VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Belagavi – 590 018.

A PROJECT REPORT

on

"BREAST CANCER DETECTION SYSTEM"

Submitted in partial fulfillment of the requirement for the award of the degree

Bachelor of Engineering

in

Computer Science and Engineering

by

ADITHYA SUNDER	**:**	**1VI18CS002**
NITHIN KUMAR B	**:**	**1VI18CS069**
SANTHOSH B RAO	**:**	**1VI18CS094**
SHASHANK P	**:**	**1VI18CS099**

Under the supervision of

Mrs. Mary Vidya John
Assistant Professor

DEPT. OF COMPUTER SCIENCE AND ENGINEERING (NBA Accredited)
VEMANA INSTITUTE OF TECHNOLOGY
BENGALURU – 560 034

2021 - 22

Karnataka ReddyJana Sangha®

VEMANA INSTITUTE OF TECHNOLOGY

(Affiliated to Visvesvaraya Technological University, Belagavi)

Koramangala, Bengaluru-34.

Department of Computer Science and Engineering

Certificate

Certified that the project work entitled "**BREAST CANCER DETECTION SYSTEM**" carried out jointly by **Adithya Sunder (1VI18CS002), Nithin Kumar B (1VI18CS069), Santhosh B Rao (1VI18CS094)** and **Shashank P (1VI18CS099),** are bonafide students **of Vemana Institute of Technology** in partial fulfillment for the award of **Bachelor of Engineering** in **Computer Science and Engineering** of the **Visvesvaraya Technological University, Belagavi** during the year 2021-22. It is certified that all corrections/suggestions indicated for internal assessment have been incorporated in the report. The project report has been approved as it satisfies the academic requirements in respect of the project work prescribed for the said degree.

Supervisor	**HOD**	**Principal**
Mrs. Mary Vidya John	Dr. M. Ramakrishna	Dr. Vijayasimha Reddy. B. G.

Submitted for the university examination (viva-voce) held on …….……

External Viva

Internal Examiner **External Examiner**

1.

2.

DECLARATION

We the undersigned solemnly declare that the project entitled "BREAST CANCER DETECTION SYSTEM" is based on our own work carried out during the course of our study under the supervision of Mrs. Mary Vidya John.

We assert the statements made and conclusions drawn are an outcome of our project work.

We further certify that,

a. The work has not been submitted to any other Institution for any other degree/diploma/certificate in this university or any other University of India or abroad.

b. We have followed the guidelines provided by the university in writing the report.

c. Whenever we have used resources (data, theoretical analysis, and text) from various sources and given due credit to them in the text of the report and their details are provided in the references.

Date:

Place: Bangalore

Project Associates:

USN	Name	Sign.
1VI18CS002	Adithya Sunder	
1VI18CS069	Nithin Kumar B	
1VI18CS094	Santhosh B Rao	
1VI18CS099	Shashank P	

ACKNOWLEDGEMENT

We sincerely thank **Visvesvaraya Technological University** for providing a platform to do the project phase - II.

Firstly, we would like to express our deep sense of gratitude to our institute "**Vemana Institute of Technology**" that provided us an opportunity to do a project phase - II entitled "**Breast Cancer Prediction System**".

We thank **Dr. Vijayasimha Reddy. B. G,** Principal, Vemana Institute of Technology, Bengaluru for providing the necessary support.

We would like to place on record our regards to **Dr. M. Ramakrishna,** Professor and Head, Department of Computer Science and Engineering for his continued support.

We would like to thank our project coordinators **Mrs. A Rosline Mary,** Assistant Professor and **Mrs. Roopalakshmi S,** Assistant Professor, Dept. of CSE for their support and coordination.

We would like to thank our project guide **Mrs. Mary Vidya John,** Assistant Professor, Dept. of CSE for her continuous support and valuable guidance towards successful completion of the project work.

We also thank all the Teaching and Non-teaching staff of Computer Science and Engineering Department, who have helped us to complete the project in time.

ADITHYA SUNDER (1VI18CS060)
NITHIN KUMAR B (1VI18CS069)
SANTHOSH B RAO (1VI18CS094)
SHASHANK P (1VI18CS099)

Date:
Place: Bengaluru

ABSTRACT

There have been several empirical studies addressing breast cancer using machine learning and soft computing techniques. Many claim that their algorithms are faster, easier, or more accurate than others are. This study is based on genetic programming and machine learning algorithms that aim to construct a system to accurately differentiate between benign and malignant breast tumors. The aim of this study was to optimize the learning algorithm. In this context, we applied the genetic programming technique to select the best features and perfect parameter values of the machine learning classifiers. The performance of the proposed method was based on sensitivity, specificity, precision, accuracy, and the roc curves. The present study proves that genetic programming can automatically find the best model by combining feature preprocessing methods and classifier algorithms. It is alarming that 55% of Indians in 2017 were pushed into poverty due to out-of-pocket medical expenses. Data on quality and accreditation of diagnostic establishments in the country have been described as scanty by many surveys conducted. These statistics are damaging considering the pernicious effects of Covid-19. The pandemic has left millions in disarray and the mounting pressure on the healthcare system isn't helping either. The population has succumbed to the fear of contracting the virus and many people make false assumptions based on their symptoms. Our goal is to get rid of these problems by attacking one major part of healthcare, diagnosis.

Keywords: Breast Cancer, Predictive Models, ROC Curves

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS AND ABBREVIATIONS

ANN	:	Artificial Neural Network
CNN	:	Convolutional Neural Network
COD	:	Cause of Death
EDA	:	Exploratory Data Analysis
PCA	:	Principal Component Analysis
SVM	:	Support Vector Machine
TIR	:	Thermal Infrared Ranging
VSC	:	Viral Status Code

CHAPTER 1

INTRODUCTION

1.1 Introduction

Breast cancer is a prevalent cause of death, and it is the only type of cancer that is widespread among women worldwide. Many imaging techniques have been developed for early detection and treatment of breast cancer and to reduce the number of deaths , and many aided breast cancer diagnosis methods have been used to increase the diagnostic accuracy . In the last few decades, several data mining and machine learning techniques have been developed for breast cancer detection and classification , which can be divided into three main stages: pre-processing, feature extraction, and classification.

To facilitate interpretation and analysis, the pre-processing of mammography films helps improve the visibility of peripheral areas and intensity distribution, and several methods have been reported to assist in this process. Feature extraction is an important step in breast cancer detection because it helps discriminate between benign and malignant tumours. After extraction, image properties such as smoothness, coarseness, depth, and regularity are extracted by segmentation.

Various transform-based texture analysis techniques are applied to convert the image into a new form using the spatial frequency properties of the pixel intensity variations. Screening programs, better treatments and a general increase in awareness have resulted in declining death rates from breast cancer over the past two decades. However, in developed countries, the disease remains the second leading cause of cancer death in women after lung cancer.

The chance of developing invasive breast cancer at some time in a woman’s life is approximately 1 in 8, and the chance that breast cancer will be responsible for a woman’s death is about 1 in 35.Effective management of this relatively common disease can therefore have a significant impact on survival at the population level, and have a profound effect on lives of those directly affected by the disease and their loved ones. Breast Cancer Management addresses key issues in disease management by exploring the best patient-centered clinical research and presenting this information both directly, as clinical findings,

and in practice-oriented formats of direct relevance in the clinic. Significant advances in basic and translational research, and places them in context for future therapy. Breast Cancer Management provides oncologists and other health professionals with the latest findings and opinions on reducing the burden of this widespread disease.

Recent research findings and advances clinical practice in the field are reported and analyzed by international experts. The journal presents this information in clear, accessible formats. All articles are subject to independent review by a minimum of two independent experts.

Breast cancer is not a transmissible or infectious disease. Unlike some cancers that have infection-related causes, such as human papillomavirus (HPV) infection and cervical cancer, there are no known viral or bacterial infections linked to the development of breast cancer.

Approximately half of breast cancers develop in women who have no identifiable breast cancer risk factor other than gender (female) and age (over 40 years). Certain factors increase the risk of breast cancer including increasing age, obesity, harmful use of alcohol, family history of breast cancer, history of radiation exposure, reproductive history (such as age that menstrual periods began and age at first pregnancy), tobacco use and postmenopausal hormone therapy.

Unfortunately, even if all of the potentially modifiable risk factors could be controlled, this would only reduce the risk of developing breast cancer by at most 30%.Female gender is the strongest breast cancer risk factor. Approximately 0.5-1% of breast cancers occur in men. The treatment of breast cancer in men follows the same principles of management as for women.Family history of breast cancer increases the risk of breast cancer, but the majority of women diagnosed with breast cancer do not have a known family history of the disease. Lack of a known family history does not necessarily mean that a woman is at reduced risk. Certain inherited “high penetrance” gene mutations greatly increase breast cancer risk, the most dominant being mutations in the genes BRCA1, BRCA2 and PALB-2. Women found to have mutations in these major genes could consider risk reduction strategies such as surgical removal of both breasts. Consideration of such a highly invasive approach only

concerns a very limited number of women, should be carefully evaluated considering all alternatives and should not be rushed.

Breast cancer most commonly presents as a painless lump or thickening in the breast.
It is important that women finding an abnormal lump in the breast consult a health practitioner without a delay of more than 1-2 months even when there is no pain associated with it. Seeking medical attention at the first sign of a potential symptom allows for more successful treatment. There are many reasons for lumps to develop in the breast, most of which are not cancer. As many as 90% of breast masses are not cancerous. Non-cancerous breast abnormalities include benign masses like fibroadenomas and cysts as well as infections.

Breast cancer can present in a wide variety of ways, which is why a complete medical examination is important. Women with persistent abnormalities (generally lasting more than one month) should undergo tests including imaging of the breast and in some cases tissue sampling (biopsy) to determine if a mass is malignant (cancerous) or benign. Advanced cancers can erode through the skin to cause open sores (ulceration) but are not necessarily painful. Women with breast wounds that do not heal should have a biopsy performed.

Breast cancer treatment can be highly effective, achieving survival probabilities of 90% or higher, particularly when the disease is identified early. Treatment generally consists of surgery and radiation therapy for control of the disease in the breast, lymph nodes and surrounding areas (locoregional control) and systemic therapy (anti-cancer medicines given by mouth or intravenously) to treat and/or reduce the risk of the cancer spreading (metastasis). Anti-cancer medicines include endocrine (hormone) therapy, chemotherapy and in some cases targeted biologic therapy (antibodies).

In the past, all breast cancers were treated surgically by mastectomy (complete removal of the breast). When cancers are large, mastectomy may still be required. Today, the majority of breast cancers can be treated with a smaller procedure called a “lumpectomy” or partial mastectomy, in which only the tumor is removed from the breast. In these cases, radiation therapy to the breast is generally required to minimize the chances of recurrence in the breast.

Lymph nodes are removed at the time of cancer surgery for invasive cancers. Complete removal of the lymph node bed under the arm (complete axillary dissection) in the past was thought to be necessary to prevent the spread of cancer. A smaller lymph node procedures called “sentinel node biopsy” is now preferred as it has fewer complications. It uses dye and/or a radioactive tracer to find the first few lymph nodes to which cancer could spread from the breast. When cancers are large, mastectomy may still be required. Today, the majority of breast cancers can be treated with a smaller procedure called lupectomy. In these cases, radiation therapy to the breast is generally required to minimize the chances of recurrence in the breast.

Medical treatments for breast cancers, which may be given before (“neoadjuvant”) or after (“adjuvant”) surgery, is based on the biological subtyping of the cancers. Cancer that express the estrogen receptor (ER) and/or progesterone receptor (PR) are likely to respond to endocrine (hormone) therapies such as tamoxifen or aromatase inhibitors. These medicines are taken orally for 5-10 years, and reduce the chance of recurrence of these “hormone-positive” cancers by nearly half. Endocrine therapies can cause symptoms of menopause but are generally well tolerated.

Cancers that do not express ER or PR are “hormone receptor negative” and need to be treated with chemotherapy unless the cancer is very small. The chemotherapy regimens available today are very effective in reducing the chances of cancer spread or recurrence and are generally given as outpatient therapy. Chemotherapy for breast cancer generally does not require hospital admission in the absence of complications.

Breast cancers may independently overexpress a molecule called the HER-2/neu oncogene. These “HER-2 positive” cancers are amenable to treatment with targeted biological agents such as trastuzumab. These biological agents are very effective but also very expensive, because they are antibodies rather than chemicals. When targeted biological therapies are given, they are combined with chemotherapy to make them effective at killing cancer cells.

Radiotherapy also plays a very important role in treating breast cancer. With early stage breast cancers, radiation can prevent a woman having to undergo a mastectomy. With later

stage cancers, radiotherapy can reduce cancer recurrence risk even when a mastectomy has been performed. For advanced stage of breast cancer, in some circumstances, radiation therapy may reduce the likelihood of dying of the disease. The effectiveness of breast cancer therapies depends on the full course of treatment. Partial treatment is less likely to lead to a positive outcome.

1.2 Scope

In 2020, there were 2.3 million women diagnosed with breast cancer and 685 000 deaths globally. As of the end of 2020, there were 7.8 million women alive who were diagnosed with breast cancer in the past 5 years, making it the world's most prevalent cancer. There are more lost disability-adjusted life years (DALYs) by women to breast cancer globally than any other type of cancer. Breast cancer occurs in every country of the world in women at any age after puberty but with increasing rates in later life.

Breast cancer mortality changed little from the 1930s through to the 1970s. Improvements in survival began in the 1980s in countries with early detection programmes combined with different modes of treatment to eradicate invasive disease. Age-standardized breast cancer mortality in high-income countries dropped by 40% between the 1980s and 2020. Countries that have succeeded in reducing breast cancer mortality have been able to achieve an annual breast cancer mortality reduction of 2-4% per year. If an annual mortality reduction of 2.5% per year occurs worldwide, 2.5 million breast cancer deaths would be avoided between 2020 and 2040.

The strategies for improving breast cancer outcomes depend on fundamental health system strengthening to deliver the treatments that are already known to work. These are also important for the management of other cancers and other non-malignant noncommunicable diseases (NCDs). For example, having reliable referral pathways from primary care facilities to district hospitals to dedicated cancer centres.The establishment of reliable referral pathways from primary care facilities to district hospitals to dedicated cancer centers is the same approach as is required for the management of cervical cancer, lung cancer, colorectal cancer and prostate cancer. To

that end, breast cancer is an “index” disease whereby pathways are created that can be followed for the management of other diseases.

The three pillars toward achieving these objectives are: health promotion for early detection; timely diagnosis; and comprehensive breast cancer management.

By providing public health education to improve awareness among women of the signs and symptoms of breast cancer and, together with their families, understand the importance of early detection and treatment.

1.3 Objectives

This analysis aims to observe which features are most helpful in predicting malignant or benign cancer and to see general trends that may aid us in model selection and hyper parameter selection. The goal is to classify whether the breast cancer is benign or malignant. To achieve this I have used machine learning classification methods to fit a function that can predict the discrete class of new input.

1.4 Organization of the project work

Table 1.1: Project Phase-1 Organization

Task	October	November	December	January	February
Stage 1					
Stage 2					
Stage 3					
Stage 4					
Stage 5					

Table 1.1 shows the progress of our first phase of our project.

Stage 1 – This is the initial stage of analyzing what is the required in the real world, the problems faced by patients, to draft a problem statement that will be helpful for the hospitals. The team identified the metrics to perform on.

Stage 2 – In this stage, the team identified the software requirements for the project, drafted a plan and collected 4 reference papers to gain information that is required to start the project.

Stage 3- In this stage, the team gained knowledge on neural networks and how to use them. We learnt about attacks, defenses, and the utility metrics for each.

Stage 4 - The phase 1 progress was proposed to the college and reviewed by the panelists.

Stage 5 – In this stage, we started with the documentation work and drafted the report for project phase-1.

Table 1.2: Project Phase-2 Organization

Task	March	April	May	June
Stage 1				
Stage 2				
Stage 3				
Stage 4				

Table 1.2 shows the progress of the second phase of the project. It has our work cut out over 4 months. It has been divided into 4 stages.

Stage 1 This is the part where we decide on the predictive models and methods.

Stage 2 – In this stage, the team identified the algorithms to be used for the project, drafted a plan and collected 4 reference papers to gain information that is required to compare algorithms.

Stage 3- In this stage, the team gained knowledge on voting classifiers and how to use it.

Stage 4 - The phase 2 progress was proposed to the college and reviewed by the panelists.

CHAPTER 2
LITERATURE SURVEY

2.1 Title: Asymmetry Analysis in Breast Cancer Detection Using Thermal Infrared Images[Phani Teja Kuruganti, Hairong Qi]

Methodology**:** This paper discusses an automated approach for breast cancer detection using Thermal Infrared(TIR) images. Breast cancer is a disease in which only the early diagnosis increases the survival hope.

Thermal imaging is a very powerful remote sensing technique for a number of reasons, particularly when used to elucidate field studies relating to animal ecology. Thermal imaging data is collected at the speed of light in real time from a wide variety of platforms, including land, water, and air-based vehicles. It is superior to visible imaging technologies because thermal radiation can penetrate smokes, aerosols, dust, and mists more effectively than visible radiation so that animals can be detected over a wide range of normally troublesome atmospheric conditions.

It is a completely passive technique capable of imaging under both daytime and night-time conditions. This minimizes disruptions and stressful disturbances to wildlife during data collection activities. It is capable of detecting animals which are colder, warmer, or the same as their background temperature because it does not compare temperatures but rather the emissivity of the animal against its background.

While the emphasis is on the counting and observation of wildlife there are other very important applications where remote sensing via thermal imaging can be of use. For example, using thermal imagers in aerial surveys of the landscape for mapping purposes can provide some unique capabilities that cannot be gained any other way. From aircraft heights and at aircraft speeds there are no fundamental problems in achieving ground resolutions down to a fraction of a meter.

