BREAST CANCER PREDECTION USING MACHINE LEARNING TECHNIQUES

A PROJECT REPORT

Submitted by

ANGADA CHANDRA MOULI [RA2011003010664]

GADHAMSETTY NAVANEETH [RA2011003011335]

Under the Guidance of

Ms.M.RANJANI

Assistant Professor, Department of Computing Technologies

in partial fulfillment of the requirements for the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

DEPARTMENT OF COMPUTING TECHNOLOGIES

COLLEGE OF ENGINEERING AND TECHNOLOGY

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR– 603 203

NOVEMBER 2023

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR–603 203

BONAFIDE CERTIFICATE

Certified that 18CSP109L / I8CSP111L project report titled "**BREAST CANCER PREDICTION USING MACHINE LEARNING**" is the bonafide work of **A.CHANDRA MOULI[RA2011003010664]** and **G.NAVANEETH[RA2011003011335]** who carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported here in does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion for this or any other candidate.

Ms. M. RANJANI
SUPERVISOR
Assistant Professor
Department of Computing Technologies

Dr. G. USHA
PANEL HEAD
Associate Professor
Department of Computing Technologies

Dr. M. PUSHPALATHA
HEAD OF THE DEPARTMENT
Department of Computing Technologies

Department of Computing Technologies

SRM Institute of Science and Technology

Own Work Declaration Form

Degree/Course **:** B.Tech in Computer Science and Engineering

S**tudent Names** **:** A. CHANDRA MOULI, G. NAVANEETH

Registration Number: RA2011003010664, RA2011003011335

Title of Work **: Breast Cancer Prediction using machine learning techniques**

I/We here by certify that this assessment compiles with the University's Rules and Regulations relating to Academic misconduct and plagiarism, as listed in the University Website, Regulations, and the Education Committee guidelines.

I / We confirm that all the work contained in this assessment is our own except where indicated, and that we have met the following conditions:

- Clearly references / listed all sources as appropriate
- Referenced and put in inverted commas all quoted text (from books, web,etc.)
- Given the sources of all pictures, data etc that are not my own.
- Not made any use of the report(s) or essay(s) of any other student(s)either past or present
- Acknowledged in appropriate places any help that I have received from others (e.g fellow students, technicians, statisticians, external sources)
- Compiled with any other plagiarism criteria specified in the Course hand book / University website

I understand that any false claim for this work will be penalized in accordance with the University policies and regulations.

DECLARATION:
I am aware of and understand the University's policy on Academic misconduct and plagiarism and I certify that this assessment is my / our own work, except where indicated by referring, and that I have followed the good academic practices noted above. **Student 1 Signature:** **Student 2 Signature:** **Date:**

ACKNOWLEDGEMENT

We express our humble gratitude to **Dr. C. Muthamizhchelvan**, Vice-Chancellor, SRM Institute of Science and Technology, for the facilities extended for the project work and his continued support.

We extend our sincere thanks to Dean-CET, SRM Institute of Science and Technology, **Dr. T. V. Gopal**, for his invaluable support.

We wish to thank **Dr. Revathi Venkataraman**, Professor and Chairperson, School of Computing, SRM Institute of Science and Technology, for her support throughout the project work.

We are incredibly grateful to our Head of the Department, **Dr. M. Pushpalatha,** Professor, Department of Computing Technologies, SRM Institute of Science and Technology, for her suggestions and encouragement at all the stages of the project work.

We want to convey our thanks to our Project Coordinators, **Dr. M. Kanchana**, **Dr. G. Usha, Dr. R. Yamini and Dr. K. Geetha**, Panel Head, **Dr. G. Usha**, Associate Professor and Panel Members, **Dr. Anto Arockia Rosaline** Assistant Professor, **Dr. Pretty Diana Cyril** Assistant Professor and **Mrs. M. Ranjani** Assistant Professor, Department of Computing Technologies, SRM Institute of Science and Technology, for their inputs during the project reviews and support.

We register our immeasurable thanks to our Faculty Advisor, **Dr. Deeba Kannan**, Assistant Professor, Department of Computing Technologies, SRM Institute of Science and Technology, for leading and helping us to complete our course

Our inexpressible respect and thanks to our guide, **Ms. M. Ranjani**, Assistant Professor, Department of Computing Technologies, SRM Institute of Science and Technology, for providing us with an opportunity to pursue our project under his / her mentorship. He / She provided us with the freedom and support to explore the research topics of our interest. His / Her passion for solving problems and making a difference in the world has always been inspiring.

We sincerely thank all the staff and students of Computing Technologies Department, School of Computing, S.R.M Institute of Science and Technology, for their help during our project. Finally, we would like to thank our parents, family members, and friends for their unconditional love, constant support and encouragement.

ANGADA CHANDRA MOULI [RA2011003010664]

GADHAMSETTY NAVANEETH[RA2011003011335]

ABSTRACT

Breast cancer is one of the most prevalent and life-threatening diseases affecting women worldwide. Early detection is crucial for improving the prognosis and survival rates of patients. In recent years, predictive modelling and machine learning techniques have emerged as promising tools for breast cancer prediction, diagnosis, and risk assessment. This abstract provides an overview of the current state of research in breast cancer prediction using predictive modelling and highlights the key findings and challenges in this field.

The primary goal of breast cancer prediction models is to identify individuals at a higher risk of developing the disease. Several risk factors, including age, family history, genetic mutations, and lifestyle choices, contribute to an individual's likelihood of developing breast cancer. Predictive models use these factors to estimate a person's risk and provide a personalized risk assessment.

Machine learning algorithms, such as logistic regression, support vector machines, random forests, and deep neural networks, have been employed to build predictive models. These models leverage extensive datasets containing clinical and demographic information, mammographic images, and genetic data. By analyzing these diverse data sources, predictive models can identify patterns and associations that might be missed by traditional methods.

A critical component of breast cancer prediction models is the integration of multiple data types. Mammographic images, for example, are essential for early detection, as they can reveal abnormal tissue changes. Genetic data, on the other hand, can help identify individuals with a predisposition to breast cancer. Combining these data sources in a single model allows for more accurate predictions.

LIST OF FIGURES

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 GENERAL

Breast cancer remains a significant global public health concern, impacting the lives of millions of women and their families. As one of the most prevalent and potentially life-threatening diseases, early detection and timely intervention are paramount in improving survival rates and patient outcomes. Predictive modelling for breast cancer has emerged as a vital area of research, offering the promise of more accurate risk assessment and early diagnosis. This introduction outlines the compelling need for breast cancer prediction and highlights the significance of leveraging advanced technologies to combat this disease.

The Burden of Breast Cancer:

Breast cancer is a formidable adversary, affecting women of all ages, ethnicities, and socioeconomic backgrounds. According to the World Health Organization (WHO), it is the most common cancer in women worldwide, with approximately 2.3 million new cases diagnosed each year. In the United States alone, breast cancer ranks as the second leading cause of cancer-related deaths among women. The emotional, physical, and economic toll of breast cancer is immense, as it not only poses a risk to life but also leads to considerable financial strain on /healthcare systems and patients alike.

The Imperative of Early Detection:

The key to mitigating the impact of breast cancer lies in early detection. Research has consistently demonstrated that when breast cancer is diagnosed at an early, localized stage, the chances of successful treatment and long-term survival increase significantly. Mammography and clinical breast examinations have been the cornerstone of early detection efforts for decades, helping to identify abnormalities

in breast tissue. However, these methods are not infallible and are subject to limitations, including false positives and false negatives.

Enter Predictive Modelling:

In recent years, predictive modelling and machine learning have emerged as valuable tools in the fight against breast cancer. These techniques have the potential to augment existing screening methods and provide more accurate, personalized assessments of an individual's risk of developing breast cancer. By analyzing a multitude of factors, such as age, family history, genetic predisposition, and lifestyle choices, predictive models can offer insights that extend beyond traditional risk assessment.

Scope and Objectives:

The field of breast cancer prediction encompasses a wide range of methodologies, including machine learning algorithms, integration of diverse data sources (such as genetic information and mammographic images), and ethical considerations regarding data privacy and equitable access to these predictive technologies. This introduction sets the stage for exploring these various aspects of breast cancer prediction, emphasizing the urgency and significance of advancing research and innovation in this critical area.

1.2 Machine Learning and Deep Learning:

Machine Learning:

Machine learning is a subset of artificial intelligence (AI) that focuses on developing algorithms and statistical models that enable computers to improve their performance on a specific task through learning from data. In traditional programming, humans explicitly define the rules and logic for solving a problem. In contrast, in machine learning, the computer learns from data, discovering patterns and making predictions or decisions without being explicitly programmed.

Machine learning algorithms can be broadly categorized into three types:

Supervised Learning:
In supervised learning, algorithms learn from labelled data, which means they are provided with input-output pairs, and they learn to map inputs to outputs. Common applications include image recognition, spam email detection, and predicting house prices.
Unsupervised Learning: Unsupervised learning involves finding patterns and structure in data without explicit labels. Clustering and dimensionality reduction are typical unsupervised learning tasks. For example, it can be used in customer segmentation or anomaly detection.
Reinforcement Learning: Reinforcement learning is about training algorithms to make sequences of decisions to maximize a reward. It is widely used in areas such as game-playing, robotics, and autonomous vehicles.

