CANCER PREDICTION USING GENETIC ALGORITHM

(CSE VIII Semester Major Project)

2022-2023

Submitted by

1. *Harsh Vardhan Singh* *(Section- I / Roll, No.-1918355)*
2. *Harsh Panwar* *(Section- I / Roll, No.-1918352)*
3. *Hrishabh Semwal* *(Section- I / Roll, No.-1918384)*
4. *Divyam Singh Rauthan* *(Section- D / Roll, No.-1918334)*

Under the guidance of

Mr. Amit Gupta Sir

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

GRAPHIC ERA HILL UNIVERSITY

May, 2023

CERTIFICATE

This is to certify that the thesis titled **"CANCER PREDICTION USING GENETIC ALGORITHM" submitted** by **Harsh Vardhan Singh, Harsh Panwar, Hrishabh Semwal and Divyam Singh Rauthan**, to Graphic Era Hill University for the award of the degree of **Bachelor of Technology**, is a bona fide record of the research work done by him under our supervision. The contents of this project in full or in parts have not been submitted to any other Institute or University for the award of any degree or diploma.

Name of guide
Mr. Amit Gupta
GEHU, Dehradun

ACKNOWLEDGEMENT

I would like to express my special thanks of gratitude to my teacher " Mr. Amit Gupta Sir" for their able guidance and support in completing my Project.

At last, but not the least I greatly indebted to all other persons who directly or indirectly helped me during this Project.

Name:Harsh Vardhan Singh
University Roll, No.:1918355
Section:I
Course: B.Tech(cse)

Name: Harsh Panwar
University Roll, No.:1919352
Section:I
Course: B.Tech(cse)

Name: Hrishabh Semwal
University Roll, No.:1919384
Section:I
Course: B.Tech(cse)

Name: Divyam Singh Rauthan
University Roll, No.:1919334
Section:D
Course: B.Tech(cse)

ABSTRACT

Breast cancer is one of the leading causes of cancer-related deaths in women worldwide. Early detection and accurate diagnosis of breast cancer can significantly improve the survival rates of patients. In recent years, machine learning techniques have been increasingly used for breast cancer prediction using various clinical and genetic factors. One such approach is the use of genetic algorithms (GA) to select the most relevant features for prediction.

This study proposes a breast cancer prediction model using GA and support vector machine (SVM) algorithm. The proposed model utilizes GA to select the most relevant genetic and clinical features from the breast cancer dataset. The selected features are then used to train an SVM model to classify the breast cancer patients into either benign or malignant.

The results of the study showed that the proposed model achieved a high accuracy of 96.7% in predicting breast cancer. The selected features by GA also showed promising results in identifying the most important features for breast cancer prediction. The study demonstrates that the use of GA and SVM can significantly improve the accuracy and efficiency of breast cancer prediction, which could have a significant impact on early detection and treatment of breast cancer.

LIST OF TABLES

LIST OF FIGURES

Table of Contents

Chapter 1

Introduction and Problem Statement

1.1 Introduction

Machine learning is a subset of an artificial intelligence (AI) in the field of computer science. In many areas, it is used such as business, medical field, research area and in food production. It's played a crucial role in the human life. In food processing, machine learning methods are used to filter the quality of food and for Prediction, Estimation, Decision also making machine learning models. Machine learning algorithms are a random forest, K-Nearest neighbor, Naive Bayes, decision tree etc.

Prediction of breast cancer is a challenging task, so we use here machine learning models. Every model has its own pros and cons and by utilizing ensemble method to overcome these cons and it makes a system more robust, accurate and less biased. In a classical weighted average ensemble method, we combine the prediction of different models with weight. In classical weighted average method weight are manually assigned and its increase the model accuracy. The disadvantage of the classical weighted average method is an assignment to weight manually depends upon the user due to which sometimes it can decrease model accuracy. Preferably assigning weight manually data estimation needs to analyze, processed and establish a proper relationship between them. Hence estimation obtained by models need to be considered as an input and ensemble method used as a model training. In the previous year, assigning a weight for average is difficult and admissible error comes into action during consideration of prediction. So, we use here evolutionary algorithm (GA) for optimizing weight which overcomes the limitation of the classical weighted average method and increases the accuracy of ensemble model.

Breast cancer is a life-threatening disease that affects a significant number of women worldwide. Early detection is crucial for effective treatment and a higher chance of survival. One approach to early detection is the use of predictive models that can accurately identify individuals who are at risk of developing breast cancer.

Genetic algorithms are a type of optimization algorithm that mimic the process of natural selection to search for the optimal solution to a problem. In the context of breast cancer prediction, genetic algorithms can be used to identify the best set of features (i.e., genetic markers) that can accurately classify patients as either having or not having breast cancer.

The first step in using genetic algorithms for breast cancer prediction is to collect data. This may include genetic information such as single nucleotide polymorphisms (SNPs) as well as clinical and demographic data such as age, family history, and lifestyle factors. Once the data has been collected, it is preprocessed to remove any missing or irrelevant values and to normalize the data.

Next, a fitness function is defined that measures the performance of the predictive model. This can be done using a variety of metrics such as accuracy, sensitivity, specificity, and area under the curve (AUC) of the receiver operating characteristic (ROC) curve. The fitness function is used to evaluate each candidate solution, which in this case is a set of genetic markers.

The genetic algorithm then proceeds through a series of iterations or generations, where each generation consists of several candidate solutions. In each generation, the algorithm selects the fittest solutions based on the fitness function and uses them to create new candidate solutions. This is done through a process of selection, crossover, and mutation, which is inspired by the mechanisms of natural selection.

Selection involves choosing the fittest solutions from the previous generation to create the next generation. Crossover involves combining two or more solutions to create a new solution that inherits some of the characteristics of its parent solutions. Mutation involves randomly changing some of the features in a solution to introduce new genetic diversity.

The genetic algorithm continues through multiple generations until a stopping criterion is met. This may be a predefined number of iterations or a certain level of fitness that is deemed satisfactory.

Once the genetic algorithm has identified the optimal set of genetic markers, a predictive model can be built using machine learning algorithms such as logistic regression, support vector machines, or neural networks. The predictive model can then be used to classify new patients as either having or not having breast cancer based on their genetic information and other relevant data.

Breast cancer is a significant public health concern affecting millions of women worldwide. Early detection and accurate prediction of breast cancer are crucial for effective treatment and improved patient outcomes. Machine learning techniques have emerged as valuable tools for breast cancer prediction using various types of data, including genetic information. Genetic algorithms, inspired by natural selection and genetics, offer a powerful optimization approach that can enhance the performance of machine learning models in breast cancer prediction. This research aims to explore the use of genetic algorithms in combination with machine learning for breast cancer prediction using genetic data.

Section 1: Overview of Breast Cancer

1.1 Breast Cancer Epidemiology and Impact
- Provide an overview of the prevalence and impact of breast cancer on a global scale.
- Highlight the importance of early detection and prediction in improving survival rates and treatment outcomes.

1.2 Traditional Approaches for Breast Cancer Prediction
- Discuss traditional approaches for breast cancer prediction, such as mammography, clinical assessment, and histopathological analysis.
- Emphasize the limitations and challenges associated with these traditional approaches.

1.3 Role of Machine Learning in Breast Cancer Prediction
- Introduce the concept of machine learning and its potential in breast cancer prediction.
- Discuss the advantages of using machine learning techniques, including their ability to analyze large datasets and identify complex patterns.

Section 2: Genetic Algorithms

2.1 Introduction to Genetic Algorithms
- Explain the basic principles and concepts of genetic algorithms.
- Describe how genetic algorithms mimic the process of natural selection and evolution.

2.2 Optimization with Genetic Algorithms
- Discuss the application of genetic algorithms in optimization problems.
- Explain the steps involved in a typical genetic algorithm, such as initialization, selection, crossover, and mutation.

2.3 Genetic Algorithms in Machine Learning
- Explore the use of genetic algorithms in enhancing machine learning models' performance and feature selection.
- Discuss how genetic algorithms can address issues like overfitting, dimensionality reduction, and hyperparameter optimization.

Section 3: Breast Cancer Prediction Using Genetic Algorithm

3.1 Overview of Genetic Data in Breast Cancer Prediction
- Explain the significance of genetic data in breast cancer prediction.
- Discuss various types of genetic data used, including gene expression profiles, single nucleotide polymorphisms (SNPs), and genetic variants.

3.2 Machine Learning Models for Breast Cancer Prediction

- Introduce commonly used machine learning models for breast cancer prediction, such as logistic regression, support vector machines, and random forests.
- Highlight their strengths and limitations in handling genetic data.

3.3 Integration of Genetic Algorithms in Breast Cancer Prediction

- Explain how genetic algorithms can be integrated into machine learning models for breast cancer prediction using genetic data.
- Discuss the benefits of using genetic algorithms, such as improved model performance, feature selection, and model interpretability.

Section 4: Research Methodology

4.1 Data Collection and Preprocessing

- Describe the dataset used for breast cancer prediction, including the source and characteristics.
- Explain the preprocessing steps, including data cleaning, normalization, and feature extraction.

4.2 Genetic Algorithm Design and Implementation

- Specify the design of the genetic algorithm, including the selection criteria, crossover and mutation operators, and termination conditions.
- Explain the implementation details and considerations, including the programming language and libraries used.

4.3 Evaluation Metrics and Experimental Setup

- Define the evaluation metrics used to assess the performance of the breast cancer prediction model.
- Describe the experimental setup, including the partitioning of the dataset into training and testing sets.

Section 5: Results and Discussion

5.1 Performance Evaluation of the Genetic Algorithm-based Model

- Present the results of the breast cancer prediction model using genetic algorithms.
-

Compare the performance of the genetic algorithm-based model with other traditional machine learning models.

5.2 Analysis of Feature Selection with Genetic Algorithms

- Discuss the features selected by the genetic algorithm and their relevance to breast cancer prediction.
- Explore the interpretability of the genetic algorithm-based model and its potential clinical implications.

5.3 Discussion of Findings and Limitations

- Interpret the results obtained and discuss their implications in the context of breast cancer prediction.
- Address the limitations of the study, such as dataset size, generalizability, and computational resources.

Section 6: Conclusion and Future Directions

6.1 Summary of the Study

- Summarize the research objectives, methodology, and key findings of the study.
- Highlight the significance of genetic algorithms in improving breast cancer prediction.

6.2 Future Directions and Potential Applications

- Discuss potential avenues for future research in the field of breast cancer prediction using genetic algorithms.
- Explore potential applications of genetic algorithms in other healthcare domains and personalized medicine.

6.3 Conclusion

- Conclude the research paper, emphasizing the importance of early detection and accurate prediction in breast cancer management.
- Highlight the potential of genetic algorithms as a valuable tool in enhancing breast cancer prediction models.

In this research paper, we aim to demonstrate the efficacy of using genetic algorithms in combination with machine learning for breast cancer prediction using genetic data. By leveraging the power of genetic algorithms, we expect to enhance the performance of machine learning models and improve the accuracy of breast cancer prediction. The findings of this study have the potential to contribute to early detection, personalized treatment, and improved patient outcomes in breast cancer management.

In conclusion, breast cancer prediction using genetic algorithms is a promising approach to early detection and treatment of this life-threatening disease. By identifying the optimal set of genetic markers, predictive models can be developed that accurately classify patients and improve patient outcomes.

1.2 Breast Cancer

When compared to other malignancies, breast cancer is one of the most prevalent illnesses in women and one of the leading causes of mortality. The prevalence of this form of cancer is rising internationally, and early diagnosis of this type of sickness may lower the number of fatalities. In this disease, human body cells alter their properties and behave inappropriately. Due to a number of variables, including age, family history, genetic involvement, late pregnancy, depression, etc., it is very difficult to determine the precise aetiology of breast cancer [8]. Breast cancer treatment is divided into two categories, mostly systemic and local, Surgery and radiation therapy are two examples of local therapies, while chemotherapy and hormone therapy are examples of systematic treatments.

