Feature Selection Using Evolutionary Techniques

A

Minor Project (CC3270)

Report

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

Bachelor of Technology

in

Computer and Communication Engineering

By:

JITESH RANJAN PRUSTY
209205027

Under the guidance of:

Mr. Monu Bhagat

MANIPAL UNIVERSITY
JAIPUR

April - 2023

Department of Computer and Communication Engineering
School of Computer and Communication Engineering
Manipal University Jaipur
VPO. Dehmi Kalan, Jaipur, Rajasthan, India – 303007

Department of Computer and Communication Engineering
School of Computer and Communication Engineering, Manipal University Jaipur,
Dehmi Kalan, Jaipur, Rajasthan, India- 303007

STUDENT DECLARATION

I hereby declare that this project ***Feature Selection Using Evolutionary Techniques*** *is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma of the University or other Institute, except where due acknowledgements have been made in the text.*

Place: Manipal University Jaipur
Date: 21st April 2023

Jitesh Ranjan Prusty
209205027
B. Tech (CCE) 6th Semester

Department of Computer and Communication Engineering
School of Computing and Communication Engineering, Manipal University Jaipur,
Dehmi Kalan, Jaipur, Rajasthan, India- 303007

Date: 21st April 2023

CERTIFICATE FROM GUIDE

*This is to certify that the work entitled "**Feature Selection Using Evolutionary Techniques"** submitted by **Jitesh Ranjan Prusty** (209205027) to **Manipal University Jaipur** for the award of the degree of **Bachelor of Technology** in **Computer and Communication Engineering** is a bonafide record of the work carried out by him under my supervision and guidance from January 2023 to April 2023.*

Mr. Monu Bhagat
Department of Computer and Communication Engineering
Manipal University Jaipur

ACKNOWLEDGEMENT

I would like to show my sincere gratitude to everyone who helped me finish my small project successfully and provided me with advice. My project mentor **Mr. Monu Bhagat**, who gave me invaluable advice and support during the project, has my sincere gratitude. This project could not have been completed without their constant inspiration and support. I also want to express my gratitude to my friends for lending me their valuable time and knowledge to finish the project. Their insightful opinions and helpful suggestions played a crucial role in the project's development. I am also appreciative of my family, who have always supported and inspired me to go towards my objectives.

Finally, I want to express my gratitude to the institution for providing me with the tools and space I needed to do the project. Once more, I would like to express my gratitude to you all for your help in making this initiative a success.

Thank You.

ABSTRACT

In this abstract, the topic of feature selection using evolutionary algorithms—a well-liked technique for choosing pertinent characteristics in huge datasets—is explored. In machine learning, feature selection is an important phase that helps to boost model efficiency, decrease overfitting, and improve model accuracy. A class of optimization techniques known as evolutionary algorithms simulates natural selection to find the best possible combination of traits. An overview of evolutionary algorithms, their benefits, and how they might be used for feature selection and further how those features can be used in prediction system is shown. Also, it covers the several evolutionary algorithm subtypes and various kinds of classifiers that are frequently employed in feature selection, including genetic algorithms. The summary ends by outlining some of the difficulties and restrictions associated with employing evolutionary algorithms for feature selection.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

INTRODUCTION

A crucial phase in the machine learning process is feature selection, which identifies the most pertinent and instructive elements in a dataset. Data that is noisy, redundant, or irrelevant is cleaned up using FS. The performance has improved as a result, by eliminating characteristics that have a negative impact on the learning or classification process, feature selection seeks to identify a feature subset. A group of optimization algorithms known as evolutionary techniques derive their inspiration from evolutionary processes and natural selection [1]. The ability of evolutionary computation (EC) approaches to provide highly optimal solutions in a variety of problem contexts has made them a standard in computer science. These methods have been demonstrated to be effective in several domains, including machine learning.

The challenge of this project is therefore to develop a machine learning model for breast cancer prognosis that addresses these challenges and provides accurate, reliable and interpretable predictions [2]. Models include a variety of data types, oversee high dimensionality and noise, address class imbalance, and provide insight into the features which are most relevant for prediction. The goal is to develop models that help healthcare professionals make faster, more accurate, and more personalized decisions for breast cancer patients [6].