Deep Learning:

Deep learning is a subfield of machine learning that focuses on neural networks with many layers, known as deep neural networks. Deep learning has gained significant attention and popularity due to its remarkable ability to learn complex patterns from large amounts of data. Deep neural networks are designed to automatically learn hierarchical features, making them well-suited for tasks such as image and speech recognition. Key characteristics of deep learning include:
Neural Networks: Deep learning models are based on artificial neural networks, which are inspired by the structure and function of the human brain. These networks consist of interconnected layers of artificial neurons.
Deep Architectures: Deep learning models have multiple hidden layers, allowing them to automatically learn and represent intricate features from raw data. This makes them highly effective for tasks like image and speech recognition, natural language processing, and autonomous driving.

Big Data: Deep learning thrives on extensive datasets. With the availability of large volumes of data and advances in computing power (such as GPUs), deep learning models can handle complex problems that were once deemed insurmountable.
Applications: Deep learning has achieved significant breakthroughs in various domains, including computer vision (object detection, image classification), natural language processing (language translation, sentiment analysis), and reinforcement learning (game-playing, robotics).

1.3 Difference between Machine Learning and Deep Learning:

Machine learning and deep learning are both subfields of artificial intelligence, but they differ in several key aspects. Here are the differences:

Machine Learning:
Machine Learning typically uses shallow models with a limited number of layers. It often requires manual feature engineering, where experts design and select relevant features. Machine Learning can work with smaller datasets effectively. It uses a variety of algorithms, including decision trees, support vector machines, and k-nearest neighbours. Generally, this provides more interpretable models, making it easier to understand how predictions are made. Machine Learning requires less computational power and is often feasible on standard CPUs. This is well-suited for a wide range of tasks, including classification, regression, clustering, and recommendation systems. Machine Learning trains faster on small to medium-sized datasets and performs well in many applications but may not match the accuracy of deep learning in specific domains like image recognition or language translation. Machine Learning often relies on domain expertise and human input to design effective models.

Deep Learning:

Deep Learning specifically focuses on deep neural networks with multiple hidden layers. It automatically learns features from raw data, reducing the need for extensive feature engineering. Deep Learning often requires large volumes of data to perform well and generalize effectively. Deep Learning primarily relies on neural networks and variations like convolutional neural networks (CNNs) and recurrent neural networks (RNNs). Deep neural networks are often considered "black boxes," making it challenging to interpret the reasoning behind their predictions. Deep Learning demands high computational resources, typically requiring GPUs or specialized hardware for training large models. It excels in specific domains, such as computer vision, natural language processing, and speech recognition. Deep Learning trains the deep neural networks can be time-consuming, especially on large datasets. Deep Learning achieves state-of-the-art performance in tasks where it's applied, outperforming traditional machine learning in certain cases.

1.4 Computer Vision:

Computer vision has played a significant role in breast cancer prediction and diagnosis. Leveraging image analysis, machine learning, and deep learning techniques, computer vision assists medical professionals in early detection and assessment of breast cancer. Below is an overview of how computer vision is applied in breast cancer prediction:

Mammographic Image Analysis: Mammography is a common screening method for breast cancer. Computer vision algorithms can analyze mammographic images to identify potential abnormalities, such as masses, calcifications, or architectural distortions. This aids in early detection by providing radiologists with more precise and objective information.

Image Segmentation: Computer vision algorithms can segment breast images to isolate regions of interest, such as lesions or tumors, from the surrounding tissue.

This enables accurate measurement of the size and characteristics of these abnormalities, which is crucial for diagnosis and monitoring.

Feature Extraction: Computer vision techniques extract quantitative features from mammographic images, such as texture, shape, and edge information. These features are then used as input for machine learning models to predict the likelihood of malignancy, thereby assisting in breast cancer risk assessment.

Computer-Aided Detection (CAD): CAD systems employ computer vision to automatically detect potential abnormalities in mammograms. These systems serve as a "second pair of eyes" for radiologists, helping them identify subtle signs of breast cancer that might be missed during manual inspection.

3D Imaging: Beyond 2D mammography, computer vision can be applied to analyze 3D breast images, such as breast tomosynthesis or magnetic resonance imaging (MRI). This provides a more comprehensive view of the breast tissue and enhances the accuracy of cancer detection.

Image Registration: Computer vision is used to align multiple breast images taken at different times or using different imaging modalities. This assists in tracking changes in abnormalities over time and determining whether they are benign or malignant.

Histopathology Image Analysis: In addition to radiological images, computer vision is applied to histopathology images of breast tissue obtained through biopsies. These algorithms aid pathologists in diagnosing breast cancer by analyzing the cellular and structural characteristics of tissue samples.

Risk Assessment: Computer vision can also be employed to assess breast cancer risk by analyzing images and other patient-specific data, such as family history and genetic markers. By combining image features with clinical information, predictive models can estimate an individual's risk of developing breast cancer.

Telemedicine: Computer vision technology enables telemedicine applications, allowing experts to remotely analyze breast images and provide diagnoses and recommendations. This is particularly valuable in underserved areas with limited

access to specialized healthcare.

Personalized Treatment: Computer vision can help determine the stage and subtype of breast cancer, which is crucial for tailoring personalized treatment plans for patients.

1.5 Computer Vision in Healthcare:

Computer vision in healthcare is a rapidly evolving field that leverages advanced image and video processing techniques to improve various aspects of healthcare, from diagnostics and treatment to administrative tasks. Here is an overview of how computer vision is transforming healthcare:

Medical Imaging Analysis: Computer vision is widely used in the analysis of medical images, including X-rays, MRIs, CT scans, and ultrasound images. It assists in the early detection and diagnosis of diseases, such as cancer, fractures, and neurological conditions.

Radiology Assistance: Radiologists benefit from computer vision by using AI-powered tools for the detection and quantification of abnormalities in medical images. These tools help in providing more accurate and efficient diagnoses.

Pathology and Histopathology: Computer vision aids pathologists in analyzing tissue samples. It can detect and classify cells and structures, helping diagnose conditions like cancer and autoimmune diseases.

Disease Detection and Monitoring: Computer vision enables continuous monitoring of patients, alerting healthcare providers to changes in a patient's condition. This is particularly valuable for monitoring chronic diseases, such as diabetes, or tracking patient movement in a hospital.

Surgical Assistance: During surgery, computer vision can provide real-time guidance to surgeons. This includes identifying critical structures, enhancing precision, and minimizing the risk of complications.

Telemedicine: Telemedicine relies on computer vision for remote consultations.

Healthcare providers can assess patients, make diagnoses, and provide treatment recommendations via video conferencing and image analysis.

Remote Monitoring: Wearable devices equipped with computer vision can monitor vital signs and health parameters. This data can be transmitted to healthcare providers in real time, allowing for early intervention if necessary.

Falls and Activity Monitoring: In elderly care, computer vision can help monitor patients to prevent falls or detect unusual activity that might indicate a health issue. This is particularly important for aging populations.

Medication Management: Computer vision systems can help ensure medication adherence by verifying that patients are taking the correct medication and dosage. They can also detect signs of medication side effects.

Administrative Efficiency: Computer vision streamlines administrative tasks, such as patient registration, medical coding, and billing. This reduces the administrative burden on healthcare staff and minimizes errors.

Drug Discovery: Computer vision is used in drug discovery to analyze the interactions of potential drug compounds with biological targets, accelerating the identification of new medications.

Healthcare Robotics: Robots with computer vision capabilities are used in healthcare settings for tasks such as medication delivery, patient transport, and disinfection, reducing the workload on healthcare staff.

Monitoring Infectious Diseases: Computer vision systems can monitor the movement of individuals in high-risk areas to track the spread of infectious diseases and ensure compliance with safety measures.

Early Warning Systems: Computer vision algorithms can identify early warning signs of various health conditions, such as changes in skin color, posture, or facial expressions, which can be indicative of specific diseases or emotional states.

1.6 Logistic Regression:

Logistic regression is a fundamental and versatile machine learning algorithm used

for binary classification tasks. It is particularly well-suited for situations where the dependent variable or target variable is categorical and has two possible classes, such as "yes" or "no," "spam" or "not spam," or "positive" or "negative." Logistic regression helps in predicting the probability of an observation belonging to one of these two classes based on one or more predictor variables. Here's a closer look at the key aspects of logistic regression:

Sigmoid Function: The logistic regression model uses the sigmoid function to transform a linear combination of predictor variables into a value between 0 and 1. This value represents the probability that an observation belongs to the positive class. The sigmoid function's formula is:

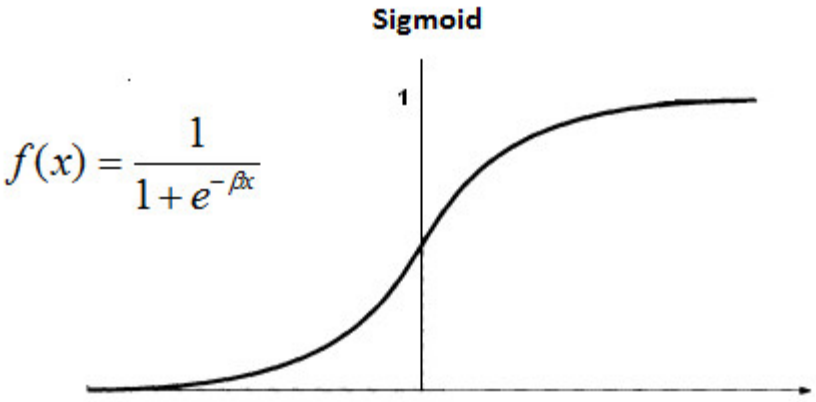


Fig 1.6 Logistic Regression

Model Parameters: Logistic regression has two main parameters: coefficients (weights) and an intercept (bias). These parameters are learned during the training process. The coefficients represent the strength and direction of the relationship between predictor variables and the target variable. The intercept is a constant that shifts the decision boundary. Maximum Likelihood Estimation: During training, logistic regression aims to find the optimal values for the model parameters by maximizing the likelihood of the observed data given the model.

