such as linear regression.

4.3 Classification

A set of numeric features can be conveniently described by a feature vector. An example of reaching a two-way classification from a feature vector (related to the perceptron) consists of calculating the scalar product between the feature vector and a vector of weights, comparing the result with a threshold, and deciding the class based on the comparison.

Algorithms for classification from a feature vector include nearest neighbor classification, neural networks, and statistical techniques such as Bayesian approaches.

4.3.1 Examples

In character recognition, features may include histograms counting the number of black pixels along horizontal and vertical directions, number of internal holes, stroke detection and many others.

In speech recognition, features for recognizing phonemes can include noise ratios, length of sounds, relative power, filter matches and many others.

In spam detection algorithms, features may include the presence or absence of certain email headers, the email structure, the language, the frequency of specific terms, the grammatical correctness of the text.

In computer vision, there are a large number of possible features, such as edges and objects.

In pattern recognition and machine learning, a **feature vector** is an n-dimensional vector of numerical features that represent some object. Many algorithms in machine learning require a numerical representation of objects, since such representations facilitate processing and statistical analysis. When representing images, the feature values might correspond to the pixels of an image, while when representing texts the features might be the frequencies of occurrence of textual terms. Feature vectors are equivalent to the vectors of explanatory variables used in statistical procedures such as linear regression. Feature vectors are often combined with weights using a dot product in order to construct a linear predictor function that is used to determine a score for making a prediction.

The vector space associated with these vectors is often called the **feature space**. In order to reduce the dimensionality of the feature space, a number of dimensionality reduction techniques can be employed.

Higher-level features can be obtained from already available features and added to the feature vector; for example, for the study of diseases the feature 'Age' is useful and is defined as *Age = 'Year of death' minus 'Year of birth'* . This process is referred to as **feature construction**. Feature construction is the application of a set of constructive operators to a set of existing features resulting in construction of new features. Examples of such constructive operators include checking for the equality conditions {=, ≠}, the arithmetic operators {+,−,×, /}, the array operators {max(S), min(S), average(S)} as well as other more sophisticated operators, for example count(S,C) that counts the number of features in the feature vector S satisfying some condition C or, for example, distances to other recognition classes generalized by some accepting device. Feature construction has long been considered a powerful tool for increasing both accuracy and understanding of structure, particularly in high-dimensional problems. Applications include studies of disease and emotion recognition from speech.

4.4 Selection and Extraction

The initial set of raw features can be redundant and too large to be managed. Therefore, a preliminary step in many applications of machine learning and pattern recognition consists of selecting a subset of features, or constructing a new and reduced set of features to facilitate learning, and to improve generalization and interpretability.

Extracting or selecting features is a combination of art and science; developing systems to do so is known as feature engineering. It requires the experimentation of multiple possibilities and the combination of automated techniques with the intuition and knowledge of the domain expert. Automating this process is feature learning, where a machine not only uses features for learning, but learns the features itself.

4.5 FORMATION OF FEATURE MATRIX

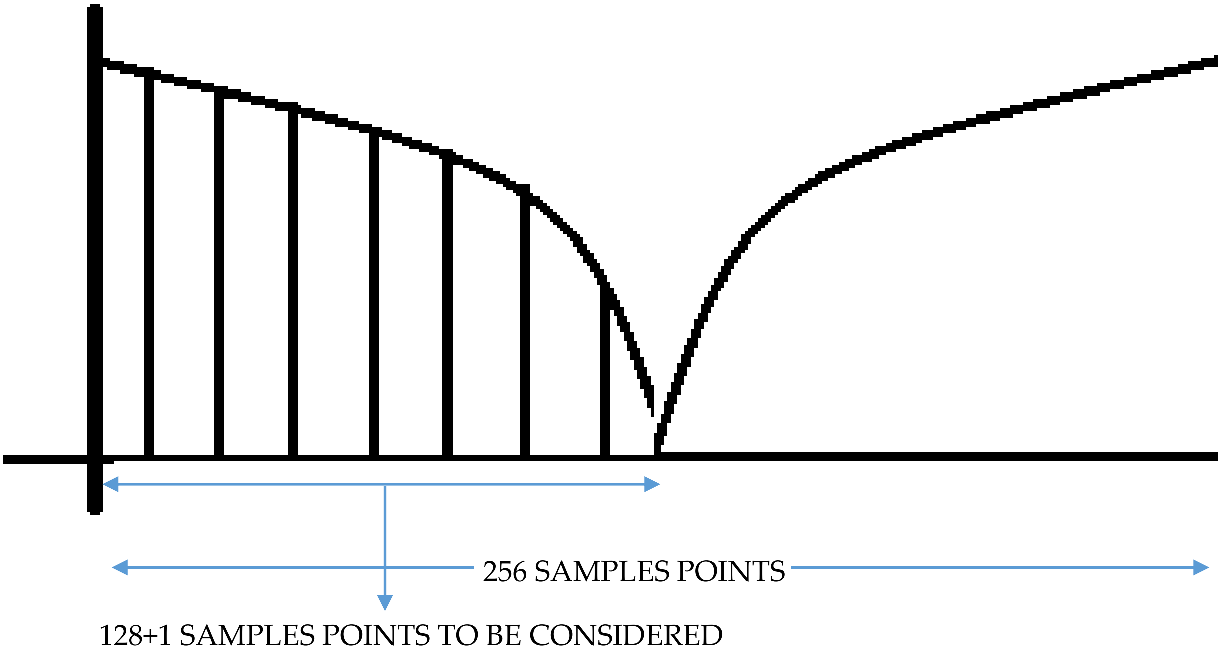
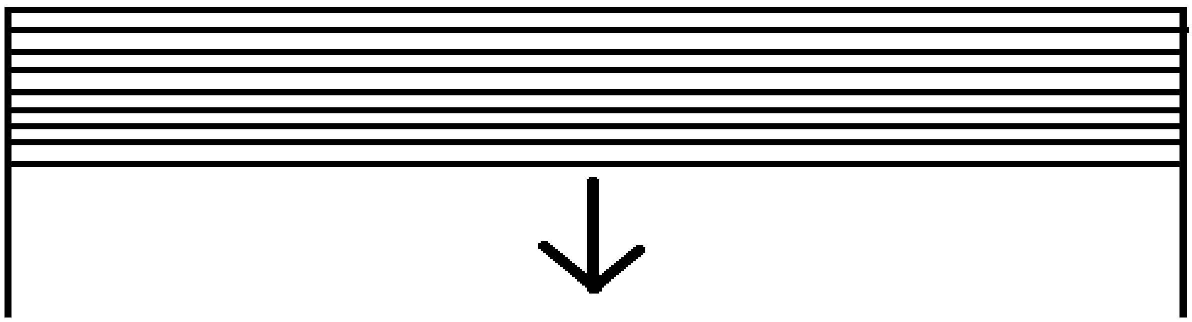


FIG : 4.5 (A)



USING THIS CONSIDERED 129 SAMPLE POINTS, WE GENERATE A PARTICULAR FEATURE MATRIX FOR EACH SET OF SAMPLES USED.

FIG : 4.5 (B)

Feature learning is the set of methods that allow to find an appropriate representation of data in order to perform a machine learning task. In other words, the goal of feature learning is to find a transformation that maps raw data into a representation that is more suitable for a machine learning task (e.g. classification).

Let's see it through an example. For this purpose, we will use a neural network, which exploits the concept of feature learning by its very nature. In a neural network, each hidden layer maps its input data to an inner representation that tends to capture a higher level of abstraction.

Suppose we want to classify the next dataset:

Dataset that we want to classify.

Note that this dataset **is not** linearly separable, because we can't separate it using a linear model (such as a feedforward neural network without hidden layers) without tweaking our input. For example, when training a feedforward neural network without any hidden layer using this dataset, we obtain the next classification boundaries:

Classification boundaries obtained when we train a feedforward neural network without any hidden layer to classify our dataset.

Here's where feature learning makes the difference. Neural networks can learn inner representations of data through hidden layers, and that's why they are so powerful. Accompanied by a proper non-linear activation function, hidden layers map its input data into a more abstract non-linear space.

By learning multiple successive inner representations of data we expect to find a transformation such that the resulting latent features are linearly separable. Once we achieve that, performing a simple logistic regression on top of the network will be enough to make an accurate classification.

Recall that we weren't able to separate one class from another with a straight line. Now, imagine you have Figure 1 plotted on a transparent sheet. What would you do to make data linearly separable?

You could fold the transparent sheet at A=B. Note that now you can draw a straight line on the transparent sheet that separates one class from the other. You've just learnt an inner representation of data that makes it linearly separable!

That's exactly what a neural network does. For example, when training a feedforward neural network with one hidden layer using our dataset, it learns the following inner representation of data:

Inner representation of data at the hidden layer. You could imagine this as a top view of the transparent sheet. Note that now we could easily draw a straight line separating one class from the other.

Classification boundaries obtained when we train a feedforward neural network with one hidden layer to classify our dataset.

Finally, it's important to add that a neural network can learn much more complex transformations than folding a sheet. Also, I hope that I could provide you with the intuition behind inner representations of data and feature learning.

Features are the variables found in the given problem set that can strongly/sufficiently help us build an accurate predictive model.

Eg : To predict the sale price of a house, size of the house is a feature.

- Features are a column of data given as the input. They are also called as attributes or might sometimes be referred as dimensions.
- A particular problem data set can have several features tagging to them. It is important to select the features that are more relevant to our problem so that the accuracy of the model improves. It also reduces the complexity of the model as we avoid the least significant / unnecessary feature data. The simpler model is simpler to understand and explain.
- This Process is called feature engineering / selection and is one of the crucial step of pre-processing. Different algorithms can be used to implement it.
- The Features can be of different types.
  - Simple Supervised selection where they are simple values like numbers and characters.
    - Eg: Size of the house (number) .
  - In unsupervised learning, the model is itself trained to recognize the features and work on it.
    - Eg: In character recognition, features may include histograms counting the number of black pixels along horizontal and vertical directions, number of internal holes, stroke detection and many others.

Eg.: Loan Granting Problem

Let us build a model that tells us if to give loan to a particular customer or not.

Now its data will have many features/attributes attached to it :

Loan id, Cust. Name, Cust. id, Cust.Addr, Employed (or) not, Age, Marital Status, Has already avail loan, Annual Income, no.of open accounts, tax liens, credit score, current balance and so on.