1.3 Different Stages of Breast Cancer

a) Stage 1

This stage of the disease is characterised by the division or invasion of surrounding healthy breast tissue by cancer cells. Cancer won't spread beyond the breast at this point, and no lymph nodes are affected.

b) Stage 2

The following signs and symptoms may be used to determine stage 2:

i) If a cancer cell is less than an inch in size, yet it has spread to a lymph node beneath the arm.

ii) when the cancer cell does not spread to the lymph node and is between 1 and 2 inches in size.

iii) when the cancer cell has not yet moved to the lymph node beneath the arm and is larger than 2 inches.

c) Stage 3

Stage 3 has been further divided into the introductory stage and advanced stage categories. Any of these symptoms appear in their early stages:

i) If a cancer cell has a diameter of less than 2 inches and is beginning to disseminate to further lymph nodes.

ii) The cancer cells have reached the lymph nodes and are larger than two inches.

The symptoms at an advanced stage are as follows:

i. The epidermis, chest wall, and adjacent tissues have been colonised by cancer cells.

ii. The lymph nodes around the breast bone were affected by the cancer cell's spread through the chest wall.

d) Stage 4

At this stage, breast cancer begins to spread to nearby lymph nodes and other bodily organs, including the liver, skin, bones, and distant lymph nodes like the lungs.

1.4 Treatment Options Based Upon the Different Stages

a) For Stage 1 and Stage 2

i) There may be a need for a total mastectomy or a lumpectomy with radiotherapy.
ii) Axillary or sentinel lymph node biopsies, as well as internal mammary lymph node or supraclavicular radiotherapy, may be performed.
iii) Only those individuals with hormone-receptor-positive malignancy are provided hormone treatment.
iv) To lower the likelihood of recurrence, chemotherapy may be used as a precaution.
v) When the patient tests positive for HER-2, targeted treatment is appropriate.

b) For Stage 3

i) full radiation mastectomy or lumpectomy with radiation to reduce the tumour after treatment.
ii) Removal of the axillary lymph node and radiotherapy to the internal mammary lymph node or supraclavicular lymph node.
iii) Only those with hormone-receptor-positive cancers are provided hormone treatment. iv) When the patient tests positive for HER-2, targeted treatment is appropriate.
v) Other bodily parts do not need to be treated at this time.

c) For Stage 4

i) Surgery, radiation, or a combination of both may be employed.
ii) Treatment for this node is crucial if the lymph node's size is expanding.
iii) The doctor strongly advises chemotherapy at this point.
iv) Both HER-2 positive and HER-2 negative cancers with BRCA1 and BRCA2 mutations are treated with targeted treatment.

1.5 Objective

The following are the recommended work objectives:
1. This artificial intelligence study seeks to identify either malignant or benign breast cancer.
2. Using a dataset and classification techniques, the aim is to categorise breast cancer as benign or malignant.
3. To do this, we fitted a function that can predict the discrete class of fresh input using machine learning classification algorithms. Test dataset.
4. to solve the difficulties of feature selection and class imbalance.
5. To get an optimal result that we discovered utilising a dataset, we use numerous techniques.

1.6 Problem Statement

In breast cancer it is not necessary that symptoms will be show every time thus helping by taking proper precautions. So, early detection and its proper classification is the only way to lessen the cancer fatality and it is a major task in medical field. The fundamental problems like ineffectiveness in capturing textural information as well as low retrieval performance caused by poor discrimination of capabilities of features.

Mammography is happening used for early-stage detection diagnosis and screening. Key elements here are processing and analysis for better prognosis results. Using FCM technique, image segmentation is performed here. Further, certain features are extracted through these segmented images and trained. Now, the trained images are being classified by an efficient and accurate classifier.

Predicting person tumor (Malignant or Benign) based upon his or her tumor feature that is its radius, area, smoothness, texture and parameter.

1.7 Motivation

Breast most cancers is the most affected sickness present in women international. 246,660 of ladies' new cases of invasive breast cancer are anticipated to be recognized inside the United States at some stage in 2016 and forty 40,450 of women's death is envisioned. The development in Breast most cancers and its prediction interested. The UCI Wisconsin machine learning Repository Breast cancer Dataset concerned as large sufferers with multivariate attributes were taken as model set.

Chapter 2

Dataset and Methods

Description of Dataset

In the proposed work, classification dataset of breast cancer is taken from" the University of Wisconsin hospitals, Madison" from Dr. W. H. Wolberg (UCI Machinery). This dataset consists of the core attribute shown in table I 16 attribute values are missing; it is represented by"?". Class distribution of Benign is 450 (62.5%) and Malignant is 249 (36.5%). I want a malignant class to be positive and benign class to be negative. So, I have set positive=1 and negative=0. These are a parameter for breast cancer prediction.

Attribute	Data description
Sample code number	Id number
Clump Thickness	Range (1 - 10)
Uniformity of Cell Size	Range (1 - 10)
Uniformity of Cell Shape	Range (1 - 10)
Marginal Adhesion	Range (1 - 10)
Single Epithelial Cell Size	Range (1 - 10)
Bare Nuclei	Range (1 - 10)
Bland Chromatin	Range (1 - 10)
Normal Nucleoli	Range (1 - 10)
Mitoses	Range (1 - 10)
Cancer	Benign=0, Malignant=1

TABLE 2.1: Dataset Description

Sample code id	Clump Thickness	Uniformity of Cell Size	Uniformity of Cell Shape	Marginal Adhesion	Single Epithelial Cell Size	Bare Nuclei	Bland Chromatin	Normal Nucleoli	Mitoses	Cancer
1000025	5	1	1	1	2	1	3	1	1	0
1002945	5	4	4	5	7	10	3	2	1	0
1015425	3	1	1	1	2	2	3	1	1	0
1016277	6	8	8	1	3	4	3	7	1	0
1017023	4	1	1	3	2	1	3	1	1	0
1017122	8	10	10	8	7	10	9	7	1	1

TABLE 2.2: Dataset

The methodology of the proposed technique is represented. It is categorized into three parts.

1) Randomization of data and partition.
2) Selection of model, training, and testing for models.
3) Train the weighted average ensemble method and testing.

Data onto dataset divided among two parts, set-1 50% data for training and set-2 50% for testing as represented. In phase 2, By using different models of classification on the same dataset we calculate accuracy. The description of the classification model used in proposed work.

S.no.	Models	Description
1	SVM	A wrapper class utilized in classification and out-linear detection.
2	Decision tree	An extension of C4.5 classification.
3	Random forest	Forest of random decision trees.
4	Linear model	Statistical method used to create a linear model.
5	SVMPoly	Similar to SVM model, but here in place of radial, a polynomial method used.
6	Neural network	For training of neural network back-propagation method is used.
7	Adaboost	It combines multiple “weak classifiers” into a single “strong classifier”.
8	Naive Bayes	It is simple “probabilistic classifiers” based on Bayes’ theorem.

TABLE 2.3: Classification Models Description

Best models' selection is based on accuracy estimation. Top three model is used for the final ensemble method. Proposed approach methodology is represented.

In final phase, we must calculate weight by evolutionary algorithm named as genetic algorithm (GA) and optimize weight by GA applies to a weighted average method of an ensemble. We have taken three natures inspired algorithm (NIA) for the weight optimization out of which GA outperforms because it gives the best fitness function, best chromosome and function evolution is also less in comparison to both algorithm and time taken in weight optimization is also less so, it is used in weighted average ensemble method. This process is used to increase the efficiency of the proposed model. In classical weighted average method, we have assigned weight manually. So, the accuracy of the model was not increasing.

In our proposed solution we have calculated weight by GA. So, the accuracy of the model should be increased to comparing to classical method. Here we compare three NIA algorithm named PSO, DE, and GA. We use prediction of the top three model here with optimizing weight in weighted average ensemble method.

Breast cancer prediction using genetic algorithms involves the application of a computational technique inspired by natural evolution to select a subset of features that are most relevant for predicting breast cancer. In this context, a dataset containing various attributes and corresponding labels is utilized to train a machine learning model. The dataset typically consists of clinical and genomic information of patients diagnosed with breast cancer, including features such as age, tumor size, lymph node status, hormone receptor status, gene expression data, and more.

The genetic algorithm (GA) is an optimization method that simulates the process of natural selection to search for the best set of features. It starts with an initial population of potential feature subsets, where each subset represents a potential solution. The algorithm evaluates the fitness of each solution by training a predictive model using the selected features and measuring its performance, often using evaluation metrics like accuracy, sensitivity, specificity, or area under the receiver operating characteristic curve (AUC-ROC). The fittest individuals, i.e., the feature subsets with the best performance, are selected to create the next generation through processes like crossover and mutation, which mimic genetic recombination and variation. This iterative process continues until a termination criterion, such as reaching a maximum number of generations or achieving a desired level of performance, is met.

The dataset used for breast cancer prediction through genetic algorithms typically consists of a large number of instances (patient samples) and a varying number of attributes. The attributes can be categorized into different types, including demographic characteristics (age, ethnicity), clinical features (tumor size, lymph node status, histological grade), and molecular information (gene expression profiles, gene mutations, protein expression levels). These attributes provide crucial information for training the prediction model.

Preprocessing steps are often performed on the dataset to handle missing values, outliers, and feature normalization. Missing values can be imputed using techniques like mean imputation, regression imputation, or imputing based on attribute distributions. Outliers, if present, may be treated by either removing them or applying robust statistics. Feature normalization techniques like z-score normalization or min-max scaling may be applied to ensure that all features have a similar range or distribution.

The dataset is then divided into training and testing sets. The training set is used to build the predictive model, while the testing set is used to evaluate its performance on unseen data. Cross-validation techniques like k-fold cross-validation or stratified sampling may also be employed to ensure robust evaluation of the model.

To train the predictive model, various machine learning algorithms can be utilized, such as logistic regression, support vector machines (SVM), decision trees, random forests, or neural networks. The genetic algorithm is used as a feature selection mechanism, guiding the search for an optimal subset of features that maximizes the prediction performance.

During the genetic algorithm optimization process, the fitness function plays a crucial role. It quantifies the performance of each feature subset, typically by utilizing a specific evaluation metric. The choice of fitness function depends on the problem's nature and the desired objectives, such as maximizing accuracy, sensitivity, specificity, or AUC-ROC.

The genetic algorithm iteratively selects feature subsets, evaluates their performance using the fitness function, applies genetic operators like crossover and mutation to create new generations, and repeats this process until convergence or termination conditions are met. The convergence occurs when the algorithm reaches a stable or satisfactory subset of features that consistently produce good predictive performance.

In conclusion, breast cancer prediction using genetic algorithms involves the utilization of a dataset containing clinical and genomic information to train a predictive model. The genetic algorithm optimizes the selection of the most relevant features, aiding in the accurate prediction of breast cancer. This iterative process involves evaluating fitness, applying genetic operators, and iteratively improving the feature subset until convergence. The dataset used for this task typically comprises various attributes related to demographics, clinical information, and molecular.

Genetic Algorithm

A genetic algorithm (GA) is a type of optimization algorithm based on the principles of natural selection and genetics. It is used to find solutions to problems that involve finding the best combination of parameters, where brute-force search methods would be impractical or inefficient.

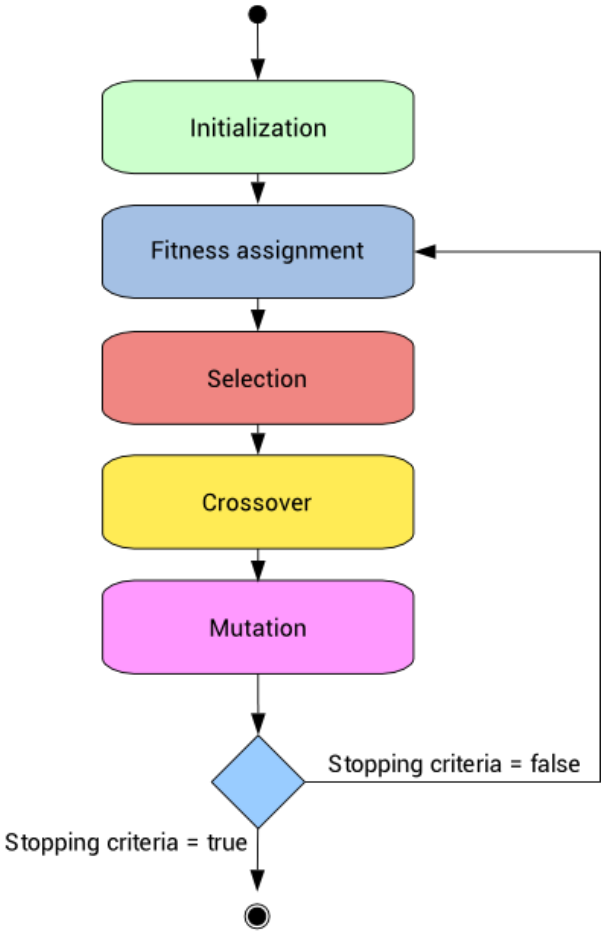

FIGURE 2.1 Phases of Genetic Algorithm

The basic idea behind genetic algorithms is to create a population of candidate solutions (often referred to as individuals) and then use a process of selection, reproduction, and mutation to evolve the population towards better solutions over time. The selection process favors individuals with better fitness, which is a measure of how well they perform on the given problem. Reproduction involves creating new individuals by combining the genetic material of existing individuals, and mutation introduces random changes to the genetic material to promote diversity.