Maximum likelihood estimation helps determine the parameters that make the observed data most probable under the model.

Decision Boundary: Logistic regression models create a decision boundary, which is a hyperplane that separates the two classes in the feature space.

The shape of the decision boundary depends on the number of predictor variables and their relationships. Regularization: Logistic regression models can be regularized to prevent overfitting. Common regularization techniques include L1 (Lasso) and L2 (Ridge) regularization, which add penalty terms to the model's cost function to discourage overly complex models.

Evaluation Metrics: Logistic regression models are assessed using various evaluation metrics, including accuracy, precision, recall, F1-score, and the area under the receiver operating characteristic curve (AUC-ROC). These metrics help determine the model's performance in classifying data correctly. Applications of logistic regression is widely used in fields such as healthcare (for disease prediction), finance (for credit scoring), marketing (for customer churn prediction), and many others where binary classification is essential.

Limitations: Logistic regression is primarily designed for binary classification tasks and may not perform well when there are more than two classes. It assumes that the relationship between predictor variables and the log-odds of the target variable is linear, which may not always be the case.

1.7 Decision Tree:

A decision tree is a fundamental machine learning algorithm used for both classification and regression tasks. It's a graphical representation of a decision-making process that resembles an inverted tree, with a root node at the top, branches, and leaves. Decision trees are easy to understand and interpret, making them valuable tools in various fields, including finance, healthcare, and business.

Structure of a Decision Tree:

Root Node: The topmost node represents the initial decision or the feature that best separates the data. These nodes represent intermediate decisions or features used for data splitting. The branches represent the outcomes or paths following each decision.
Leaves (Terminal Nodes): The terminal nodes represent the final outcome or class label, in the case of classification, or a numerical value, in the case of regression.

How Decision Trees Work:

Decision trees work by recursively partitioning the dataset based on the most informative features. They aim to create subsets of data that are as pure as possible concerning the target variable (i.e., they minimize classification or regression errors).
At each node, the decision tree algorithm selects the feature that provides the best split, usually based on metrics like Gini impurity, information gain, or mean squared error.
This process continues until certain stopping criteria are met, such as a maximum tree depth, a minimum number of samples in a leaf, or a minimum impurity threshold.

Advantages of Decision Trees:

Interpretability: Decision trees are easy to understand and visualize. They provide a transparent decision-making process that can be comprehended by non-technical stakeholders.
Applicability: Decision trees can be applied to both classification and regression tasks, making them versatile for various machine learning problems.
Data Preprocessing: They can handle missing values and don't require extensive data preprocessing.
Feature Selection: Decision trees can identify and prioritize important features, aiding in feature selection.
Scalability: Decision trees can handle both small and large datasets efficiently.

Challenges and Limitations:

Overfitting: Decision trees can easily overfit the data by creating overly complex trees that perform well on the training data but poorly on unseen data. This can be mitigated using techniques like pruning.

Bias: Decision trees can be biased if they select certain features as more important due to their position in the tree. Random Forest and Gradient Boosting algorithms can address this bias.

Inherent Error: Decision trees are susceptible to errors as they make decisions based on a single feature at a time, which may not capture complex relationships in the data.

Instability: Small changes in the data can result in significantly different decision trees, making them less stable compared to some other algorithms.

1.8 Random Forest Classifier:

The Random Forest Classifier is a versatile and powerful machine learning algorithm used for both classification and regression tasks. It belongs to the ensemble learning family, which means it combines the predictions of multiple individual models (decision trees) to make more accurate and robust predictions. Here's an overview of the Random Forest Classifier and its key characteristics:

Ensemble Learning: Random Forest is an ensemble of decision trees. Each tree in the forest is trained on a random subset of the training data and a random subset of the features. This reduces the risk of overfitting and makes the model more robust.

Decision Trees: Decision trees are the base learners in a Random Forest. These are simple yet effective models that split the data into subsets based on feature conditions, creating a hierarchical structure of decisions. Random Forest combines the predictions from multiple decision trees to make a final prediction.

Bootstrap Aggregating (Bagging): Random Forest employs a technique called bagging, where it generates multiple subsets (bootstrapped samples) from the

training data. Each decision tree is trained on one of these subsets. This diversity helps reduce the variance and improve the model's generalization.

Feature Randomness: In addition to using bootstrapped data samples, Random Forest introduces randomness in feature selection for each split within each decision tree. This decorrelates the trees, making them more independent and further reducing overfitting.

Prediction Process: When making predictions, Random Forest aggregates the predictions from individual trees by taking a majority vote (for classification tasks) or an average (for regression tasks). This ensemble approach often results in more accurate and stable predictions.

Robust to Overfitting: Random Forest is inherently resistant to overfitting, which is a common problem in decision trees. By averaging the predictions of multiple trees, it tends to produce more reliable and generalizable results.

Feature Importance: Random Forest provides a measure of feature importance, allowing you to understand which features have the most influence on the model's predictions. This information is valuable for feature selection and understanding the underlying data.

Versatility: Random Forest is applicable to a wide range of problems, including classification and regression tasks. It is commonly used in various domains, such as finance, healthcare, marketing, and more.

Handling Imbalanced Data: Random Forest is robust when dealing with imbalanced datasets, as it can weigh the importance of each class during training and make better predictions for minority classes.

CHAPTER 2
LITERATURE SURVEY

2.1 Motivation:

[1] Early Detection and Improved Survival Rates: One of the primary motivations is the potential to save lives through early detection. Research into breast cancer prediction aims to identify methods and models that can detect the disease at its earliest stages when treatment is most effective, leading to improved survival rates.
Reducing the Burden of Breast Cancer: Breast cancer is a significant public health concern worldwide. Motivated by the desire to reduce the physical, emotional, and economic burden of breast cancer on individuals and healthcare systems, researchers seek more accurate and efficient prediction methods.
Enhancing Screening and Diagnosis: The existing methods for breast cancer screening and diagnosis, such as mammography, have limitations, including false positives and false negatives. Motivated by the need for more reliable and precise methods, research in breast cancer prediction aims to complement or improve upon current practices.

[4] Personalized Medicine: The motivation for personalized breast cancer prediction is driven by the desire to tailor treatment plans to individual patients. By predicting a person's risk of developing breast cancer or assessing the likelihood of cancer subtypes, treatment decisions can become more precise and effective.
Advances in Machine Learning and Data Science: With the rapid advancement of machine learning and data science techniques, researchers are motivated to leverage these tools to better understand and predict breast cancer. Machine learning allows for the integration of diverse data sources, including genetic, clinical, and imaging data, to enhance prediction models.
Ethical and Equitable Healthcare: The motivation for breast cancer prediction research extends to ensuring equitable access to early detection and risk assessment.

Efforts are made to bridge healthcare disparities, making predictive models accessible to diverse populations and socioeconomic groups.

Improved Quality of Life: Early detection through breast cancer prediction can lead to less aggressive treatment and reduced side effects, ultimately improving the quality of life for patients. This is a significant motivating factor for both researchers and healthcare practitioners.

[7] Cost-Effective Healthcare: By accurately predicting breast cancer risk and early stages, healthcare costs associated with late-stage treatments and hospitalization can be reduced. This cost-effectiveness serves as a practical motivation for both healthcare providers and policymakers.

Progress in Research and Innovation: Researchers are motivated to continually advance the field of breast cancer prediction through innovation. Each new study contributes to the collective understanding, pushing the boundaries of what is possible in terms of prediction accuracy and risk assessment.

[3] Global Health Impact: The impact of breast cancer on women's health is not limited to one region. Research in breast cancer prediction is motivated by the opportunity to make a global impact, reducing the disease's prevalence and mortality on a worldwide scale.

2.2 Objective

Comprehensive Understanding: To gain a comprehensive understanding of the state of the art in breast cancer prediction, including the methodologies, algorithms, and datasets used in previous research.

[9] Identify Key Trends: To identify and highlight the key trends, emerging techniques, and breakthroughs in the field of breast cancer prediction. This includes understanding the evolution of predictive models and their performance over time.

Assess Performance: To evaluate the performance of different breast cancer prediction models and techniques, taking into account metrics such as sensitivity,

specificity, accuracy, and the area under the receiver operating characteristic curve (AUC-ROC).

Comparison of Approaches: To compare various machine learning and deep learning approaches, highlighting the strengths and weaknesses of each method in terms of predictive accuracy, interpretability, and generalizability.

Data Sources and Preprocessing: To examine the sources of data used in breast cancer prediction research, including mammographic images, genetic information, clinical data, and other relevant sources. Also, to investigate how data preprocessing and feature extraction impact the predictive models.

[2] Model Interpretability: To explore the level of interpretability of predictive models, as model transparency and understanding are essential in medical applications. This includes discussing techniques that make models more interpretable.

Ethical and Privacy Considerations: To address ethical considerations in breast cancer prediction, including issues related to data privacy, informed consent, and the potential biases that might exist in the datasets used.

Clinical Relevance: To assess the clinical relevance of the predictive models. This involves understanding whether the models developed in research studies have the potential to be integrated into clinical practice to aid in early detection, risk assessment, and treatment planning.