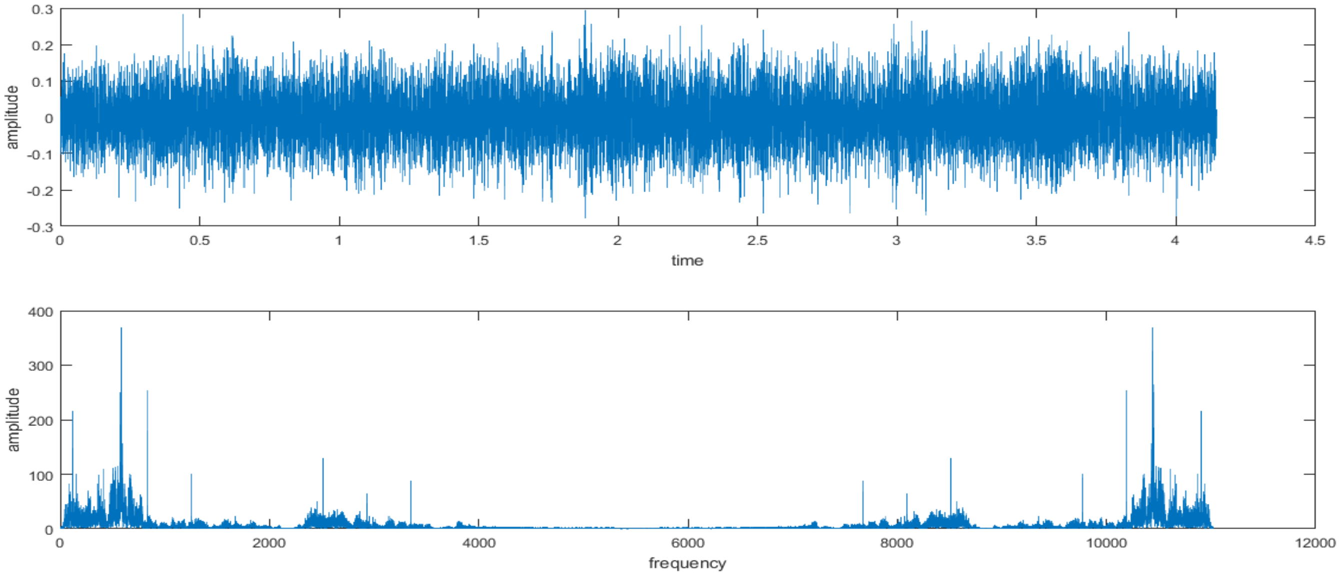
Using the feature selection it can be observed that for a particular Customer,

Employed (or) not, age, current credit score , annual income, already availed loancan significantly explain / contribute to the model accuracy better than the others.

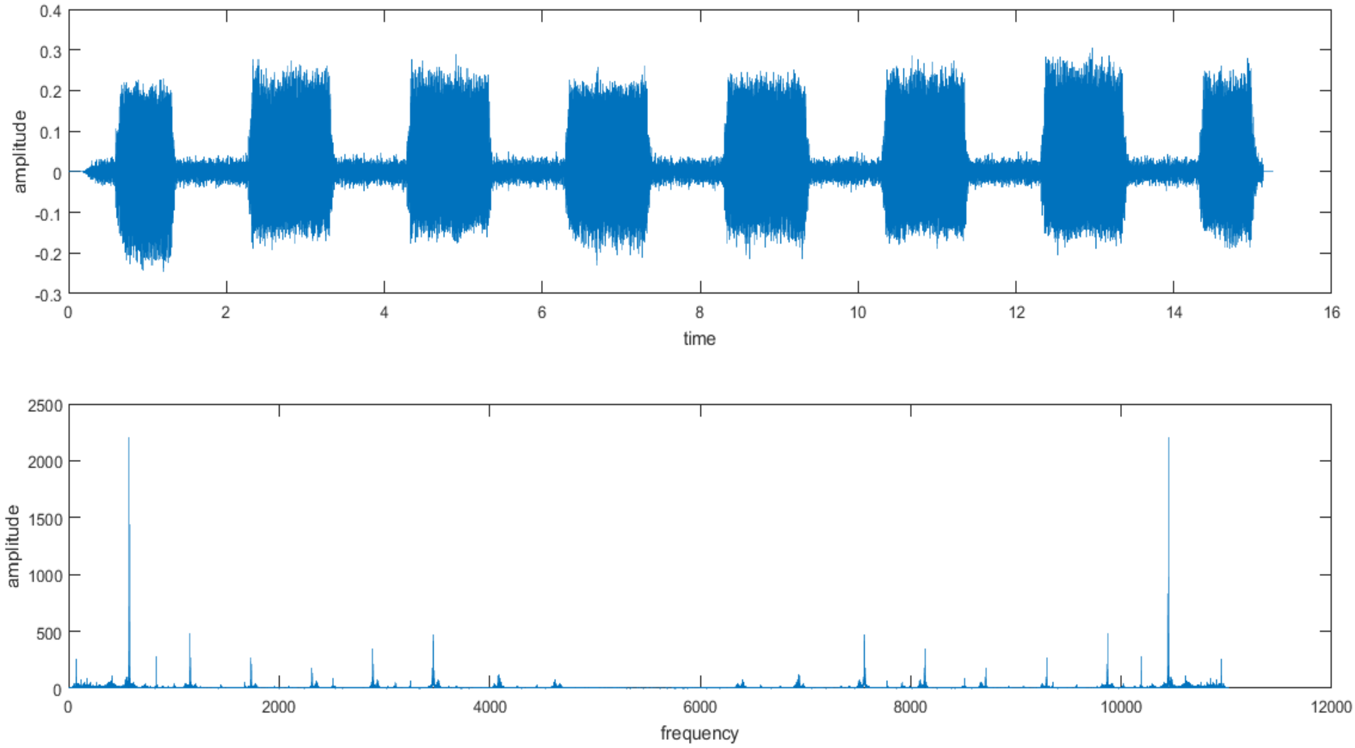
Thus they become the features for building our model for this particular problem.

OUTCOME (SPECTRUM ANALYSIS)

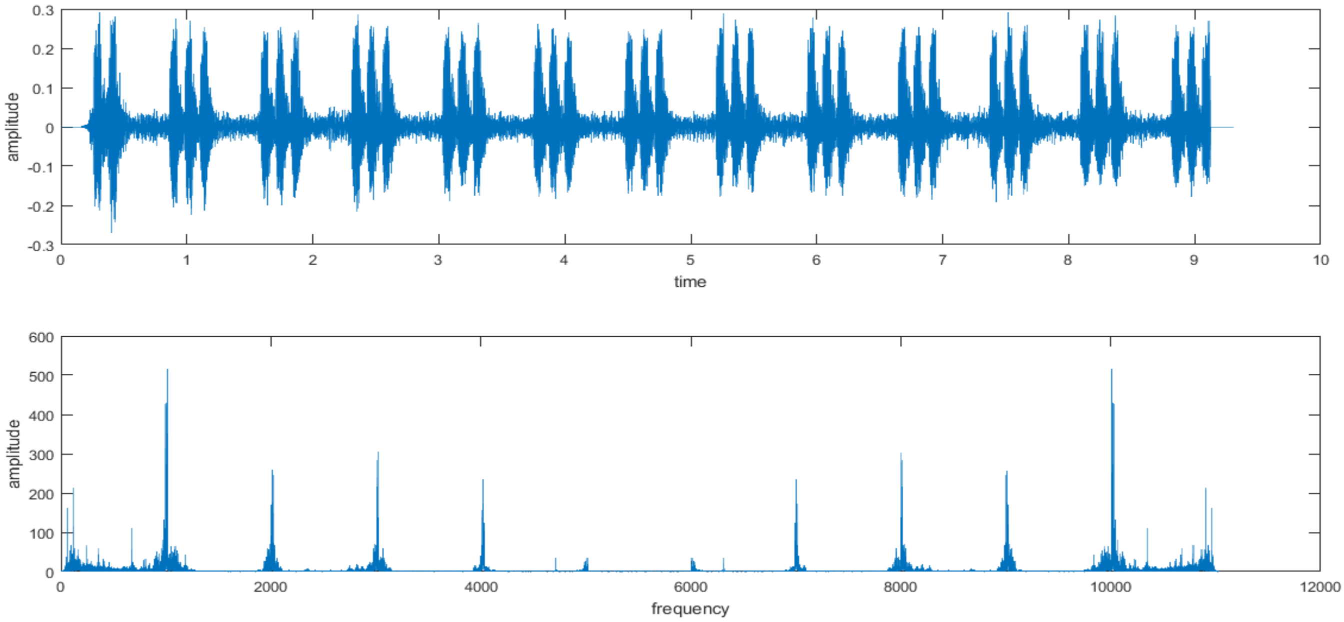
4.6.1 SPECTRUM OF FIRST CLASS (COCKPIT 1)



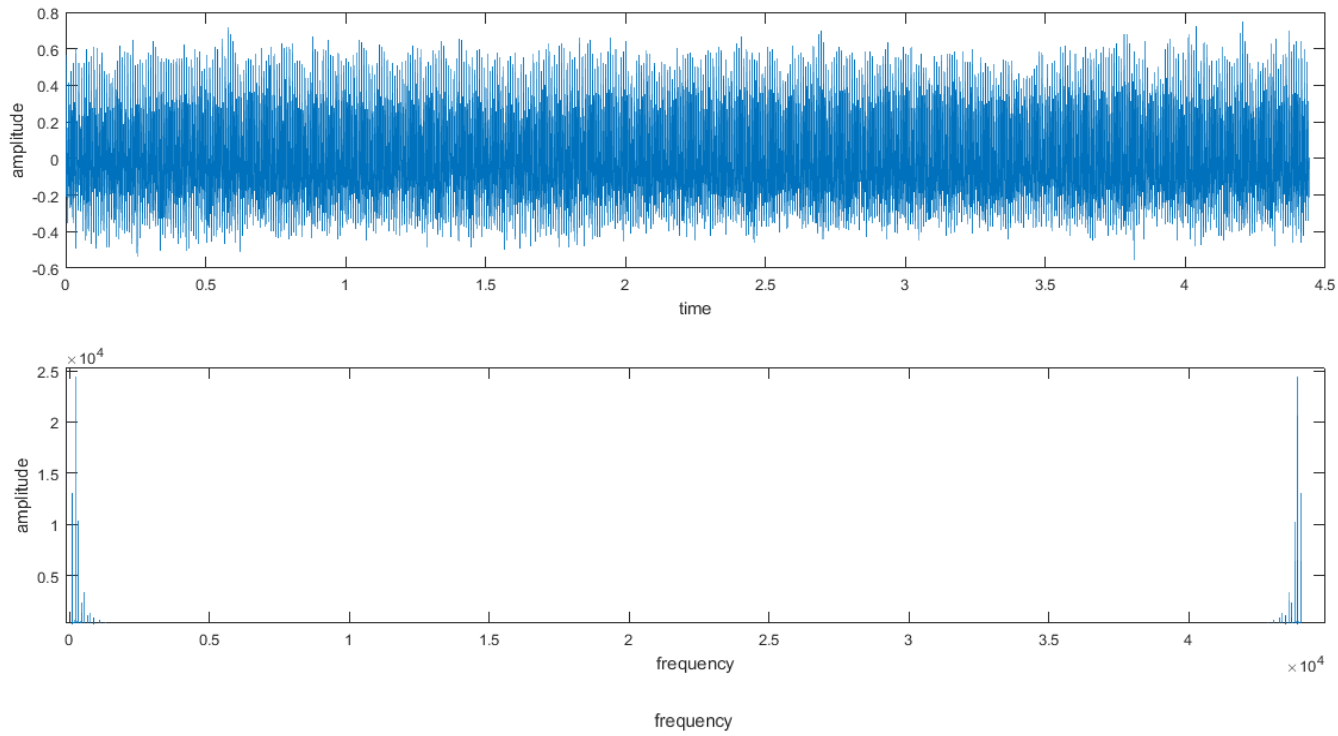
4.6.2SPECTRUM OF SECOND CLASS (COCKPIT 2)