In this paper we have explored the use of evolutionary techniques for feature selection, and how they can be applied to perform feature selection in various machine learning tasks. We will also examine the strengths and limitations of evolutionary techniques for feature selection and provide examples and case studies to illustrate their use in real world applications such as early breast cancer prediction.

PROBLEM STATEMENT

Breast cancer is the leading cause of cancer-related death in women worldwide. Early detection and accurate diagnosis of disease are critical to improving survival and providing effective treatment. However, breast cancer diagnosis and prognosis are a challenging task that relies on a combination of medical imaging, clinical examination, and histopathological analysis. Machine learning has the potential to improve breast cancer diagnosis and prognosis by leveraging the large and complex data sets generated by medical imaging, genomics, and other diagnostic techniques [2]. However, developing accurate and reliable machine learning models for breast cancer diagnosis and prognosis is a complex task that presents several challenges. One of the main challenges is the lack of large and representative data sets containing various data types such as imaging data, clinical data, and genomic data. The availability of large and diverse datasets is critical for training and validating machine learning models that can generalize to new data and provide accurate predictions [6]. Another challenge is the high dimensionality and noise of the data. This can lead to overfitting and poor model performance. Feature selection and feature engineering techniques are essential to reduce the dimensionality of the data and identify the features that are most relevant for prediction.

Furthermore, the presence of class imbalance, where the number of cancer cases is much lower than the number of non-cancer cases, can make the development of accurate and reliable models difficult. Balancing the dataset or using a good sampling technique can overcome this challenge.

Finally, we need interpretable and transparent models that can provide insight into the features most relevant to prediction. Interpretable models help healthcare professionals understand how models make predictions and make informed decisions for their patients.

LITERATURE REVIEW

This section gives a review of literature work in the field of "Feature Selection Using Evolutionary Techniques" It is formulated as shown in Table 1.

Sr. No.	Study	Contribution	Technique Used	Strength	Weakness
1.	D. Pradip et al [1]	Data analysis using a combination of the ABC and DE algorithms, has focused on integrating exploratory capabilities with exploitative ones.	Artificial Bee Colony Optimization and Differential Evolution	Offers an additional advantage of being applicable to other binary optimization problems across various fields.	It can be computationally expensive, especially when dealing with large datasets or complex search spaces.
2.	R. Boggia et al [2]	This paper shows that GA can be a valuable approach in the problem of determining relevant variables.	Genetic Algorithm	It can be effectively applied to problems of regression, classification, and modelling, with the results produced in reasonable time.	Challenging to troubleshoot and may require a substantial number of computational resources. Moreover, these problems can be susceptible to the starting point.
3.	Rawal et al [3]	Aim of research categories in three categories. First is prediction of cancer before diagnosis, second is prediction of diagnosis and treatment and third domain focuses on outcome during treatment.	Logistic Regression, SVM, KNN.	The outcome through SVM was giving best outcome among other techniques used.	KNN was used which is a lazy learner hence can be time taking for large number of datasets.
4.	Mehdi Hosseinzadeh et al [4]	This paper evaluates the performance of ACO as a FS method for text categorization. It is found that ACO can quickly converge and effectively search the problem space to identify a minimal feature subset.	Ant Colony Optimization	The experimental results showed that ACO performed competitively compared to other methods.	This method frequently fails in identifying the most optimal feature subsets, as it is impossible to devise a perfect rule that can ensure achieving the best possible outcome.

5.	P. Ghamisi et al [5]	The algorithm presented in this paper is based on evolutionary methods, which are significantly faster than other widely used feature selection techniques that require a time	Hybrid of GA and PSO	Has an excellent classification accuracy and does in reasonable amount of time.	Requires additional programming and computational efforts to combine which results in increased complexity and has inability to guarantee global optimization.
6.	H. Asri et al [6]	Comparison of efficiency and effectiveness of those algorithms in terms of accuracy, precision, sensitivity, and specificity to find the best classification accuracy	C4.5, SVM, KNN.	Gives an accuracy of 97.13%	High computational cost
7	Venkatesh et al [7]	This paper drags the attention of the researchers to produce the best feature selection model that suits for any applications irrespective of the constraints.	Dimensionality Reduction and Hybrid algorithms	Easy to implement and get the desired results	Increased Computational cost, Increased risk of overfitting and Increased complexity due to high computational time.
8	S. Sharma et al [8]	The proposed model in this paper presents a comparative study of different machine learning algorithms, for the detection of breast cancer	Naïve Bayes, k fold cross validation.	KNN was most effective in this model whereas other techniques gave an accuracy of more than 94 %.	Unsupervised learning algorithms are avoided.