[6] Challenges and Open Questions: To identify challenges, limitations, and open questions in the field of breast cancer prediction, and to suggest areas where further research is needed.

Guidelines and Best Practices: To derive guidelines and best practices for future research in breast cancer prediction, including recommendations for data collection, model evaluation, and ethical considerations.

CHAPTER 3
ARCHITECTURE DIAGRAM AND FOUNDATION FOR CANCER DETECTION

3.1 Architecture Diagram

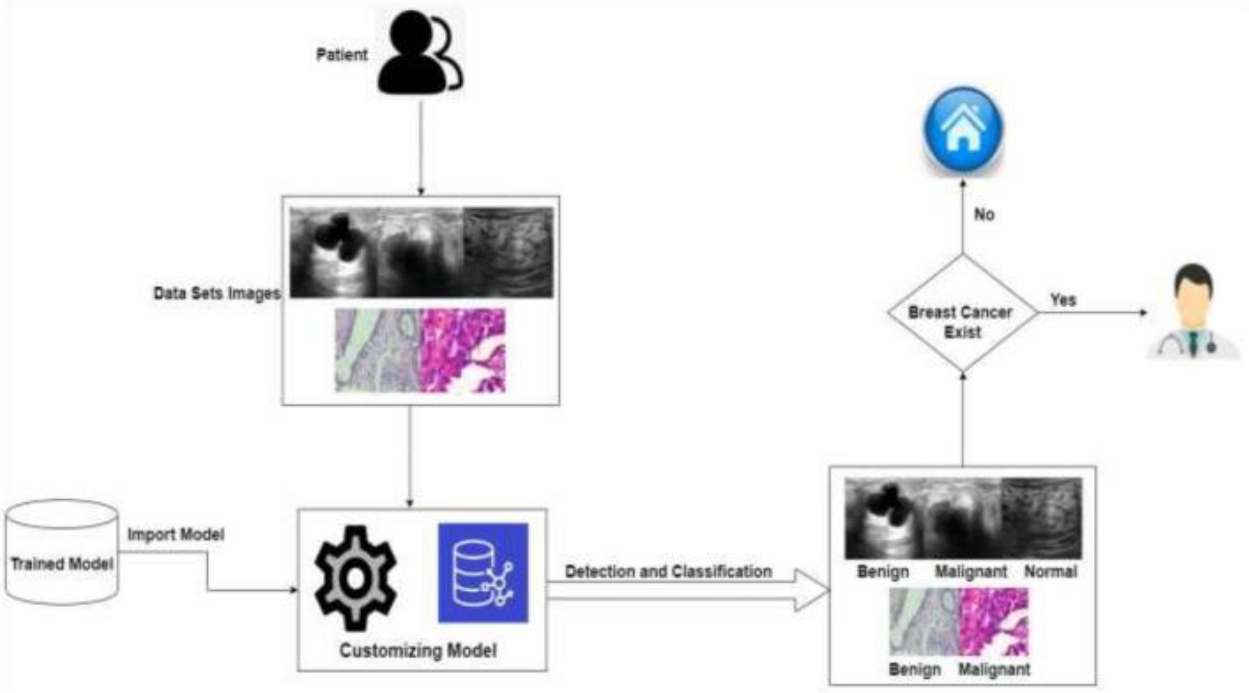


Fig 3.1 Architecture Diagram

This architecture diagram shows us the mechanism of how this project works. Initially we have a dataset which consists of information regarding the tissues associated with the cancer and many other parameters such as radius, concavity mean etc. At first we had inserted the dataset and had labelled them into 0's and 1's from benign and malignant tissue. This helps the model to classify among them. Now, we had split the dataset into two halves named training set and testing test which has 3:1 ratio. As the project needs more than one model for good accuracy, we have considered 3 models. The considered machine learning Techniques are Logistic Regression, Decision Tree, Random Forest Classifier.

Now we'll train the model with training set and test the model on the testing dataset. At-last by using the confusion matrix we found the precision, F1 score, recall, support, accuracy for all the three models. The one which had got more accuracy is considered and at our end we got "Random Forest Classifier" as the best one with 96.7 accuracy.

3.2 Importing Libraries

In a machine Learning project, the most fundamental step is to import Libraries. The imported libraries for this project are as follows:

```
[ ]  import pandas as pd
     import numpy as np
     import matplotlib.pyplot as plt
     import seaborn as sns
     import scipy.stats as s
     from sklearn.metrics import classification_report
```

Fig 3.2 Importing the libraries

Pandas (import pandas as pd): Pandas is a popular data manipulation library that provides data structures and functions to work with structured data, such as tables and time series. It allows you to read, write, filter, clean, and analyze data efficiently. The primary data structures in Pandas are Data frames and Series.

NumPy (import numpy as np): NumPy, short for Numerical Python, is a fundamental library for numerical and scientific computing in Python. It provides support for large, multi-dimensional arrays and matrices, along with a wide range of mathematical functions to operate on these arrays. It is the foundation for various numerical and scientific libraries in Python.

Matplotlib (import matplotlib.pyplot as plt): Matplotlib is a comprehensive 2D plotting library for creating static, animated, or interactive visualizations in Python. It allows you to create a wide range of charts, graphs, and plots, making it a valuable tool for data visualization.

Seaborn (import seaborn as sns): Seaborn is a high-level data visualization library built on top of Matplotlib. It is designed for creating informative and attractive statistical graphics. Seaborn simplifies the process of creating complex plots, including heatmaps, pair plots, and violin plots, with just a few lines of code.

Scipy (import scipy.stats as s): SciPy is an open-source library for mathematics, science, and engineering. The 'stats' module within SciPy provides a wide range of statistical functions, probability distributions, and statistical tests. It is particularly useful for statistical analysis and hypothesis testing.

Scikit-Learn (from sklearn.metrics import classification_report): Scikit-Learn, often referred to as sklearn, is a machine learning library in Python. The 'classification_report' function from Scikit-Learn's 'metrics' module is used for reporting performance metrics of a classification model. It calculates values such as precision, recall, F1-score, and support for each class in a classification problem, providing insights into the model's performance.

3.3 Importing Dataset:

The Second step is importing the dataset.

Fig 3.3 Importing the dataset

These commands help's us to insert the dataset and in the 3rd line i.e after the brackets we give the path of the dataset in the computer and the last command helps us to display the contents of the dataset in the google collab.

CHAPTER 4

DESIGN AND IMPLEMENTATION OF DETECTION MODEL

4.1 Dataset:

In a breast cancer prediction project, the dataset plays a pivotal role as it forms the foundation for training and evaluating machine learning models. These datasets typically contain a wide range of information, including clinical, genetic, and imaging data related to breast cancer. The data is meticulously collected and curated to enable the development of predictive models for early detection, risk assessment, and treatment planning. Clinical data in the dataset may include patient demographics, family history of cancer, and lifestyle factors. Genetic data often involves information about genetic mutations or variations that may influence a person's susceptibility to breast cancer. Additionally, mammographic images and other medical imaging data are crucial components of the dataset, as they provide detailed insights into breast tissue structure and abnormalities.

Furthermore, these datasets are typically labeled, which means they include information about whether a patient has been diagnosed with breast cancer or not. This labeling allows for the supervised training of machine learning models, enabling them to learn patterns and associations that can aid in predicting breast cancer cases accurately.

The quality and size of the dataset are critical factors that can significantly impact the performance of predictive models. A larger dataset with diverse samples can lead to more robust and accurate models. Feature engineering, data preprocessing, and thorough data analysis are often conducted to ensure that the dataset is well-prepared for model training. The data used for the experiments was acquired from Kaggle. This dataset is Break-Hist Dataset consisting of four directories representing the magnification of the images respectively i.e 100X, 200X, 400X and 40X. The dataset consists of 7,858 instances in total which are divided into the four magnification directories. Each magnification directory consists of two directories representing the

tumours i.e Benign and Malignant.

.

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	...	texture_worst	perimeter_worst
0	842302	M	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.3001	0.14710	...	17.33	184.60
1	842517	M	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.0869	0.07017	...	23.41	158.80
2	84300903	M	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.1974	0.12790	...	25.53	152.50
3	84348301	M	11.42	20.38	77.58	386.1	0.14250	0.28390	0.2414	0.10520	...	26.50	98.87
4	84358402	M	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.1980	0.10430	...	16.67	152.20

5 rows × 33 columns

Fig 4.1 Visualizing the dataset

4.2 Data Preprocessing:

Feature Selection The importance of feature selection in a machine learning model is inevitable. It turns the data to be free from ambiguity and reduces the complexity of the data. Also, it reduces the size of the data, so it is easy to train the model and reduces the training time. It avoids over fitting of data. Selecting the best feature subset from all the features increases the accuracy. Some feature selection methods are wrapper methods, filter methods, and embedded methods.

Recursive Feature Elimination:

RFE is a wrapper-type feature selection algorithm. This means that a different machine learning algorithm is given and used in the core of the method, is wrapped by RFE, and used to help select features. This is in contrast to filter-based feature selections that score each feature and select those features with the largest (or smallest) score. Technically, RFE is a wrapper-style feature selection algorithm that also uses filter-based feature selection internally. RFE works by searching for a subset of features by starting with all features in the training dataset and successfully removing features until the desired number remains. This is achieved by fitting the given machine learning algorithm used in the core of the model, ranking features by importance, discarding the least important features, and re-fitting the model.