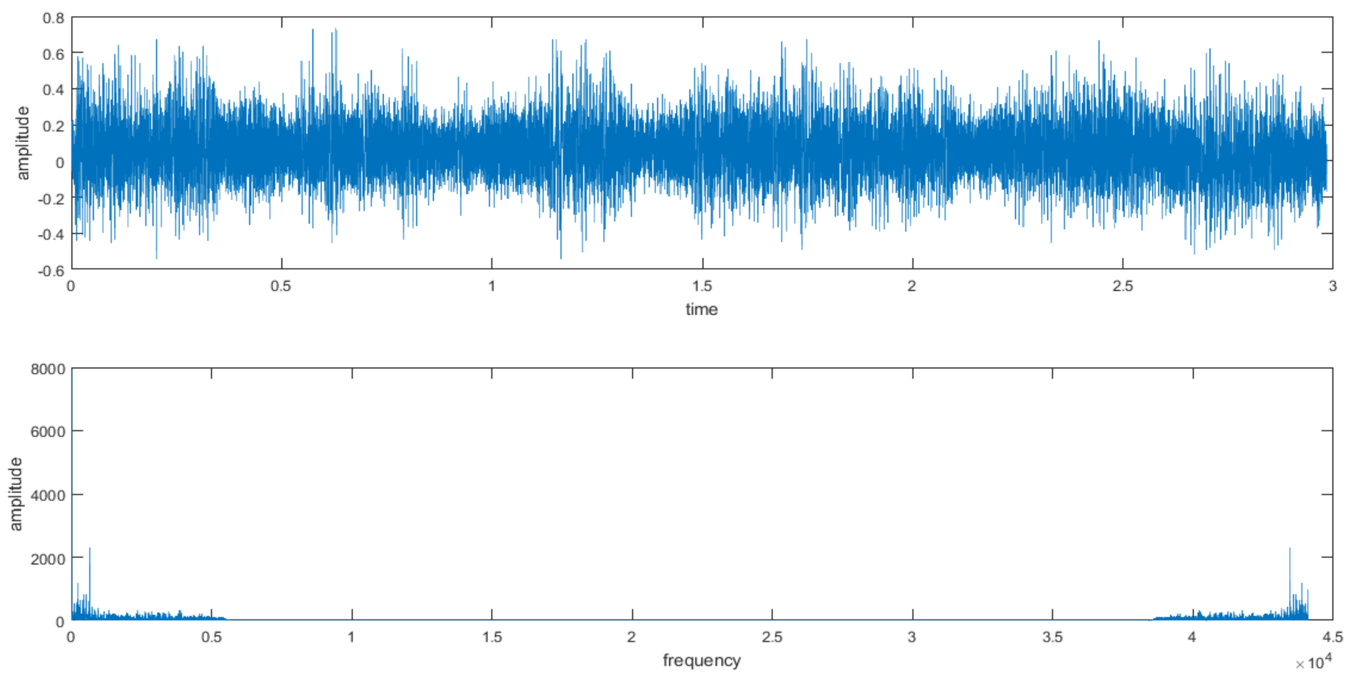
4.6.3.SPECTRUM OF THIRD CLASS (COCKPIT 3)



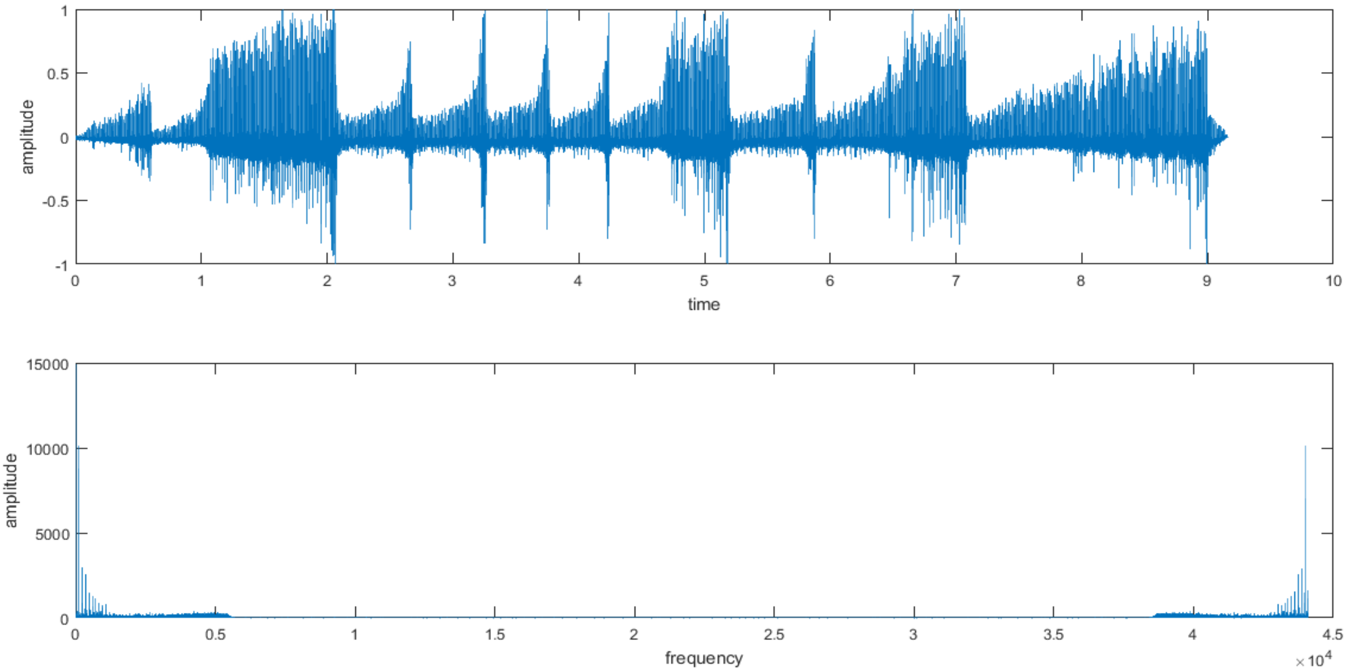
4.6.5.SPECTRUM OF FIFTH CLASS (ENGINE 2)



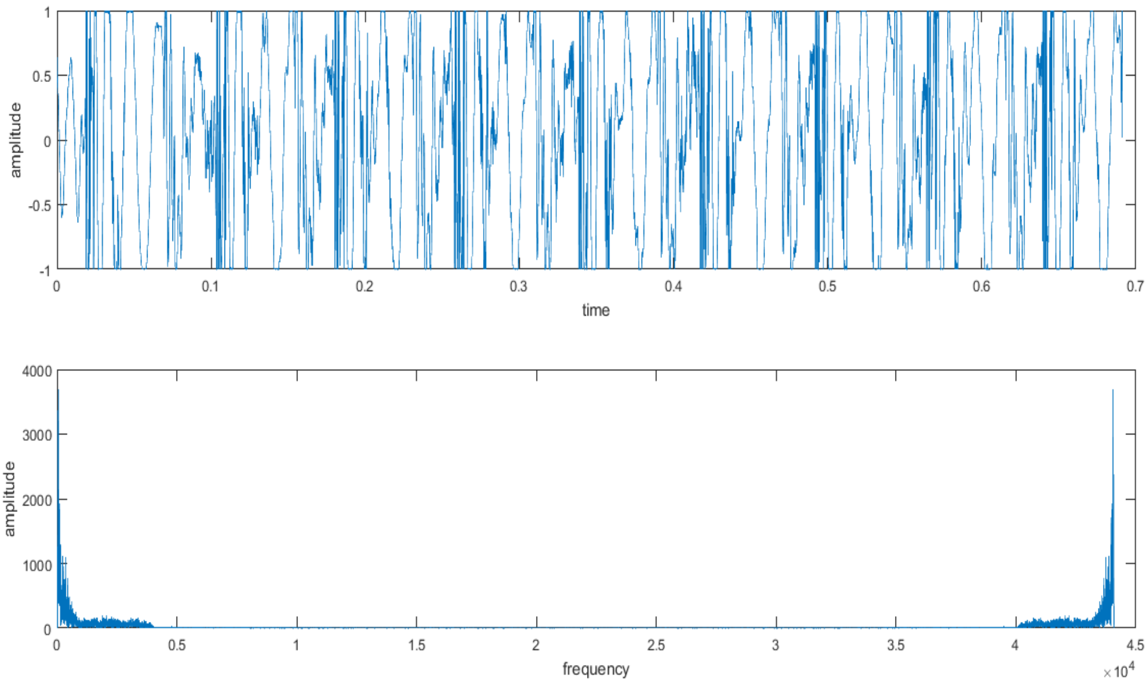
4.6.6.SPECTRUM OF SIXTH CLASS (FACTORY 1)