The main advantage of thermal images over visible aerial photography is that they can sense heat. For example, soil types that are absorbing differing amounts of solar radiation can be mapped as well as shading effects on north/south facing slopes on hilly or

mountainous terrain. Shading can also be used to help map features of dry washes, forest edges, fence lines, agriculture fields, drainage ditches, variations in soil moisture, and evaporation and even to determine wind direction in many cases.

It is interesting to note that Quattrochi and Luvall (1999) identify a similar reluctance on the part of remote sensing scientists to adopt the powerful resources offered by thermal imaging as do we on the part of wildlife scientists engaged in studying and monitoring wildlife populations. Although numerous articles have appeared in the professional literature that have employed thermal infrared (TIR) data for the use in studying specific aspects of landscape-related processes (e.g., evapotranspiration), the direct application of TIR data for assessment of landscape processes and patterns within a landscape ecological purview is lacking.

They argue that the use of TIR data from airborne and satellite sensors could be very useful for parameterizing surface moisture conditions and developing better simulations of landscape energy exchange over a variety of conditions and space and time scales. They postulate that TIR remote sensing data can significantly contribute to the observation, measurement, and analysis of energy balance characteristics (i.e., the fluxes and redistribution of thermal energy within and across the land surface) as an implicit and important aspect of landscape dynamics and landscape function.

There are three primary reasons for the lack of enthusiasm to use TIR remote sensing data for landscape ecological studies. First, TIR data are little understood from both a theoretical and applications perspective within the landscape ecological community. Second, TIR data are perceived as being difficult to obtain and work with to those researchers who are uninitiated to the characteristics and attributes of these data for applications in landscape ecological research. Although numerous articles have appeared in the professional literature that have employed thermal infrared (TIR) data for the use in studying specific aspects of landscape-related processes.

This process is very important as it can make our product much better and take us ahead.

The literature survey helps us in finding out new problems faced by researchers so we can implement them later in the project. This helps us find out the needs of the industry sooner.

Finally, the spatial resolution of TIR data, primarily from satellites, is viewed as being too coarse for landscape ecological applications (e.g., Landsat Thermatic Mapper data at 120 m spatial resolution) and calibration of these data for deriving measurements of landscape thermal energy fluxes is seen as problematic. Interestingly, these reasons are very similar to those given for the limited use of thermal imagers by wildlife scientists in the preface of this book. Quattrochi and Luvall (1999) proposed ways to overcome these misconceptions regarding the use of TIR remote sensing data in landscape ecological research by providing supporting evidence from a sampling of work that has employed TIR remote sensing data for analysis of landscape characteristics.

Several problems must be addressed when considering the design of automated thermal imaging detection applications. Thermal imagers collect radiation from animals and their background, and if there are sufficient differences in the apparent temperatures between the two then quality imagery can be obtained. This imagery can contain sufficient information to count and identify a large number of species and, in many cases, the user is able to make accurate evaluations regarding the activity, age, sex, and physical condition of the animal. These evaluations are not automatically determined and require the detailed examination of the thermographer.

This being the case, we assume that there is adequate information obtained from the imagery, but the data processing was carried out in a human brain. The complexity and difficulty of processing thermal images so that robots can see well enough to make decisions in outdoor surroundings is truly a difficult thing to even imagine. Going from the raw imagery to just counting the number of individual animals in the imagery has taken a significant effort on the part of many researchers and field scientists.

It has been realized that the only way to count large numbers of animals during a short period of time is through automating the data extraction from the imagery. The basic problem that needs to be solved is how the number of animals in the collected imagery can be counted so that none are missed. We reviewed a number of efforts in Chapter 10 that used digital image processing, computer vision analysis, superposition of detection and tracking algorithms, and automated motion detection to solve some of these problems.

We can have a new aspect to our project considering the metrics mentioned above.

The application of thermal infrared (TIR) imaging in breast cancer study started as early as 1961. However, it has not been widely recognized due to the premature use of the technology, the superficial understanding of the infrared (IR) images, and its poorly controlled introduction into breast cancer detection in the 1970s. Interestingly, these reasons are very similar to those given for the limited use of thermal imagers by wildlife scientists in the preface. Quattrochi and Luvall (1999) proposed ways to overcome these misconceptions regarding the use of TIR remote sensing data in landscape ecological research by providing supporting evidence from a sampling of work that has employed TIR remote sensing data for analysis of landscape characteristics.

Advantage: False positive cases are reduced.
Disadvantage: Chances of activation function are higher, which reduces accuracy

2.2 Title: Review of Electromagnetic Techniques for Breast Cancer Detection[Ahmed M. Hassan, Magda El-Shenawee]

Methodology: Numerous electromagnetic techniques used for detection of disease have been studied and the feasibility has been examined. Techniques like Bio magnetic Detection and Magnetic Resonance Techniques have been studied and clinical records are examined.

An MRI (magnetic resonance imaging) scan is a painless test that produces very clear images of the organs and structures inside your body. MRI uses a large magnet, radio waves and a computer to produce these detailed images. It doesn't use X-rays (radiation).

Because MRI doesn't use X-rays or other radiation, it's the imaging test of choice when people will need frequent imaging for diagnosis or treatment monitoring, especially of their brain. An open (or "open bore") MRI refers to the type of machine that takes the images. Typically, an open MRI machine has two flat magnets positioned over and under you with a large space between them for you to lie. This allows for open space on two sides and alleviates much of the claustrophobia many people experience with closed-bore MRI machines. Although numerous articles have appeared in the professional literature that have employed thermal infrared (TIR) data for the use in studying specific aspects of landscape-related processes

MRIs do have a drawback as seen here. There are many issues when it comes to imaging.

However, open MRIs don't take as clear images as closed-bore MRI machines. Closed-bore MRI machines have a ring of magnets that forms an open hole or tube in the middle where you'd lie to get the images. Closed-bore MRIs are narrow with tight head-to-ceiling space. This can cause anxiety and discomfort for some people, but these MRI machines take the best quality images. Magnetic resonance imaging (MRI) uses magnets, radio waves and a computer to create images of the inside of your body, whereas computed tomography (CT) uses X-rays and computers.

Healthcare providers often prefer to use MRI scans instead of CT scans to look at the non-bony parts or soft tissues inside your body. MRI scans are also safer since they don't use the damaging ionizing radiation of X-rays.MRI scans also take much clearer pictures of your brain, spinal cord, nerves, muscles, ligaments and tendons than regular X-rays and CT scans.However, not everyone can undergo an MRI. The magnetic field of MRI can displace metal implants or affect the function of devices such as pacemakers and insulin pumps. If this is the case, a CT scan is the next best option. MRI scanning is usually more expensive than X-ray imaging or CT scanning. Magnetic resonance imaging (MRI) produces detailed images of the inside of your body. Healthcare providers can "look at" and evaluate several different structures inside your body using MRI. Healthcare providers use magnetic resonance imaging (MRI) to help diagnose or monitor the treatment for many different conditions. There are also different types of MRIs based on which area of your body your provider wants to examine.

Advantage: High potential to be best predicted system
Disadvantage: Can't be implemented in a wide scale yet

2.3 Title: Comparison of data mining classification algorithms for breast cancer prediction[Mr. Chintan Shah , Dr. Anjali Jivani]

Methodology: Three different data mining classification methods for prediction of breast cancer. Different parameters have been compared to come to this conclusion. The algorithms used are Decision Tree, K-nearest Neighbor and Bayes Classification.
Data Mining Algorithms are a particular category of algorithms useful for analyzing data and developing data models to identify meaningful patterns. These are part of machine learning algorithms. These algorithms are implemented through various programming like R

language, Python, and data mining tools to derive the optimized data models. Some of the popular data mining algorithms are C4.5 for decision trees, K-means for cluster data analysis, Naive Bayes Algorithm, Support Vector Mechanism Algorithms,

The Apriori algorithm for time series data mining. These algorithms are part of data analytics implementation for business. These algorithms are based upon statistical and mathematical formulas which applied to the data set.

1. C4.5 Algorithm

Some constructs are used by classifiers which are tools in data mining. These systems take inputs from a collection of cases where each case belongs to one of the small numbers of classes and are described by its values for a fixed set of attributes. The output classifier can accurately predict the level to which it belongs. It uses decision trees where the first initial tree is acquired by using a divide and conquer algorithm.

Suppose S is a class and the tree is leaf labelled with the most frequent type in S. Choosing a test based on a single attribute with two or more outcomes than making this test as root one branch for each work of the test can be used. The partitions correspond to subsets S1, S2, etc., which are outcomes for each case. C4.5 allows for multiple products. C4.5 has introduced an alternative formula in thorny decision trees, which consists of a list of rules, where these rules are grouped for each class. To classify the case, the first class whose conditions are satisfied is named as the first one. If the patient meets no power, then it is assigned a default class. The C4.5 rulesets are formed from the initial decision tree. C4.5 enhances the scalability by multi-threading.

2. The k-means Algorithm

This algorithm is a simple method of partitioning a given data set into the user-specified number of clusters. This algorithm works on d-dimensional vectors, $D=\{xi \mid i= 1, \ldots N\}$ where i is the data point. To get these initial data seeds, the data has to be sampled at random. This sets the solution of clustering a small subset of data, the global mean of data k times. This algorithm can be paired with another algorithm to describe non-convex clusters. It creates k groups from the given set of objects. It explores the entire data set with its cluster

analysis. It is simple and faster than other algorithms when it is used with different algorithms. This algorithm is mostly classified as semi-supervised. Along with specifying the number of clusters, it also keeps learning without any information. It observes the group and learns.

3. Naive Bayes Algorithm

This algorithm is based on Bayes theorem. This algorithm is mainly used when the dimensionality of inputs is high. This classifier can easily calculate the next possible output. New raw data can be added during the runtime, and it provides a better probabilistic classifier. Each class has a known set of vectors that aim to create a rule that allows the objects to be assigned to classes in the future. The vectors of variables describe the future things. This is one of the most comfortable algorithms as it is easy to construct and does not have any complicated parameter estimation schemas. It can be easily applied to massive data sets as well. It does not need any elaborate iterative parameter estimation schemes, and hence unskilled users can understand why the classifications are made.

4. Support Vector Machines Algorithm

If a user wants robust and accurate methods, then Support Vector machines algorithm must be tried. SVMs are mainly used for learning classification, regression or ranking function. It is formed based on structural risk minimization and statistical learning theory. The decision boundaries must be identified, which is known as a hyperplane. It helps in the optimal separation of classes. The main job of SVM is to identify the maximizing the margin between two types. The margin is defined as the amount of space between two types. A hyperplane function is like an equation for the line, y= MX + b. SVM can be extended to perform numerical calculations as well. SVM makes use of kernel so that it operates well in higher dimensions. This is a supervised algorithm, and the data set is used first to let SVM know about all the classes. Once this is done then, SVM can be capable of classifying this new data.

5. The Apriori Algorithm

The Apriori algorithm is widely used to find the frequent itemsets from a transaction data set and derive association rules. To find frequent itemsets is not difficult because of its combinatorial explosion. Once we get the frequent itemsets, it is clear to generate association

rules for larger or equal specified minimum confidence. Apriori is an algorithm which helps in finding routine data sets by making use of candidate generation. It assumes that the item set or the items present are sorted in lexicographic order. After the introduction of Apriori data mining research has been specifically boosted.

Advantage: Low computation time

Disadvantage: More superior prediction can be achieved. Full potential is not yet reached

2.4 Title: Breast Cancer Prediction Analysis using Machine Learning Algorithms[Vinayak A Telsang,Kavyashree Hegde]

Methodology: Dataset of breast cancer is taken. During the preprocessing stage, attributes, targets and normalization is done. Correlation matrix is plotted.After looking at the positive and negative correlated features, linear kernel function is used for prediction.

Random Forest: Random Forest is perhaps the most popular classification algorithm, capable of both classification and regression. It can accurately classify large volumes of data.

The name “Random Forest” is derived from the fact that the algorithm is a combination of decision trees. Each tree depends on the values of a random vector sampled independently with the same distribution for all trees in the “forest.” Each one is grown to the largest extent possible. Correlation matrix is plotted.After looking at the positive and negative correlated features, linear kernel function is used for prediction.

Predictive analytics algorithms try to achieve the lowest error possible by either using “boosting” (a technique which adjusts the weight of an observation based on the last classification) or “bagging” (which creates subsets of data from training samples, chosen randomly with replacement). Random Forest uses bagging. If you have a lot of sample data, instead of training with all of them, you can take a subset and train on that, and take another subset and train on that (overlap is allowed). All of this can be done in parallel. Multiple samples are taken from your data to create an average.

K-Means: A highly popular, high-speed algorithm, K-means involves placing unlabeled data points in separate groups based on similarities. This algorithm is used for the clustering model. For example, Tom and Rebecca are in group one and John and Henry are in group

two. Tom and Rebecca have very similar characteristics but Rebecca and John have very different characteristics. K-means tries to figure out what the common characteristics are for individuals and groups them together. This is particularly helpful when you have a large data set and are looking to implement a personalized plan—this is very difficult to do with one million people.

In the context of predictive analytics for healthcare, a sample size of patients might be placed into five separate clusters by the algorithm. One particular group shares multiple characteristics: they don't exercise, they have an increasing hospital attendance record (three times one year and then ten times the next year), and they are all at risk for diabetes. Based on the similarities, we can proactively recommend a diet and exercise plan for this group.

CNN:Neural Networks are used in various classification tasks like image, audio, words. Different types of Neural Networks are used for different purposes, for example for predicting the sequence of words we use Recurrent Neural Networks more precisely an LSTM, similarly for image classification we use Convolution Neural networks. In this blog, we are going to build a basic building block for CNN.

Before diving into the Convolution Neural Network, let us first revisit some concepts of Neural Network. In a regular Neural Network there are three types of layers:

Input Layers: It's the layer in which we give input to our model. The number of neurons in this layer is equal to the total number of features in our data (number of pixels in the case of an image).

Hidden Layer: The input from the Input layer is then feed into the hidden layer. There can be many hidden layers depending upon our model and data size. Each hidden layer can have different numbers of neurons which are generally greater than the number of features. The output from each layer is computed by matrix multiplication of output of the previous layer with learnable weights of that layer and then by the addition of learnable biases followed by activation function which makes the network nonlinear. Output Layer: The output from the hidden layer is then fed into a logistic function like sigmoid or softmax which converts the output of each class into the probability score of each class.

Advantage: Tests multiple algorithms. All of them have an accuracy of over 90%.

Disadvantage: Detailed analysis should be done before implementing on a large scale .

2.1 Comparative Analysis

Table 2.1: Comparative Analysis

Reference	Algorithm/ Technique	Platform used	Performance Metrics	Advantage	Drawback
[1]	CNN	MATLAB	Time consumption, accuracy, space consumption	Less time consumption and more accuracy	Not enough research ongoing about graphic processing
[2]	KNN,Naïve Bayes,LR,SVM,RF	Web	Precision, Recall,F-1 Score	Tests multiple algorithms. All of them have an accuracy of over 90%.	Detailed analysis should be done before implementing on a large scale
[3]	KNN,Naïve Bayes,LR,SVM,RF,DT	Web	Precision,Sensitivity,Specificity, F-1 score	All algorithms have been tested and highest accuracy is from the SVM with 97% accuracy	Since the highest accuracy is 97%, more work has to be done to implement this in a larger scale.
[4]	KNN,Naïve Bayes,LR,SVM,RF,DT,.Ensemble algorithms	MATLAB,SPYDER	Acuuracy	Effective way of prediction of malignant tumors	Dataset issues need solving.

Table 2.1 shows the comparative analysis of the literature papers we have considered. They vary from different algorithms used for the prediction of breast cancer. Most of these papers have comparable accuracies when it comes to prediction. It must however, account for the disadvantages each paper has. Accuracy isn't the major aspect here. It's got to do with how implementable it is. They have different drawbacks ranging from dataset issues to research deficiencies in major areas.

CHAPTER 3

SYSTEM ANALYSIS

3.1 Existing System

There has been extensive research done in the field of breast cancer prediction. Researchers have proposed solutions to detect malignant and benign tumours. The model developed provides high accuracies but it must however, account for the disadvantages each paper has. Accuracy isn't the major aspect here. It's got to do with how implementable it is. They have different drawbacks ranging from dataset issues to research deficiencies in major areas.

3.1.1 Drawbacks

Doctors cannot always tell if a cancer will go on to be life-threatening or not. So treatment is always offered if you're diagnosed with breast cancer.This means some cancers that are diagnosed and treated would not have been life-threatening. Treatment of non life-threatening cancers is the main risk of breast screening.

Other risks of breast screening include:

- Cancer being missed – mammograms do not always find a cancer that is there
- X-rays – having a mammogram every 3 years for 20 years gives you a very slightly higher chance of getting cancer over your lifetime
- Most people feel the benefits of breast screening outweigh the possible risks.

3.2 Proposed System

The proposed system is a robustness analyzer that provides a web application as an interface to communicate with the users. It is a platform where users can test how robust their model is against adversarial attacks. These attacks can make the user model misclassify the data and thus leading to drop the accuracy of the model. The proposed system works for a face recognition application. It takes the dataset for the application, performs 4 different attacks on it, generates adversarial samples. Then, the original user model is fed with the adversarial samples and the accuracy is calculated. Here, it is observed how the accuracy

of the original model and the new accuracy varies. This makes the accuracy drop to a certain level. Once the attacks are applied and accuracy is found. The defense module is applied. In the proposed system, adversarial training is used to provide defenses. This is one of the many defense modules that are present. Using adversarial training we shall train the original model on the adversarial samples to make it robust enough to classify the noised data correctly. Once this is completed, we perform evaluation of the model. We compare the results of the model after attack and after defense modules are applied and represent it in a form of a graph so that it is easier for the users to visualize the difference and how robust the model is.