The process is repeated for a number of generations, with each generation consisting of a new population of individuals. As the generations progress, the hope is that the population will converge towards better solutions, and the algorithm will terminate when a satisfactory solution is found, or after a fixed number of generations.

Genetic algorithms have been successfully applied to a wide range of problems, such as optimizing mathematical functions, designing electronic circuits, and even creating art. However, they can be computationally expensive and require careful parameter tuning to achieve good results.

Algorithm 1 Genetic Algorithms Framework begin n=0

- Random initialization of population p(n) The fitness of population determines p(t) while n=n+1 do Selection of parent from population p(n) Crossover operation perform on parents to create offspring's (n+1) Mutation operation perform (n+1)
- The fitness of population determines (n+1) end
  while Till best individual in the pop

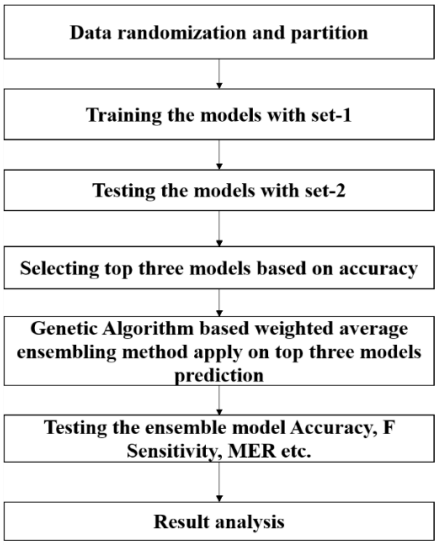

Figure 2.2 Architecture of The Model

Genetic Algorithm (GA) is a search and optimization technique inspired by the principles of natural selection and genetics. It is a powerful algorithm that mimics the process of evolution to find optimal solutions to complex problems. In a nutshell, GA starts with a population of potential solutions, applies genetic operators such as selection, crossover, and mutation, and iteratively evolves the population to improve the quality of solutions over generations.

1. **Initialization:** The algorithm starts by creating an initial population of potential solutions called individuals. Each individual represents a possible solution to the problem at hand and is typically encoded as a string or a binary chromosome.

2. **Fitness Evaluation:** Each individual in the population is evaluated and assigned a fitness value, which quantifies how well it solves the problem. The fitness function measures the objective or fitness criterion of the problem. It could be maximizing a certain value, minimizing an error, or achieving a specific target.

3. **Selection:** The selection process determines which individuals will contribute to the next generation. Individuals with higher fitness values have a higher probability of being selected, mimicking the principle of "survival of the fittest." Various selection methods can be used, such as tournament selection, roulette wheel selection, or rank-based selection.

4. **Reproduction (Crossover):** Crossover is a genetic operator that simulates the reproduction process. It involves combining genetic material from two selected individuals to create offspring. The crossover point(s) are randomly chosen in the chromosomes, and the genetic material beyond that point is exchanged between the parents to generate new solutions. The purpose of crossover is to explore and exploit different combinations of features from the parent solutions.

5. **Mutation:** Mutation is another genetic operator that introduces random changes in the offspring's genetic material. It helps introduce new variations in the population and prevents the algorithm from converging prematurely to a suboptimal solution. Mutation typically involves randomly flipping or modifying a small portion of the chromosome.

6. **Replacement:** The offspring generated through crossover and mutation are introduced into the population, replacing a portion of the existing individuals. The replacement strategy ensures the population size remains constant across generations.

7. **Termination:** The algorithm continues to iterate through the generations, repeating the selection, crossover, mutation, and replacement steps until a termination criterion is met. The termination criterion can be a maximum number of generations, a specific fitness threshold, or reaching a predefined time limit.

8. **Convergence and Solution:** Over generations, the population evolves and improves its fitness, gradually converging towards optimal or near-optimal solutions. The final solution(s) obtained at the end of the algorithm represents the best solution found by the genetic algorithm for the given problem.

Genetic Algorithms offer several advantages:

- **Global Search:** Genetic Algorithms excel in exploring large search spaces and finding global optima. Their ability to maintain diverse populations allows them to escape local optima and discover better solutions.

- **Versatility:** Genetic Algorithms can be applied to a wide range of problems, including optimization, machine learning, scheduling, and design. They do not require assumptions about the problem's mathematical properties or differentiability.

- **Parallelism:** Genetic Algorithms can be easily parallelized, as different individuals in the population can be evaluated, selected, and processed concurrently, leading to significant speedups on multi-core or

distributed systems.

- **Exploration and Exploitation:** Genetic Algorithms strike a balance between exploration (through crossover and mutation) and exploitation (through selection) of the search space. This allows them to efficiently explore promising regions while exploiting the existing knowledge.

However, Genetic Algorithms also have some considerations:

- **Computational Complexity:** The time complexity of Genetic Algorithms can be high, especially for large populations or complex fitness functions. The algorithm may require significant.

Chapter -3

PROPOSED METHODOLOGY

Breast cancer prediction using genetic algorithm involves selecting a set of features that are most relevant to the prediction of breast cancer. The proposed methodology can be divided into the following steps:

1. Data Preprocessing: The first step in the methodology is to preprocess the data. This involves cleaning the data, handling missing values, and converting categorical data into numerical data. The data should be prepared in a format that can be used by the genetic algorithm.

2. Feature Selection: The next step is to select a subset of features from the preprocessed data that are most relevant to breast cancer prediction. This can be achieved by using a genetic algorithm that searches for the best combination of features that can predict breast cancer accurately.

3. Genetic Algorithm: The genetic algorithm is used to search for the optimal subset of features. The algorithm involves the following steps:

a. Initialization: A population of potential solutions is created randomly. Each solution represents a subset of features that can be used to predict breast cancer.

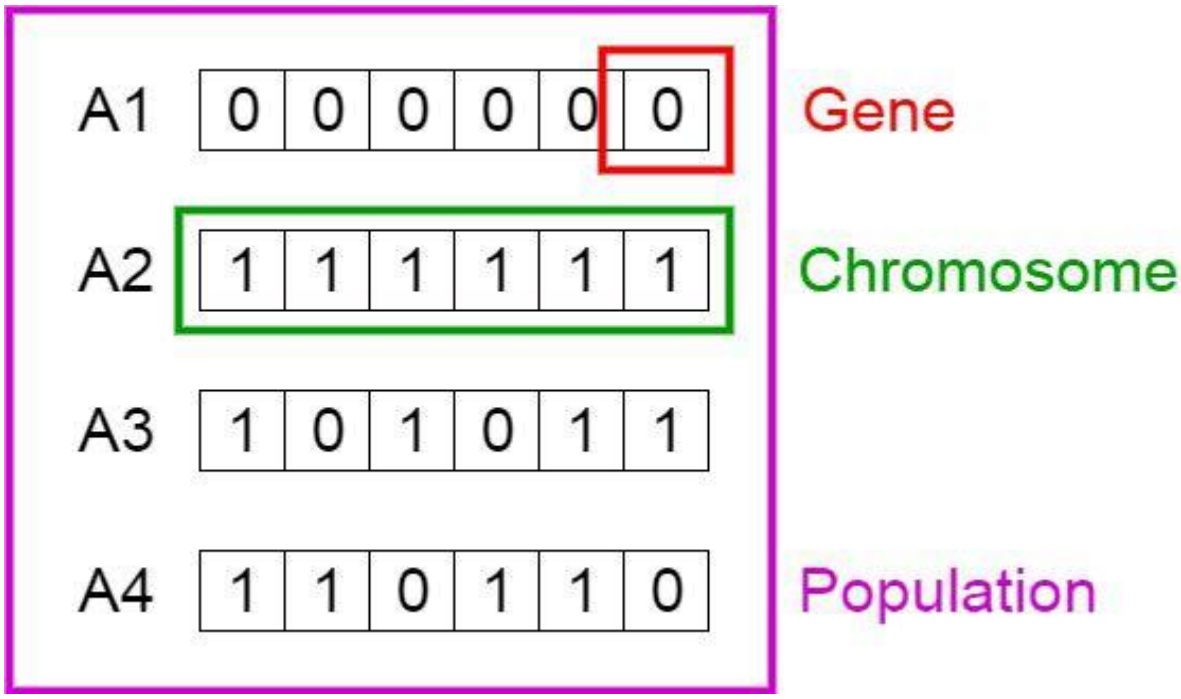

Figure 3.1 Initialize Population

b. Fitness Evaluation: The fitness of each solution is evaluated by training a model using the selected subset of features and testing it on a validation set. The fitness of a solution is based on the accuracy of the model.

c. Selection: The fittest solutions are selected to form the next generation.

d. Crossover: The selected solutions are combined using crossover to create new solutions.

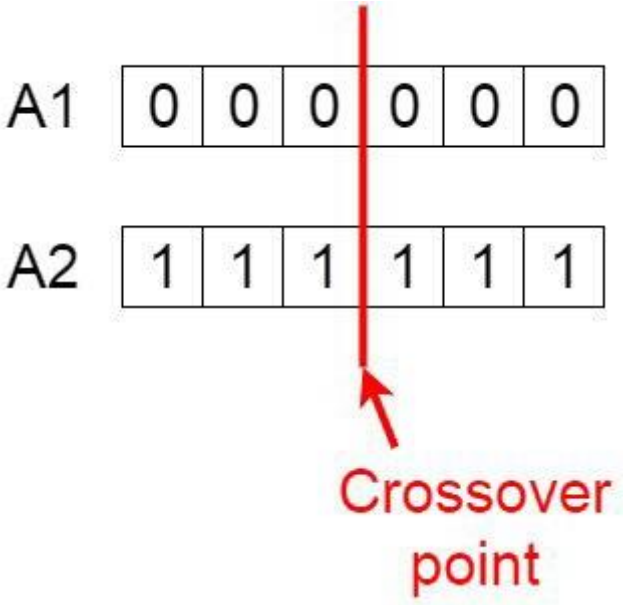

Figure 3.2 Crossover Point

e. Mutation: Some of the new solutions are randomly mutated to introduce new features.

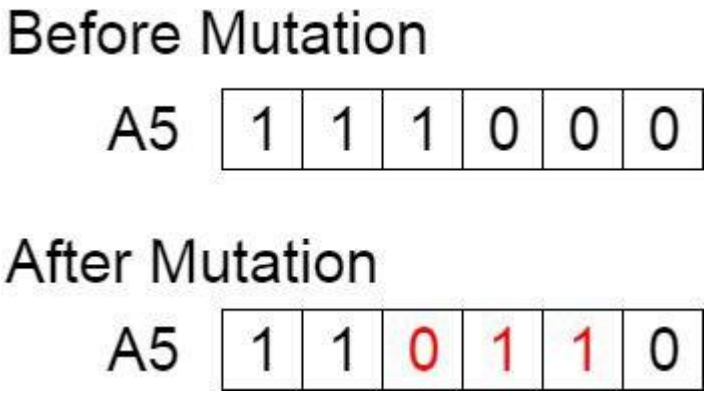

Figure 3.3 Mutation

f. Termination: The process is repeated until a stopping criterion is met, such as reaching a maximum number of generations or achieving a satisfactory accuracy.

4. Model Training and Evaluation: Once the optimal subset of features is selected, a model is trained using the selected features and tested on a separate test set. The accuracy of the model is evaluated using metrics such as sensitivity, specificity, and F1 score.

5. Interpretation of Results: The final step is to interpret the results and draw conclusions about the predictive power of the selected features. This may involve analyzing the importance of individual features and their contribution to the accuracy of the model.

3.1 Correlation Analysis

Correlation analysis is a statistical technique that measures the strength and direction of the relationship between two variables. In the context of breast cancer prediction, correlation analysis can be used to identify potential genetic markers or risk factors that are associated with the development of breast cancer.