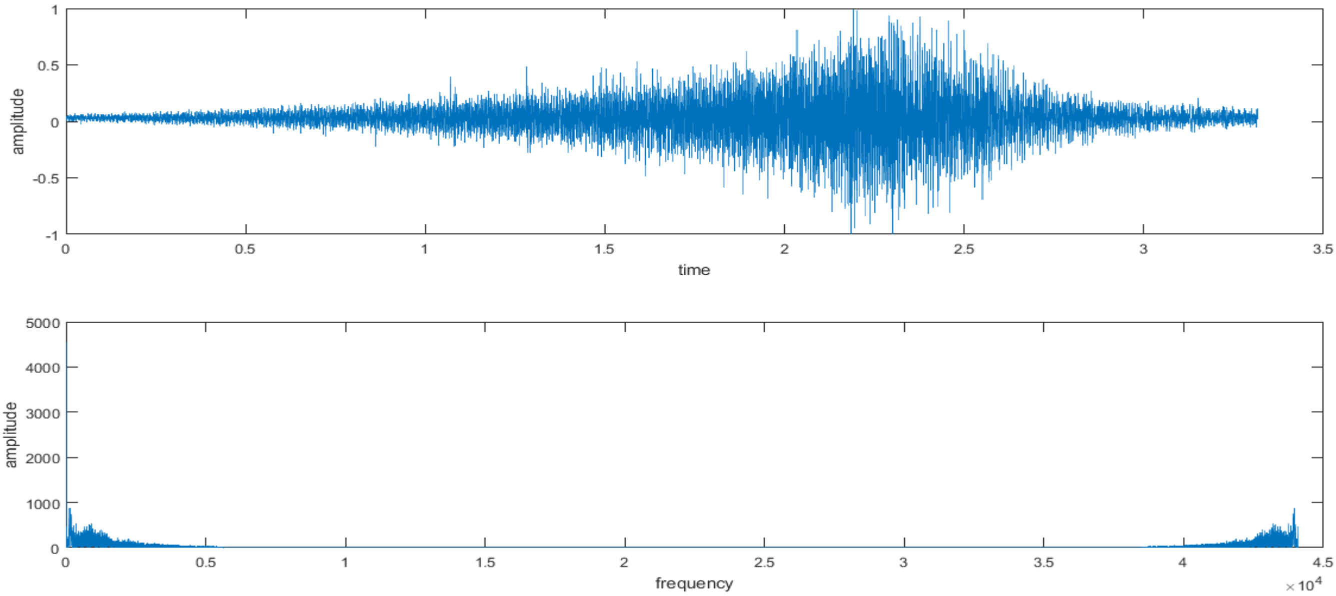
4.6.7.SPECTRUM OF SEVENTH CLASS (FACTORY 2)



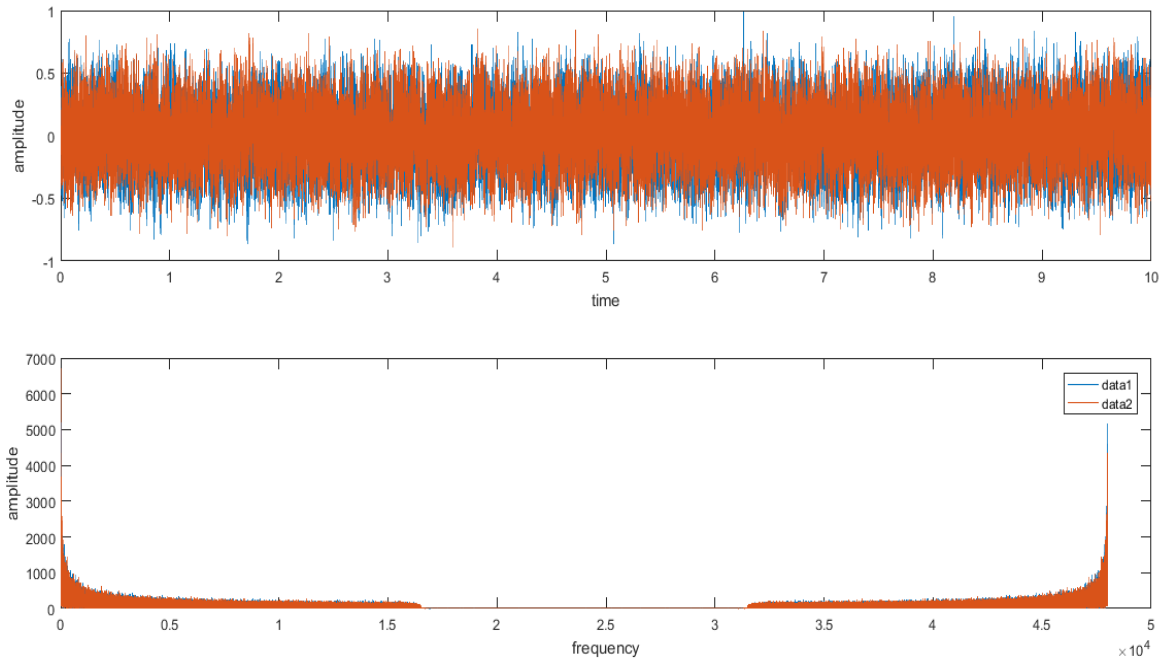
4.6.8.SPECTRUM OF SEVENTH CLASS (MACHINE GUN)



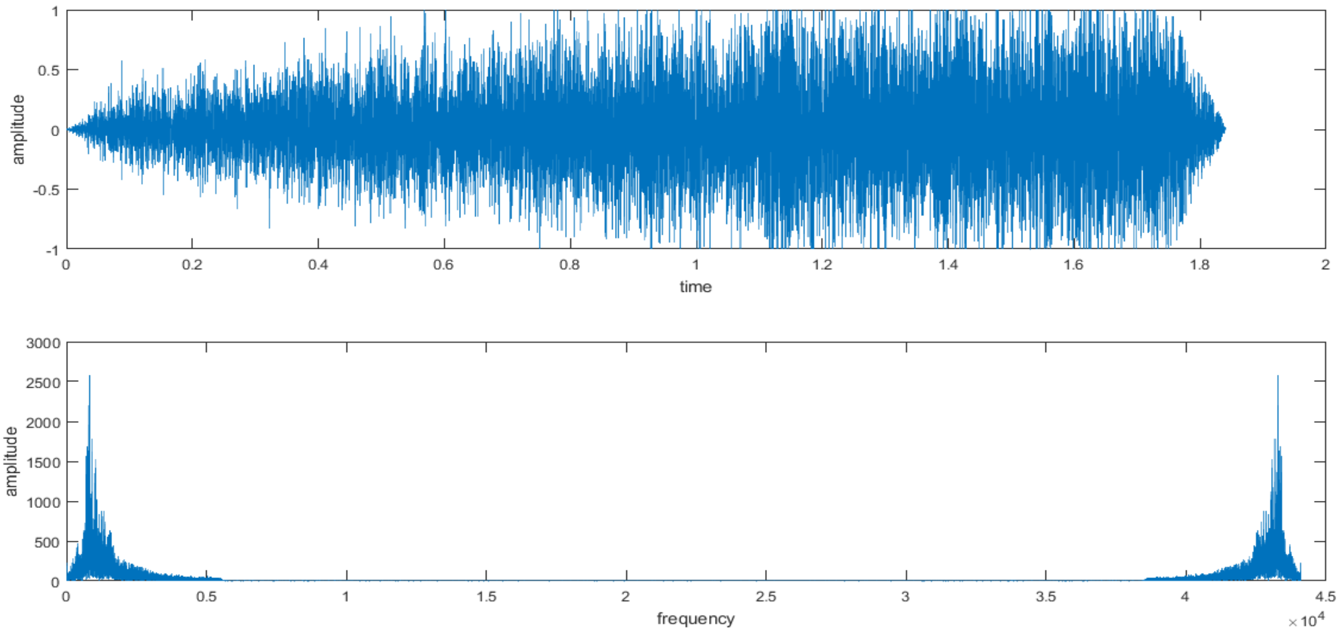
4.6.9.SPECTRUM OF NINTH CLASS (VOLVO BUS)



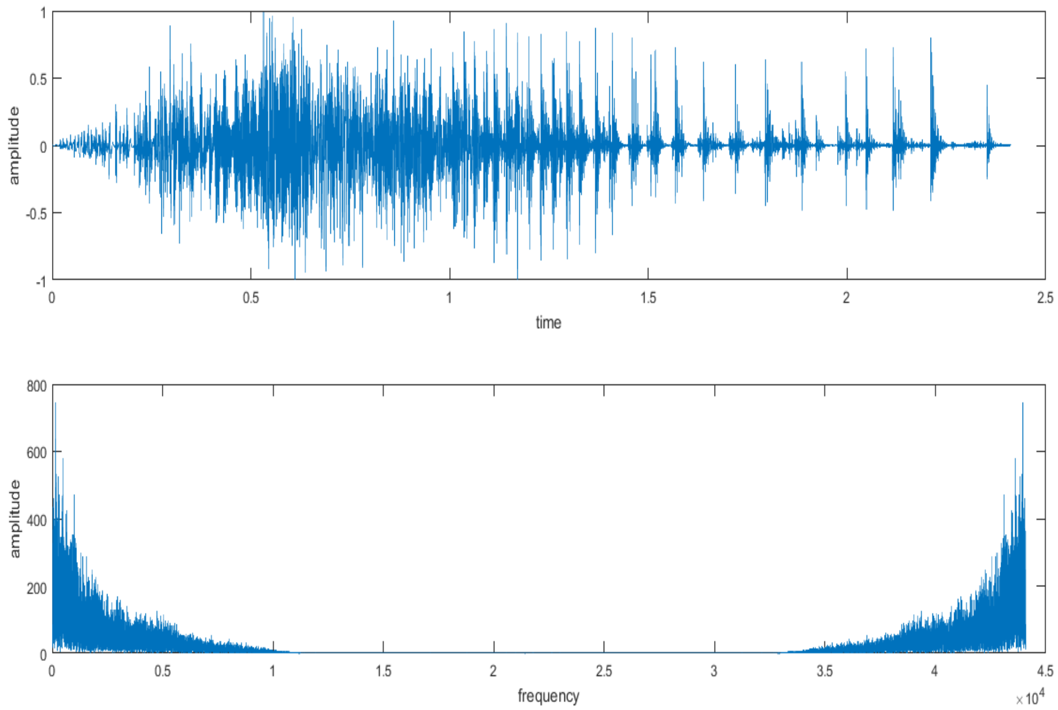
4.6.10.SPECTRUM OF TENTH CLASS (WHITE NOISE)



4.6.11.SPECTRUM OF ELEVENTH CLASS (CLASSROOM BABEL NOISE)