3.3 Feasibility Study

The feasibility study explored the technological needs, the benefits, and the long-term viability of the proposed system, Breast Cancer Detection System, to explore if this platform would help users to test and evaluate the accuracy of the model against attest datasets. The feasibility study demonstrated that the project was viable, paving the way to enacting the development plans of the proposed system. Without conducting a feasibility study it is impossible to understand how it can be implemented. This feasibility study aims to objectively uncover the strengths of the system developed.

3.3.1 Technical Feasibility

The technical feasibility examines the functional, non-functional, software and hardware requirements of the proposed system. The software requirements for the proposed solution includes an operating system [Windows 8 or higher], a platform to develop [Jupyter Notebook], Python language. The hardware requirements include a computer having a RAM of 2 GB or higher and a CPU of Intel i3 processor 10th generation or higher. Non-Functional requirements define desired qualities of the system to be developed and often influence the system architecture more than functional requirements do. The Non-Functional requirement of the project is to meet the accuracy, recall and precision. The system is proposed to be reliable. The proposed system must be compatible with different browsers. As the user interface for the project will be a web application, it must be built to be user friendly, and the operations can be well understood by the user. A functional

requirement is a description of the service that the software must offer. The functional requirements of the proposed system include:

In conclusion, after a thorough study it is noted that the proposed system is technically feasible as the resources required to develop are available and minimal. The proposed system must be compatible with different browsers. As the user interface for the project will be a web application.

3.3.2 Operational Feasibility

The operational feasibility examines how the project plan satisfies the needs and the objectives by solving the problem or not. Machine learning has been successfully applied to a wide variety of fields ranging from information retrieval, data mining, and speech recognition, to computer graphics, visualization, and human–computer interaction. However, most users often treat a machine learning model as a black box because of its incomprehensible functions and unclear working mechanism. Without a clear understanding of how and why a model works, the development of high-performance models typically relies on a time-consuming trial-and-error process. As a result, academic researchers and industrial practitioners are facing challenges that demand more transparent and explainable systems for better understanding and analyzing machine learning models. The proposed system is mainly focused to help the industries that use mission critical deep learning applications. The project aims to create the solution for the challenges faced by the medical field by providing an algorithm that works and applies utility metrics of accuracy,precision and recall to give a positive or negative result. This platform enables users to:

1. Train the model as per their requirements.
2. Provide high accuracy, precision, and recall

The plan of action was to come up with two predictive models.Both have them have similar and comparable accuracies. They have classified the test cases right above 97% of the times.. This project will be very helpful in the healthcare industry. This solution makes it easy to determine the if a tumor is benign or malignant.

Without a clear understanding of how and why a model works, the development of high-performance models typically relies on a time-consuming trial-and-error process.

We need to establish some metrics before hand to make sure nothing goes wrong.

3.3.3 Economical Feasibility

The economic feasibility examines the financial viability of the proposed system. Detailed analysis was carried out to determine the financial costs and benefits of the proposed solution.This model mainly focuses on accuracy, precision and recall. It has high performance rates for these metrics. The need for such models is very high in today's world because ML, DL, and AI are being used to solve consequential real-world problems. In such scenarios, even the slightest inaccuracy may cause immutable repercussions.. It is cost-effective as only scaling the database will require a small capital. The healthcare industry is an avenue that doesn’t accept minor mistakes as it is a matter of life and death.Since costs are very low, the project is economically feasible.

CHAPTER 4

SYSTEM SPECIFICATION

4.1 Functional Requirements

A Functional Requirement is a description of the service that the software must offer. It describes a software system or its component. Requirements are that specifies a function that a system or system component must be able to perform. The functional requirement describes a functionality to be made available to the users of the system, characterizing partially its behavior as an answer to the stimulus that it is subjected to. This type of requirement should not mention any technological issue, that is, ideally functional requirements must be independent of design and implementation aspects.

- **The project must be able to analyze the data in different ways to find trends**

Datasets need to be cleaned, labeled and organized to conduct Exploratory Data Analysis and Principal component Analysis.

- **It should be capable of providing service even for bigger datasets**

The model should be able to provide solutions and predictions to bigger datasets. The data for cancer rates keeps increasing day by day and the model should be able to handle huge datasets.

- **The model should provide high accuracy**

  Accuracy is one metric for evaluating classification models. Informally, accuracy is the fraction of predictions our model got right. This is one of the most important metrics

- **The model should also provide high precision and recall**

Precision and recall fall just behind accuracy but are equally important in testing a model's performance. Precision refers to the number of true positives divided by the total number of positive predictions. It basically refers to the quality of the decision made. Recall means the percentage of a certain class correctly identified. It is also important for measuring the success rate of the model. These metrics make up the most important aspects considered

when making the product. These metrics help in not going wrong with the process of solving the problem. We have to choose them carefully for higher success rate.

4.2 Non-Functional Requirements

Non-Functional requirements define desired qualities of the system to be developed and often influence the system architecture more than functional requirements do. A non-functional requirement corresponds to a set of restrictions imposed on the system to be developed, establishing, for how attractive, useful, fast, or reliable it is. This category of requirement is a set of required overall attributes of the system, including portability, reliability, efficiency, human engineering, testability, understandability, and modifiability.

The Non-Functional requirement of the project is to meet the accuracy and execution speed expected by the users. It is built to be reliable and does not consist of any sensitive information of a user, thus making it more secure. The project is to be compatible with different browsers. As the user interface for the project will be a web application it is built to be user friendly, and the operations can be well understood by the user.

Availability (A): The system must be available for wide range of users from various fields.

Maintainability (MN): The system must be effectively and efficiently modified.

Performance (PE): The project must provide high performance compared to the existing systems.

Reliability (R): The system must be consistent without any failure to provide result at any given time.

Scalability (SC): The system must be effectively modified for multiple datasets to obtain accurate results.

Usability (US): The system must be user friendly and easy to access.

4.3 Software Requirements

- Operating System – Windows 8 or higher
- Platform Used – Kaggle Notebook
- Languages – Python

4.4 Hardware Requirements

- RAM – 2 GB or higher
- GPU – 1050 or higher

CHAPTER 5

PROJECT DESCRIPTION

5.1 Problem Definition

In the ML/DL algorithms developed by researchers to predict breast cancer, majority of them have used archaic methods.In the newer times, AI and ML have been used to make right predictions. The model may be experimented with certain data sets but for the model to get tested with all the permutation and combination of data sets under different conditions is a challenge. For the model to give efficient and accurate results under all circumstances, the users require more accuracy, precision and recall

- Mission : To create an algorithm that is unique, efficient and solves a real-world problem.

- Vision : To build Breast Cancer Detection framework that works on regression, applies utility metrics to give a decision on the tumours.

5.2 Overview of the Project

With the proposed project we provide an adversarial robustness analysis framework for visual recognition that applies utility metrics of attacks and defense to give a comprehensive robustness report. The outcome of the robustness analyzer will have three parts, the first one being a robust model of the application that has been given to analyze. The existing model with defense metrics which can handle attacks will be available for the user to publish it.

The second part consists of a confined document which contains all the adversary samples that were used by our analyzer to check the robustness of the applications model. All the adversary samples are noted to which the model has malfunctioned and the same is given to the user for the better understanding of the model's weak points. The last one being a complete comprehensive robustness report that gives a detailed description of the model's accuracy and efficiency for utility attack metrics.

5.3 System Architecture

The Breast Cancer Coimbra Data set consists of all the routine blood analysis. • In the per-processing phase using the process Normalization, the data-set is structured in order

to reduce data redundancy and improve data integrity • In the Feature Extraction phase, using PCA the large data-set is summarized into smaller set of summary indices that can be easily visualized and analyzed. • The Median Filtering removes the noise from the data-set images. • The Data dividing phase uses, the k-fold cross validation scheme to classify the data into training data and testing data • Using the ANN-based classification, it is determined if the person is healthy or is a patient who is suffering.

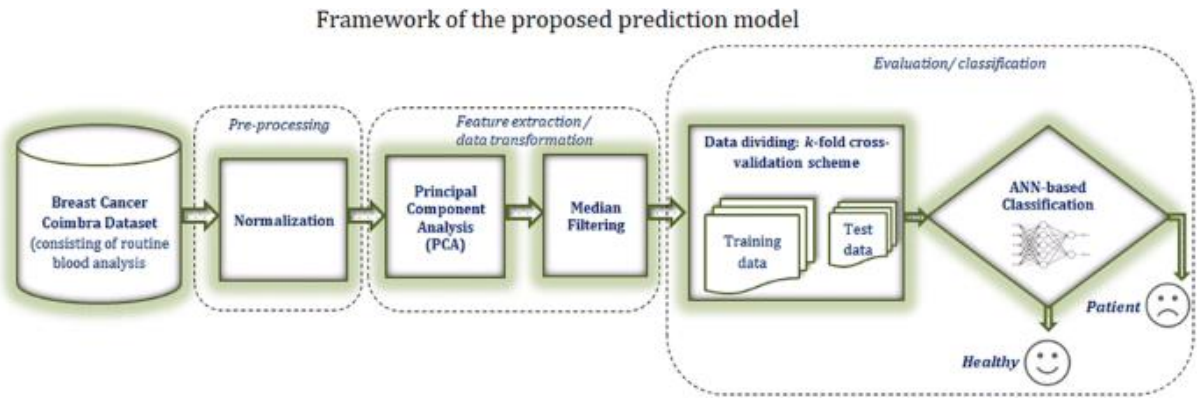


Fig 5.3: System Architecture

Fig 5.3 shows the overview of the proposed system's architecture. Here, the layers of the system and how each of these layers interact with one another, along with the working flow of the system is depicted.

5.4 Data Flow Diagram

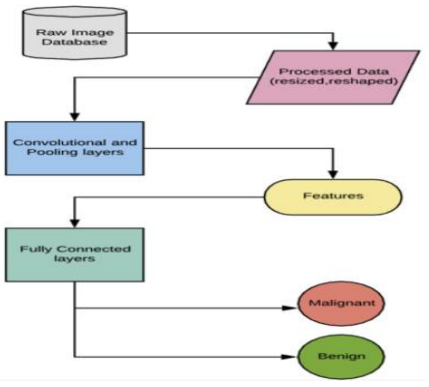


Fig 5.4 : Data Flow Diagram

Fig 5.4 shows the dataflow diagram of the project. The raw image database collects all the raw images, and these are resized and reshaped to become Processed Data. The generalized

features are extracted by convolutional and pooling filters given by the convulutional layers, which helps the network recognise features independent of their location in the image.

• The fully connected layers connects features from one layer to another layers.

• This is further grouped into two groups called ‘Malignant’ and ‘Benign’.

5.5 Module Description

5.5.1 Exploratory Data Analysis

In statistics, exploratory data analysis is an approach of analyzing data sets to summarize their main characteristics, often using statistical graphics and other data visualization methods. A statistical model can be used or not, but primarily EDA is for seeing what the data can tell us beyond the formal modeling and thereby contrasts traditional hypothesis testing. Exploratory data analysis has been promoted to encourage statisticians to explore the data, and possibly formulate hypotheses that could lead to new data collection and experiments. EDA is different from initial data analysis which focuses more narrowly on checking assumptions required for model fitting and hypothesis testing, and handling missing values and making transformations of variables as needed. EDA encompasses IDA.

The objectives of EDA are to:

- Enable unexpected discoveries in the data
- Suggest hypotheses about the causes of observed phenomena
- Assess assumptions on which statistical inference will be based
- Support the selection of appropriate statistical tools and techniques
- Provide a basis for further data collection through surveys or experiments

Univariate analysis is the simplest form of data analysis, where the data being analyzed consists of only one variable. Since it’s a single variable, it doesn’t deal with causes or relationships. The main purpose of univariate analysis is to describe the data and find patterns that exist within it. Multivariate data analysis refers to any statistical technique used to analyze data that arises from more than one variable. This models more realistic applications, where each situation, product, or decision involves more than a single variable. Let us look at a few visualizations used for performing multivariate analysis. A scatter plot is a two-dimensional data visualization that uses dots to represent the values obtained for two different variables – one plotted along the x-axis and the other plotted

along the y-axis. A bar chart represents categorical data, with rectangular bars having lengths proportional to the values that they represented. EDA is a crucial step to take before diving into machine learning or statistical modeling because it provides the context needed to develop an appropriate model in the future. EDA is valuable to create the project.

Fig 5.5: EDA

Fig 5.5 shows the EDA of the attribute radius_worst. It shows us the comparison of the normal distribution and normal sample that is in the data.

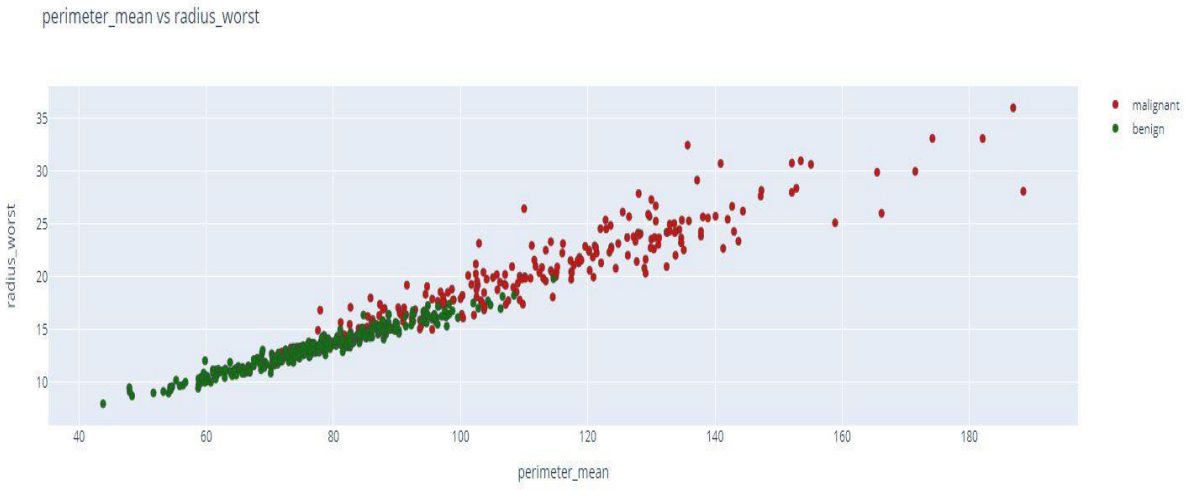


Fig 5.6: Radius vs Perimeter Comparison

Fig 5.6 shows the comparison of mean perimeter and worst radius. It helps us understand the extremities of the dataset when it comes to these 2 attributes. It helps us find trends and

come to conclusions. It is a statistical process that converts the observations of correlated features into a set of linearly uncorrelated features with the help of orthogonal transformation. These new transformed features are called the Principal Components.

5.5.2 Principal Component Analysis

Principal Component Analysis is an unsupervised learning algorithm that is used for the dimensionality reduction in machine learning. It is a statistical process that converts the observations of correlated features into a set of linearly uncorrelated features with the help of orthogonal transformation. These new transformed features are called the Principal Components. It is one of the popular tools that is used for exploratory data analysis and predictive modeling. It is a technique to draw strong patterns from the given dataset by reducing the variances. PCA generally tries to find the lower-dimensional surface to project the high-dimensional data.

PCA works by considering the variance of each attribute because the high attribute shows the good split between the classes, and hence it reduces the dimensionality. Some real-world applications of PCA are image processing, movie recommendation system, optimizing the power allocation in various communication channels. It is a feature extraction technique, so it contains the important variables and drops the least important variable.

The transformed new features or the output of PCA are the Principal Components. The number of these PCs are either equal to or less than the original features present in the dataset. Some properties of these principal components are given below:

1. The principal component must be the linear combination of the original features.
2. These components are orthogonal, i.e., the correlation between a pair of variables is zero.
3. The importance of each component decreases when going to 1 to n, it means the 1 PC has the most importance, and n PC will have the least importance.

Steps for PCA algorithm

- Getting the dataset

- Firstly, we need to take the input dataset and divide it into two subparts X and Y, where X is the training set, and Y is the validation set.
- Representing data into a structure
- Now we will represent our dataset into a structure. Such as we will represent the two-dimensional matrix of independent variable X. Here each row corresponds to the data items, and the column corresponds to the Features. The number of columns is the dimensions of the dataset.
- Standardizing the data
- In this step, we will standardize our dataset. Such as in a particular column, the features with high variance are more important compared to the features with lower variance.
- If the importance of features is independent of the variance of the feature, then we will divide each data item in a column with the standard deviation of the column. Here we will name the matrix as Z.
- Calculating the Covariance of Z
- To calculate the covariance of Z, we will take the matrix Z, and will transpose it. After transpose, we will multiply it by Z. The output matrix will be the Covariance matrix of Z.
- Calculating the Eigen Values and Eigen Vectors
- Now we need to calculate the eigenvalues and eigenvectors for the resultant covariance matrix Z. Eigenvectors or the covariance matrix are the directions of the axes with high information. And the coefficients of these eigenvectors are defined as the eigenvalues.
- Sorting the Eigen Vectors
- In this step, we will take all the eigenvalues and will sort them in decreasing order, which means from largest to smallest. And simultaneously sort the eigenvectors accordingly in matrix P of eigenvalues. The resultant matrix will be named as P*.
- Calculating the new features Or Principal Components
- Here we will calculate the new features. To do this, we will multiply the P* matrix to the Z. In the resultant matrix Z*, each observation is the linear combination of original features. Each column of the Z* matrix is independent of each other.
- Remove less or unimportant features from the new dataset.

- The new feature set has occurred, so we will decide here what to keep and what to remove. It means, we will only keep the relevant or important features in the new dataset, and unimportant features will be removed out.