9	M. Ghosh et al [9]	This paper proposed the new idea for calculation of heuristic desirability in ACO. The deposit of pheromones on feature instead of path is an important part of the idea which can be extended by fixing an adaptive ordering of features.	Ant colony Optimization based on Wrapper Filter method.	Computationally efficient	Stagnation phase. Exploration rate is less. Convergence speed is less.
10	Emrah Hancer et al [10]	A fuzzy local search module was proposed by mimicking typical forward and backward selection strategies to improve the design of the evolutionary process.	Differential Evolution based Fuzzy Wrapper-Filter approach.	Provides with more accuracy and precision	It has High Computational cost and performs limited handling of noisy or redundant feature. Moreover, it is sensitive to parameter values.

Table 1. Related Works

PROPOSED METHODOLOGY

1. Datasets

The dataset taken here is of Breast Cancer, Wisconsin breast cancer (original) datasets from the UCI machine learning repository are used. This research. There are 569 breast cancer cases in Wisconsin. There are thirty major factors involved in the process of feature selection. Some of the major parameters in the datasets to judge are fractal dimension mean, area mean, compactness mean, symmetry mean, etc.

2. Classifiers

(a) XGBoost

XGBoost is a popular open-source gradient boosting framework that uses a tree-based ensemble learning method for both regression and classification tasks. It is designed to be scalable, efficient, and flexible, making it a powerful tool for many applications. XGBoost is based on the boosting concept, where weak models are iteratively combined into a strong model. It uses several techniques such as gradient boosting, smoothing and parallel processing to improve model accuracy and reduce overfitting. XGBoost has gained popularity in machine learning competitions due to its high performance and ability to process large and rich data sets.

(b) AdaBoost

Adaptive Boosting is an ML algorithm used for classification tasks. It works by combining multiple "weak" classifiers to form a "strong" classifier that can make accurate predictions based on new data. AdaBoost assigns a higher weight to misclassified samples in previous rounds, increasing the importance of those samples in subsequent rounds. This approach helps the model focus on hard-to-classify examples, improving overall performance. AdaBoost has been used in many applications, including object detection, face recognition and text classification. It is a popular algorithm due to its simplicity, efficiency, and ability to manage high-dimensional data.

(c) Logistic

Logistic regression is a supervised learning algorithm used for binary classification tasks that aim to predict whether a given input belongs to one of two classes. It models the probability of an input belonging to a given class as a function of its characteristics using a logistic function. Logistic regression is a linear model, but unlike linear regression, it uses a sigmoid activation function to transform the linear output into a probability value between 0 and 1 [4]. The decision boundary of a logistic classifier is usually a line or hyperplane that divides the input. space into regions corresponding to two classes.

(d) Linear SVM

Linear SVM (Support Vector Machine) is a supervised learning algorithm used for binary and multiclass classification tasks. It works by finding the optimal hyperplane that separates the input data into different classes. The best hyperplane is chosen to maximize the margin or distance between the hyperplane and the closest data point for each class. SVM can be used with both linear and nonlinear kernel functions to find the optimal separating hyperplane in high-dimensional feature space [5]. A linear SVM uses a linear kernel function to transform the input data into a high-dimensional space that can be more easily separated by hyperplanes.

(e) Random Forest

Random Forest is a supervised learning algorithm used for classification, regression, and other tasks. It works by building an ensemble of decision trees, each trained on a random subset of data and features. During training, the algorithm randomly selects a subset of features for each tree [7]. This reduces overfitting and improves generalization. The final prediction is made by combining the predictions of all individual trees in the ensemble. Random forests can manage both categorical and continuous data and can manage missing data without requiring imputation.

(f) KNN

K-Nearest Neighbors is a supervised learning algorithm used for classification and regression tasks. It works by finding the K closest data points to a given input and using their labels to predict the label of the input. KNN is a nonparametric algorithm. That is, it makes no assumptions about the underlying distribution of the data. KNN uses a distance metric like Euclidean distance or Manhattan distance to measure similarity between data points [7]. The value of K, which represents the number of neighbors to consider, is usually chosen by cross-validation.