SPECTRUM OF TWELEVETH CLASS (HIGH FREQUENCY CHANNEL



OBSERVATIONS

After the simulation , we get our confusion matrix as - :

Confusion_Matrix =

	I	II	III	IV	V	VI	VII	VIII	IX	X	XI	XII
I	899	1	0	3	31	42	135	100	1	37	0	0
II	0	1025	0	37	0	0	148	22	17	0	0	0
III	0	0	1212	2	0	2	0	0	0	0	0	33
IV	6	27	0	1198	0	0	9	9	0	0	0	0
V	12	0	0	0	1213	0	0	14	10	0	0	0
VI	50	0	31	2	1	1055	54	2	0	51	3	0
VII	68	7	3	3	0	114	873	113	0	50	7	11
VIII	115	0	0	2	0	4	70	1032	0	10	16	0
IX	0	9	0	0	3	0	0	4	1233	0	0	0
X	6	0	0	0	0	15	0	1	0	1206	21	0
XI	0	0	0	0	0	0	0	0	0	246	1003	0
XII	0	0	96	0	0	2	1	0	0	1	0	1149

From above , we observe the folloing scenario-:

Class - babble.wav[CLS ID : I]

HIT = 71.98 %

FA = 1.87 %

ACC = 95.95 %

Class - cockpit_buccaneer1.wav[CLS ID : II]

HIT = 82.07 %

FA = 0.32 %

ACC = 98.21 %

Class - cockpit_buccaneer2.wav[CLS ID : III]

HIT = 97.04 %

FA = 0.95 %

ACC = 98.89 %

Class - cockpit_f16.wav[CLS ID : IV]

HIT = 95.92 %

FA = 0.36 %

ACC = 99.33 %

Class - destroyerengine.wav[CLS ID : V]

HIT = 97.12 %

FA = 0.25 %

ACC = 99.53 %

Class - destroyerops.wav[CLS ID : VI]

HIT = 84.47 %

FA = 1.30 %

ACC = 97.51 %

Class - factory1.wav[CLS ID : VII]

HIT = 69.90 %

FA = 3.04 %

ACC = 94.71 %

Class - factory2.wav[CLS ID : VIII]

HIT = 82.63 %

FA = 1.93 %

ACC = 96.78

Class - hfchannel.wav[CLS ID : IX]

HIT = 98.72 %

FA = 0.20 %

ACC = 99.71 %

Class - machinegun.wav[CLS ID : X]

HIT = 96.56 %

FA = 2.88 %

ACC = 97.08 %

Class - volvo.wav[CLS ID : XI]

HIT = 80.30 %

FA = 0.34 %

ACC = 98.05 %

Class - white.wav[CLS ID : XII]

HIT = 91.99 %

FA = 0.32 %

ACC = 99.04 %

RECEIVER OPERATING CHARACTERISTICS

INTRODUCTION

In statistics, a **receiver operating characteristic curve**, i.e. **ROC curve**, is a graphical plot that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied. The Total Operating Characteristic (TOC) expands on the idea of ROC by showing the total information in the two-by-two contingency table for each threshold. ROC gives only two bits of relative information for each threshold, thus the TOC gives strictly more information than the ROC.

The ROC curve is created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings. The true-positive rate is also known as sensitivity, recall or *probability of detection* in machine learning. The false-positive rate is also known as the fall-out or *probability of false alarm* and can be calculated as (1 − specificity). It can also be thought of as a plot of the Power as a function of the Type I Error of the decision rule (when the performance is calculated from just a sample of the population, it can be thought of as estimators of these quantities). The ROC curve is thus the sensitivity as a function of fall-out. In general, if the probability distributions for both detection and false alarm are known, the ROC curve can be generated by plotting the cumulative distribution function (area under the probability distribution from minus infinity to the discrimination threshold) of the detection probability in the y-axis versus the cumulative distribution function of the false-alarm probability on the x-axis.

ROC analysis provides tools to select possibly optimal models and to discard suboptimal ones independently from (and prior to specifying) the cost context or the class distribution. ROC analysis is related in a direct and natural way to cost/benefit analysis of diagnostic decision making.

The ROC curve was first developed by electrical engineers and radar engineers during World War II for detecting enemy objects in battlefields and was soon introduced to psychology to account for perceptual detection of stimuli. ROC analysis since then has been

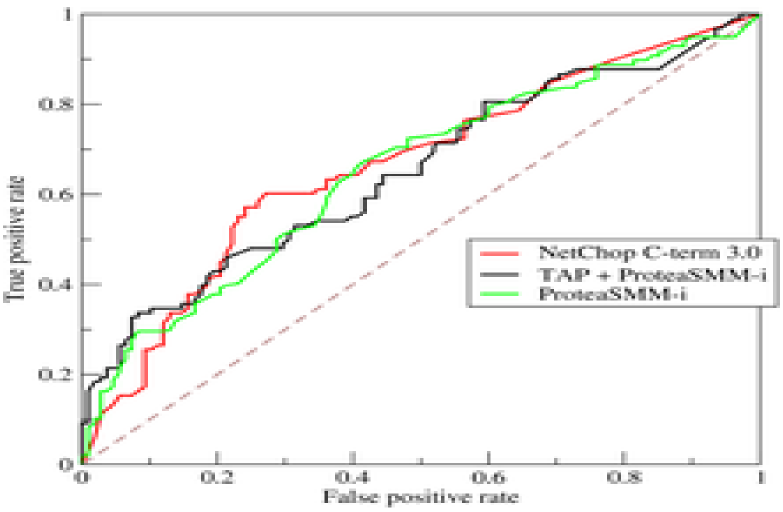


FIG : 6.1(A)

used in medicine, radiology, biometrics, forecasting of natural hazards, meteorology, model performance assessment, and other areas for many decades and is increasingly used in machine learning and data mining research.

The ROC is also known as a relative operating characteristic curve, because it is a comparison of two operating characteristics (TPR and FPR) as the criterion changes.

In a ROC curve the true positive rate (Sensitivity) is plotted in function of the false positive rate (100-Specificity) for different cut-off points of a parameter. Each point on the ROC curve represents a sensitivity/specificity pair corresponding to a particular decision threshold. The area under the ROC curve (AUC) is a measure of how well a parameter can distinguish between two diagnostic groups (diseased/normal).

The diagnostic performance of a test, or the accuray of a test to discriminate diseased cases from normal cases is evaluated using Receiver Operating Characteristic (ROC) curve analysis (Metz, 1978; Zweig & Campbell, 1993). ROC curves can also be used to compare the diagnostic performance of two or more laboratory or diagnostic tests (Griner et al., 1981).

When you consider the results of a particular test in two populations, one population with a disease, the other population without the disease, you will rarely observe a perfect separation between the two groups. Indeed, the distribution of the test results will overlap, as shown in the following figure.

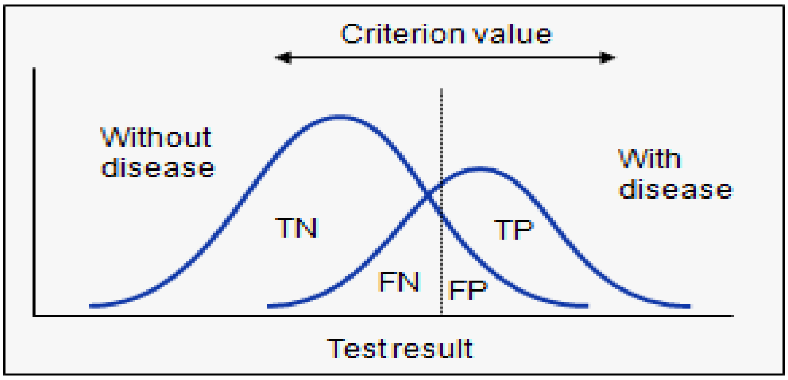


FIG : 6.2

For every possible cut-off point or criterion value you select to discriminate between the two populations, there will be some cases with the disease correctly classified as positive (TP = True Positive fraction), but some cases with the disease will be classified negative (FN = False Negative fraction). On the other hand, some cases without the disease will be correctly classified as negative (TN = True Negative fraction), but some cases without the disease will be classified as positive (FP = False Positive fraction).

The different fractions (TP, FP, TN, FN) are represented in the following table.

	Disease				
Test	**Present**	**n**	**Absent**	**n**	**Total**
Positive	True Positive (TP)	*a*	False Positive (FP)	*c*	*a + c*
Negative	False Negative (FN)	*b*	True Negative (TN)	*d*	*b + d*
Total		*a + b*		*c + d*	

The following statistics can be defined:

Sensitivity	$\frac{a}{a + b}$	**Specificity**	$\frac{d}{c + d}$
Positive Likelihood Ratio	$\frac{\text{Sensitivity}}{1 - \text{Specificity}}$	**Negative Likelihood Ratio**	$\frac{1 - \text{Sensitivity}}{\text{Specificity}}$
Positive Predictive Value	$\frac{a}{a + c}$	**Negative Predictive Value**	$\frac{d}{b + d}$

- *Sensitivity*: probability that a test result will be positive when the disease is present (true positive rate, expressed as a percentage).
= a / (a+b)
- *Specificity*: probability that a test result will be negative when the disease is not present (true negative rate, expressed as a percentage).
= d / (c+d)
- *Positive likelihood ratio*: ratio between the probability of a positive test result given the *presence* of the disease and the probability of a positive test result given the *absence* of the disease, i.e.
= True positive rate / False positive rate = Sensitivity / (1-Specificity)
- *Negative likelihood ratio*: ratio between the probability of a negative test result given the *presence* of the disease and the probability of a negative test result given the *absence* of the disease, i.e.
= False negative rate / True negative rate = (1-Sensitivity) / Specificity
- *Positive predictive value*: probability that the disease is present when the test is positive (expressed as a percentage).
= a / (a+c)
- *Negative predictive value*: probability that the disease is not present when the test is negative (expressed as a percentage).
= d / (b+d)

When you select a higher criterion value, the false positive fraction will decrease with increased specificity but on the other hand the true positive fraction and sensitivity will decrease:

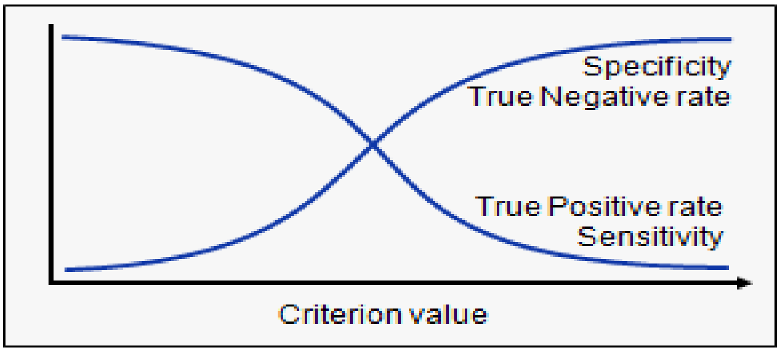


FIG : 6.3

When you select a lower threshold value, then the true positive fraction and sensitivity will increase. On the other hand the false positive fraction will also increase, and therefore the true negative fraction and specificity will decrease.