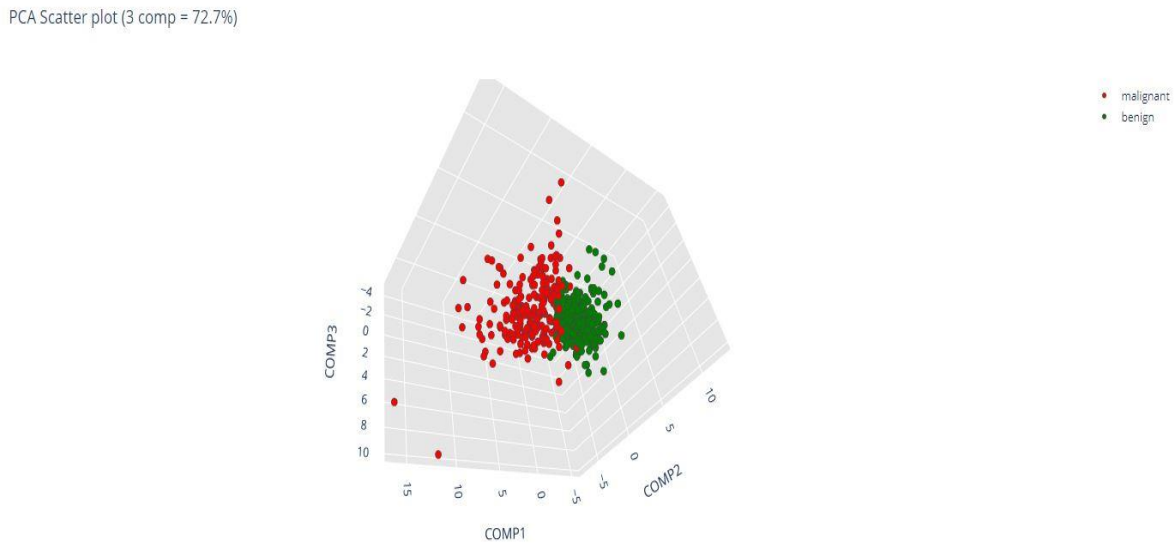


Fig 5.7: PCA with 3 components

Fig 5.7 shows the PCA 3D graph with 3 components. 31 components have been reduced to 3.

5.5.3 Predictive Model 1 and 2

We have developed two predictive models and compared their accuracy, recall and precision.

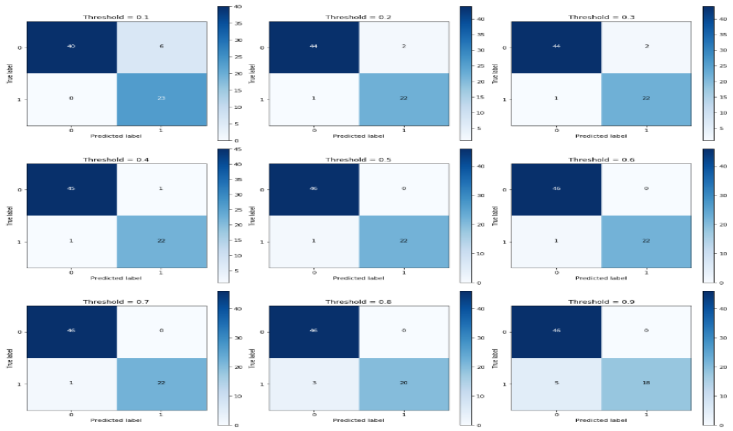


Fig 5.8: Predictive model 1

The Fig 5.8 shows the predictive model 1's performance in terms of metrics considered.

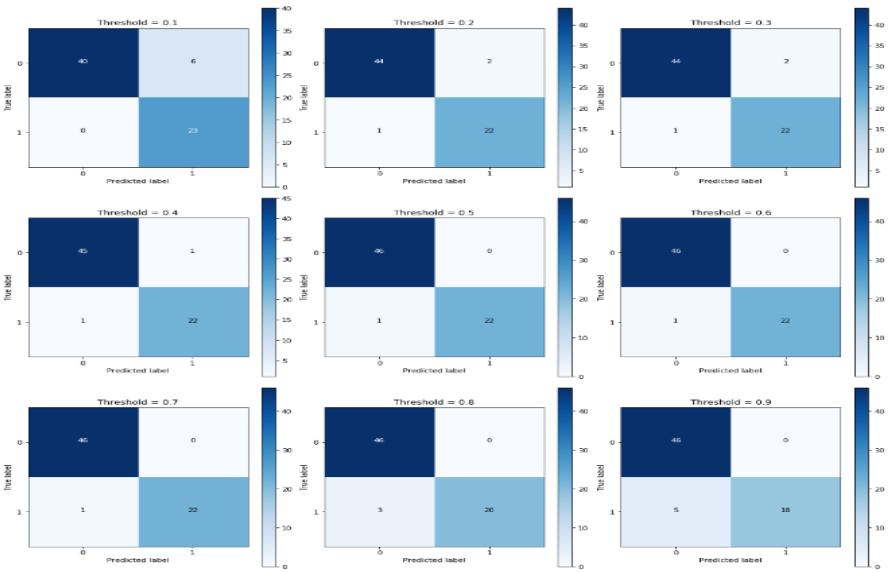


Fig 5.9 : Predictive model 2

Fig 5.9 shows the working of Predictive model 2.

A Voting Classifier is a machine learning model that trains on an ensemble of numerous models and predicts an output (class) based on their highest probability of chosen class as the output.

It simply aggregates the findings of each classifier passed into Voting Classifier and predicts the output class based on the highest majority of voting. The idea is instead of creating separate dedicated models and finding the accuracy for each them, we create a single model which trains by these models and predicts output based on their combined majority of voting for each output class.

Voting Classifier supports two types of votings:

- Hard Voting: In hard voting, the predicted output class is a class with the highest majority of votes i.e the class which had the highest probability of being predicted by each of the classifiers. Suppose three classifiers predicted the output class(A, A, B), so here the majority predicted A as output. Hence A will be the final prediction.
- Soft Voting: In soft voting, the output class is the prediction based on the average of probability given to that class. Suppose given some input to three models, the prediction probability for class A = (0.30, 0.47, 0.53) and B = (0.20, 0.32, 0.40). So the average for class A is 0.4333 and B is 0.3067, the winner is clearly class A because it had the highest probability averaged by each classifier.

CHAPTER 6

SYSTEM TESTING

6.1 Introduction

Many practitioners may rely solely on machine learning model performance evaluation. However, evaluation is not the same as testing. It is important to identify their differences. Machine learning model evaluation focuses on the overall performance of the model. Such evaluations can consist of performance metrics and curves, and perhaps examples of incorrect predictions.

This way of model evaluation is a great way to monitor your model's outcome between different versions. However, it does not tell us a lot about the reasons behind the failures and the specific model behaviors.

For example, your model might suffer a performance drop in a critical data subset while its overall performance doesn't change or even improves. Or, in another case, model retraining on new data does not produce performance change but introduces unnoticed social bias towards a specific demographic group**.**

Testing is not easy, and testing machine learning models is even harder. You need to prepare your workflow for unexpected events while working with dynamic inputs, black-box models, and shifting input/output relationships.

For this reason, it is worth following established best practices in software testing:

- Test after introducing a new component, model, or data, and after model retraining
- Test before deployment and production
- Write tests to avoid recognized bugs in the future

6.2 Predictive Model 1 and 2

Logistic regression is a process of modeling the probability of a discrete outcome given an input variable. The most common logistic regression models a binary outcome; something that can take two values such as true/false, yes/no, and so on. Multinomial logistic regression can model scenarios where there are more than two possible discrete outcomes. Logistic regression is a useful analysis method for classification problems, where we are trying to determine if a new sample fits best into a category. As aspects of cyber security are classification problems, such as attack detection, logistic regression is a useful analytic technique. The second technique used was voting classifier.

CHAPTER 7

RESULTS AND DISCUSSION

7.1 EDA

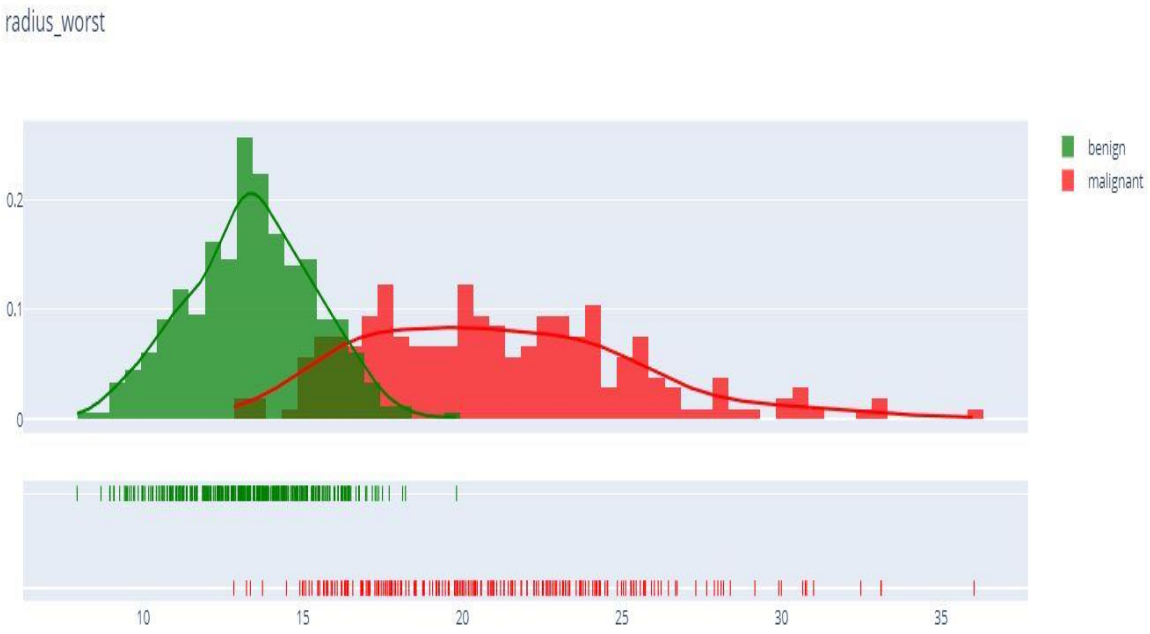


Fig 7.1: EDA Radius Worst

Fig 7.1 shows the graph of the attribute radius_worst. It shows us the comparison of the normal distribution and normal sample that is in the data.

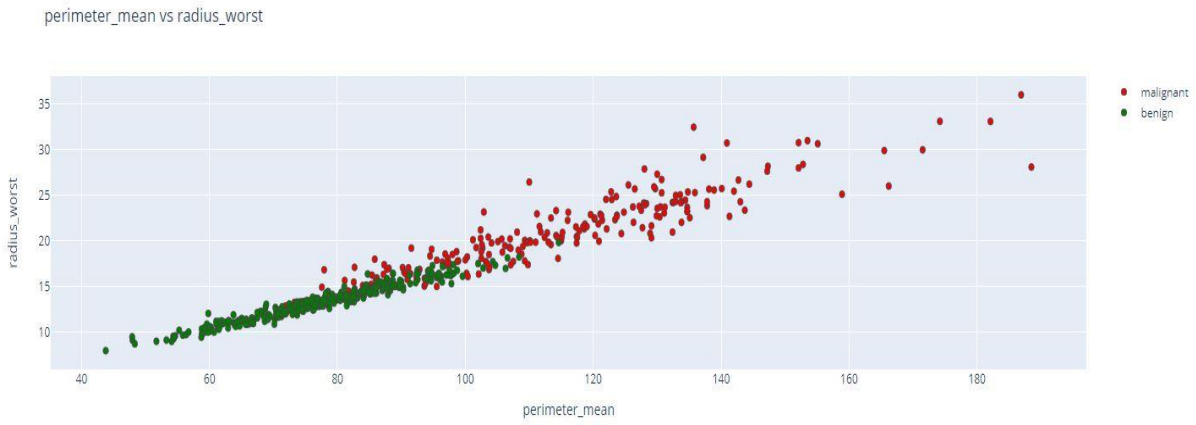


Fig 7.2: Radius vs Perimeter

Fig 7.2 shows the graph of the attribute radius_worst and mean perimeter. It shows us the comparison of the normal distribution and normal sample that is in the data.

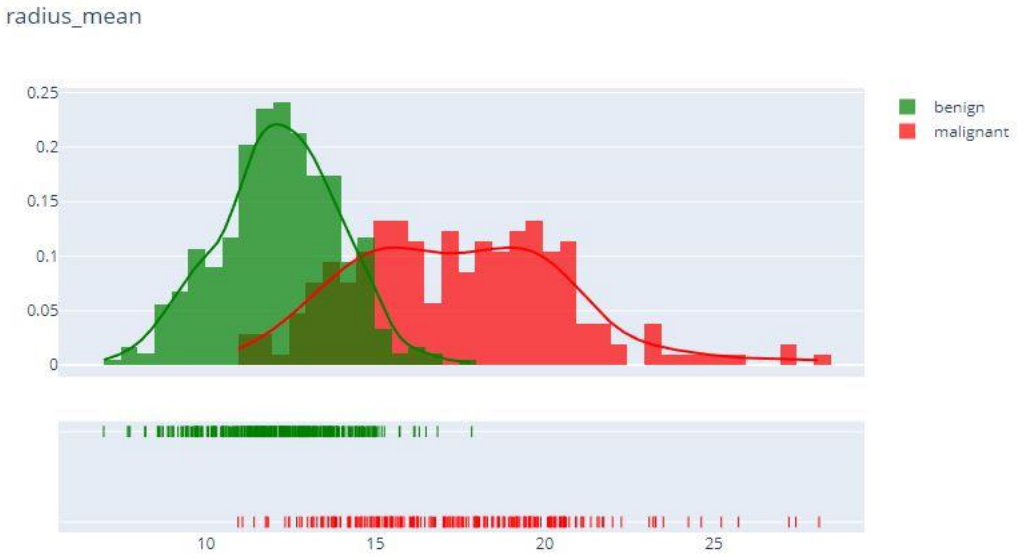


Fig 7.3: EDA radius worst attribute

Fig 7.3 shows the EDA of the attribute radius_worst. It shows us the comparison of the normal distribution and normal sample that is in the data.

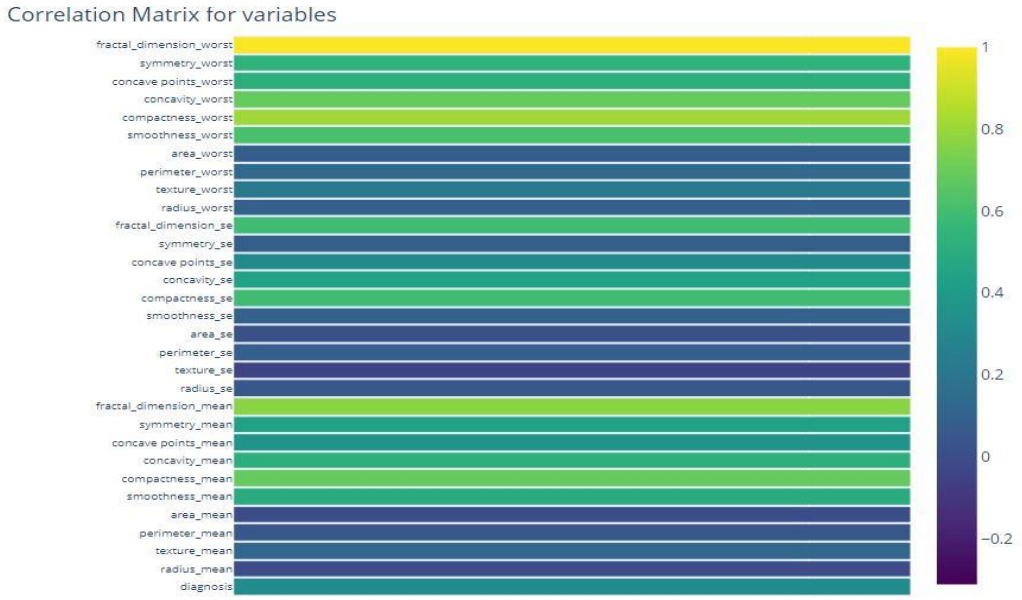


Fig 7.4: Correlation Matrix

Fig 7.4 shows correlation matrix. It shows us the comparison of how each variable value varies with respect to another variable. Logistic regression uses this method for classification problems, where we are trying to determine if a new sample fits best into a

category. As aspects of cyber security are classification problems, such as attack detection, logistic regression is a useful analytic technique.

7.2 PCA

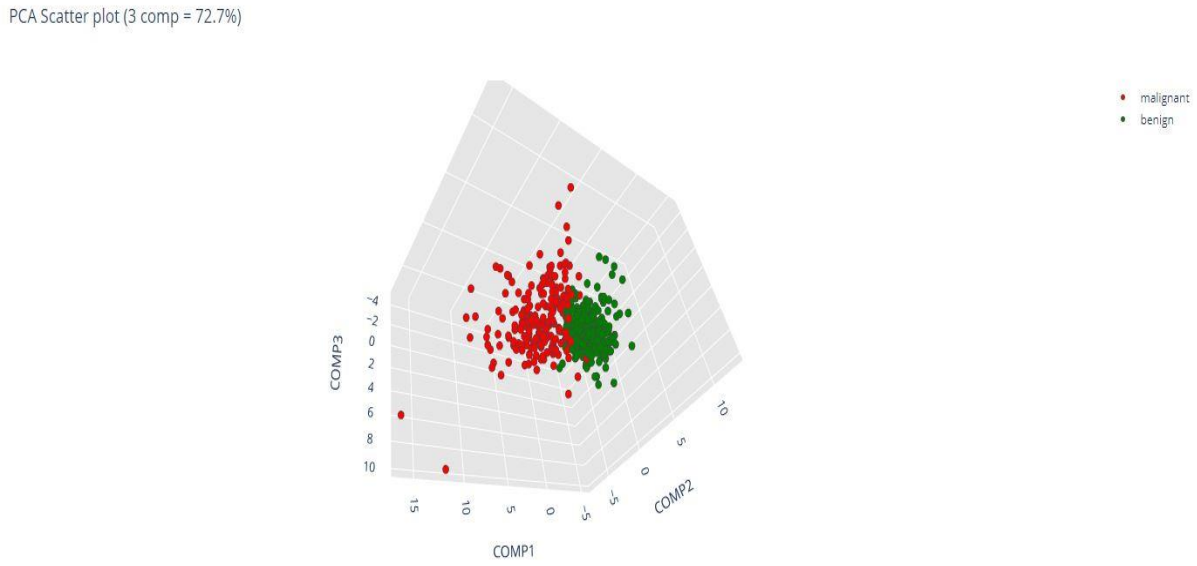


Fig 7.5 :PCA with 3 components

Fig 7.5 shows the PCA 3D graph of 3 components. 31 components are reduced to 3.