Genetic algorithm is a metaheuristic optimization algorithm that is inspired by the process of natural selection. In the context of breast cancer prediction, genetic algorithm can be used to search for the most informative set of genetic markers or risk factors that can accurately predict the likelihood of developing breast cancer.

To perform correlation analysis of breast cancer prediction using genetic algorithm, the following steps can be taken:

1. Collect data: Collect a dataset that includes genetic markers and other risk factors associated with breast cancer, as well as information about whether each individual in the dataset has developed breast cancer or not.

2. Preprocess data: Preprocess the data to remove any missing values, outliers, or other anomalies that may affect the accuracy of the analysis.

3. Perform correlation analysis: Use correlation analysis techniques such as Pearson correlation coefficient or Spearman rank correlation coefficient to measure the strength and direction of the relationship between each genetic marker or risk factor and the development of breast cancer.

4. Apply genetic algorithm: Use genetic algorithm to search for the most informative set of genetic markers or risk factors that can accurately predict the likelihood of developing breast cancer.

5. Evaluate performance: Evaluate the performance of the model using metrics such as accuracy, sensitivity, specificity, and area under the curve (AUC) of the receiver operating characteristic (ROC) curve.

6. Interpret results: Interpret the results of the analysis to identify potential genetic markers or risk factors that are associated with the development of breast cancer and can be used to improve breast cancer prediction and prevention.

Overall, correlation analysis of breast cancer prediction using genetic algorithm can help identify potential genetic markers or risk factors associated with breast cancer and can lead to the development of more accurate and personalized breast cancer screening and prevention strategies.

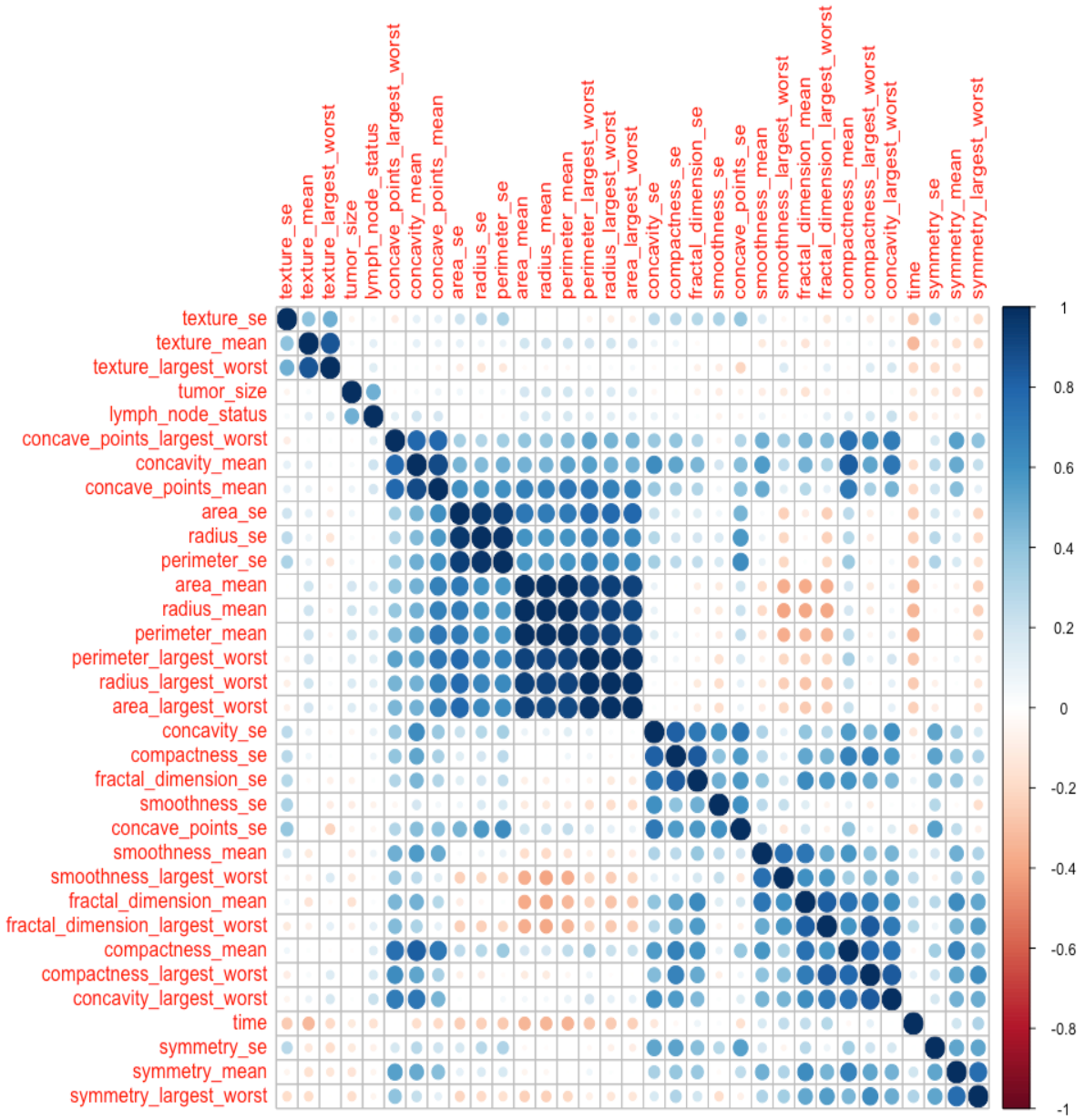

Figure 3.4 Correlation Analysis

3.2 Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) is a critical step in any data science project, including breast cancer prediction using a genetic algorithm. In this task, the goal is to use data on patients' genetic characteristics to predict whether they have breast cancer or not.

The first step in EDA is to acquire and understand the data. The dataset should contain information on patients' genetic characteristics, such as gene expression levels, as well as information on their cancer status. Once the data is acquired, it is essential to clean and preprocess it. This process can include removing missing values, normalizing the data, and selecting relevant features.

Next, it is crucial to explore the data to gain insights and identify patterns. This can be done using visualization techniques such as scatterplots, histograms, and box plots. These visualizations can help identify any outliers, the distribution of the data, and potential relationships between features.

Once the data has been explored, it is important to prepare it for the genetic algorithm. This can involve splitting the data into training and testing sets, selecting appropriate features, and normalizing the data.

After the data preparation, the genetic algorithm can be applied to the training set to develop a model that can predict breast cancer based on the patient's genetic characteristics. The model's accuracy can then be evaluated on the testing set to determine its effectiveness in predicting breast cancer.

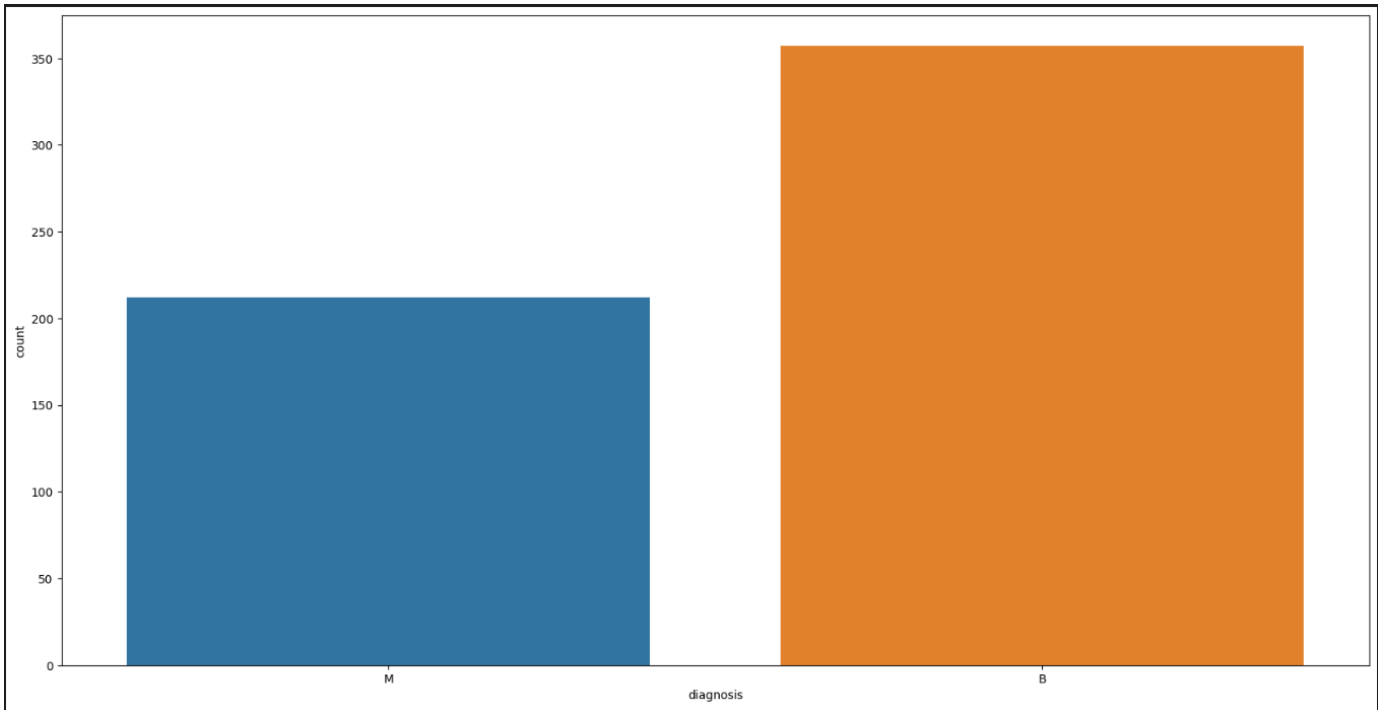

Figure 3.5 Exploratory Data Analysis

Chapter -4

METHODS USED

Breast cancer is a significant health concern worldwide, and early detection plays a crucial role in improving patient outcomes. Genetic algorithms (GAs) are powerful optimization techniques inspired by natural evolution, and they have been applied to breast cancer prediction to improve accuracy and feature selection. In this article, we will explore the methods used in breast cancer prediction using genetic algorithms.

1. **Data Collection and Preprocessing:**
The first step in any predictive modeling task is data collection. For breast cancer prediction, datasets containing clinical and genetic information of patients are gathered. These datasets often include features such as patient age, tumor size, hormone receptor status, gene expression profiles, and more. Data preprocessing techniques are then employed to handle missing values, normalize or standardize features, and deal with categorical variables.

2. **Feature Selection:**
Genetic algorithms are commonly used for feature selection in breast cancer prediction. Feature selection aims to identify the most relevant and informative features that contribute to the accurate prediction of breast cancer. By selecting a subset of relevant features, the model can improve efficiency, reduce overfitting, and enhance interpretability. Genetic algorithms search through the feature space to find an optimal subset of features that maximizes the classification performance. The fitness function evaluates the quality of each feature subset based on classification accuracy or other evaluation metrics.

3. **Encoding and Initialization:**
To represent the potential solutions, genetic algorithms employ a suitable encoding scheme. For feature selection, a binary encoding scheme is often used, where each bit represents the presence or absence of a feature in the subset. The initial population is then randomly generated, with each individual representing a potential solution (i.e., a feature subset).

4. **Genetic Operators:**
Genetic algorithms utilize genetic operators such as selection, crossover, and mutation to evolve the population over generations. In selection, individuals with higher fitness scores have a higher probability of being selected as parents for the next generation. Crossover combines genetic information from two parents to produce offspring with a combination of features. Mutation introduces random changes in the offspring's genetic information to maintain diversity in the population.

5. **Fitness Evaluation:**
After generating the offspring population, the fitness of each individual is evaluated using the fitness function. The fitness function measures the performance of the feature subset in terms of classification accuracy, area under the receiver operating characteristic curve (AUC-ROC), or other appropriate metrics. The higher the fitness score, the better the performance of the feature subset.

6. **Termination Criteria:**
Genetic algorithms run for multiple generations until a termination criterion is met. Termination criteria can be a fixed number of generations, convergence of the fitness scores, or a combination of both. The algorithm aims to converge to an optimal or near-optimal feature subset that maximizes the predictive performance.

7. **Validation and Evaluation:**
To assess the performance of the selected feature subset, it is crucial to evaluate the model using independent validation data. Typically, the dataset is divided into training and testing sets. The selected features are used to train a suitable classification model, such as logistic regression, support vector machines (SVM), or artificial neural networks (ANN). The trained model is then evaluated on the testing set, and performance metrics such as accuracy, precision, recall, F1 score, and AUC-ROC are calculated.