ROC -:BASIC CONCEPTS

A classification model (classifier or diagnosis) is a mapping of instances between certain classes/groups. The classifier or diagnosis result can be a real value (continuous output), in which case the classifier boundary between classes must be determined by a threshold value (for instance, to determine whether a person has hypertension based on a blood pressure measure). Or it can be a discrete class label, indicating one of the classes.

Let us consider a two-class prediction problem (binary classification), in which the outcomes are labeled either as positive (*p*) or negative (*n*). There are four possible outcomes from a binary classifier. If the outcome from a prediction is *p* and the actual value is also *p*, then it is called a *true positive* (TP); however if the actual value is *n* then it is said to be a *false positive* (FP). Conversely, a *true negative* (TN) has occurred when both the prediction outcome and the actual value are *n*, and *false negative* (FN) is when the prediction outcome is *n* while the actual value is *p*.

To get an appropriate example in a real-world problem, consider a diagnostic test that seeks to determine whether a person has a certain disease. A false positive in this case occurs when the person tests positive, but does not actually have the disease. A false negative, on the other hand, occurs when the person tests negative, suggesting they are healthy, when they actually do have the disease.

Let us define an experiment from **P** positive instances and **N** negative instances for some condition. The four outcomes can be formulated in a 2×2 *contingency table* or *confusion matrix*, as follows:

		True condition				
	Total population	Condition positive	Condition negative	Prevalence = Σ Condition positive/Σ Total population	Accuracy (ACC) = Σ True positive + Σ True negative/Σ Total population	
Predicted condition	Predicted condition positive	**True positive**, Power	**False positive**, Type I error	Positive predictive value (PPV), Precision = Σ True positive/Σ Predicted condition positive	False discovery rate (FDR) = Σ False positive/Σ Predicted condition positive	
	Predicted condition negative	**False negative**, Type II error	**True negative**	False omission rate (FOR) = Σ False negative/Σ Predicted condition negative	Negative predictive value (NPV) = Σ True negative/Σ Predicted condition negative	
		True positive rate (TPR), Recall, Sensitivity, probability of detection = Σ True positive/Σ Condition positive	False positive rate (FPR), Fall-out, probability of false alarm = Σ False positive/Σ Condition negative	Positive likelihood ratio (LR+) = TPR/FPR	Diagnostic odds ratio (DOR) = LR+/LR−	F_1 score = 2/1/Recall + 1/Precision
		False negative rate (FNR), Miss rate = Σ False negative/Σ Condition positive	True negative rate (TNR), Specificity(SPC) = Σ True negative/Σ Condition negative	Negative likelihood ratio (LR−) = FNR/TNR		

TABLE : 2

In a Receiver Operating Characteristic (ROC) curve the true positive rate (Sensitivity) is plotted in function of the false positive rate (100-Specificity) for different cut-off points. Each point on the ROC curve represents a sensitivity/specificity pair corresponding to a particular decision threshold. A test with perfect discrimination (no overlap in the two distributions) has a ROC curve that passes through the upper left corner (100% sensitivity, 100% specificity). Therefore the closer the ROC curve is to the upper left corner, the higher the overall accuracy of the test

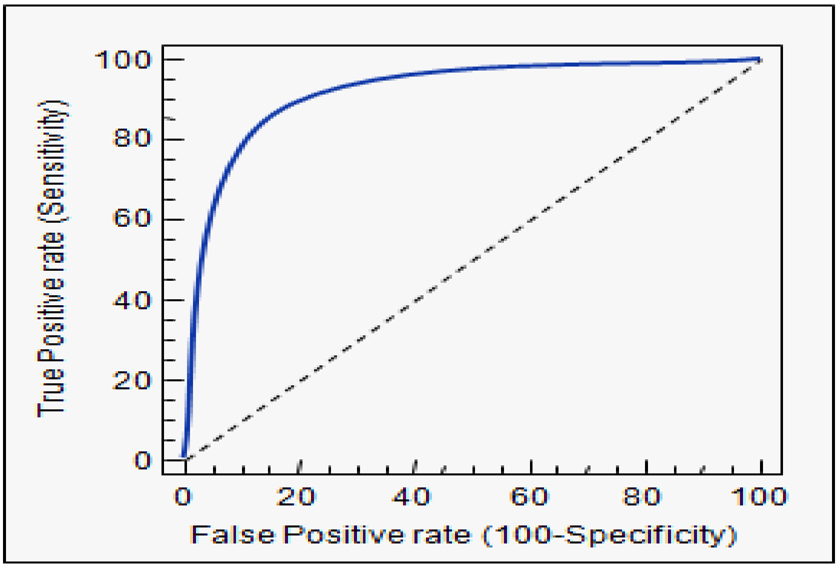
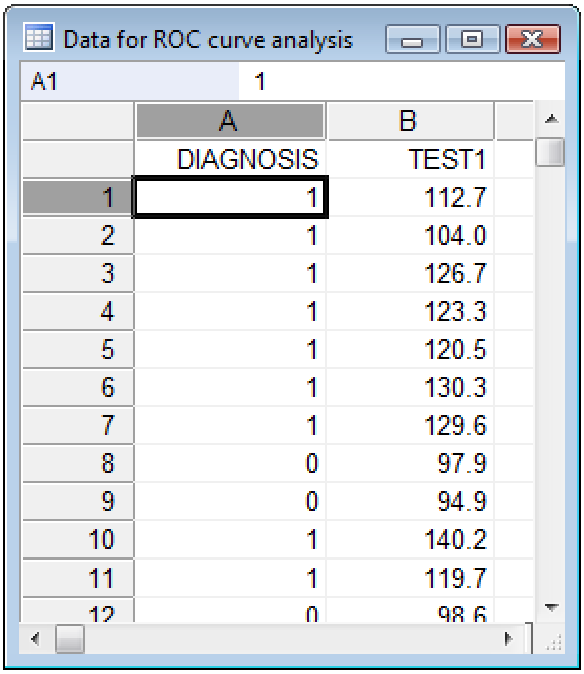


FIG : 6.5

In order to perform ROC curve analysis in MedCalc you should have a measurement of interest (= the parameter you want to study) and an independent diagnosis which classifies your study subjects into two distinct groups: a diseased and non-diseased group. The latter diagnosis should be independent from the measurement of interest.

In the spreadsheet, create a column DIAGNOSIS and a column for the variable of interest, e.g. TEST1. For every study subject enter a code for the diagnosis as follows: 1 for the diseased cases, and 0 for the non-diseased or normal cases. In the TEST1 column, enter the measurement of interest (this can be measurements, grades, etc. - if the data are categorical, code them with numerical values).



	A	B
	DIAGNOSIS	TEST1
1	1	112.7
2	1	104.0
3	1	126.7
4	1	123.3
5	1	120.5
6	1	130.3
7	1	129.6
8	0	97.9
9	0	94.9
10	1	140.2
11	1	119.7
12	0	98.6

FIG : 6.6

Data

- **Variable**: select the variable of interest.
- **Classification variable**: select or enter a a dichotomous variable indicating diagnosis (0=negative, 1=positive).

  If your data are coded differently, you can use the Define status tool to recode your data.

- **Filter**: (optionally) a filter in order to include only a selected subgroup of cases (e.g. AGE>21, SEX="Male").

Methodology:

- **DeLong et al.**: use the method of DeLong et al. (1988) for the calculation of the Standard Error of the Area Under the Curve (recommended).
- **Hanley & McNeil**: use the method of Hanley & McNeil (1982) for the calculation of the Standard Error of the Area Under the Curve.
- **Binomial exact Confidence Interval for the AUC**: calculate an exact Binomial Confidence Interval for the Area Under the Curve (recommended). If this option is not selected, the Confidence Interval is calculated as AUC ± 1.96 its Standard Error.

Disease prevalence

Whereas *sensitivity* and *specificity*, and therefore the ROC curve, and *positive* and *negative likelihood ratio* are independent of disease prevalence, *positive* and *negative predictive values* are highly dependent on disease prevalence or prior probability of disease. Therefore when disease prevalence is unknown, the program cannot calculate *positive* and *negative predictive values*.

Clinically, the disease prevalence is the same as the probability of disease being present before the test is performed (prior probability of disease).

- **Unknown**: select this option when the disease prevalence is unknown, or irrelevant for the current statistical analysis.
- **The ratio of cases in the positive and negative groups reflects the prevalence of the disease**: if the sample sizes in the positive and the negative group reflect the real prevalence of the disease in the population, this can be indicated by selecting this option.
- **Other value (%)**: alternatively you can enter a value for the disease prevalence, expressed as a percentage.

Options

- **List criterion values with test characteristics**: option to create a list of criterion values corresponding with the coordinates of the ROC curve, with associated sensitivity, specificity, likelihood ratios and predictive values (if disease prevalence is known).
  - **Include all observed criterion values**: When you select this option, the program will list sensitivity and specificity for all possible threshold values. If this option is not selected, then the program will only list the more important points of the ROC curve: for equal sensitivity/specificity it will give the threshold values (criterion values) with the highest specificity/sensitivity.
- **95% Confidence Interval** for sensitivity/specificity, likelihood ratio and predictive values: select the Confidence Intervals you require.
- **Calculate optimal criterion value taking into account costs**: option to calculate the optimal criterion value taking into account the disease prevalence and cost of false and true positive and negative decisions (Zweig & Campbell, 1993). This option is only available if disease prevalence is known (see above).
  - FPc: the cost of a false positive decision.
  - FNc: the cost of a false negative decision.
  - TPc: the cost of a true positive decision.
  - TNc: the cost of a true negative decision.

  These data are used to calculate a parameter S as follows:

$$S = \left(\frac{FP_c - TN_c}{FN_c - TP_c}\right) \times \left(\frac{1-P}{P}\right)$$

  where P denotes the prevalence in the target population (Greiner et al., 2000). The point on the ROC curve where a line with this slope S touches the curve is the optimal operating point, taking into account prevalence and the costs of the different decisions.