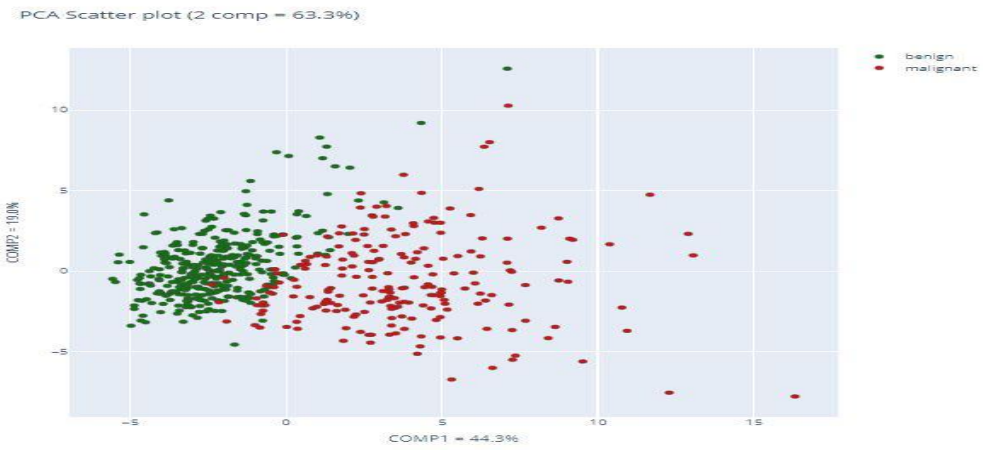


Fig 7.6: PCA Scatter Plot

Fig 7.6 shows PCA Scatter Plot of 2 components. 31 components are reduced to 2.

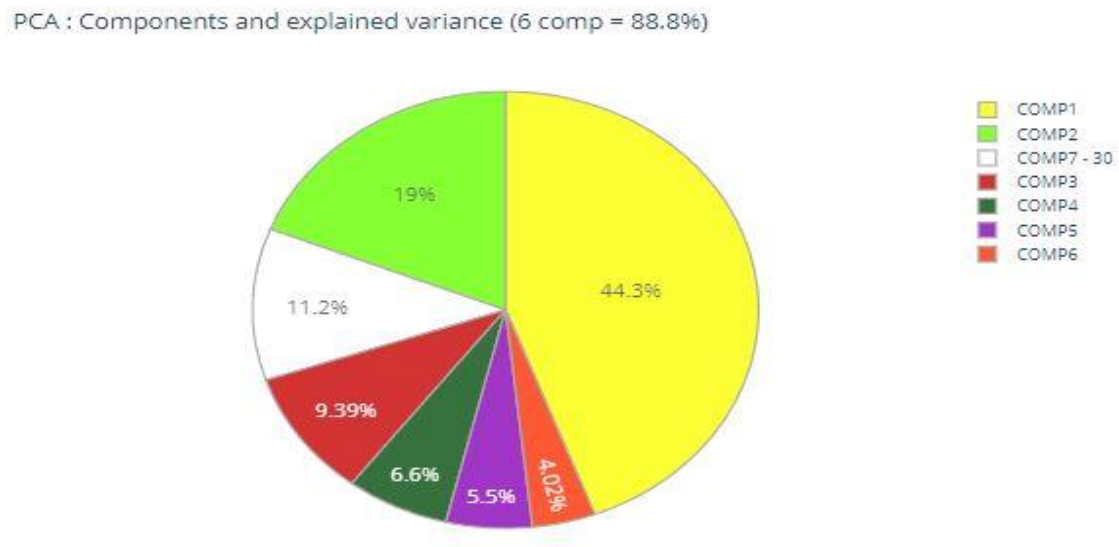


Fig 7.7 : PCA Pie Plot

Fig 7.7 shows the components and explained variance. This is a pie-plot of the PCA.

7.3 Predictive models

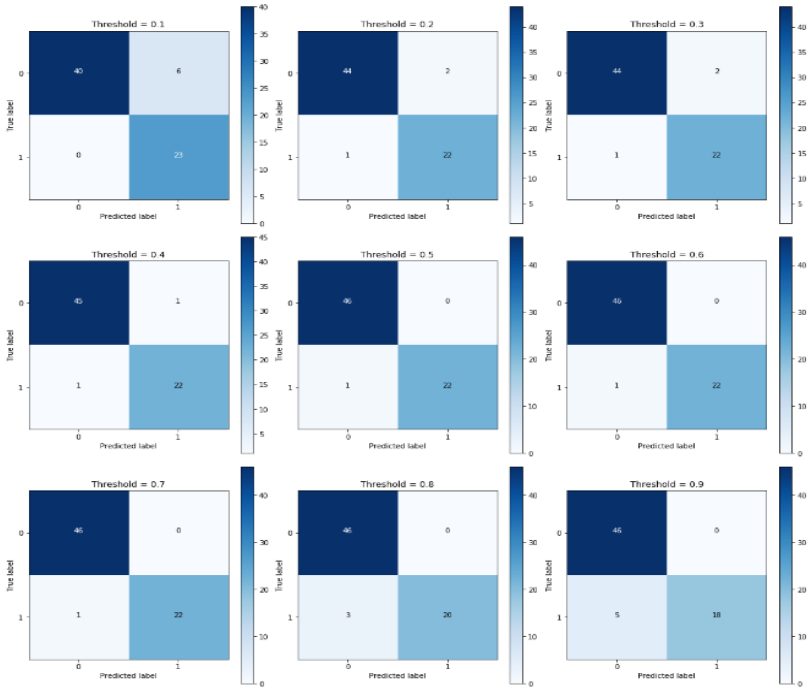


Fig 5.8: Predictive model 1

The Fig 7.8 shows the predictive model 1's performance in terms of metrics considered.

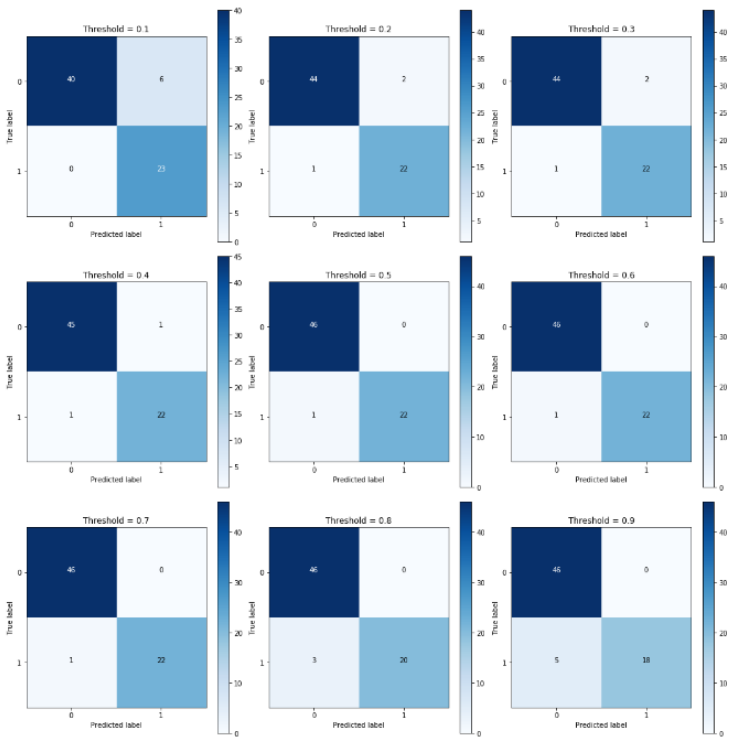


Fig 7.9: Predictive model 2

The Fig 5.8 shows the predictive model 2's performance in terms of metrics considered.

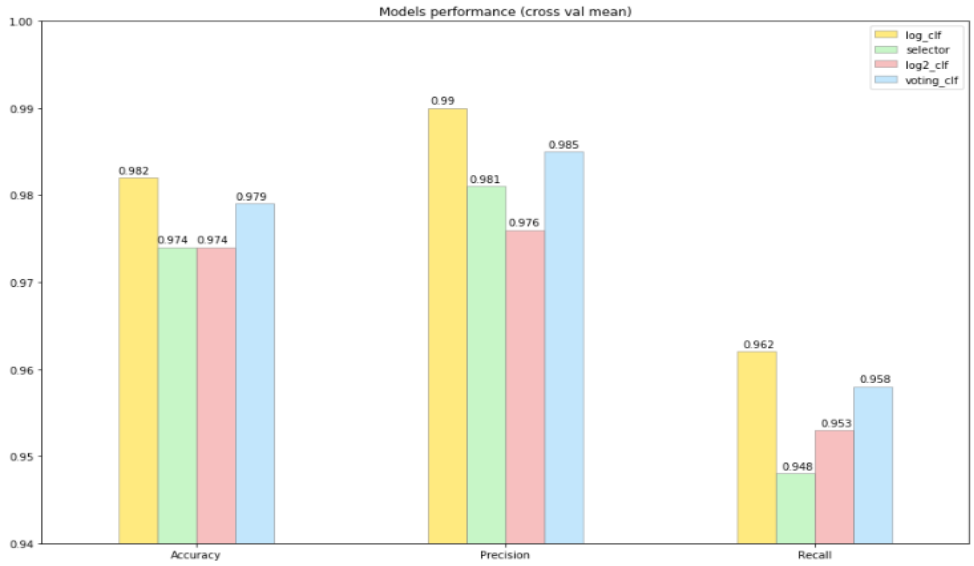


Fig 7.10: Comparison of metrics

The Fig 7.10 shows the comparison between the metrics considered. It displays the accuracy , precision and recall of each model.

The metrics considered here are accuracy, precision and recall. This helps us set a proper path for our project.

CONCLUSION AND FUTURE WORK

Breast Cancer has been a long standing issue which has been plaguing humans world-wide every single day. We are grateful to have this opportunity to work on a social cause and try to provide a solution. We have always looked at the problem at hand and tried to solve it pragmatically. Any solution provided to the society adds to the measures that can taken to get through impediments. We have come up with 2 predictive models which have performed very well according to the metrics we have considered. In the future, this system can be implemented in a wide scale in every health centre. Since the disease is omnipresent, the challenge of scalability is one issue we can not afford to put on the backburner. It has to be developed in such a way that the database is ever increasing and the servers are able to handle it.

REFERENCES

[1] Fatima, Noreen; Liu, Li; Sha, Hong; Ahmed, Haroon (2020). *Asymmetry Analysis in Breast Cancer Detection Using Thermal Infrared Images[Phani Teja Kuruganti, Hairong Qi] (), 1–1.* doi:10.1109/ACCESS.2020.3016715

[2] Review of Electromagnetic Techniques for Breast Cancer Detection[Ahmed M. Hassan, Magda El-Shenawee] (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC) - Palladam, India (2020.10.7-2020.10.9)] 2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC) - *Machine Learning model for Breast Cancer Prediction.* (), 472–477. doi:10.1109/I-SMAC49090.2020.9243323

[3] Comparison of data mining classification algorithms for breast cancer prediction[Mr. Chintan Shah , Dr. Anjali Jivani] *Computing and Industry* 4.0 (C2I4), (), –. doi:10.1109/c2i451079.2020.9368911

[4] Breast Cancer Prediction Analysis using Machine Learning Algorithms[Vinayak A Telsang,Kavyashree Hegde] *Breast Cancer Risk Prediction based on Six Machine Learning Algorithms . 2020 IEEE Asia-Pacific Conference on Computer Science and Data Engineering* (CSDE), (), –. doi:10.1109/csde50874.2020.9411572

[5] www.google.com – Search Engine

[6] www.kaggle.com – Online community of data scientists and machine learning practitioners

[7] www.ieeexplore.ieee.org – IEEE Papers

APPENDIX

A.1 Frontend

The frontend of the project was developed using Django a framework on python. It follows a model-template-views architectural pattern. This is built to provide a web interface to the user to gain awareness about the disease and generic information about the model.

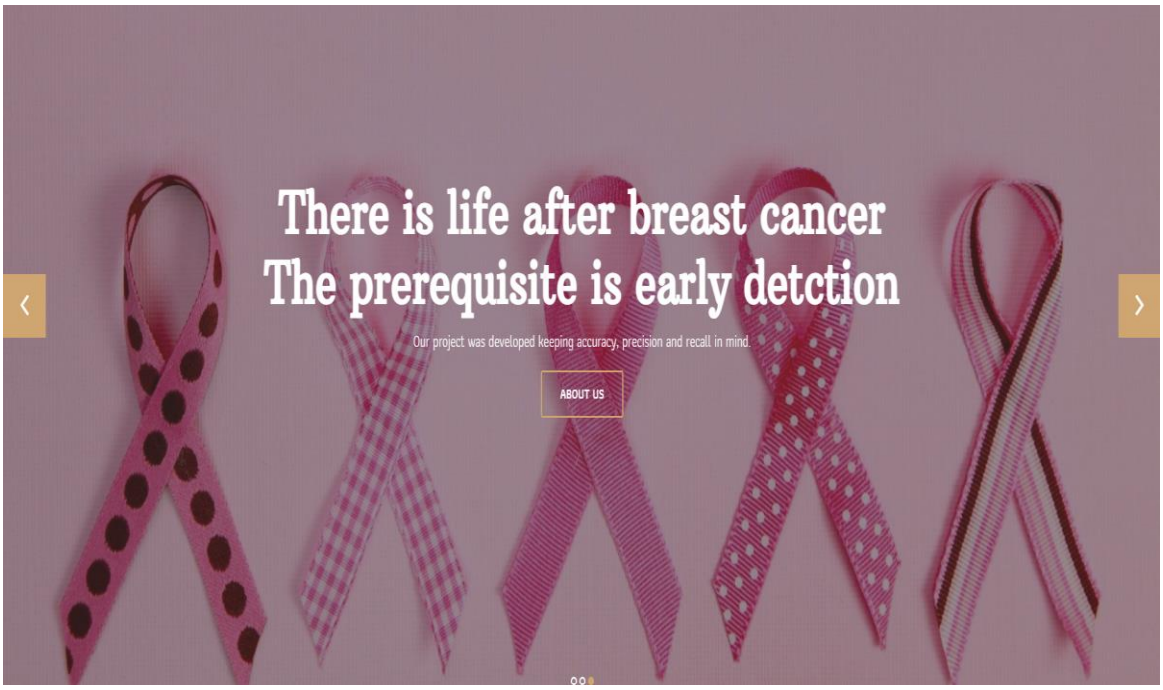


Fig 8.1: Home Page Scroll 1

The Figure 8.1 shows the view of Home Scroll 1.

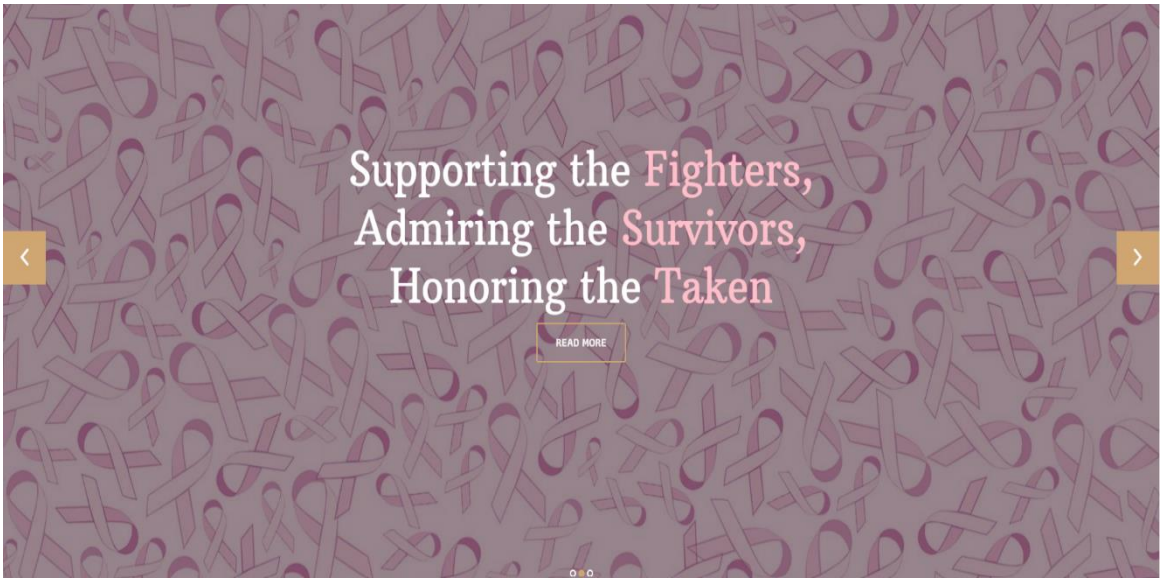


Fig 8.2: Home Page Scroll 2

Figure 8.2 shows the view of Home Page Scroll 2

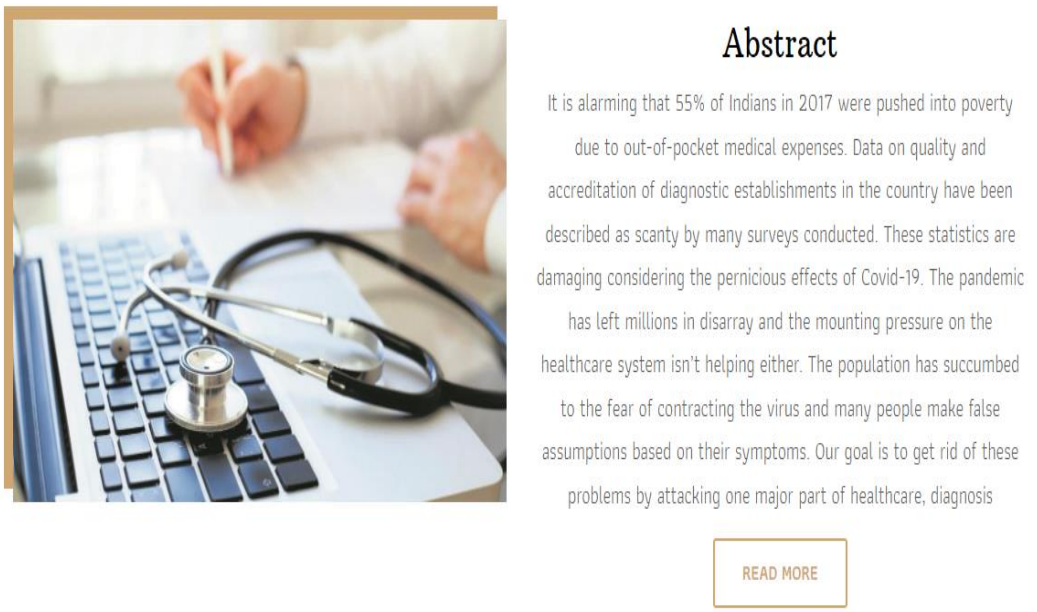


Fig 8.3: Abstract

The Figure 8.3 shows the view of the abstract of the project.