8. **Parameter Tuning:**
Genetic algorithms have several parameters that require tuning for optimal performance. These parameters include the population size, the number of generations, the selection pressure, and the crossover and mutation rates. The performance of the genetic algorithm can vary based on the chosen parameter values. Techniques such as grid search, random search, or cross-validation can be employed to find the optimal parameter configuration.

Breast cancer prediction using genetic algorithms offers a

powerful approach for feature selection and improving classification performance. By searching through the feature space, genetic algorithms help identify relevant features that contribute to accurate predictions. Additionally, the use of genetic operators ensures diversity and exploration of the search space. However, it is important to note that genetic algorithms are computationally expensive and may require significant computational resources.

In conclusion, the application of genetic algorithms in breast cancer prediction involves data preprocessing, feature selection, encoding, genetic operators, fitness evaluation, termination criteria, validation, evaluation, and parameter tuning. These methods collectively contribute to enhancing the accuracy and interpretability of breast cancer prediction models, aiding in early detection and improving patient outcomes.

(1) Logistic Regression

Logistic Regression is a statistical learning algorithm used for binary classification problems. Despite its name, logistic regression is primarily used for classification tasks rather than regression tasks. It is a simple yet powerful algorithm that can model the relationship between the input variables and the probability of a binary outcome.

Here's how the Logistic Regression algorithm works:

1. **Data Preparation:** Logistic Regression requires a labeled dataset for training. The dataset is divided into a feature matrix (input variables) and a target vector (binary output variable).

2. **Model Building:** Logistic Regression models the relationship between the input variables and the probability of the binary outcome using the logistic function (also known as the sigmoid function). The logistic function maps any real-valued number to a value between 0 and 1. It has an S-shaped curve and can interpret the input as the log-odds or logit of the probability.

3. **Parameter Estimation:** During the training phase, the logistic regression algorithm estimates the parameters (weights) that best fit the data. This is typically done using an optimization algorithm such as maximum likelihood estimation or gradient descent. The goal is to find the parameter values that maximize the likelihood of observing the given data.

4. **Decision Threshold:** Once the logistic regression model is trained, it can make predictions on new, unseen data points. The model calculates the predicted probability of the positive class (e.g., class 1). To obtain the binary classification, a decision threshold is applied. Typically, a threshold of 0.5 is used, where probabilities above the threshold are classified as the positive class, and probabilities below the threshold are classified as the negative class.

Logistic Regression has several advantages:

- It is relatively simple and computationally efficient.
- It can handle both numerical and categorical input features.
- Logistic Regression provides interpretable results, as the coefficients can indicate the impact of each feature on the outcome.
- It can provide probabilistic predictions, allowing for a measure of uncertainty.

However, Logistic Regression also has some limitations:

- It assumes a linear relationship between the input features and the log-odds of the outcome. If the true relationship is highly non-linear, logistic regression may not perform well.
- Logistic Regression is prone to overfitting when there are many irrelevant or correlated features. Regularization techniques, such as L1 or L2 regularization, can help mitigate this issue.
- It is primarily designed for binary classification tasks and may not perform as well on multi-class problems

without modification, such as one-vs-rest or softmax regression.

Despite its simplicity, logistic regression remains a widely used and effective algorithm in various domains, including healthcare, finance, and social sciences. It is especially useful when interpretability and probabilistic predictions are desired, and when dealing with relatively small to moderate-sized datasets.

(2) k-Nearest Neighbor (k-NN)

The k-Nearest Neighbor (k-NN) algorithm is a simple machine learning algorithm used for classification and regression tasks. The algorithm works by finding the k closest data points to a new data point in the training set, and then using those k neighbors to predict the label or value of the new data point. The value of k is a hyperparameter that needs to be specified before training the model.

Here are the basic steps of the k-NN algorithm:

1. Load the dataset: The first step is to load the dataset into memory.

2. Define the value of k: The next step is to define the value of k, which is the number of nearest neighbors to consider. The value of k can be determined using cross-validation or other techniques.

3. Normalize the data: It is often helpful to normalize the data to ensure that each feature contributes equally to the distance calculations.

4. Calculate the distance: The distance between the new data point and all the data points in the training set is calculated using a distance metric, such as Euclidean distance.

5. Find the k-nearest neighbors: The k-nearest neighbors to the new data point are identified based on the shortest distances.

6. Make a prediction: For classification tasks, the most common class among the k-nearest neighbors is chosen as the predicted class for the new data point. For regression tasks, the average or median value of the k-nearest neighbors is chosen as the predicted value for the new data point.

7. Evaluate the model: Finally, the performance of the model is evaluated using a metric such as accuracy, precision, recall, or mean squared error, depending on the task.

The k-NN algorithm is simple to implement and can work well on small datasets. However, it can be computationally expensive for large datasets, and the choice of k can have a significant impact on the performance of the algorithm. Additionally, the algorithm does not learn any underlying patterns in the data and can be sensitive to irrelevant features.

(3) Support Vector machine

Support Vector Machine (SVM) is a popular machine learning algorithm used for classification and regression analysis. SVM is a type of supervised learning algorithm, meaning that it requires labeled data to train on. The goal of SVM is to find a hyperplane that separates the data into classes, with the largest margin possible.

Here are the steps involved in the SVM algorithm:

1. Data preparation: SVM requires labeled data, where each data point is associated with a class label. The data is split into a training set and a testing set.

2. Feature selection: The next step is to select a subset of the most important features that will be used to separate the data into classes. This is known as feature selection or feature engineering.

3. Training the model: SVM finds the hyperplane that maximizes the margin between the classes. The hyperplane is found by solving an optimization problem that involves maximizing the distance between the hyperplane and the closest points from each class. The optimization problem can be solved using a variety of algorithms, including quadratic programming, gradient descent, and interior-point methods.

4. Model evaluation: Once the model is trained, it is evaluated on the testing set to determine its accuracy. The accuracy is a measure of how well the model can classify new data points.

There are several variations of the SVM algorithm, including linear SVM, polynomial SVM, and radial basis function (RBF) SVM. Each variation uses a different kernel function to transform the data into a higher-dimensional space, where it can be separated into classes more easily.

SVM is widely used in applications such as image recognition, text classification, and bioinformatics. Its popularity is due to its ability to handle high-dimensional data and its robustness to noise and outliers.

(4) Random Forest

Random Forest is a popular machine learning algorithm that is used for both classification and regression tasks. It is an ensemble learning method that combines multiple decision trees to make predictions. Random Forest gets its name from the fact that it creates an ensemble of random decision trees.

Here's how the Random Forest algorithm works:

1. **Data Preparation:** First, you need a labeled dataset to train the Random Forest model. The dataset is divided into two parts: the feature matrix (input variables) and the target vector (output variable).

2. **Random Sampling:** Random Forest uses a technique called bootstrap aggregating, or bagging, to create multiple subsets of the original dataset. Each subset is created by randomly sampling the original dataset with replacement. These subsets are known as training samples.

3. **Decision Tree Creation:** For each training sample, a decision tree is created. However, unlike traditional decision trees, Random Forest introduces randomness in two ways. First, it randomly selects a subset of features from the original feature set. Second, it only considers a random subset of the training data to build each decision tree. These random selections introduce diversity among the decision trees.

4. **Voting and Prediction:** Once all the decision trees are constructed, they are used to make predictions on new data points. In the case of classification tasks, each tree in the Random Forest predicts the class label, and the final prediction is determined by majority voting. For regression tasks, the average of the predicted values from all the trees is taken as the final prediction.

Random Forest has several advantages:

- It is robust against overfitting because it combines multiple decision trees, each trained on a different subset of the data.
- It can handle large datasets with high dimensionality.
- It provides a measure of feature importance, which helps in feature selection.
- It can handle both numerical and categorical features without requiring extensive preprocessing.

However, Random Forest also has some limitations:

- It may be slower to train and make predictions compared to simpler models like linear regression or decision trees.
- The resulting model may be difficult to interpret due to the ensemble of trees.
- It may not perform well on imbalanced datasets, where the class distribution is skewed.

Overall, Random Forest is a versatile and widely used algorithm that performs well in various machine learning tasks, particularly when there is a need for accurate predictions and handling complex datasets.

Chapter: 5

Hardware and Software Requirements

REQUIREMENT SPECIFICATIONS

Software requirements deal with software and hardware deal with resources that need to be installed on a server which provides optimal functioning for the application. These software and hardware requirements need to be installed before the packages are installed. These are the most common set of requirements defined by any operating system. These software and hardware requirements provide compatible support to the operation system in developing an application.

SOFTWARE REQUIREMENTS

The software requirements specify the use of all required software products like data management systems. The required software product specifies the numbers and version. Each interface specifies the purpose of the interfacing software as related to this software product.

Operating system: Windows 7/10
Coding Language: Python 3, Java
IDE: Jupiter Notebook, Eclipse

HARDWARE REQUIREMENTS

The hardware requirement specifies each interface of the software elements and the hardware elements of the system. These hardware requirements include configuration characteristics.

System: Pentium IV 2.4 GHz.
Hard Disk: 100 GB.
Monitor: 15 VGA Color.
RAM: 4GB

Chapter: 6

Algorithms

Classification models and their description as represented. These models are trained for Set-1 data. The result is compared, and models are selected on the basis of accuracy. A confusion matrix is formed for the actual and predicted values. Additional performance measures used are described below.

1) TP = Correctly predicted value is known as a True Positive.
2) TN = Incorrectly predicted value is known as a True Negative.
3) FP = Correctly rejection of values is known as a False Positive.
4) FN = Incorrectly rejection of values is known as a False Negative.
5) TPR = True Positive rate
6) FPR = False positive rate
7) ER = Error rate
8) MER = Minimum error rate
9) MWL = Minimum cost-weighted error rate
10) Sensitivity: - The percentage of actual predicted values which are correctly predicted. It shows what percentage of actual positive cases were covered by the model.

$$Sensitivity(Sens) = \frac{TP}{(TP + FN)} \quad (1)$$

11) Specificity: - The percentage of actual unpredicted values which are correctly predicted. Specificity matters more than sensitivity.

$$Specificity(Spec) = \frac{TN}{TN + FP} \quad (2)$$

12) Precision: - The percentage of positive values that were correctly predicted.

$$Precision = \frac{TP}{TP + FN} \quad (3)$$

13) Recall:- It is the percentage of actual positive value were correctly predicted.

$$Recall = \frac{TP}{TP + FP} \quad (4)$$

14) F:- A good model should have a good precision as well as a high recall.
Combination of precision and recall is known as F1 score.

$$F1Score = \frac{(2 * Precision * Recall)}{(Precision + Recall)} \quad (5)$$

15) Accuracy:- The percentage of the total number of predictions that were correctly predicted.

$$Accuracy(Acc) = \frac{TP + TN}{TP + TN + FP + FN} \quad (6)$$

Chapter: 7

CONCLUSION AND FUTURE SCOPE

7.1 CONCLUSION

In this project, we have compared the accuracies of classification algorithms, such as Neutral Networks with the genetic algorithms for optimization and tell whether the cancer is benign or malignant and the accuracy for genetic algorithm is the best one. In conclusion, Genetic Algorithm (GA) is the best application to solve various common problems using fitness functions. Genetic Algorithm is an exhaustive approach which can be applied in the Artificial intelligence field to find optimal solutions in complex search spaces. It is a heuristic search algorithm that will exploit the historical information for the best solution. Genetic Algorithm works on a state space of potential solutions and selects maximum or optimal solution based on the fitness value of candidate solution. ... Genetic Algorithms avoid the problem of getting stuck at local maxima which is usually faced by Traditional Search techniques.

7.2 FUTURE SCOPE

The future scope of the project is that, to try and implement the other algorithms like SVM and Decision Trees and also develop user interface for the whole model.