  Costs can be financial costs or health costs, but all 4 cost factors need to be expressed on a common scale. Benefits can be expressed as negative costs. Suppose a false negative (FN) decision is judged to be twice as costly as a false positive (FP) decision, and no assumptions are made about the costs for true positive and true negative decisions. Then for FNc you enter 2, for FPc enter 1 and enter 0 for both TPc and TNc.

  Because the slope S must be a positive number:

  - FPc cannot be equal to TNc
  - FNc cannot be equal to TPc
  - When TNc is larger than FPc then TPc must be larger than FNc

- When TNc is smaller than FPc then TPc must be smaller than FNc

  The parameter S is "cost-neutral" when (FPc-TNc)/(FNc-TPc) evaluates to 1, that is when FPc-TNc equals FNc-TPc. In this case S, and the "optimal criterion value" depends only on the disease prevalence.

- **Advanced**: click this button for some advanced options:

  These options require bootstrapping and are computationally intensive and time consuming.

  - **Estimation of sensitivity and specificity at fixed specificity and sensitivity**: compile a table with estimation of sensitivity and specificity, with a BC_abootstrapped 95% confidence interval (Efron, 1987; Efron & Tibshirani, 1993), for a fixed and prespecified specificity and sensitivity of 80%, 90%, 95% and 97.5% (Zhou et al., 2002).
  - **Bootstrap Youden index confidence interval**: calculate a BC_a bootstrapped 95% confidence interval for the Youden index and its associated criterion value.
  - **Bootstrap replications**: enter the number of bootstrap replications. 1000 replications is a number commonly encountered in the literature. High numbers increase accuracy but also increase processing time.
  - **Random-number seed**: this is the seed for the random number generator. Enter 0 for a random seed; this can result in different confidence intervals when the procedure is repeated. Any other value will give a repeatable "random" sequence, which will result in repeatable values for the confidence intervals.

ROC graph

- Select **Display ROC curve window** to obtain the graph in a separate window.

  Options:

  - mark points corresponding to criterion values.
  - display 95% Confidence Bounds for the ROC curve
- The prevalence of a disease may be different in different clinical settings. For instance the pre-test probability for a positive test will be higher when a patient consults a specialist than when he consults a general practitioner. Since positive and negative predictive values are sensitive to the prevalence of the disease, it would be misleading to compare these values from different studies where the prevalence of the disease differs, or apply them in different settings.
- The data from the results window can be summarized in a table. The sample size in the two groups should be clearly stated. The table can contain a column for the different criterion values, the corresponding sensitivity (with 95% CI), specificity (with 95% CI), and possibly the positive and negative predictive value. The table should not only contain the test's characteristics for one single cut-off value, but preferably there should be a row for the values corresponding with a sensitivity of 90%, 95% and 97.5%, specificity of 90%, 95% and 97.5%, and the value corresponding with the Youden index or highest accuracy.

In a ROC curve the true positive rate (Sensitivity) is plotted in function of the false positive rate (100-Specificity) for different cut-off points. Each point on the ROC curve represents a sensitivity/specificity pair corresponding to a particular decision threshold. A test with perfect discrimination (no overlap in the two distributions) has a ROC curve that passes through the upper left corner (100% sensitivity, 100% specificity). Therefore the closer the ROC curve is to the upper left corner, the higher the overall accuracy of the test -:

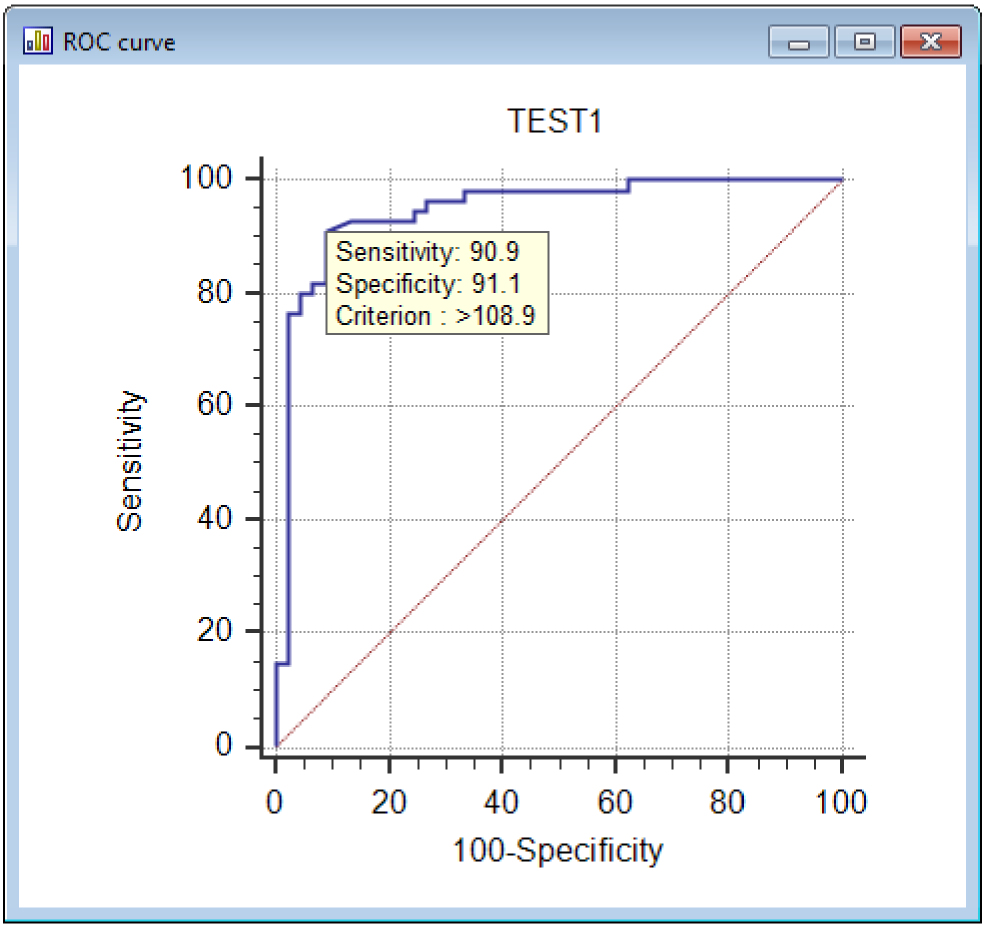


FIG : 6.7

<u>ROC SPACE</u>

The ROC space and plots of the four prediction examples.

The contingency table can derive several evaluation "metrics" (see infobox). To draw a ROC curve, only the true positive rate (TPR) and false positive rate (FPR) are needed (as functions of some classifier parameter). The TPR defines how many correct positive results occur among all positive samples available during the test. FPR, on the other hand, defines how many incorrect positive results occur among all negative samples available during the test.

A ROC space is defined by FPR and TPR as x and y axes, respectively, which depicts relative trade-offs between true positive (benefits) and false positive (costs). Since TPR is equivalent to sensitivity and FPR is equal to 1 − specificity, the ROC graph is sometimes called the sensitivity vs (1 − specificity) plot. Each prediction result or instance of a confusion matrix represents one point in the ROC space.

The best possible prediction method would yield a point in the upper left corner or coordinate (0,1) of the ROC space, representing 100% sensitivity (no false negatives) and 100% specificity (no false positives). The (0,1) point is also called a *perfect classification*. A random guess would give a point along a diagonal line (the so-called *line of no-discrimination*) from the left bottom to the top right corners (regardless of the positive and negative base rates). An intuitive example of random guessing is a decision by flipping coins. As the size of the sample increases, a random classifier's ROC point migrates towards the diagonal line. In the case of a balanced coin, it will migrate to the point (0.5, 0.5).

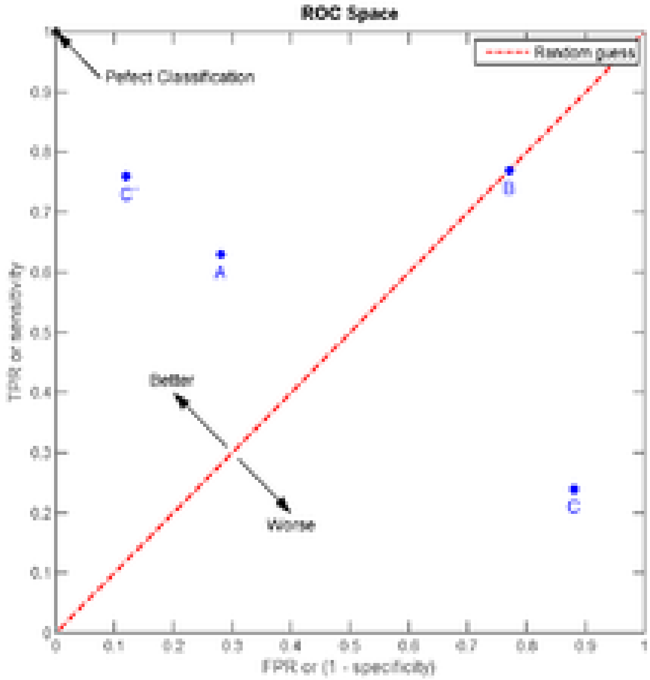


FIG : 6.8

The diagonal divides the ROC space. Points above the diagonal represent good classification results (better than random), points below the line represent poor results (worse than random). Note that the output of a consistently poor predictor could simply be inverted to obtain a good predictor.

Let us look into four prediction results from 100 positive and 100 negative instances:

A	B	C	C′
TP=63 FP=28 91 FN=37 TN=72 109 100 100 200	TP=77 FP=77 154 FN=23 TN=23 46 100 100 200	TP=24 FP=88 112 FN=76 TN=12 88 100 100 200	TP=76 FP=12 88 FN=24 TN=88 112 100 100 200
TPR = 0.63	TPR = 0.77	TPR = 0.24	TPR = 0.76
FPR = 0.28	FPR = 0.77	FPR = 0.88	FPR = 0.12
PPV = 0.69	PPV = 0.50	PPV = 0.21	PPV = 0.86
F1 = 0.66	F1 = 0.61	F1 = 0.22	F1 = 0.81
ACC = 0.68	ACC = 0.50	ACC = 0.18	ACC = 0.82

Plots of the four results above in the ROC space are given in the figure. The result of method **A** clearly shows the best predictive power among **A**, **B**, and **C**. The result of **B** lies on the random guess line (the diagonal line), and it can be seen in the table that the accuracy of **B** is 50%. However, when **C** is mirrored across the center point (0.5,0.5), the resulting method **C′** is even better than **A**. This mirrored method simply reverses the predictions of whatever method or test produced the **C** contingency table. Although the original **C** method has negative predictive power, simply reversing its decisions leads to a new predictive method **C′** which has positive predictive power. When the **C** method predicts **p** or **n**, the **C′** method would predict **n** or **p**, respectively. In this manner, the **C′** test would perform the best. The closer a result from a contingency table is to the upper left corner, the better it predicts, but the distance from the random guess line in either direction is the best indicator of how much predictive power a method has. If the result is below the line (i.e. the method is worse than a random guess), all of the method's predictions must be reversed in order to utilize its power, thereby moving the result above the random guess line.