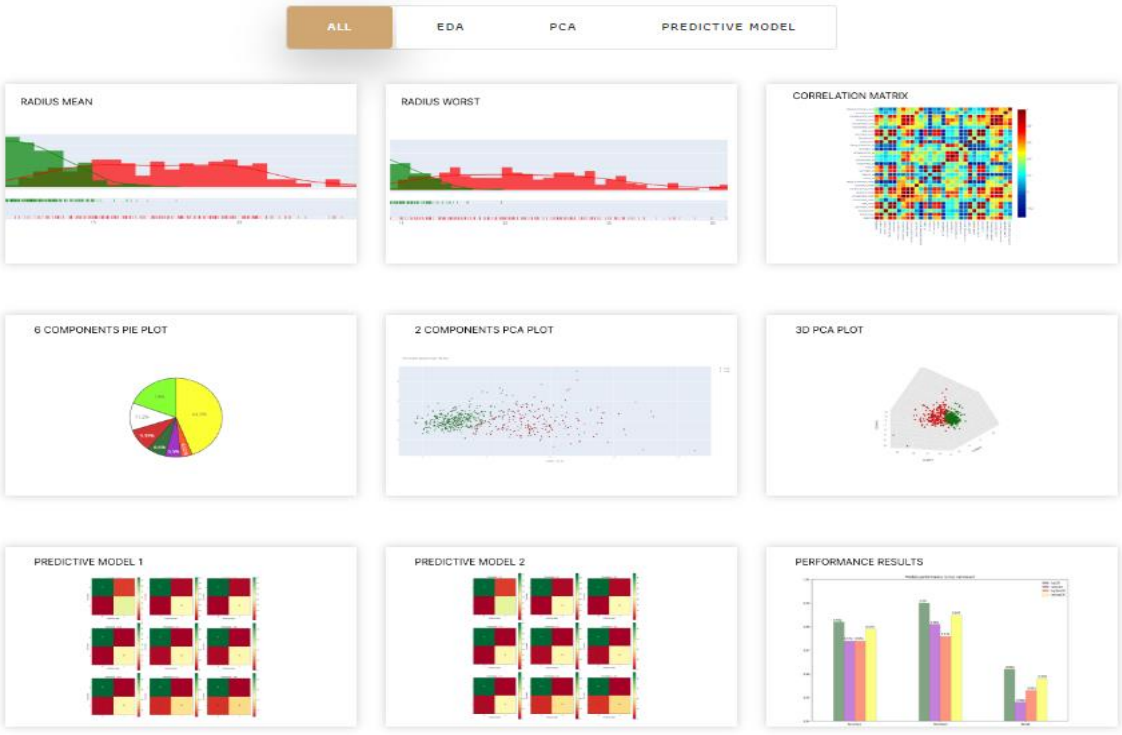


Fig 8.4: Snippets of results

The Figure 8.4 shows the view of the Snippets of results.

Fig 8.5: EDA Snippets

The Figure 8.5 shows the view of EDA Snippets.

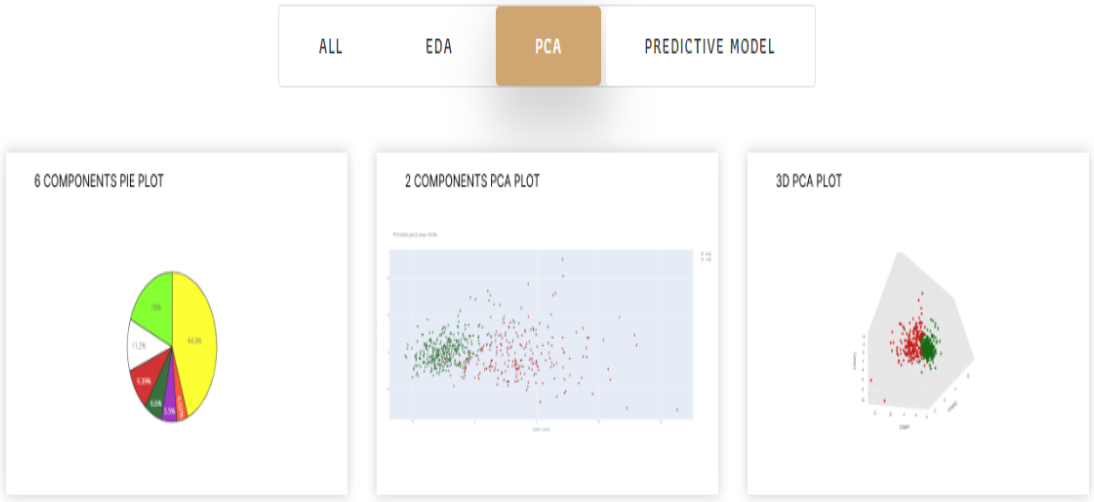


Fig 8.6: PCA Snippets

The Figure 8.6 shows the view of PCA Snippets.

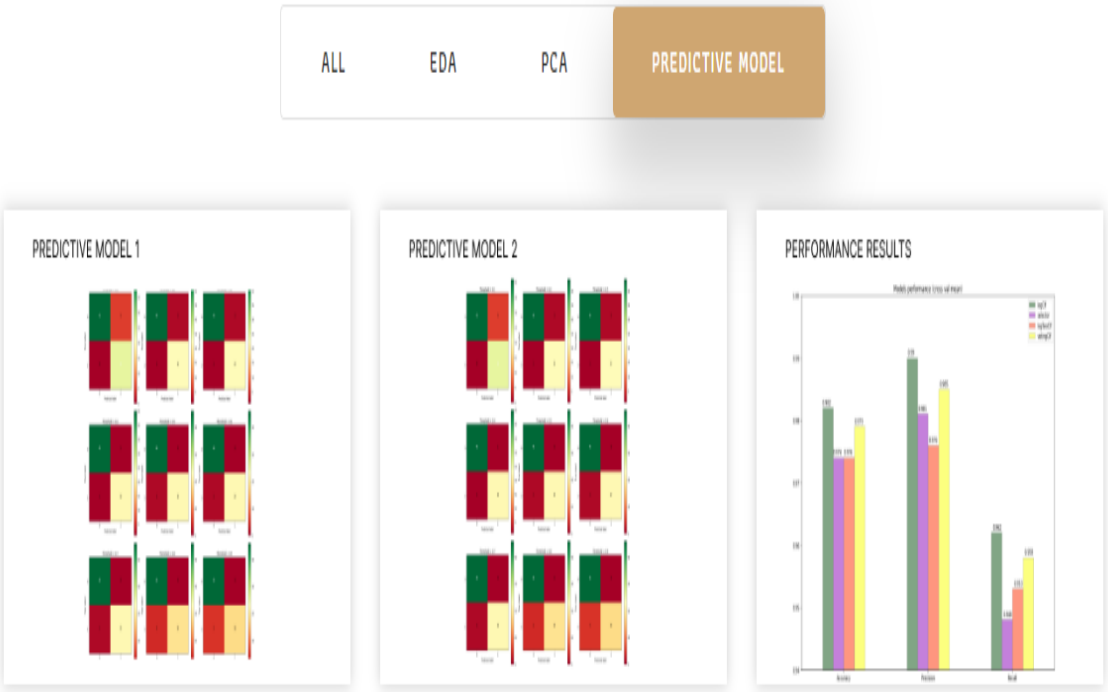


Fig 8.7: Predictive Model Snippets

The Figure 8.7 shows the view of Predictive Model Snippets.

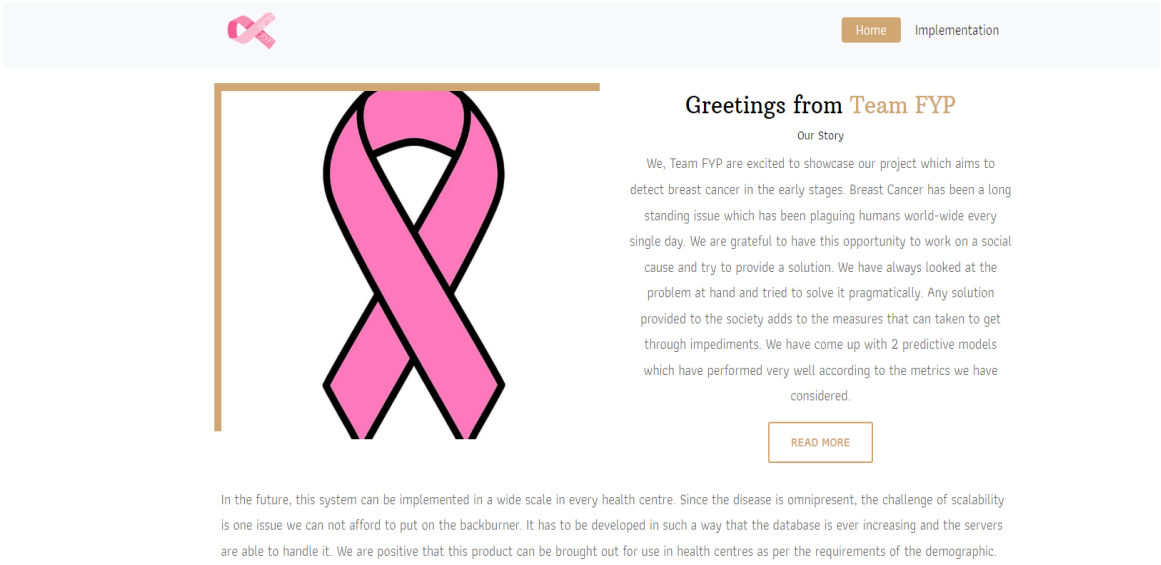


Fig 8.8: Message from Team

The Figure 8.8 shows the view of Message from Team.

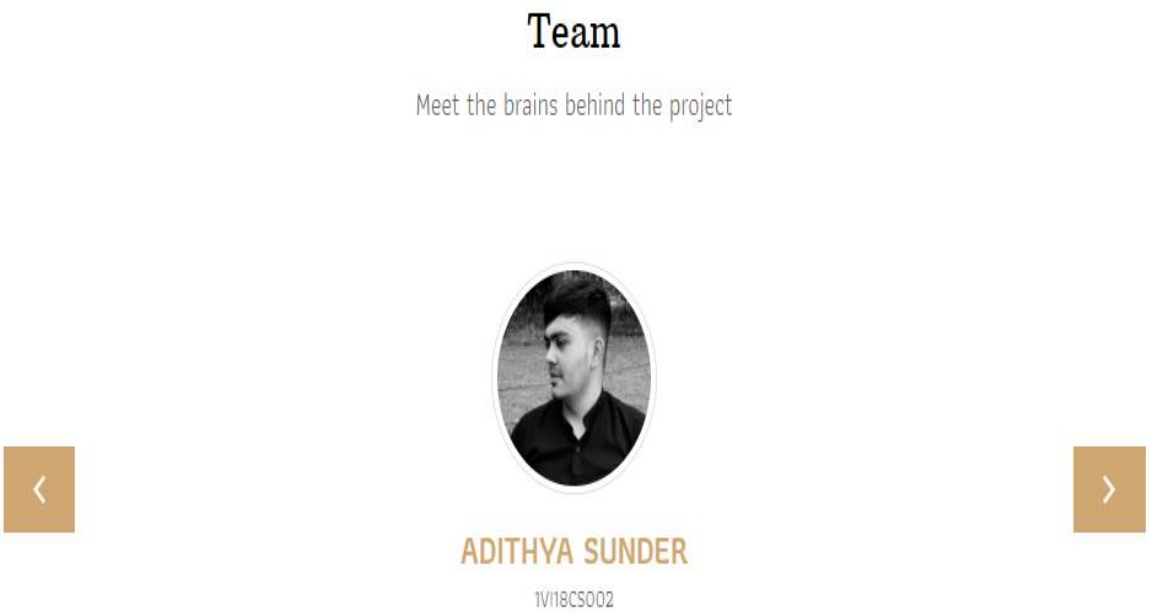
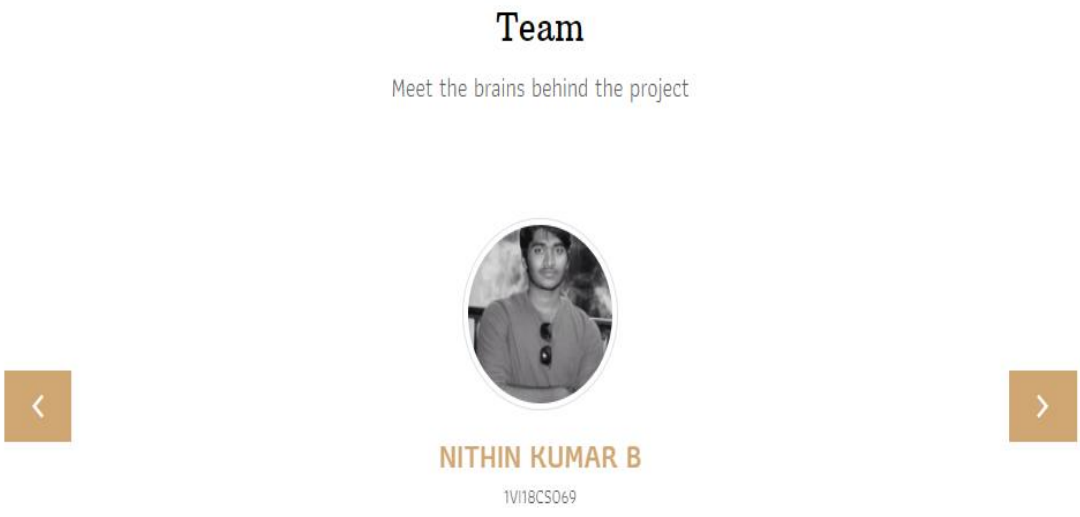


Fig 8.9:Team Member 1

The Figure 8.9 shows the view of Team Member 1.



8.91: Team Member 2

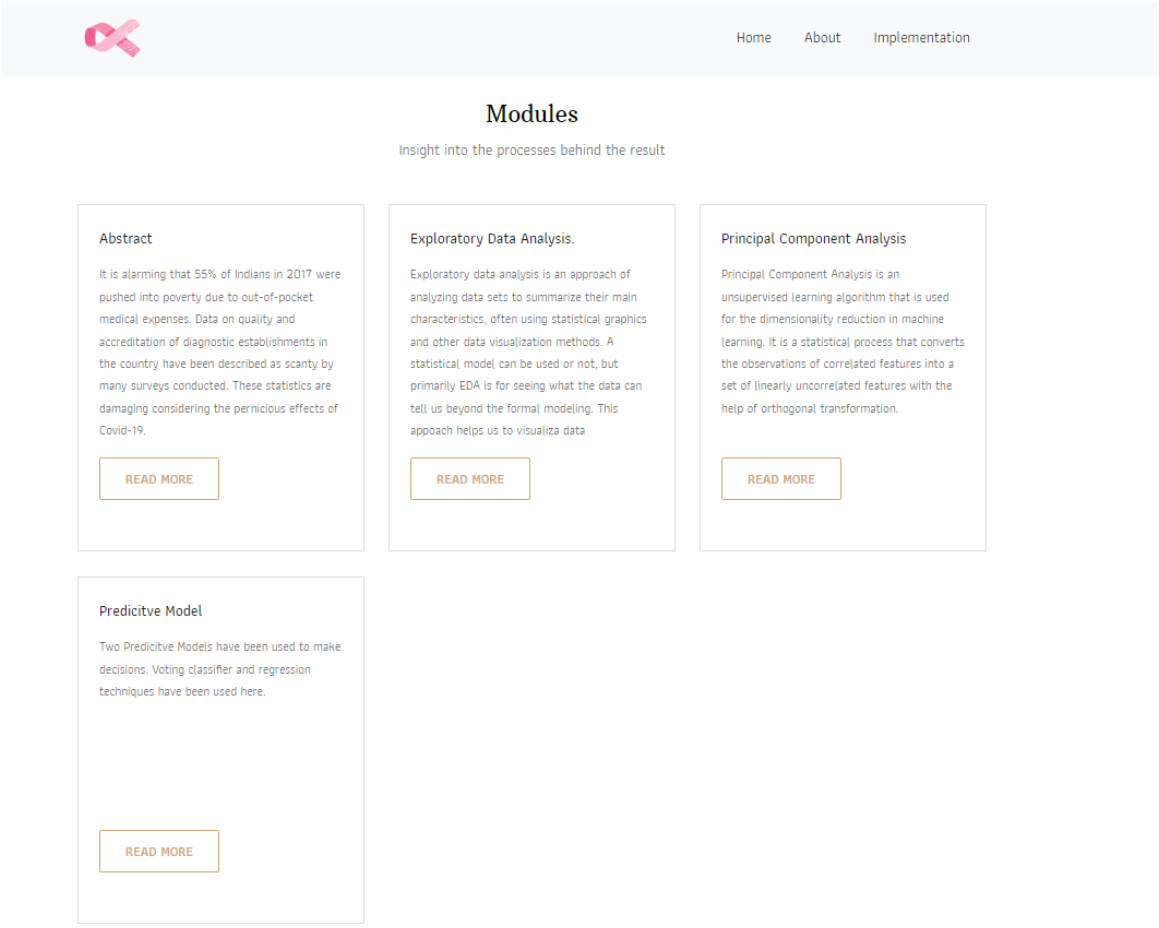
The Figure 8.2 shows the view of Team Member 2.