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 modelling. 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

CROSS-VALIDATION

In machine learning, we couldn't fit the model on the training data and can't say that the model will work accurately for the real data. For this, we must assure that our model got the correct patterns from the data, and it is not getting up too much noise. For this purpose, we use the cross-validation technique. Cross validation is a technique used in machine learning to evaluate the performance of a model on unseen data. It involves dividing the available data into multiple folds or subsets, using one of these folds as a validation set, and training the model on the remaining folds. This process is repeated multiple times, each time using a different fold as the validation set. Finally, the results from each validation step are averaged to produce a more robust estimate of the model's performance.

The main purpose of cross validation is to prevent over fitting, which occurs when a model is trained too well on the training data and performs poorly on new, unseen

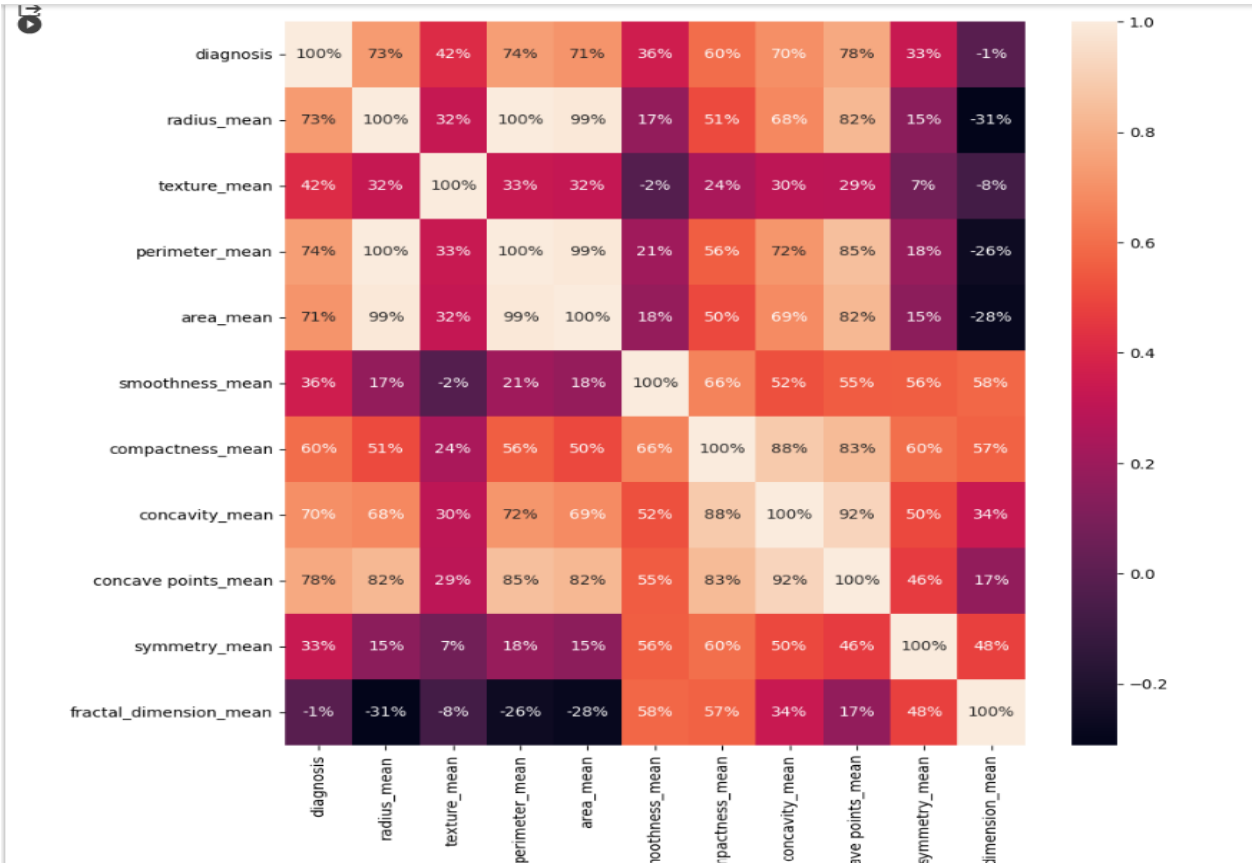
data. By evaluating the model on multiple validation sets, cross validation provides a more realistic estimate of the model's generalization performance, i.e., its ability to perform well on new, unseen data. There are several types of cross validation techniques, including k-fold cross validation, leave-one-out cross validation, and stratified cross validation. The choice of technique depends on the size and nature of the data, as well as the specific requirements of the modelling problem.

Cross-validation is a technique in which we train our model using the subset of the data-set and then evaluate using the complementary subset of the dataset. The three steps involved in cross-validation are as follows:

Reserve some portion of sample data-set.

Using the rest data-set train the model.

Test the model using the reserve portion of the data-set.



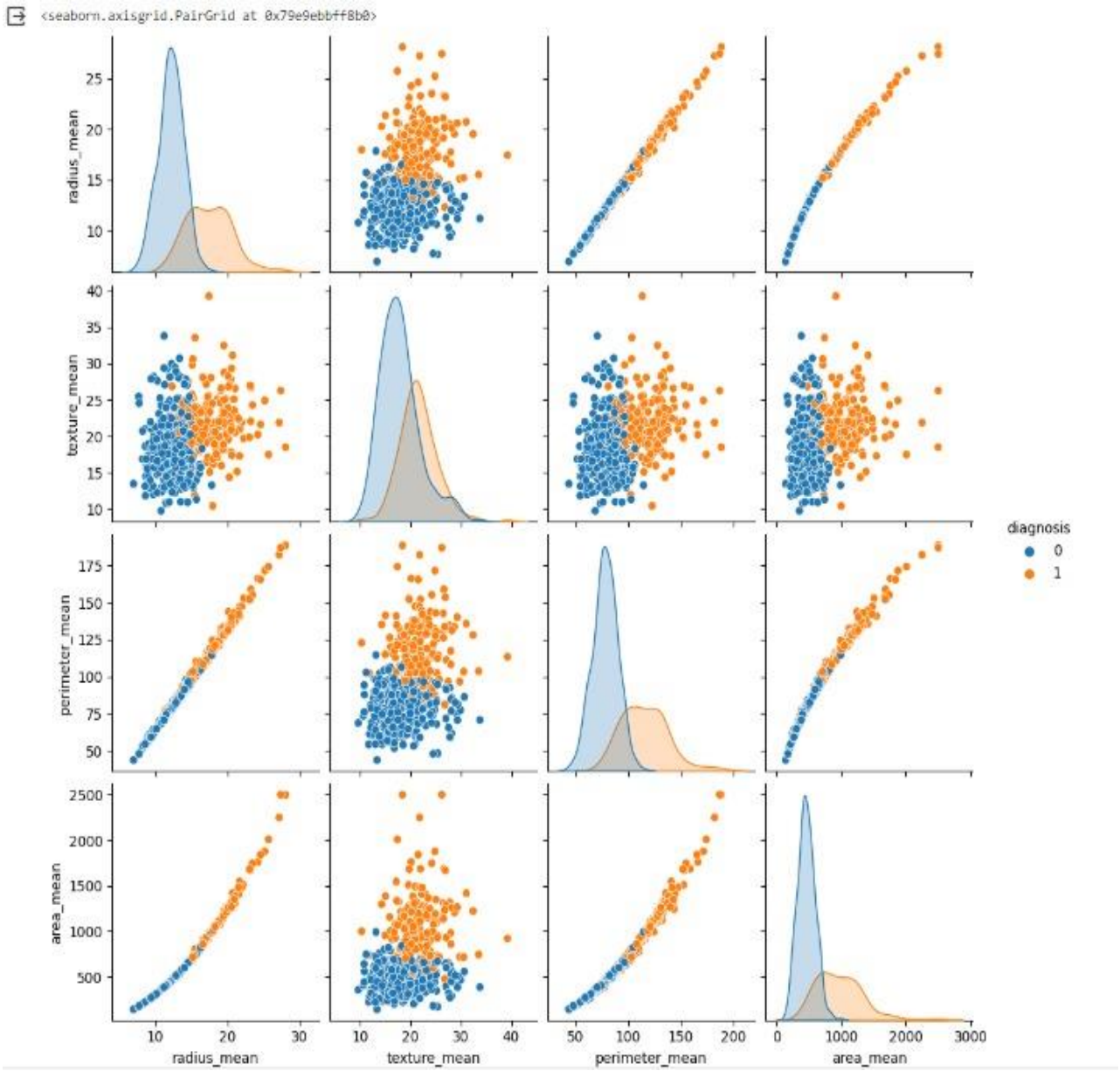


Fig 4.2 Heat map and Data preprocessing

4.3 Logistic Regression Model:

```
#Logistic Regression
from sklearn.linear_model import LogisticRegression
log = LogisticRegression(random_state=0)
log.fit(X_train,Y_train)
```

Fig 4.3 Logistic Regression model

The first line imports the Logistic Regression model class from the linear model module of the scikit-learn library. scikit-learn is a popular Python library for machine

learning and data science, and Logistic Regression is a classification algorithm used for binary and multiclass classification tasks. The second line, a logistic regression model is created and assigned to the variable log. The random_state parameter is set to 0. The random_state parameter is used to control the randomness involved in the logistic regression algorithm, such as data shuffling and random initialization of model parameters. Setting it to a specific value (in this case, 0) ensures that the results will be reproducible.