(g) Gradient Boosting

Gradient boosting is a supervised learning algorithm used for classification and regression tasks. It works using a weak learner such as a decision tree that iteratively improves by minimizing the error of previous iterations. The algorithm starts by training a weak learner on the original data and computes the prediction error. In the next iteration, the algorithm trains a new weak learner with errors and combines the predictions of the previous and current models [9]. This process is repeated until a certain number of iterations is reached or the error is minimized to a certain level. Gradient boosting can be used with various loss functions such as: Mean squared error for regression and binary cross-entropy for classification.

(h) Radial SVM

Radial SVM is a supervised learning algorithm used for classification tasks. It is a variation of the linear SVM algorithm that uses kernel functions to transform the input data into a high-dimensional feature space, which can be more easily separated by nonlinear hyperplanes. The Radial SVM algorithm uses a Radial Basis Function (RBF) kernel that measures the similarity between two data points based on distance.

RBF kernels have a tunable parameter called the gamma parameter that controls the smoothness of the decision boundary [10]. Larger gamma values lead to more irregular decision boundaries, which can lead to overfitting. Smaller gamma values result in smoother decision boundaries, which can lead to underfitting.

(i) Decision Tree

These are supervised learning algorithms used for classification and regression tasks. It works by splitting the data into subsets based on the values of the input features and recursively splitting the data until the subsets are as pure as possible. Each split is based on criteria such as the Gini impurity and entropy, which measure the homogeneity of the subset. Decision trees can oversee both categorical and continuous data and can manage missing data. Decision trees are interpretable, easy to understand, and can be used to visualize the decision-making process. However, decision trees are prone to overfitting, especially if the tree is too deep or the data is noisy. Various regularization techniques such as pruning and ensemble techniques such as random forests and gradient boosting have been developed to address this problem.

3. Methods

The methodology to perform the proposed work is explained in this section. The pro to type is built by combining the part of Machine Learning algorithm and classifiers together. Components of this project are – Datasets, Algorithms, Python code, Code editor. The whole system will work in the following way:

Here is a method for using a hybrid of genetic algorithm and XGBoost for selecting the features:

Preprocessing: Clean and preprocess the data set by managing missing values, coding categorical variables, and scaling numerical functions as needed.

Feature Coding: Convert features and target variables into a binary format suitable for use in a genetic algorithm.

Genetic Algorithm: Use Genetic Algorithm to search for optimal functions. This involves selecting a subset of features from the original dataset and evaluating the fit of each subset based on the performance of the XGBoost model.