8.92: Team Member 3

The Figure 8.92 shows the view of Team Member 3.

8.93: Team Member 4

The Figure 8.93 shows the view of Team Member 4.

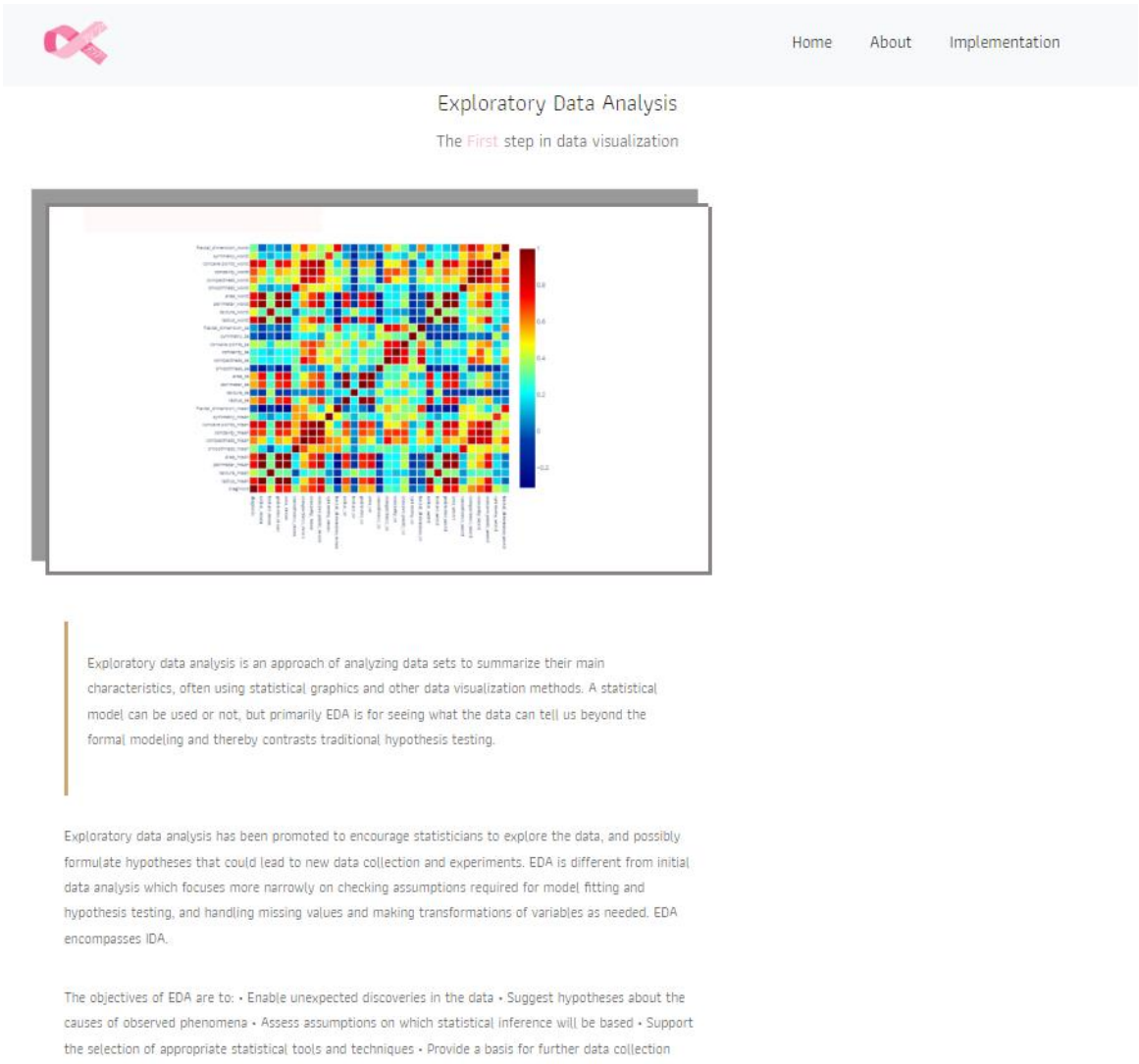


8.94: Modules

The Figure 8.94 shows the view of Modules.

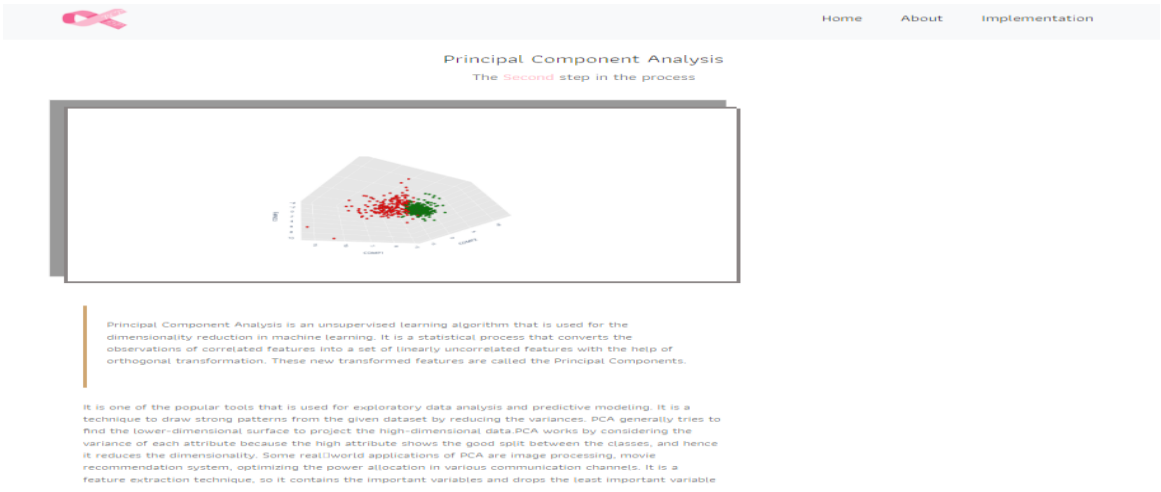
8.95: Abstract Page

The Figure 8. shows the view of Team Member 2.



8.96: EDA Page

The Figure 8.96 shows the view of EDA Page.

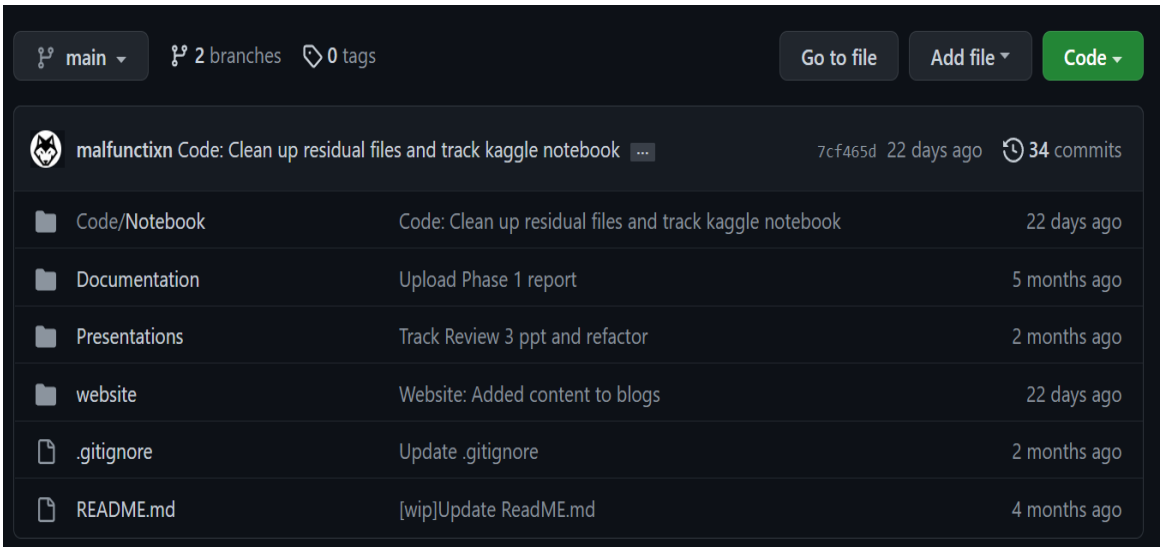


8.97: PCA Page

The Figure 8.97 shows the view of PCA Page.

8.98: Result Page

The Figure 8.98 shows the view of Result Page.



8.99: Implementation

The Figure 8.99 shows the view of Implementation.

A.2 Backend

The backend of the project uses python language for computing. We use various machine learning libraries for classification .

A.3 Source Code

A.3.1 EDA

```
data.head()
#describe
data.describe()
# We have two datasets one is Malignant and the other is benign
M = data[(data['diagnosis'] != 0)]
B = data[(data['diagnosis'] == 0)]
# count of the data avaialble
trace = go.Bar(x = (len(M), len(B)), y = ['malignant', 'benign'], orientation = 'h', opacity
= 0.8, marker=dict(
        color=[ 'red', 'green'],
        line=dict(color='#000000',width=1.5)))
layout = dict(title =  'Count of diagnosis variable')
              fig = dict(data = [trace], layout=layout)
py.iplot(fig)
correlation = data.corr()
#tick labels
matrix_cols = correlation.columns.tolist()
#convert to array
corr_array  = np.array(correlation)
plotLearningCurve(CVLogClf, 'Learning Curve For Logistic Model', X, y, (0.85,1.05),
10)
plt.savefig('7')
plt.show()
print("="*15)
print("Cross val Log ")
```

```
print("="*15)
crossLog = crossValMetrics(CVLogClf)
print("="*36)
print("Processing some more values and calulating, please hold...")
plotLearningCurve(selector, 'Learning Curve For Logistic Model with RFE', X, y, (0.85,1.05), 10)
plt.show()
print("="*15)
print("Cross val Log with RFE")
print("="*15)
cross_selector = crossValMetrics(selector)
print("="*36)
print("Process completed with no errors")
print("="*36)

def plotDistribution(dataSelect, sizeBin) :
   tmp1 = M[dataSelect]
   tmp2 = B[dataSelect]
   histData = [tmp1, tmp2]
   group_labels = ['malignant', 'benign']
   colors = ['red', 'green']
fig = ff.create_distplot(histData, group_labels, colors = colors, show_hist = True, bin_size = sizeBin, curve_type='kde')
     fig['layout'].update(title = dataSelect)
     py.iplot(fig, filename = 'Density plot')
plotLearningCurve(CVLogClf, 'Learning Curve For Logistic Model', X, y, (0.85,1.05), 10)
plt.savefig('7')
plt.show()
print("="*15)
print("Cross val Log ")
print("="*15)
```

```
crossLog = crossValMetrics(CVLogClf)
print("="*36)
print("Processing some more values and calulating, please hold...")
plotLearningCurve(selector, 'Learning Curve For Logistic Model with RFE', X, y, (0.85,1.05), 10)
plt.show()
print("="*15)
print("Cross val Log with RFE")
print("="*15)
cross_selector = crossValMetrics(selector)
print("="*36)
print("Process completed with no errors")
print("="*36)
#plot distribution 'mean'
plotDistribution('radius_mean', .5)
 plotDistribution('texture_mean', .5)
 plotDistribution('perimeter_mean', 5)
 plotDistribution('area_mean', 10)
plotDistribution('smoothness_mean', .5)
plotDistribution('compactness_mean' .5)
plotDistribution('concavity_mean' .5)
plotDistribution('concave points_mean' .5)
plotDistribution('symmetry_mean' .5)
plotDistribution('fractal_dimension_mean' .5)
trace = go.Bar(x = (len(M), len(B)), y = ['malignant', 'benign'], orientation = 'h', opacity = 0.8, marker=dict(
        color=[ 'red', 'green'],
        line=dict(color='#000000',width=1.5)))
layout = dict(title =  'Count of diagnosis variable')
            fig = dict(data = [trace], layout=layout)
py.iplot(fig)
```

```
#correlation
correlation = data.corr()
#tick labels
matrix_cols = correlation.columns.tolist()
#convert to array
corr_array  = np.array(correlation)

#Plotting
trace = go.Heatmap(z = corr_array,x = matrix_cols,y = matrix_cols,xgap = 2,ygap =
2,colorscale='jet',colorbar   = dict() ,)
layout = go.Layout(dict(title = 'Correlation Matrix for variables',autosize =
False,height  = 720,width   = 800,margin  = dict(r = 0 ,l = 210,t = 25,b = 210,),yaxis
= dict(tickfont = dict(size = 9)),xaxis   = dict(tickfont = dict(size = 9)),))
fig = go.Figure(data = [trace],layout = layout)
py.iplot(fig)
plotLearningCurve(CVLogClf, 'Learning Curve For Logistic Model', X, y,
(0.85,1.05), 10)
plt.savefig('7')
plt.show()
print("="*15)
print("Cross val Log ")
print("="*15)
crossLog = crossValMetrics(CVLogClf)
print("="*36)
print("Processing some more values and calulating, please hold...")
plotLearningCurve(selector, 'Learning Curve For Logistic Model with RFE', X, y,
(0.85,1.05), 10)
plt.show()
print("="*15)
print("Cross val Log with RFE")
print("="*15)
cross_selector = crossValMetrics(selector)
```

```
print("="*36)
print("Process completed with no errors")
print("="*36)
correlation = data.corr()
#tick labels
matrix_cols = correlation.columns.tolist()
#convert to array
corr_array  = np.array(correlation)

plotDistribution('texture_mean', .5)
 plotDistribution('perimeter_mean', 5)
 plotDistribution('area_mean', 10)
plotDistribution('smoothness_mean', .5)
plotDistribution('compactness_mean' .5)
plotDistribution('concavity_mean' .5)
plotDistribution('concave points_mean' .5)
plotDistribution('symmetry_mean' .5)
plotDistribution('fractal_dimension_mean' .5)
correlation = data.corr()
#tick labels
matrix_cols = correlation.columns.tolist()
#convert to array
corr_array  = np.array(correlation)

def plotFeatureOneFeatureTwo(featureOne, featureTwo) :
   trace0 = go.Scatter(
      x = M[featureOne],
      y = M[featureTwo],
      name = 'malignant',
      mode = 'markers',
      marker = dict(color = 'red',
         line = dict(
```

```
                width = 1)))

    trace1 = go.Scatter(
        x = B[featureOne],
        y = B[featureTwo],
        name = 'benign',
        mode = 'markers',
        marker = dict(color = 'green',
            line = dict(
                width = 1)))

    layout = dict(title = featureOne +" "+"vs"+" "+ featureTwo,
                  yaxis = dict(title = featureTwo,zeroline = False),
                  xaxis = dict(title = featureOne, zeroline = False)
                 )

    plots = [trace0, trace1]

    fig = dict(data = plots, layout=layout)
    py.iplot(fig)
correlation = data.corr()
#tick labels
matrix_cols = correlation.columns.tolist()
#convert to array
corr_array  = np.array(correlation)

plotFeatureOneFeatureTwo('perimeter_mean','radius_worst')
plotFeatureOneFeatureTwo('area_mean','radius_worst')
plotFeatureOneFeatureTwo('texture_mean','texture_worst')
plotFeatureOneFeatureTwo('area_worst','radius_worst')
plotFeatureOneFeatureTwo('smoothness_mean','texture_mean')
plotFeatureOneFeatureTwo('radius_mean','fractal_dimension_worst')
```

```
plotFeatureOneFeatureTwo('texture_mean','symmetry_mean')
plotFeatureOneFeatureTwo('texture_mean','symmetry_se')
plotFeatureOneFeatureTwo('area_mean','fractal_dimension_mean')
 plotFeatureOneFeatureTwo('radius_mean','fractal_dimension_mean')
plotFeatureOneFeatureTwo('area_mean','smoothness_se')
plotFeatureOneFeatureTwo('smoothness_se','perimeter_mean')
trace0 = go.Scatter(
      x = M[featureOne],
      y = M[featureTwo],
      name = 'malignant',
      mode = 'markers',
      marker = dict(color = 'red',
        line = dict(
          width = 1)))

    trace1 = go.Scatter(
      x = B[featureOne],
      y = B[featureTwo],
      name = 'benign',
      mode = 'markers',
      marker = dict(color = 'green',
        line = dict(
          width = 1)))

trace2 = go.Scatter3d(x = B_pca['COMP1'],
                y = B_pca['COMP3'],
                z = B_pca['COMP2'],
                name = 'benign',
                mode = 'markers',
                marker = dict(size = 4,color= 'green',line = dict(width = 1))
               )
trace0 = go.Scatter(
```

```
        x = M[featureOne],
        y = M[featureTwo],
        name = 'malignant',
        mode = 'markers',
        marker = dict(color = 'red',
          line = dict(
            width = 1)))

    trace1 = go.Scatter(
        x = B[featureOne],
        y = B[featureTwo],
        name = 'benign',
        mode = 'markers',
        marker = dict(color = 'green',
          line = dict(
            width = 1)))
```

This code shows the exploratory data analysis conducted.It displays the visualized data. This is plotted using the library.