The third line fits (trains) the logistic regression model log using the training data.
X_train typically represents the feature matrix, which contains the input features or independent variables used to make predictions.
Y_train represents the target values or labels associated with the training data, which are the values the model is trying to predict or classify.
The `fit` method adjusts the model's parameters to find the best fit for the training data. In the case of logistic regression, it finds the optimal weights for the linear combination of features to make predictions about the target variable.

After training, the log model is ready to make predictions on new, unseen data. You can use the predict method to obtain predictions for the target variable based on a given set of features.

4.4 Decision Tree Model:

```
#Decision Tree
from sklearn.tree import DecisionTreeClassifier
tree=DecisionTreeClassifier(criterion = 'entropy',random_state=0)
tree.fit(X_train,Y_train)
```

Fig 4.4 Decision Tree Model

The first line imports the Decision Tree Classifier class from the tree module of the scikit-learn library. A Decision Tree is a supervised learning algorithm used for both classification and regression tasks. It builds a tree-like structure that recursively

splits the data into subsets based on the values of features to make predictions. In the second line, a Decision Tree Classifier is created and assigned to the variable `tree`. The constructor of the Decision Tree Classifier class is called with the following parameters:

criterion='entropy': This parameter specifies the criterion used to measure the quality of a split. In this case, 'entropy' is used, which measures the impurity of a node in the decision tree. Other common options include 'gini' and 'mse' for classification and regression tasks, respectively.

`random_state=0`: The `random_state` parameter is used to ensure reproducibility. Setting it to a specific value (in this case, 0) means that the randomness involved in the decision tree's internal operations, like feature selection and data shuffling, will produce consistent results when the code is run multiple times.

`tree.fit(X_train, Y_train)`: This line fits (trains) the Decision Tree Classifier `tree` using the training data.

`X_train` typically represents the feature matrix, which contains the input features or independent variables used for making predictions.

`Y_train` represents the target values or labels associated with the training data, which are the values the model is trying to predict or classify.

The `fit` method builds the decision tree by recursively splitting the data based on the chosen criterion (in this case, 'entropy') and finding the optimal decision boundaries that best separate the classes in the training data. After training, the `tree` model is ready to make predictions on new, unseen data using the `predict` method.

4.5 Random Forest Classifier Model:

```
#Forest Classifier
from sklearn.ensemble import RandomForestClassifier
forest=RandomForestClassifier(n_estimators = 10, criterion = 'entropy',random_state=0)
forest.fit(X_train, Y_train)
```

Fig 4.5 Random Forest Classifier Model

The first line imports the `Random Forest Classifier` class from the `ensemble` module of the `scikit-learn` library. Random Forest is an ensemble learning algorithm that combines the predictions of multiple decision trees to improve the accuracy and robustness of classification tasks. The second line, a Random Forest Classifier is created and assigned to the variable `forest`. The constructor of the `Random Forest Classifier` class is called with the following parameters:

`n_estimators=10`: This parameter specifies the number of decision trees to be included in the random forest. In this case, `n_estimators` is set to 10, meaning the random forest will consist of 10 decision trees.

`criterion='entropy'`: This parameter specifies the criterion used for measuring the quality of splits when building individual decision trees within the random forest. 'Entropy' measures the impurity of a node. Other common options include 'gini' for measuring Gini impurity. `random_state=0`: The `random_state` parameter is used to ensure reproducibility. Setting it to a specific value (in this case, 0) makes the results reproducible, as the randomness in the random forest's internal operations, such as data shuffling and feature selection, will produce consistent results when the code is run multiple times.

The third line fits (trains) the Random Forest Classifier `forest` using the training data. `X_train` typically represents the feature matrix, which contains the input features or independent variables used to make predictions. `Y_train` represents the target values or labels associated with the training data, which are the values the model is trying to predict or classify. The `fit` method builds the random forest by training multiple decision trees on bootstrapped subsets of the training data and selecting random subsets of features. The final prediction is made by aggregating the predictions of individual decision trees through a majority vote (for classification tasks) or averaging (for regression tasks).

CHAPTER 5

CODING AND TESTING

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import scipy.stats as s
from sklearn.metrics import classification_report
from google.colab import files

uploaded = files.upload()
df = pd.read_csv('data (1).csv')
df.head()

df.shape
df.isna().sum()
df = df.dropna(axis=1)
df.shape
df['diagnosis'].value_counts()
df.dtypes

from sklearn.preprocessing import LabelEncoder

labelencoder_Y = LabelEncoder()

df.iloc[:,1] = labelencoder_Y.fit_transform(df.iloc[:,1].values)


df.iloc[:,1]

sns.pairplot(df.iloc[:,1:6], hue='diagnosis')

df.head(5)

df.iloc[:,1:12].corr()

plt.figure(figsize=(10,10))

sns.heatmap(df.iloc[:,1:12].corr(), annot=True, fmt = '.0%')

X = df.iloc[:,2:31].values

Y = df.iloc[:,1].values

from sklearn.model_selection import train_test_split

X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.25 , random_state = 0)

from sklearn.preprocessing import StandardScaler

sc = StandardScaler()

X_train = sc.fit_transform(X_train)

X_test = sc.fit_transform(X_test)
```

```
X_train
def models(X_train, Y_train):
  #Logistic Regression
  from sklearn.linear_model import LogisticRegression
  log = LogisticRegression(random_state=0)
  log.fit(X_train,Y_train)

  #Decision Tree
  from sport RandomForestClassifier
  forest=RandomForestClassifier(n_estimators = 10, criterion = 'entropy',random_state=0)
  forest.fit(X_train, Y_train)klearn.tree import DecisionTreeClassifier
  tree=DecisionTreeClassifier(criterion = 'entropy',random_state=0)
  tree.fit(X_train,Y_train)

  #Forest Classifier
  from sklearn.ensemble im

  #Printing the models accuracy for training the data
  print('[0]Logistic Regression Training Accuracy:', log.score(X_train, Y_train))
  print('[1]Decision Tree Classifier Training Accuracy:', tree.score(X_train, Y_train))
  print('[2]Random Forest Classifier Training Accuracy:', forest.score(X_train, Y_train))

  return log, tree, forest
  model = models(X_train, Y_train)
from sklearn.metrics import confusion_matrix
for i in range(len(model)):
  print('Model ',i)
  cm = confusion_matrix(Y_test, model[i].predict(X_test))

  TP = cm[0][0]
  TN = cm[1][1]
  FN = cm[1][0]
  FP = cm[0][1]
```

```
print(cm)
print('Testing Accuracy = ', (TP + TN)/(TP + TN + FN + FP))
print()
from sklearn.metrics import classification_report
from sklearn.metrics import accuracy_score
for i in range(len(model)):
print('Model ',i)
print( classification_report(Y_test, model[i].predict(X_test)))
print(accuracy_score(Y_test, model[i].predict(X_test)))
pred = model[2].predict(X_test)
print(pred)
print()
print(Y_test)
```

CHAPTER 6

RESULTS AND DISCUSSION

6.1 Performance Analysis using various metrics:

As we mentioned, we used 3 models for this project and by using the confusion matrix we came to a conclusion that “Random Forest Classifier” has performed the best among the three models with an accuracy of 96.5 percentage.

```
Model  0
              precision    recall  f1-score   support

           0       0.97      0.96      0.96        90
           1       0.93      0.94      0.93        53

    accuracy                           0.95       143
   macro avg       0.95      0.95      0.95       143
weighted avg       0.95      0.95      0.95       143

0.951048951048951
Model  1
              precision    recall  f1-score   support

           0       0.98      0.92      0.95        90
           1       0.88      0.96      0.92        53

    accuracy                           0.94       143
   macro avg       0.93      0.94      0.93       143
weighted avg       0.94      0.94      0.94       143

0.9370629370629371
Model  2
              precision    recall  f1-score   support

           0       0.98      0.97      0.97        90
           1       0.94      0.96      0.95        53

    accuracy                           0.97       143
   macro avg       0.96      0.96      0.96       143
weighted avg       0.97      0.97      0.97       143
```

The above picture represents the precision, recall,F1-score,support of respective models.

Precision, recall, F1-score, and support are essential metrics used to evaluate the performance of classification models, especially in scenarios where imbalanced datasets or differing class distributions are present. These metrics offer insights into the model's ability to correctly identify and classify instances of each class. Precision is a metric that measures the accuracy of positive predictions made by the model. It is calculated as the number of true positives (correctly predicted positive instances) divided by the sum of true positives and false positives (incorrectly predicted positive instances). Precision assesses the model's ability to avoid false alarms and ensure that when it predicts a positive result, it is highly likely to be correct.

Recall, also known as sensitivity or true positive rate, quantifies the model's ability to capture all actual positive instances. It is computed as the number of true positives divided by the sum of true positives and false negatives (instances that were actually positive but incorrectly predicted as negative). Recall is crucial in applications where missing positive cases can have severe consequences, such as medical diagnoses, ensuring that the model doesn't overlook relevant instances. F1-score is the harmonic mean of precision and recall. It combines both metrics into a single value that balances the trade-off between precision and recall. F1-score is particularly useful when precision and recall are in conflict. A higher F1-score indicates a good balance between precision and recall, meaning the model is making accurate positive predictions while minimizing false negatives.

Support represents the number of instances in each class in the dataset. It provides context for the other metrics, helping you understand the distribution of actual instances in each class. Support can be used to identify class imbalances and is particularly valuable when working with multi-class classification, where different classes may have varying amounts of data.

6.2 Comparison between Existing Models:

There isn't any comparison between the models because one model can attain high accuracy based upon its requirement for the project. As we developed three models we tested their accuracy and found Random Forest Classifier as the best one.