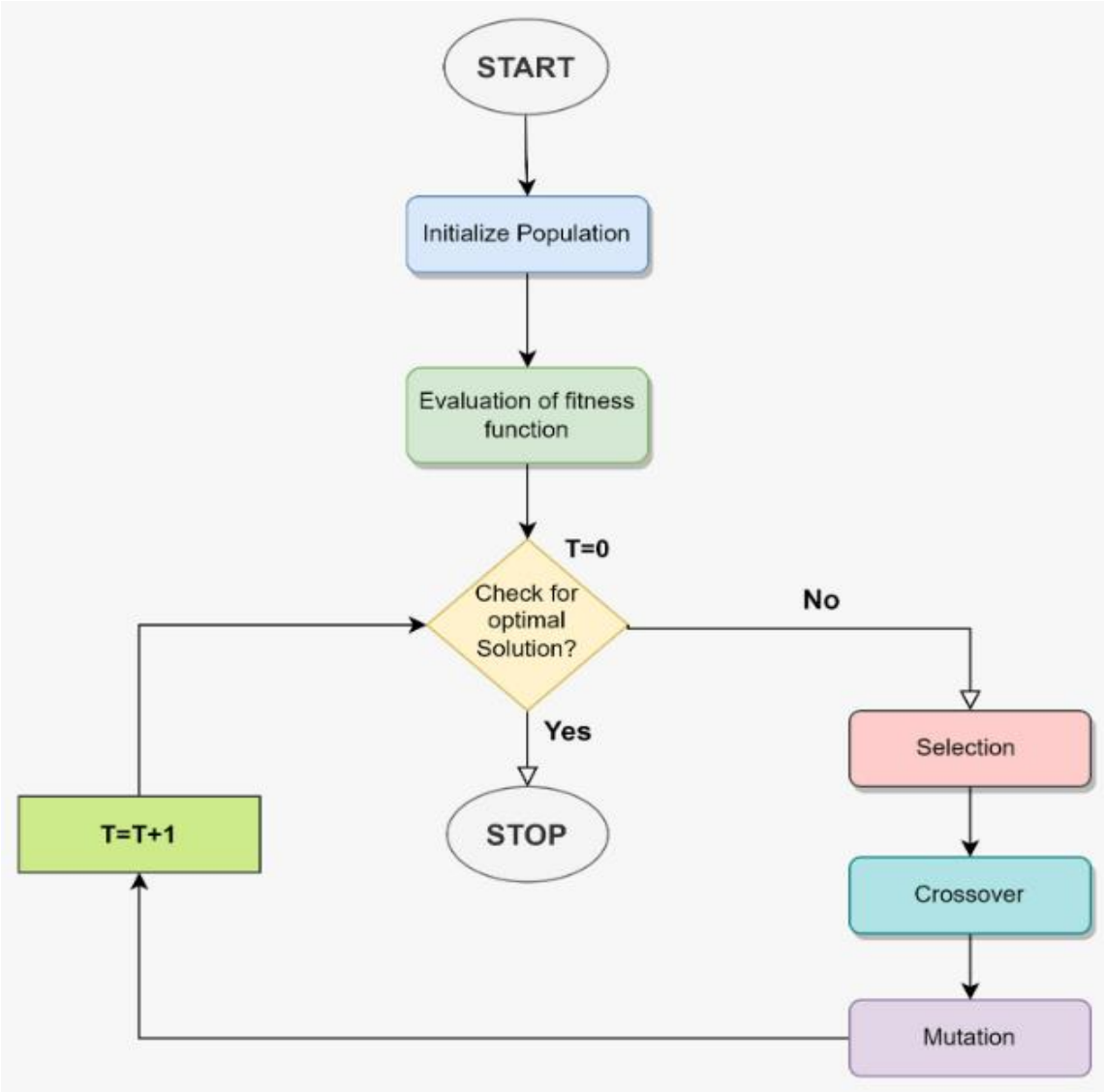

Figure 1. Block diagram of GA.

XGBoost Model: Train the XGBoost model using a subset of selected features and evaluate its performance against a set of retention tests. The performance of the model is used as an indicator of the suitability of the genetic algorithm.

Training function: Specify a fitness function that evaluates the performance of the XGBoost model using a subset of features. This could be accuracy, F1 score, or any other metric appropriate for a particular task.