A.3.2 PCA

```
targetPCA = data['diagnosis']
dataPCA = data.drop('diagnosis', axis=1)

targetPCA = pd.DataFrame(targetPCA)
#To make a PCA, normalization of the data is essential
X_pca = dataPCA.values
X_std = StandardScaler().fit_transform(X_pca)
pca = PCA(svd_solver='full')
pca_std = pca.fit(X_std, targetPCA).transform(X_std)
pca_std = pd.DataFrame(pca_std)
pca_std = pca_std.merge(targetPCA, left_index = True, right_index = True, how = 'left')
pca_std['diagnosis'] = pca_std['diagnosis'].replace({1:'malignant',0:'benign'})
```

```
#explained_variance
    correlation = data.corr()
    #tick labels
    matrix_cols = correlation.columns.tolist()
    #convert to array
    corr_array  = np.array(correlation)


var_pca = pd.DataFrame(pca.explained_variance_ratio_)
var_pca = var_pca.T
col_list = list(v for v in chain(pca_std.columns[6:30]))
var_pca['OTHERS_COMP'] = var_pca[col_list].sum(axis=1)
var_pca.drop(var_pca[col_list],axis=1,inplace=True)
var_pca = var_pca.T
labels = ['COMP1','COMP2','COMP3','COMP4','COMP5','COMP6', 'COMP7 - 30']
colors = ['#FBFF00', '#68FF00', '#C80000', '#054D0C', '#8A03B9', '#FE2E01',
'#FFFFFF']

trace = go.Pie(labels = labels, values = var_pca[0].values, opacity = 0.8,
            textfont=dict(size=15),
            marker=dict(colors=colors,
                        line=dict(color='#9A9695', width=1.5)))
layout = dict(title =  'PCA : Components and explained variance (6 comp = 88.8%)')
  fig = dict(data = [trace], layout=layout)
py.iplot(fig)
plotFeatureOneFeatureTwo('perimeter_mean','radius_worst')
# plotFeatureOneFeatureTwo('area_mean','radius_worst')
# plotFeatureOneFeatureTwo('texture_mean','texture_worst')
# plotFeatureOneFeatureTwo('area_worst','radius_worst')
plotFeatureOneFeatureTwo('perimeter_mean','radius_worst')
# plotFeatureOneFeatureTwo('area_mean','radius_worst')
# plotFeatureOneFeatureTwo('texture_mean','texture_worst')
# plotFeatureOneFeatureTwo('area_worst','radius_worst')
```

```
plotFeatureOneFeatureTwo('perimeter_mean','radius_worst')
# plotFeatureOneFeatureTwo('area_mean','radius_worst')
# plotFeatureOneFeatureTwo('texture_mean','texture_worst')
# plotFeatureOneFeatureTwo('area_worst','radius_worst')
pca = PCA(n_components = 2)
pca_std = pca.fit(X_std, targetPCA).transform(X_std)
pca_std = pd.DataFrame(pca_std,columns = ['COMP1','COMP2'])
pca_std = pca_std.merge(targetPCA,left_index = True,right_index = True,how = 'left')
pca_std['diagnosis'] = pca_std['diagnosis'].replace({1:'malignant',0:'benign'})
def pca_scatter(target,color) :
   tracer = go.Scatter(x = pca_std[pca_std['diagnosis'] == target]['COMP1'] ,
                  y = pca_std[pca_std['diagnosis'] == target]['COMP2'],
                  name = target, mode = 'markers',
                  marker = dict(color = color,line = dict(width = 1))
                 )
   return tracer
layout = go.Layout(dict(title = 'PCA Scatter plot (2 comp = 63.3%)',
                  xaxis = dict(gridcolor = 'rgb(255, 255, 255)',
                        title = 'COMP1 = 44.3%',
                        zerolinewidth=1,ticklen=5,gridwidth=2),
                  yaxis = dict(gridcolor = 'rgb(255, 255, 255)',
                        title = 'COMP2 = 19.0%',
                        zerolinewidth=1,ticklen=5,gridwidth=2),
                  height = 800
                 ))
trace1 = pca_scatter('malignant','red')
trace2 = pca_scatter('benign','green')
plots = [trace2,trace1]
fig = go.Figure(data = plots,layout = layout)
py.iplot(fig)
pca = PCA(n_components = 3)
pca_std = pca.fit(X_std, targetPCA).transform(X_std)
```

```
pca_std = pd.DataFrame(pca_std,columns = ['COMP1','COMP2','COMP3'])
pca_std = pca_std.merge(targetPCA, left_index = True, right_index = True,how = 'left')
pca_std['diagnosis'] = pca_std['diagnosis'].replace({1:'malignant',0:'benign'})
M_pca = pca_std[(pca_std['diagnosis'] == 'malignant')]
B_pca = pca_std[(pca_std['diagnosis'] == 'benign')]
    correlation = data.corr()
    #tick labels
    matrix_cols = correlation.columns.tolist()
    #convert to array
    corr_array  = np.array(correlation)
trace1 = go.Scatter3d(x = M_pca['COMP1'],
                      y = M_pca['COMP3'],
                      z = M_pca['COMP2'],
                      mode = "markers",
                      name = "malignant",
                      marker = dict(size = 4,color = 'red',line = dict(width = 1))
                     )
trace2 = go.Scatter3d(x = B_pca['COMP1'],
                      y = B_pca['COMP3'],
                      z = B_pca['COMP2'],
                      name = 'benign',
                      mode = 'markers',
                      marker = dict(size = 4,color= 'green',line = dict(width = 1))
                     )



layout = go.Layout(dict(title = 'PCA Scatter plot (3 comp = 72.7%)',
                        scene = dict(camera = dict(up=dict(x= 0 , y=0, z=0),
                                                   center=dict(x=0, y=0, z=0),
                                                   eye=dict(x=1.25, y=1.25, z=1.25)),
                                     xaxis  = dict(title = 'COMP1',
```

```
                                gridcolor='rgb(255, 255, 255)',
                                zerolinecolor='rgb(255, 255, 255)',
                                showbackground=True,
                                backgroundcolor='rgb(230, 230,230)'),
                    yaxis  = dict(title = 'COMP3',
                                gridcolor='rgb(255, 255, 255)',
                                zerolinecolor='rgb(255, 255, 255)',
                                showbackground=True,
                                backgroundcolor='rgb(230, 230,230)'
                                ),
                    zaxis  = dict(title = 'COMP2',
                                gridcolor='rgb(255, 255, 255)',
                                zerolinecolor='rgb(255, 255, 255)',
                                showbackground=True,
                                backgroundcolor='rgb(230, 230,230)'
                                )),height = 600))
        plots = [trace1,trace2]
    fig  = go.Figure(data = plots,layout = layout)
    py.iplot(fig)
plotFeatureOneFeatureTwo('perimeter_mean','radius_worst')
# plotFeatureOneFeatureTwo('area_mean','radius_worst')
# plotFeatureOneFeatureTwo('texture_mean','texture_worst')
# plotFeatureOneFeatureTwo('area_worst','radius_worst')
plotFeatureOneFeatureTwo('perimeter_mean','radius_worst')
# plotFeatureOneFeatureTwo('area_median,'radius_worst')
# plotFeatureOneFeatureTwo('texture_median,'texture_worst')
# plotFeatureOneFeatureTwo('area_median,'radius_worst')
```

This code shows the Principal Component analysis. Principal Component Analysis is an unsupervised learning algorithm that is used for the dimensionality reduction in machine learning. It is a statistical process that converts the observations of correlated features into a set of linearly uncorrelated features with the help of orthogonal transformation

A.3.3 Confusion Matrix

```
# Confusion matrix
def plot_confusion_matrix(cm, classes,
                          normalize = False,
                          title = 'Confusion matrix"',
                          cmap = plt.cm.Blues) :
    plt.imshow(cm, interpolation = 'nearest', cmap = cmap)
    plt.title(title)
    plt.colorbar()
    tick_marks = np.arange(len(classes))
    plt.xticks(tick_marks, classes, rotation = 0)
    plt.yticks(tick_marks, classes)

    thresh = cm.max() / 2.
    for i, j in itertools.product(range(cm.shape[0]), range(cm.shape[1])) :
        plt.text(j, i, cm[i, j],
                 horizontalalignment = 'center',
                 color = 'white' if cm[i, j] > thresh else 'black')

    plt.tight_layout()
    plt.ylabel('True label')
    plt.xlabel('Predicted label')
        trace0 = go.Scatter(
            x = M[featureOne],
            y = M[featureTwo],
            name = 'malignant',
            mode = 'markers',
            marker = dict(color = 'red',
                line = dict(
                    width = 1)))
 df = pd.DataFrame(data = models_metrics)
df.rename(index={0:'Accuracy',1:'Precision', 2: 'Recall'},
```

```
            inplace=True)
ax = df.plot(kind='bar', figsize = (15,10), ylim = (0.94, 1),
      color = ['gold', 'lightgreen', 'lightcoral', 'lightskyblue'],
      rot = 0, title ='Models performance (cross val mean)',
      edgecolor = 'grey', alpha = 0.5)
for p in ax.patches:
   ax.annotate(str(p.get_height()), (p.get_x() * 1.01, p.get_height() * 1.0005))
plt.show()


# Show metrics
def show_metrics():
   tp = cm[1,1]
   fn = cm[1,0]
   fp = cm[0,1]
   tn = cm[0,0]
   print('Accuracy  =    {:.3f}'.format((tp+tn)/(tp+tn+fp+fn)))
   print('Precision =    {:.3f}'.format(tp/(tp+fp)))
   print('Recall    =    {:.3f}'.format(tp/(tp+fn)))
   print('F1_score  =    {:.3f}'.format(2*(((tp/(tp+fp))*(tp/(tp+fn)))/
                                 ((tp/(tp+fp))+(tp/(tp+fn))))))
```

This code shows the confusion matrix. The confusion matrix, also known as the error matrix, allows visualization of the performance of an algorithm.

A.3.4 Predictive Model 1

```
# Find best hyperparameters (accuracy)
log_clf = LogisticRegression(random_state = random_state)
param_grid = {
          'penalty' : ['l2','l1'],
          'C' : [0.001, 0.01, 0.1, 1, 10, 100, 1000]
          }
```

```
CV_log_clf = GridSearchCV(estimator = log_clf, param_grid = param_grid , scoring =
'accuracy', verbose = 1, n_jobs = -1)
CV_log_clf.fit(X_train, y_train)

best_parameters = CV_log_clf.best_params_
print('The best parameters for using this model is', best_parameters)
```

Here, in this code, the results of the first predictive model will be displayed.

A.3.5 Predictive model 2

```
# Find the best parameters (recall)
log2_clf = LogisticRegression(random_state = random_state)
param_grid = {
        'penalty' : ['l2','l1'],
        'C' : [0.001, 0.01, 0.1, 1, 10, 100, 1000],
        }

CV_log2_clf = GridSearchCV(estimator = log2_clf, param_grid = param_grid , scoring =
'recall', verbose = 1, n_jobs = -1)
CV_log2_clf.fit(X_train, y_train)

best_parameters = CV_log2_clf.best_params_
print('The best parameters for using this model is', best_parameters)
# Log w best hyperparameters (recall)
CV_log2_clf = LogisticRegression(C = best_parameters['C'],
                        penalty = best_parameters['penalty'],
                        random_state = random_state)

CV_log2_clf.fit(X_train, y_train)
layout = go.Layout(dict(title = 'PCA Scatter plot (3 comp = 72.7%)',
                scene = dict(camera = dict(up=dict(x= 0 , y=0, z=0),
                                        center=dict(x=0, y=0, z=0),
                                        eye=dict(x=1.25, y=1.25, z=1.25)),
```

```
                    xaxis  = dict(title = 'COMP1',
                                gridcolor='rgb(255, 255, 255)',
                                zerolinecolor='rgb(255, 255, 255)',
                                showbackground=True,
                                backgroundcolor='rgb(230, 230,230)'),
                    yaxis  = dict(title = 'COMP3',
                                gridcolor='rgb(255, 255, 255)',
                                zerolinecolor='rgb(255, 255, 255)',
                                showbackground=True,
                                backgroundcolor='rgb(230, 230,230)'
                               ),
                    zaxis  = dict(title = 'COMP2',
                                gridcolor='rgb(255, 255, 255)',
                                zerolinecolor='rgb(255, 255, 255)',
                                showbackground=True,
                                backgroundcolor='rgb(230, 230,230)'
                               )),height = 600))
        plots = [trace1,trace2]


y_pred = CV_log2_clf.predict(X_test)
y_score = CV_log2_clf.decision_function(X_test)
# Confusion maxtrix & metrics
cm = confusion_matrix(y_test, y_pred)
class_names = [0,1]
```

This code shows us results of predictive model 2

A.3.6 Voting classifier

```
# Ensemble, recall = 1.
y_score = voting_clf.predict_proba(X_test)[:,1] > 0.23
cm = confusion_matrix(y_test, y_score)
class_names = [0,1]
plt.figure()
```

```
plot_confusion_matrix(cm,
                      classes = class_names,
                      title = 'Ensemble Clf CM : recall = 100%')
plt.savefig('8')
plotLearningCurve(CVLogClf, 'Learning Curve For Logistic Model', X, y, (0.85,1.05),
10)
plt.savefig('7')
plt.show()
print("="*15)
print("Cross val Log ")
print("="*15)
crossLog = crossValMetrics(CVLogClf)
print("="*36)
print("Processing some more values and calulating, please hold...")
plotLearningCurve(selector, 'Learning Curve For Logistic Model with RFE', X, y,
(0.85,1.05), 10)
plt.show()
print("="*15)
print("Cross val Log with RFE")
print("="*15)
cross_selector = crossValMetrics(selector)
print("="*36)
print("Process completed with no errors")
print("="*36)
plt.show()
y_score = votingClf.predict_proba(X_test)[:,1] > 0.23
cm = confusion_matrix(y_test, y_score)
class_names = [0,1]
plt.figure()
plotConfusionMatrix(cm,
                      classes = class_names,
                      title = 'Ensemble Clf CM : recall = 100%')
```

```
plt.savefig('8')
plt.show()
print("="*21)
showMetrics()
print("="*21)
fpr, tpr, t = roc_curve(y_test, y_score)
plotRoc()

precision, recall, thresholds = precision_recall_curve(y_test, y_score)
plotPrecisionRecall()
show_metrics()
log_clf = LogisticRegression(random_state = random_state)
param_grid = {
          'penalty' : ['l2','l1'],
          'C' : [0.001, 0.01, 0.1, 1, 10, 100, 1000]
          }
models_metrics = {'logClf': [0.982, 0.990, 0.962],
              'selector': [0.974, 0.981, 0.948],
              'logTwoClf' : [0.974,0.976,0.953],
              'votingClf' : [0.979,0.985,0.958]
             }
df = pd.DataFrame(data = models_metrics)
df.rename(index={0:'Accuracy',1:'Precision', 2: 'Recall'},
              inplace=True)
ax = df.plot(kind='bar', figsize = (15,10), ylim = (0.94, 1),
       color = ['#054D0C', '#8A03B9', '#FE2E01', '#FBFF00'],
       rot = 0, title ='Models performance (cross val mean)',
       edgecolor = 'grey', alpha = 0.5)
for p in ax.patches:
   ax.annotate(str(p.get_height()), (p.get_x() * 1.01, p.get_height() * 1.0005))
plt.show()
# ROC curve
```

```
fpr, tpr, t = roc_curve(y_test, y_score)
plot_roc()
# Precision-recall curve
precision, recall, thresholds = precision_recall_curve(y_test, y_score)
plot_precision_recall()
```

This code displays how the voting classifier works.

A.3.7 Comparison of the accuracy, recall and precision

```
models_metrics = {'log_clf': [0.982, 0.990, 0.962],
                  'selector': [0.974, 0.981, 0.948],
                  'log2_clf' : [0.974,0.976,0.953],
                  'voting_clf' : [0.979,0.985,0.958]
                 }
plotLearningCurve(CVLogClf, 'Learning Curve For Logistic Model', X, y, (0.85,1.05), 10)
plt.savefig('7')
plt.show()
print("="*15)
print("Cross val Log ")
print("="*15)
crossLog = crossValMetrics(CVLogClf)
print("="*36)
print("Processing some more values and calulating, please hold...")
plotLearningCurve(selector, 'Learning Curve For Logistic Model with RFE', X, y, (0.85,1.05), 10)
plt.show()
print("="*15)
print("Cross val Log with RFE")
print("="*15)
cross_selector = crossValMetrics(selector)
print("="*36)
print("Process completed with no errors")
print("="*36)
df = pd.DataFrame(data = models_metrics)
```

```
df.rename(index={0:'Accuracy',1:'Precision', 2: 'Recall'},
          inplace=True)
ax = df.plot(kind='bar', figsize = (15,10), ylim = (0.94, 1),
     color = ['gold', 'lightgreen', 'lightcoral', 'lightskyblue'],
     rot = 0, title ='Models performance (cross val mean)',
     edgecolor = 'grey', alpha = 0.5)
for p in ax.patches:
   ax.annotate(str(p.get_height()), (p.get_x() * 1.01, p.get_height() * 1.0005))
plt.show()
```

This code shows the comparison between accuracy, recall and precision.

A.4 Installation Procedure

1)Any Web Browser

Install any web browser run the project and host the website.

2)Import the necessary libraries

We have used several libraries to plot graphs and detect if a tumour is malignant or not.For these features to be used, the libraries have to be imported.

3)XAMPP

The website requires XAMPP which helps us host the website.