CURVES IN ROC SPACES

In binary classification, the class prediction for each instance is often made based on a continuous random variable **X** , which is a "score" computed for the instance (e.g. estimated probability in logistic regression). Given a threshold parameter **T** , the instance is classified as "positive" if **X>T** , and "negative" otherwise. **X** follows a probability density **f(x)** if the instance actually belongs to class "positive", and **g(x)** if otherwise. Therefore, the true positive rate is given by **TPR** and the false positive rate is given by **FPR**. The ROC curve plots parametrically TPR(T) versus FPR(T) with T as the varying parameter.

For example, imagine that the blood protein levels in diseased people and healthy people are normally distributed with means of 2 g/dL and 1 g/dL respectively. A medical test might measure the level of a certain protein in a blood sample and classify any number above a certain threshold as indicating disease. The experimenter can adjust the threshold (black vertical line in the figure), which will in turn change the false positive rate. Increasing the threshold would result in fewer false positives (and more false negatives), corresponding to a leftward movement on the curve. The actual shape of the curve is determined by how much overlap the two distributions have. These concepts are demonstrated in the Receiver Operating Characteristic (ROC) Curves Applet.

FURTHER INTERPRETATIONS

Sometimes, the ROC is used to generate a summary statistic. Common versions are:

- the intercept of the ROC curve with the line at 45 degrees orthogonal to the no-discrimination line - the balance point where Sensitivity = Specificity
- the intercept of the ROC curve with the tangent at 45 degrees parallel to the no-discrimination line that is closest to the error-free point (0,1) - also called Youden's J statisticand generalized as Informedness[7]
- the area between the ROC curve and the no-discrimination line - Gini Coefficient
- the area between the full ROC curve and the triangular ROC curve including only (0,0), (1,1) and one selected operating point (tpr,fpr) - Consistency
- the area under the ROC curve, or "AUC" ("Area Under Curve"), or A' (pronounced "a-prime"), or "c-statistic".
- the sensitivity index d' (pronounced "d-prime"), the distance between the mean of the distribution of activity in the system under noise-alone conditions and its distribution under signal-alone conditions, divided by their standard deviation, under the assumption that both these distributions are normal with the same standard deviation. Under these assumptions, the shape of the ROC is entirely determined by d'.

However, any attempt to summarize the ROC curve into a single number loses information about the pattern of tradeoffs of the particular discriminator algorithm.

Area under the curve

When using normalized units, the area under the curve (often referred to as simply the AUC) is equal to the probability that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative one (assuming 'positive' ranks higher than 'negative').

It is also common to calculate the Area Under the ROC Convex Hull (ROC AUCH = ROCH AUC) as any point on the line segment between two prediction results can be achieved by randomly using one or other system with probabilities proportional to the relative length of the opposite component of the segment. Interestingly, it is also possible to invert concavities – just as in the figure the worse solution can be reflected to become a better solution; concavities can be reflected in any line segment, but this more extreme form of fusion is much more likely to overfit the data.

The machine learning community most often uses the ROC AUC statistic for model comparison. However, this practice has recently been questioned based upon new machine learning research that shows that the AUC is quite noisy as a classification measure[18] and has some other significant problems in model comparison. A reliable and valid AUC estimate can be interpreted as the probability that the classifier will assign a higher score to a randomly chosen positive example than to a randomly chosen negative example. However, the critical research suggests frequent failures in obtaining reliable and valid AUC estimates. Thus, the practical value of the AUC measure has been called into question, raising the possibility that the AUC may actually introduce more uncertainty into machine learning classification accuracy comparisons than resolution. Nonetheless, the coherence of AUC as a measure of aggregated classification performance has been vindicated, in terms of a uniform rate distribution, and AUC has been linked to a number of other performance metrics such as the Brier score.

One recent explanation of the problem with ROC AUC is that reducing the ROC Curve to a single number ignores the fact that it is about the tradeoffs between the different systems or performance points plotted and not the performance of an individual system, as well as ignoring the possibility of concavity repair, so that related alternative measures such as Informedness[7] or DeltaP are recommended. These measures are essentially equivalent to the Gini for a single prediction point with DeltaP' = Informedness = 2AUC-1, whilst DeltaP = Markedness represents the dual (viz. predicting the prediction from the real class) and their geometric mean is the Matthews correlation coefficient.[7]

Other measures

Whereas ROC AUC varies between 0 and 1 — with an uninformative classifier yielding 0.5 — the alternative measures Informedness and Gini Coefficient (in the single parameterization or single system case)[7] all have the advantage that 0 represents chance performance whilst 1 represents perfect performance, and −1 represents the "perverse" case of full informedness always giving the wrong response. Bringing chance performance to 0 allows these alternative scales to be interpreted as Kappa statistics. Informedness has been shown to have desirable characteristics for Machine Learning versus other common definitions of Kappa such as Cohen Kappa and Fleiss Kappa.

Sometimes it can be more useful to look at a specific region of the ROC Curve rather than at the whole curve. It is possible to compute partial AUC. For example, one could focus on the region of the curve with low false positive rate, which is often of prime interest for population screening tests. Another common approach for classification problems in which $P \ll N$ (common in bioinformatics applications) is to use a logarithmic scale for the x-axis.

OUR OUTPUT

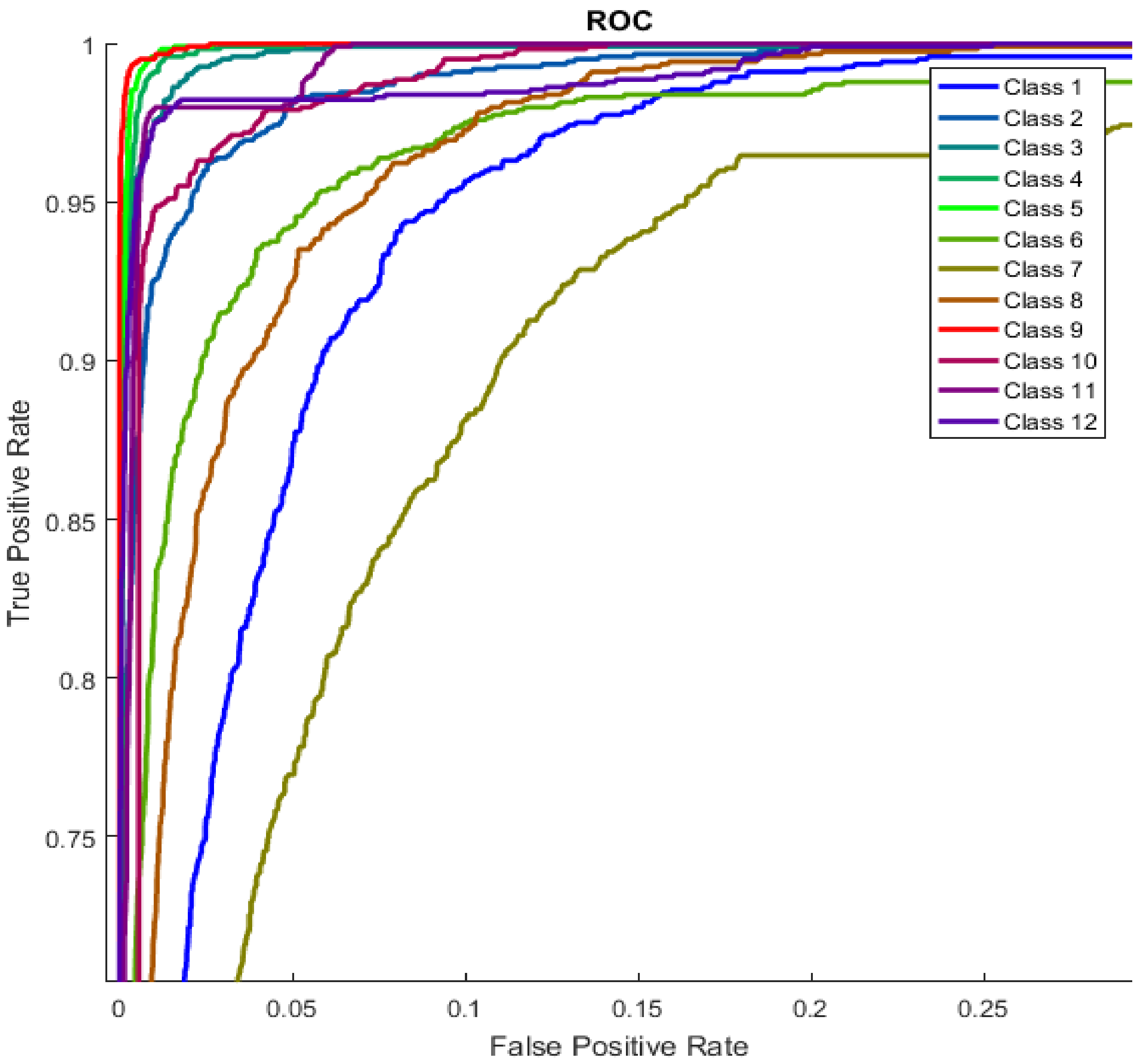


FIG : 6.9

Class seven(factory 1) has least hit point whereas class nine(hf channel) has highest it point ratio. The Receiver Operating Characteristic (ROC) curve is created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings. The true-positive rate is also known as sensitivity, recall or *probability of detection, whereas* the false-positive rate is also known as the fall-out or *probability of false alarm.*

In a ROC curve the true positive rate (Sensitivity) is plotted in function of the false positive rate (100-Specificity) for different cut-off points.. A test with perfect discrimination (no overlap in the two distributions) has a ROC curve that passes through the upper left corner (100% sensitivity, 100% specificity). Therefore the closer the ROC curve is to the upper left corner, the higher the overall accuracy of the test.

REFERENCE

❖ Our Project Guide respected Sir Mr. Sujoy Mondal, Asst. Professor , Dept. of ECE

RCC Institute of Information Technology.

❖ Other resources- Internet.

THANK YOU……….