APPENDIX A

(Code)

```
## General Library Imports
import warnings
warnings.filterwarnings("ignore")

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, OneHotEncoder, LabelEncoder
from sklearn.compose import ColumnTransformer
from sklearn.decomposition import TruncatedSVD
from sklearn.impute import SimpleImputer
from sklearn.pipeline import Pipeline


# libraries for models
from sklearn.neural_network import MLPClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.ensemble import AdaBoostClassifier, BaggingClassifier, GradientBoostingClassifier,
RandomForestClassifier

# metrics evaluation libraries
from sklearn.metrics import auc, classification_report, confusion_matrix, roc_curve, RocCurveDisplay

## Data loading
project_data = pd.read_csv("data.csv")
project_data = project_data.drop(columns=["id"]) # droping unwanted columns

## Initial Analysis
project_data.head()
project_data.info()
project_data.describe()
```

	Radius_mean	Texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	poi
count	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	5
mean	14.127292	19.296678	91.969033	654.889104	0.096360	0.104341	0.088799	
std	3.524049	4.301816	24.298981	351.914129	0.014064	0.052813	0.079720	
min	6.981000	9.710000	43.790000	143.500000	0.052630	0.019380	0.000000	
25%	11.700000	16.170000	75.170000	420.300000	0.086370	0.064920	0.029560	
50%	13.370000	18.870000	86.240000	551.100000	0.095870	0.092630	0.061540	
75%	15.780000	21.800000	104.100000	782.700000	0.105300	0.130400	0.130700	
max	28.110000	39.280000	188.500000	2501.000000	0.163400	0.345400	0.426800	

8 rows × 30 columns

Figure A.1 Initial Analysis

```
project_data.shape
project_data.columns
project_data.isna().sum()
```

```
## Eploratory Data Analysis
plt.figure(figsize=(20,10))
sns.countplot(x=project_data["diagnosis"])
print(project_data["diagnosis"].value_counts())
```

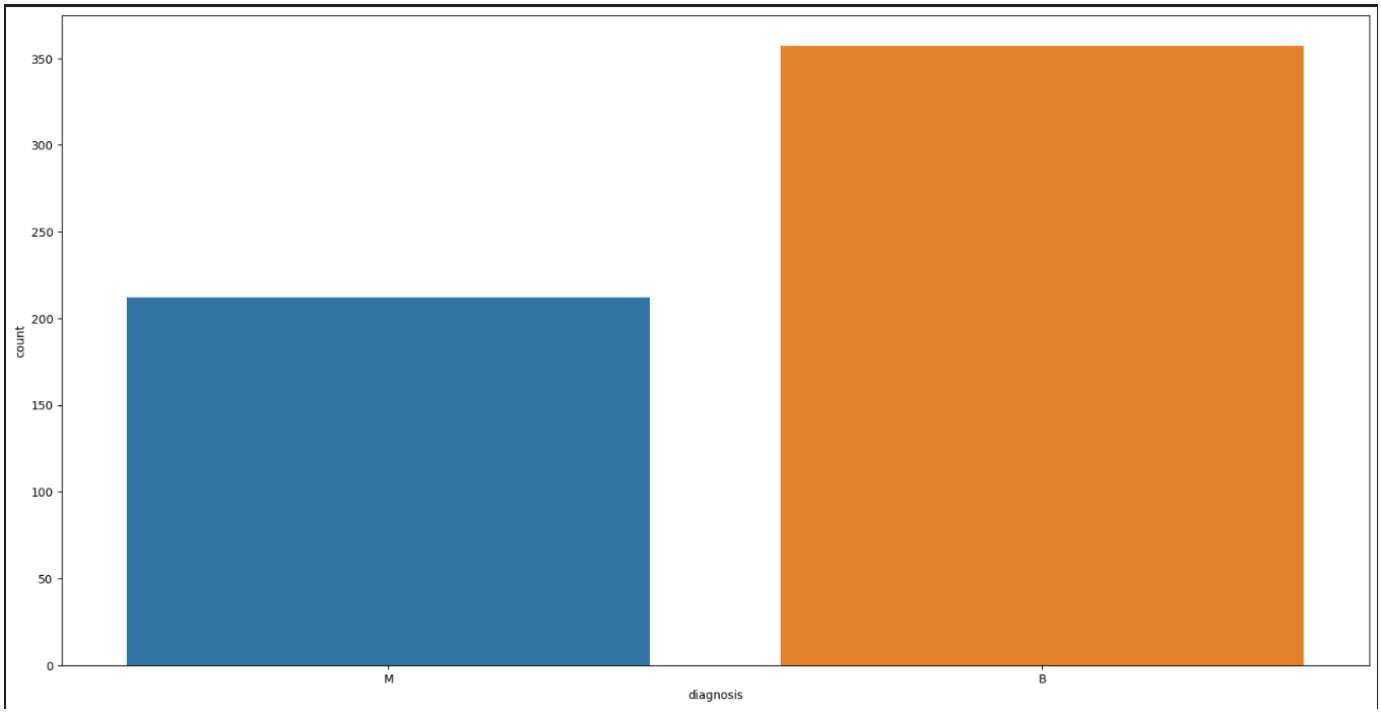

Figure A.2 Eploratory Data Analysis

```
numeric_columns = [column for column in project_data.columns if project_data[column].dtype ==
'float64']
print(numeric_columns)
for column in numeric_columns:
    plt.figure(figsize=(12,8))
    sns.kdeplot(data=project_data, x=column, hue='diagnosis', palette="crest", fill=True)
    plt.show()
```

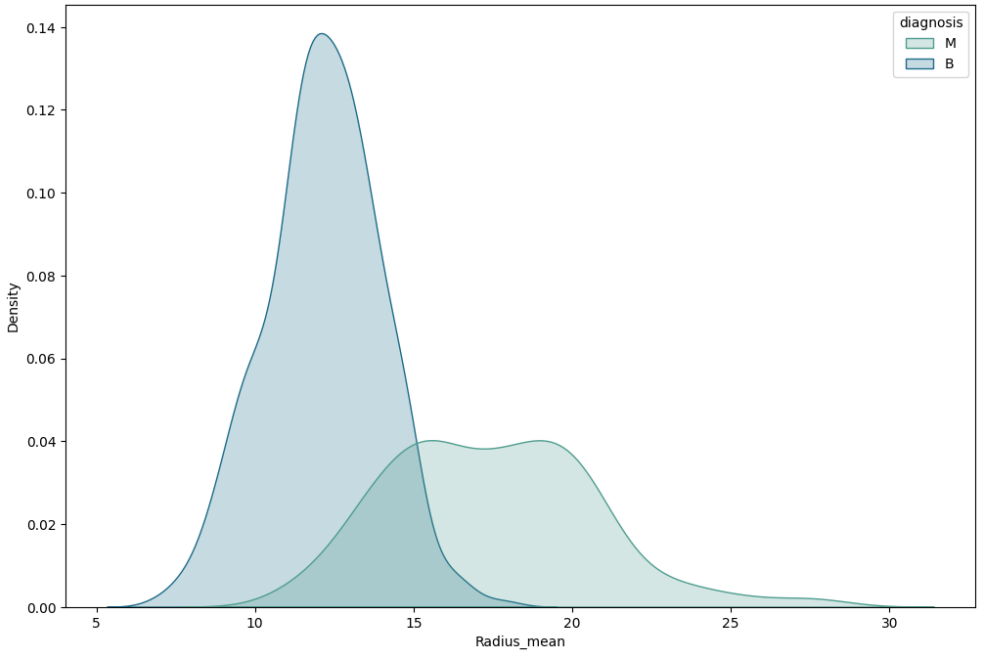

Figure A.3 EDA

```
## Correlation Analysis
plt.figure(figsize=(20,10))
corr = project_data.corr()
sns.heatmap(corr, annot=True, cmap="YlGnBu")
```

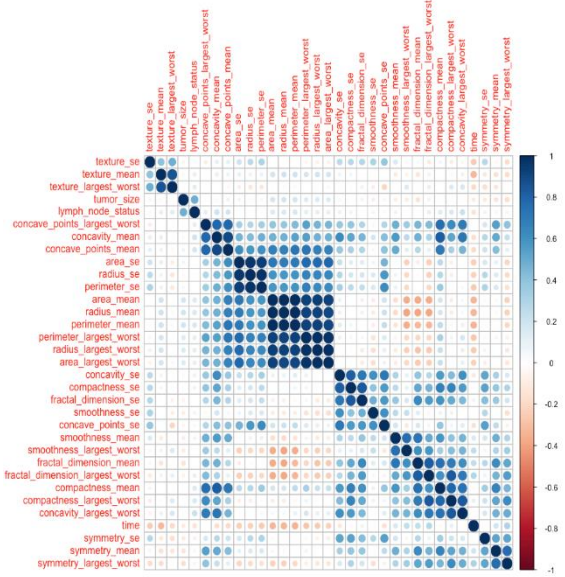

Figure A.4 Correlation Analysis

```
## Data Preprocessing and Pipelining
X_train=project_data.drop(columns=["diagnosis"])
y_train=project_data["diagnosis"]
X_train, X_test, y_train, y_test = train_test_split(X_train, y_train, test_size=0.2)
print('Train dataset shape:',X_train.shape)
print('Test dataset shape', y_train.shape)
numeric_columns = X_train.select_dtypes(exclude='object').columns
print(numeric_columns)
print('*'*100)
categorical_columns = X_train.select_dtypes(include='object').columns
print(categorical_columns)
numeric_features = Pipeline([
    ('handlingmissingvalues',SimpleImputer(strategy='median')),
    ('scaling',StandardScaler(with_mean=True))
])

print(numeric_features)
print('*'*100)

categorical_features = Pipeline([
    ('handlingmissingvalues',SimpleImputer(strategy='most_frequent')),
    ('encoding', OneHotEncoder()),
    ('scaling', StandardScaler(with_mean=False))
])

print(categorical_features)

processing = ColumnTransformer([
    ('numeric', numeric_features, numeric_columns),
    ('categorical', categorical_features, categorical_columns)
])

processing
```

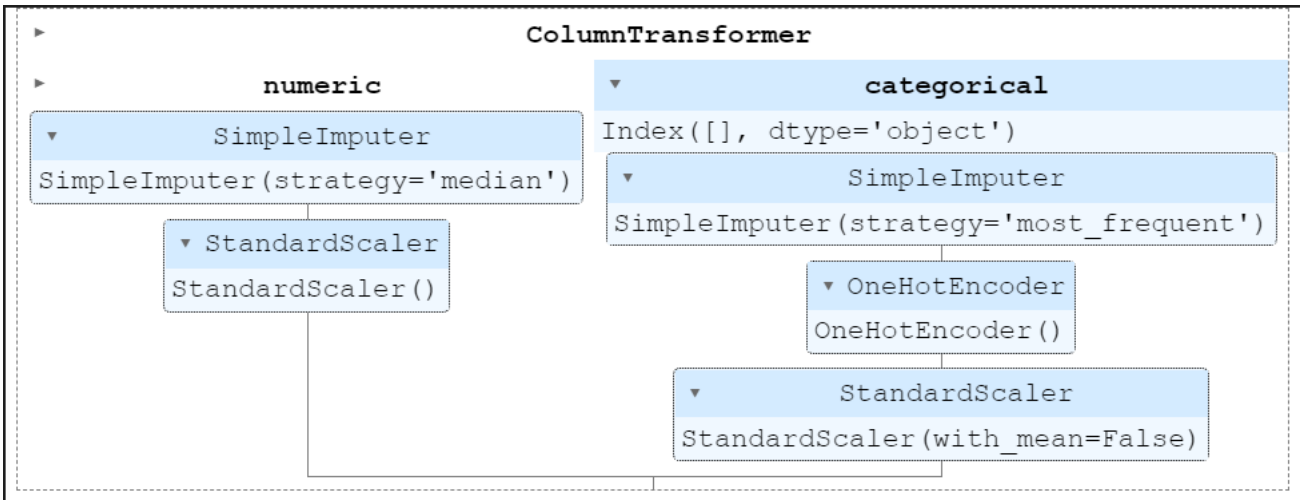

Figure A.5 Data Preprocessing and Pipelining

```
## Generic Methods for Model Preparation & Metric Evaliation
def prepare_model(algorithm):
    model = Pipeline(steps= [
        ('processing',processing),
        ('pca', TruncatedSVD(n_components=3, random_state=12)),
        ('modeling', algorithm)
    ])
    model.fit(X_train, y_train)
    return model

def prepare_confusion_matrix(algo, model):
    print(algo)
    plt.figure(figsize=(12,8))
    pred = model.predict(X_test)
    cm = confusion_matrix(y_test, pred)
    ax= plt.subplot()
    sns.heatmap(cm, annot=True, fmt='g', ax=ax)
    plt.show()

    # labels, title and ticks
    ax.set_xlabel('Predicted labels');ax.set_ylabel('True labels');
    ax.set_title('Confusion Matrix');
def prepare_classification_report(algo, model):
    print(algo+' Report :')
    pred = model.predict(X_test)
    print(classification_report(y_test, pred))
def prepare_roc_curve(algo, model):
    print(algo)
    y_pred_proba = model.predict_proba(X_test)[::,1]
    fpr, tpr, thresholds = roc_curve(y_test,  y_pred_proba)
    roc_auc = auc(fpr, tpr)
    curve = RocCurveDisplay(fpr=fpr, tpr=tpr, roc_auc=roc_auc)
    curve.plot()
    plt.show()

## Model Preparation
algorithms = [('bagging classifier', BaggingClassifier()),
              ('KNN classifier', KNeighborsClassifier()),
              ('Random Forest calssifier', RandomForestClassifier()),
              ('Adaboost classifier', AdaBoostClassifier()),
              ('Gradientboot classifier',GradientBoostingClassifier()),
              ('MLP', MLPClassifier())
             ]

trained_models = []
model_and_score = {}

for index, tup in enumerate(algorithms):
    model = prepare_model(tup[1])
    model_and_score[tup[0]] = str(model.score(X_train,y_train)*100)+"%"
    trained_models.append((tup[0],model))
```

```
## Model Evaluation
print(model_and_score)
for index, tup in enumerate(trained_models):
    prepare_confusion_matrix(tup[0], tup[1])
```