```
#test model accuracy on test data on confusion matrix
from sklearn.metrics import confusion_matrix
for i in range(len(model)):
  print('Model ',i)
  cm = confusion_matrix(Y_test, model[i].predict(X_test))

  TP = cm[0][0]
  TN = cm[1][1]
  FN = cm[1][0]
  FP = cm[0][1]

  print(cm)
  print('Testing Accuracy = ', (TP + TN)/(TP + TN + FN + FP))
  print()
```

```
Model  0
[[86  4]
 [ 3 50]]
Testing Accuracy =  0.951048951048951

Model  1
[[83  7]
 [ 2 51]]
Testing Accuracy =  0.9370629370629371

Model  2
[[87  3]
 [ 2 51]]
Testing Accuracy =  0.965034965034965
```

Fig 6.2 Predicting Accuracy using Confusion Matrix

This figure shows the code which tests model accuracy on test data using confusion matrix.

CHAPTER 7
CONCLUSION AND FUTURE SCOPE

7.1 Conclusion:

In conclusion, breast cancer prediction using machine learning techniques represents a significant step forward in the early detection and risk assessment of this prevalent disease. These machine learning techniques have demonstrated their efficacy in leveraging various data sources, such as clinical features, genetic markers, and mammographic images, to provide valuable insights and predictions.

Logistic regression, as a straightforward and interpretable model, is well-suited for assessing the likelihood of breast cancer based on individual risk factors. It offers a transparent view of how each feature contributes to the prediction, making it valuable for both healthcare professionals and patients seeking personalized risk assessments. Decision tree models, on the other hand, capture complex relationships in the data by recursively splitting features to create a hierarchical structure of decisions. They are particularly useful for interpreting feature importance and understanding how different variables influence breast cancer risk. Their visualization capabilities aid in communicating findings to non-technical stakeholders. Random forest classifiers take the concept of decision trees a step further by building an ensemble of trees. They provide higher accuracy and robustness by aggregating the predictions of multiple decision trees. Random forests excel in handling both structured and unstructured data, making them powerful tools for breast cancer prediction.

7.2 Future Scope:

In terms of future enhancements, ongoing research in the field of breast cancer prediction can focus on several areas. Firstly, the integration of more diverse and extensive datasets, including genetic information, environmental factors, and patient history, can lead to more comprehensive risk assessments. Secondly, the

development of hybrid models that combine the strengths of logistic regression, decision trees, and random forests can potentially improve the accuracy of predictions. Such models can leverage the transparency of logistic regression and the predictive power of decision trees and random forests. Moreover, the incorporation of deep learning techniques, especially for image analysis in mammograms, presents an exciting avenue for future research. Convolutional neural networks (CNNs) can automatically extract intricate features from images, potentially improving the detection of breast abnormalities. Ensuring the ethical use of patient data, maintaining data privacy, and addressing issues related to model interpretability will also be essential in future developments.

In conclusion, the combination of logistic regression, decision tree, and random forest classifier models offers a multifaceted approach to breast cancer prediction. Future enhancements in data integration, model hybridization, and the application of deep learning can further advance the accuracy and reliability of breast cancer risk assessment, ultimately contributing to early detection and improved patient outcomes.

REFERENCES

[1]Afshar, Lotfnezhad, et al. “Prediction of Breast Cancer Survival by Machine Learning Methods: An Application of Multiple Imputation”. Iran J Public Health (2021).

[2]Bardou, Dalal, Kun Zhang, and Sayed Mohammad Ahmad. “Classification of Breast Cancer Based on Histology Images Using Convolutional Neural Networks”. IEEE Access 6 (2018): 24680–24693.

[3]Derangula, Anusha, et al. “Feature Selection of Breast Cancer Data Using GradientBoosting”.(2020).

[4]Octaviani, T L and Z Rustam. “Random forest for breast cancer prediction”. Proceedings of the 4th International symposium on current progress in Mathematics and Sciences (2019)

[5]Few-Shot Breast Cancer Metastases Classification via unsupervised Cell Ranking – Jiaojiao Chen – IEEE EXPLORE 2021

[6]Remote Computer-Aided Breast Cancer prediction and diagnosis system based on cytological images – Yasmeen Mourice George-IEEE Explore 2014

Breast Cancer prediction using random tree classifier–JaySingh IEEEExplore2019.Benediktsson Published in: IEEE Geoscience and Remote Sensing Magazine, 2018

[7]C. Shorten and T. M. Khoshgoftaar, “A survey on image data augmentation for deep learning,” Journal of Big Data, vol. 6, no. 1, p. 60, 2019.

[8]M. Desai and M. Shah, “An anatomization on breast cancer detection and diagnosis employing multi-layer perceptron neural network (MLP) and Convolutional neural network (CNN),” Clinical eHealth, vol. 4, pp. 1–11, 2021.

[9]M. S. Hossain, G. Muhammad, and N. Guizani, “Explainable AI and mass surveillance systembased healthcare framework to combat COVID-I9 like pandemics,” IEEE Network, vol. 34, no. 4, pp. 126–132, 2020.

APPENDIX

CODE SNIPPETS

Breast Cancer prediction using machine learning

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import scipy.stats as s
from sklearn.metrics import classification_report
```

```
from google.colab import files


uploaded = files.upload()
df = pd.read_csv('data (1).csv')
df.head(20)
```

Choose Files data (1).csv

- **data (1).csv**(text/csv) - 124635 bytes, last modified: 8/10/2023 - 100% done

Saving data (1).csv to data (1) (1).csv

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	...	texture_worst	perimeter_worst
0	842302	M	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.30010	0.14710	...	17.33	184.60
1	842517	M	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.08690	0.07017	...	23.41	158.80
2	84300903	M	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.19740	0.12790	...	25.53	152.50
3	84348301	M	11.42	20.38	77.58	386.1	0.14250	0.28390	0.24140	0.10520	...	26.50	98.87
4	84358402	M	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.19800	0.10430	...	16.67	152.20
5	843786	M	12.45	15.70	82.57	477.1	0.12780	0.17000	0.15780	0.08089	...	23.75	103.40
6	844359	M	18.25	19.98	119.60	1040.0	0.09463	0.10900	0.11270	0.07400	...	27.66	153.20
7	84458202	M	13.71	20.83	90.20	577.9	0.11890	0.16450	0.09366	0.05985	...	28.14	110.60