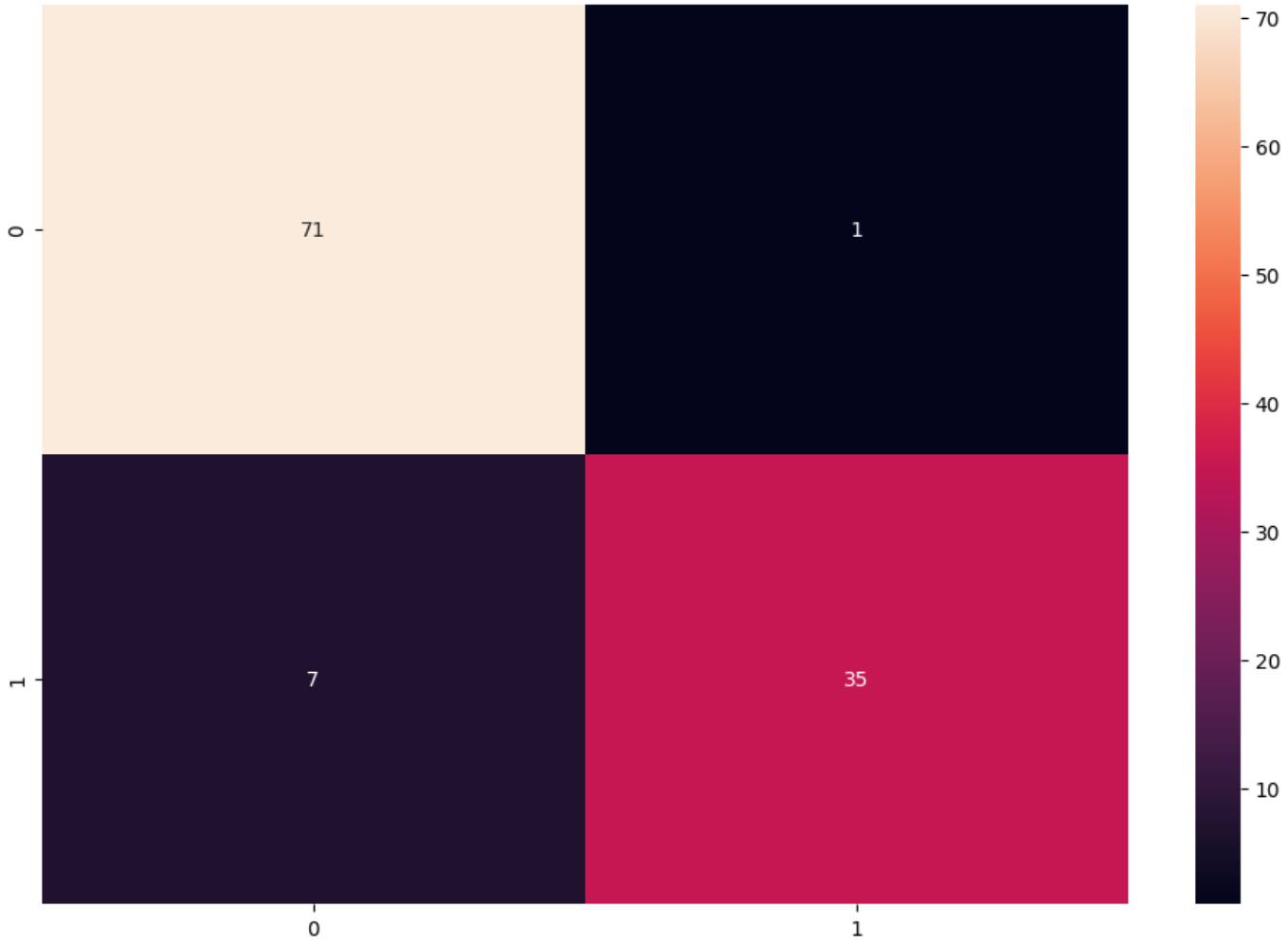

Figure A.6 Model Preparation

```
for index, tup in enumerate(trained_models):
    prepare_classification_report(tup[0], tup[1])
    print("\n")
```

bagging classifier Report :

	precision	recall	f1-score	support
B	0.91	0.99	0.95	72
M	0.97	0.83	0.90	42
accuracy			0.93	114
macro avg	0.94	0.91	0.92	114
weighted avg	0.93	0.93	0.93	114

KNN classifier Report :

	precision	recall	f1-score	support
B	0.90	0.97	0.93	72
M	0.94	0.81	0.87	42
accuracy			0.91	114
macro avg	0.92	0.89	0.90	114
weighted avg	0.91	0.91	0.91	114

Random Forest calssifier Report :

...

weighted avg	0.93	0.93	0.93	114

```
encoder = LabelEncoder()
y_test = encoder.fit_transform(y_test)

for index, tup in enumerate(trained_models):
    prepare_roc_curve(tup[0], tup[1])
```

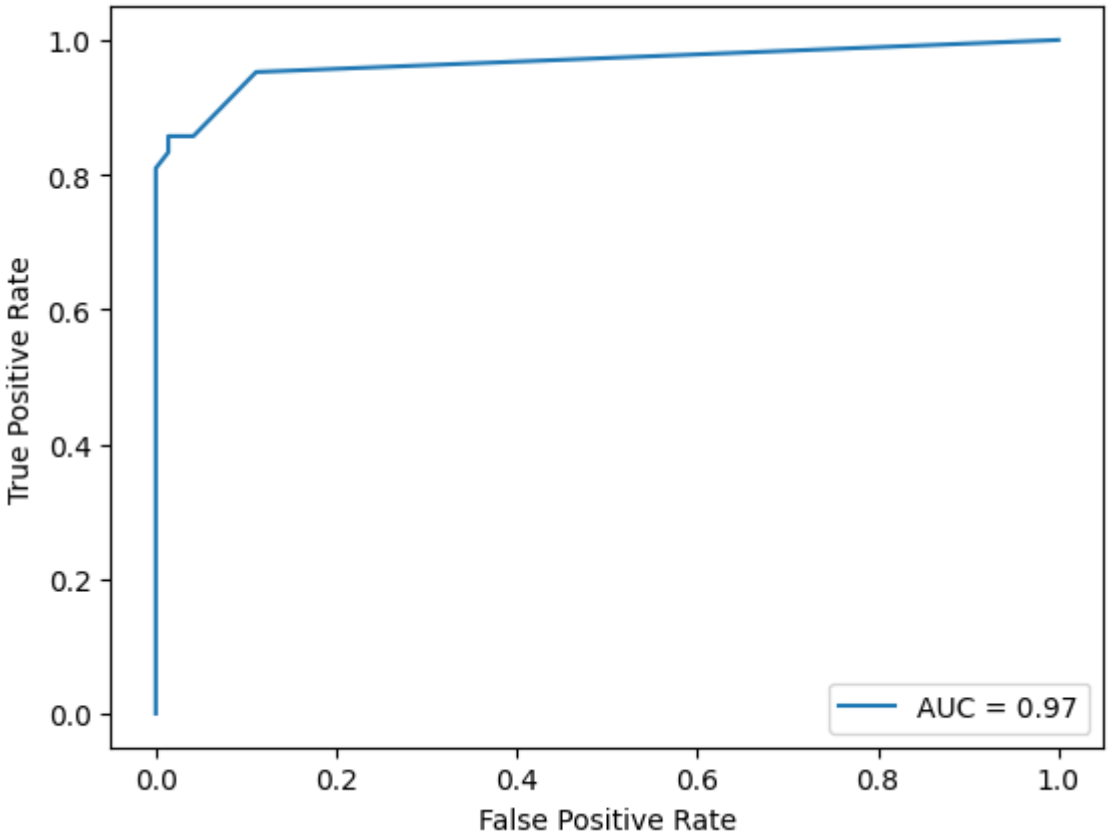

Figure A.7 Rate

Summary Of the Analysis

```
from prettytable import PrettyTable

x = PrettyTable(["Model", "Train Accuracy", "AUC SCORE"])
x.add_row(["bagging classifier","99.56","0.97"])
x.add_row(["KNN classifier","95.82","0.97"])
x.add_row(["Random Forest calssifier","100","0.97"])
x.add_row(["Adaboost classifier","100","0.96"])
x.add_row(["Gradientboot classifier","100","0.97"])
x.add_row(["MLP Classifier","96.92","0.98"])
print(x)
```

Model	Train Accuracy	AUC SCORE
bagging classifier	99.56	0.97
KNN classifier	95.82	0.97
Random Forest calssifier	100	0.97
Adaboost classifier	100	0.96
Gradientboot classifier	100	0.97
MLP Classifier	96.92	0.98

TABLE A.1: Analysis

SNAPSHOT

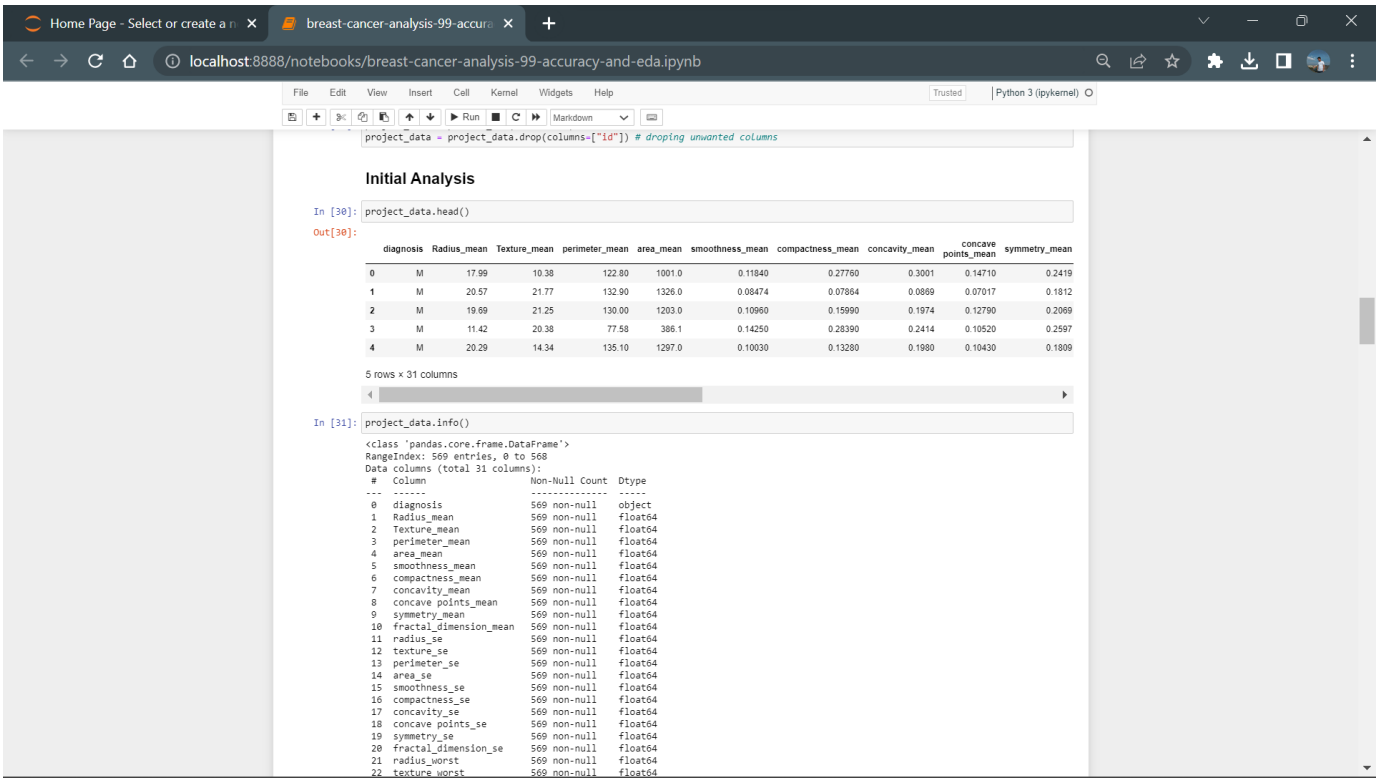

Figure A.8 Snapshot 1

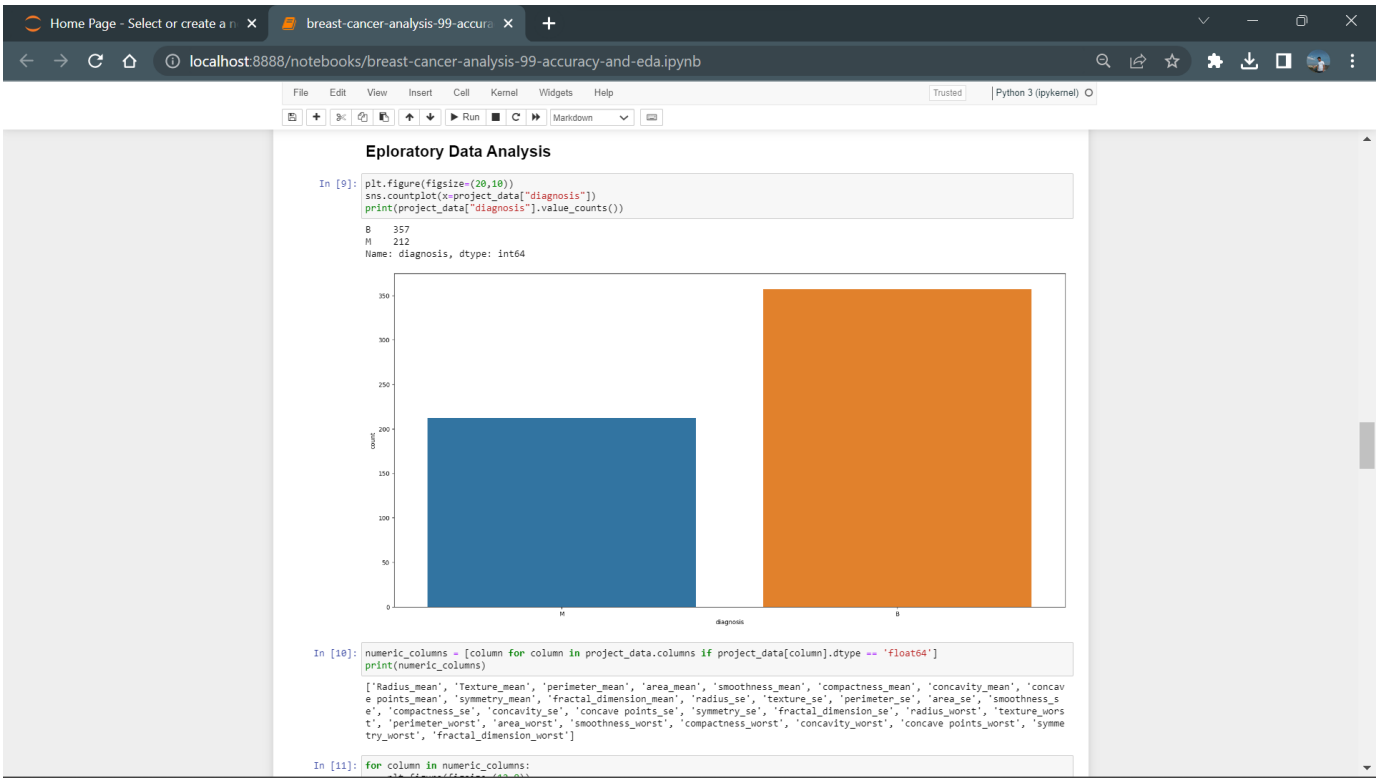

Figure A.9 Snapshot 2

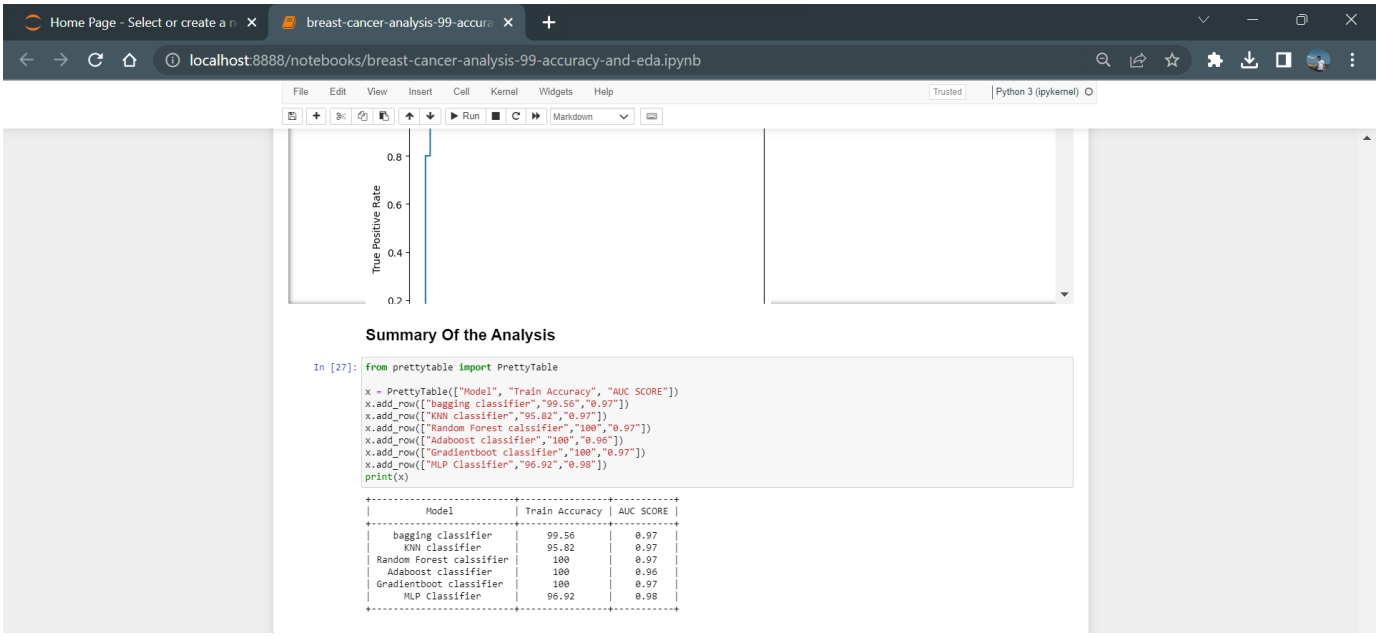

Figure A.10 Snapshot 3

References

[1] V Adegoke, Daqing Chen, Ebad Banissi, and Safia Barikzai. Prediction of breast cancer survivability using ensemble algorithms. 2017.

[2] Abdulsalam Alarabeyyat, Mohannad Alhanahnah, et al. Breast cancer detection using k-nearest neighbor machine learning algorithm.

[3] Dania Abed Aljawad, Ebtesam Alqahtani, AL-Kuhaili Ghaidaa, Nada Qamhan, Noof Alghamdi, Saleh Alrashed, Jamal Alhiyafi, and Sunday O Olatunji. Breast cancer surgery survivability prediction using bayesian network and support vector machines. In *Informatics, Health & Technology (ICIHT), International Conference on*, pages 1–6. IEEE, 2017.

[4] Hiba Asri, Hajar Mousannif, Hassan Al Moatassime, and Thomas Noel. Using machine learning algorithms for breast cancer risk prediction and diagnosis. *Procedia Computer Science*, 83:1064–1069, 2016.

[5] Mohamed Ettaouil and Youssef Ghanou. Neural architectures optimization and genetic algorithms. *Wseas Transactions On Computer*, 8(3):526–537, 2009.

[6] Pedro Ferreira, Ines Dutra, Rogerio Salvini, and Eliz-ˆ abeth Burnside. Interpretable models to predict breast cancer.

[7] Lydia D Isaac and C Sureshkumar. Diagnosis prognosis and prevention of breast cancer based on present scenario of human life.

[8] Md Milon Islam, Hasib Iqbal, Md Rezwanul Haque, and Md Kamrul Hasan. Prediction of breast cancer using support vector machine and k-nearest neighbors. In *Humanitarian Technology Conference (R10-HTC), 2017 IEEE Region 10*, pages 226–229. IEEE, 2017.

[9] Smita Jhajharia, Harish Kumar Varshney, Seema Verma, and Rajesh Kumar. A neural network based breast cancer prognosis model with pca processed features. In *Advances in Computing, Communications and Informatics (ICACCI), 2016 International Conference on*, pages 1896–1901.

[10] Divyansh Kaushik and Karamjit Kaur. Application of data mining for high accuracy prediction of breast tissue biopsy results. In *Digital Information Processing, Data Mining, and Wireless Communications (DIPDMWC), 2016 Third International Conference on*, pages 40–45. IEEE, 2016.

[11] Duc-Hau Le, Van-Huy Pham, and Thuy Thi Nguyen. An ensemble learning-based method for prediction of novel disease-microrna associations. In *Knowledge and Systems Engineering (KSE), 2017 9th International Conference on*, pages 7–12. IEEE, 2017.

[12] Bin Liu, Shanyi Wang, Qiwen Dong, Shumin Li, and Xuan Liu. Identification of dna-binding proteins by combining auto-cross covariance transformation and ensemble learning. *IEEE transactions on nanobioscience*, 15(4):328–334, 2016.

[13] Bin Liu, Shanyi Wang, Ren Long, and Kuo-Chen Chou. irspot-el: identify recombination spots with an ensemble learning approach. *Bioinformatics*, 33(1):35–41, 2016.

[14] Paritosh Pantola, Anju Bala, and Prashant Singh Rana. Consensus based ensemble model for spam detection. In *Advances in Computing, Communications and Informatics (ICACCI), 2015 International Conference on*, pages 1724–1727.

[15] Ahmed Iqbal Pritom, Md Ahadur Rahman Munshi, Shahed Anzarus Sabab, and Shihabuzzaman Shihab. Predicting breast cancer recurrence using effective classification and feature selection technique. In *Computer and Information Technology (ICCIT), 2016 19th International Conference on*, pages 310–314.

[16] Neha Rathore, Divya Tomar, and Sonali Agarwal. Predicting the survivability of breast cancer patients using ensemble approach. In *Issues and Challenges in Intelligent Computing Techniques (ICICT), 2014 International Conference on*, pages 459–464.

[17] Rishith Rayal, Divya Khanna, Jasminder Kaur Sandhu, Nishtha Hooda, and Prashant Singh Rana. N-semble: neural network based ensemble approach. *International Journal of Machine Learning and Cybernetics*, pages 1– 9, 2017.

[18] S Sathya, Sundeep Joshi, and S Padmavathi. Classification of breast cancer dataset by different classification algorithms. In *Advanced Computing and Communication Systems (ICACCS), 2017 4th International Conference on*, pages 1–4.

[19] Dongdong Sun, Minghui Wang, Huanqing Feng, and Ao Li. Prognosis prediction of human breast cancer by integrating deep neural network and support vector machine: Supervised feature extraction and classification for breast cancer prognosis prediction. In *Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), 2017 10th International Congress on*, pages 1–5. IEEE, 2017.

[20] Palli Suryachandra and P Venkata Subba Reddy. Comparison of machine learning algorithms for breast cancer. In *Inventive Computation Technologies (ICICT), International Conference on*, volume 3, pages 1–6.