✓ 55s completed at 3:46 PM

```
[5] df.shape #shows rows and columns
```

```
(569, 33)
```

```
df.isna().sum() #shows empty values for each column
```

```
id                         0
diagnosis                  0
radius_mean                0
texture_mean               0
perimeter_mean             0
area_mean                  0
smoothness_mean            0
compactness_mean           0
concavity_mean             0
concave points_mean        0
symmetry_mean              0
fractal_dimension_mean     0
radius_se                  0
texture_se                 0
perimeter_se               0
area_se                    0
smoothness_se              0
compactness_se             0
concavity_se               0
concave points_se          0
symmetry_se                0
fractal_dimension_se       0
radius_worst               0
texture_worst              0
perimeter_worst            0
area_worst                 0
smoothness_worst           0
compactness_worst          0
concavity_worst            0
```

```
[7] df = df.dropna(axis=1) #deletes all the unnessecary colums
```

```
df.shape
```

```
[9] df['diagnosis'].value_counts() #counts who are M and B
```

```
[10] df.dtypes
```

```
[11] from sklearn.preprocessing import LabelEncoder
     labelencoder_Y = LabelEncoder()
     df.iloc[:,1] = labelencoder_Y.fit_transform(df.iloc[:,1].values)

     df.iloc[:,1]
```

```
#Coding splitting into x(indep) and y(dep)
X = df.iloc[:,2:31].values
Y = df.iloc[:,1].values
```

```
[ ] #split the dataset into 75% training and 25% testing
    from sklearn.model_selection import train_test_split
    X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.25 , random_state = 0)
```

```
#feature scaling
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.fit_transform(X_test)

X_train
```

```
array([[-0.65079907, -0.43057322, -0.68024847, ..., -0.69592933,
        -0.36433881,  0.32349851],
       [-0.82835341,  0.15226547, -0.82773762, ..., -1.29277423,
        -1.45036679,  0.62563098],
       [ 1.68277234,  2.18977235,  1.60009756, ...,  0.26255563,
         0.72504581, -0.51329768],
       ...,
       [-1.33114223, -0.22172269, -1.3242844 , ..., -0.78274313,
        -0.98806491, -0.69995543],
       [-1.25110186, -0.24600763, -1.28700242, ..., -1.36015587,
        -1.75887319, -1.56206114],
       [-0.74662205,  1.14066273, -0.72203706, ...,  0.47201917,
        -0.2860679 , -1.24094654]])
```

```
#Create a function for the model
def models(X_train, Y_train):
  #Logistic Regression
  from sklearn.linear_model import LogisticRegression
  log = LogisticRegression(random_state=0)
  log.fit(X_train,Y_train)

  #Decision Tree
  from sport RandomForestClassifier
  forest=RandomForestClassifier(n_estimators = 10, criterion = 'entropy',random_state=0)
  forest.fit(X_train, Y_train)klearn.tree import DecisionTreeClassifier
  tree=DecisionTreeClassifier(criterion = 'entropy',random_state=0)
  tree.fit(X_train,Y_train)

  #Forest Classifier
  from sklearn.ensemble im

  #Printing the models accuracy for training the data
  print('[0]Logistic Regression Training Accuracy:', log.score(X_train, Y_train))
  print('[1]Decision Tree Classifier Training Accuracy:', tree.score(X_train, Y_train))
  print('[2]Random Forest Classifier Training Accuracy:', forest.score(X_train, Y_train))

  return log, tree, forest
```

```
[ ]  #Getting all of the models
     model = models(X_train, Y_train)
```

```
#test model accuracy on test data on confusion matrix
from sklearn.metrics import confusion_matrix
for i in range(len(model)):
  print('Model ',i)
  cm = confusion_matrix(Y_test, model[i].predict(X_test))

  TP = cm[0][0]
  TN = cm[1][1]
  FN = cm[1][0]
  FP = cm[0][1]

  print(cm)
  print('Testing Accuracy = ', (TP + TN)/(TP + TN + FN + FP))
  print()
```

```
Model  0
[[86  4]
 [ 3 50]]
Testing Accuracy =  0.951048951048951

Model  1
[[83  7]
 [ 2 51]]
Testing Accuracy =  0.9370629370629371

Model  2
[[87  3]
 [ 2 51]]
Testing Accuracy =  0.965034965034965
```

```
[ ]  #Another way to get matrix of the models
     from sklearn.metrics import classification_report
     from sklearn.metrics import accuracy_score
     for i in range(len(model)):
       print('Model ',i)
       print( classification_report(Y_test, model[i].predict(X_test)))
       print(accuracy_score(Y_test, model[i].predict(X_test)))
```

```
Model  0
              precision    recall  f1-score   support

           0       0.97      0.96      0.96        90
           1       0.93      0.94      0.93        53

    accuracy                           0.95       143
   macro avg       0.95      0.95      0.95       143
weighted avg       0.95      0.95      0.95       143

0.951048951048951
Model  1
              precision    recall  f1-score   support

           0       0.98      0.92      0.95        90
           1       0.88      0.96      0.92        53

    accuracy                           0.94       143
   macro avg       0.93      0.94      0.93       143
weighted avg       0.94      0.94      0.94       143

0.9370629370629371
Model  2
              precision    recall  f1-score   support

           0       0.98      0.97      0.97        90
           1       0.94      0.96      0.95        53

    accuracy                           0.97       143
   macro avg       0.96      0.96      0.96       143
weighted avg       0.97      0.97      0.97       143

0.965034965034965
```

```
#Print the prediction of random forest classifier model
pred = model[2].predict(X_test)
print(pred)
print()
print(Y_test)
```

```
[1 0 0 0 0 0 0 0 0 0 1 0 0 1 1 1 0 1 1 1 1 1 0 0 1 0 0 1 0 1 0 1 0 1 0 1 0
 1 0 1 0 0 1 0 0 1 0 0 0 1 1 1 1 0 0 0 0 0 0 1 1 1 0 0 1 0 1 1 1 1 0 0 1 0 0
 1 0 0 0 0 0 1 1 1 0 1 0 0 0 1 1 0 1 0 1 0 0 1 0 0 0 0 0 0 0 1 0 1 0 1 1 0
 1 1 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 1 0 0 0 0 0 0 1 1 0 0 0 1]

[1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1 1 1 1 0 0 1 0 0 1 0 1 0 1 0 1 0 1 0
 1 0 1 1 0 1 0 0 1 0 0 0 1 1 1 1 0 0 0 0 0 0 1 1 1 0 0 1 0 1 1 1 1 0 0 1 0 1
 1 0 0 0 0 0 1 1 1 0 1 0 0 0 1 1 0 1 0 1 0 0 1 0 0 0 0 0 0 0 1 0 1 0 1 1 0
 1 1 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 1 0 0 0 0 0 0 1 1 0 0 0 1]
```

Format - I

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Deemed to be University u/s 3 of UGC Act, 1956)

Office of Controller of Examinations

REPORT FOR PLAGIARISM CHECK ON THE DISSERTATION/PROJECT REPORTS FOR UG/PG PROGRAMMES

(To be attached in the dissertation/ project report)

1	Name of the Candidate **(IN BLOCK LETTERS)**	ANGADA CHANDRA MOULI GADHAMSETTY NAVANEETH
2	Address of the Candidate	Narayanapuram, Rajahmundry Rajagopalapuram, Naidupeta
3	Registration Number	RA2011003010664 RA2011003011335
4	Date of Birth	22 September 2003 02 September 2003
5	Department	Computer Science and Engineering
6	Faculty	Engineering and Technology, School of Computing
7	Title of the Dissertation/Project	Breast Cancer prediction using machine learning techniques
8	Whether the above project /dissertation is done by	~~Individual~~ or group : (Strike whichever is not applicable) **a)** If the project/ dissertation is done in group, then how many students together completed the project : 2 b) Mention the Name & Register number of other candidates : Angada Chandra Mouli, RA2011003010664 Gadhamsetty Navaneeth, RA2011003011335
9	Name and address of the Supervisor / Guide	M.Ranjani, Assistant Professor, Dept. of Computer Science, SRM Institute of science and Technology, Kattankulathur ,Tamil Nadu- 603203 **Mail ID: ranjanim1@srmist.edu.in** **Mobile Number: 9976808790**
10	Name and address of Co-Supervisor / Co- Guide (if any)	**NIL**

11	Software Used	Turnitin		
12	Date of Verification	14 November 2023		
13	**Plagiarism Details: (to attach the final report from the software)**			
Chapter	**Title of the Chapter**	**Percentage of similarity index (including self citation)**	**Percentage of similarity index (Excluding self-citation)**	**% of plagiarism after excluding Quotes, Bibliography, etc.,**
1	INTRODUCTION	0	0	0
2	LITERATURE SURVEY	2	2	2
3	ARCHITECTURE DIAGRAM AND FOUNDATION FOR CANCER PREDICTION	3	3	3
4	DESIGN AND IMPLEMENTATION OF DETECTION MODEL	2	2	2
5	CODING AND TESTING	2	2	2
6	RESULTS AND DISCUSSION	1	1	1
7	CONCLUSION AND FUTURE SCOPE	1	1	1
Appendices		1	1	1

I / We declare that the above information have been verified and found true to the best of my / our knowledge.

Signature of the Candidate	**Name & Signature of the Staff (Who uses the plagiarism check software)**
Name & Signature of the Supervisor/ Guide	**Name & Signature of the Co-Supervisor/Co-Guide**
Name & Signature of the HOD	

report-2

ORIGINALITY REPORT

11%	9%	8%	9%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to King's College Student Paper	2%
2	Submitted to SRM University Student Paper	1%
3	Submitted to Queen Mary and Westfield College Student Paper	1%
4	assets.researchsquare.com Internet Source	1%
5	ajomc.asianpubs.org Internet Source	<1%
6	ijisae.org Internet Source	<1%
7	plosjournal.deepdyve.com Internet Source	<1%
8	Ali Fallahi Rahmatabadi, Azam Bastanfard, Amineh Amini, Hadi Saboohi. "Building Movie Recommender Systems Utilizing Poster's Visual Features: A Survey Study", 2022 10th	<1%

RSI International Conference on Robotics and Mechatronics (ICRoM), 2022
Publication

9	Submitted to University of Wales Swansea Student Paper	<1 %
10	Bogdan Walek, Petr Fajmon. "A hybrid recommender system for an online store using a fuzzy expert system", Expert Systems with Applications, 2022 Publication	<1 %
11	mafiadoc.com Internet Source	<1 %
12	Priyash Verma, Shilpi Sharma. "Artificial Intelligence based Recommendation System", 2020 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN), 2020 Publication	<1 %
13	journal.uokufa.edu.iq Internet Source	<1 %
14	S. Aramuthakannan, M. Ramya Devi, S. Lokesh, R. Kumar. "Movie recommendation system using taymon optimized deep learning network", Journal of Intelligent & Fuzzy Systems, 2023 Publication	<1 %

dokumen.pub

15	Internet Source	<1%
16	Submitted to University of Glamorgan Student Paper	<1%
17	Submitted to Victorian Institute of Technology Student Paper	<1%
18	ijstr.org Internet Source	<1%
19	www.e3s-conferences.org Internet Source	<1%
20	ijrpr.com Internet Source	<1%
21	Submitted to Toronto Business College Student Paper	<1%
22	www.ijsr.net Internet Source	<1%
23	docu.tips Internet Source	<1%
24	core.ac.uk Internet Source	<1%
25	link.springer.com Internet Source	<1%
26	www.slideshare.net Internet Source	<1%

27	"Data Management, Analytics and Innovation", Springer Science and Business Media LLC, 2023 Publication	<1%
28	Ton Duc Thang University Publication	<1%
29	www.researchandmarkets.com Internet Source	<1%

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	Off		

ICCCI 2024 - Paper Submitted Successfully - Paper ID: 494

External Inbox ×

ICCCI <info@iccci.in> Sat, Nov 11, 10:50 AM (3 days ago)
to me, info

Dear Angada Chandra Mouli

Paper ID	494
Default Password	am2898@srmist.edu.in
Name	Angada Chandra Mouli
Paper Title	Breast Cancer prediction using machine learning Techniques
Author Mobile	9705136067
Paper Category	Others
Location:	Rajahmundry,Andhra Pradesh,India

-
Thank you
ICCCI Team.