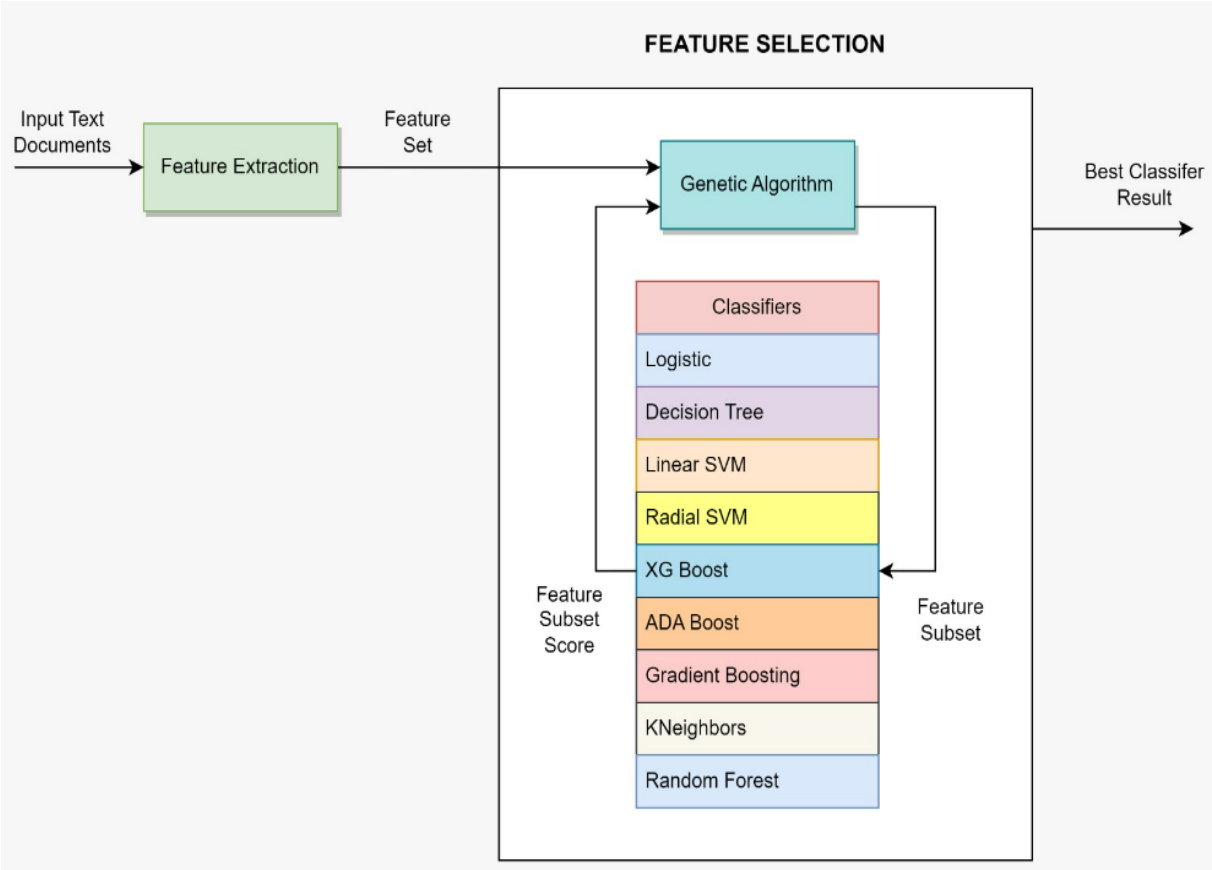

Figure 2. Block diagram of proposed feature selection method

Genetic Operators: Define genetic operators such as selection, crossover, and mutation to develop a subset of traits over generations. We have taken the mutation rate as 20 % and up to 10^{th} generation for training the model (figure 2).

Stopping Criteria: Set stopping criteria for the genetic algorithm, such as the maximum number of generations, a fitness threshold, or a fitness curve plateau (figure 1).

Performance Evaluation: Evaluate the performance of selected features and the XGBoost model on the Holdout test suite and report results including accuracy, precision, recall, F1 score and other related metrics.

Sensitivity analysis: Perform a sensitivity analysis to assess the robustness of the selected features and XGBoost model to changes in the genetic algorithm and XGBoost model parameters.

By iteratively selecting, evaluating, and evolving a feature subset, a genetic algorithm can efficiently explore the feature space and identify the best features for a given task (figure 2).

RESULTS AND DISCUSSIONS

In this section, we report the results of our data analysis. To apply and evaluate a classifier, we apply GA + XGBoost. After applying preprocessing and preparation methods, we try to analyze and understand the data visually and figure out the distribution of values in terms of effectiveness and efficiency.

1) Execution and Effectiveness.

Sr. No.	Classifier	Accuracy	Precision
1	XgBoost	0.982456	0.983871
2	AdaBoost	0.976608	0.968254
3	Logistic	0.970760	0.967742
4	Linear SVM	0.964912	0.967213
5	Random Forest	0.964912	0.967213
6	KNN	0.959064	0.982759
7	Gradient Boosting	0.959064	0.951613
8	Radial SVM	0.935673	1.000000
9	Decision Tree	0.923977	0.857143

Table 2.

We can observe that the accuracy obtained by using GA with all the other classifiers, the XGBoost predicts the highest accuracy of 98.24 % whereas the highest precision is given by Radial SVM. However, this cannot be considered as our results, for that we need to evaluate our model for desired number of generations. For less complexity, in this paper we chose to go until 10^{th} generation of the GA. After iterating through all the ten generation we obtained the following results as tabulated in Table 3.

Generation	Best Score
1	0.9883040935672515
2	0.9824561403508771
3	0.9883040935672515
4	0.9941520467836257
5	0.9883040935672515
6	0.9883040935672515
7	0.9883040935672515
8	1.0000000000000000
9	0.9941520467836257
10	0.9883040935672515

Table 3.

Here the mutation rate we had taken was 0.2 i.e., 20%. We must remember that if we take lower mutation rate, we must increase the mutation rate and if we choose to go with higher rate the number of iterations should be minimum. This is because at lower rate we observe less variations and vice versa. Below in Figure 3. is the comparison of the top three classifier's score.

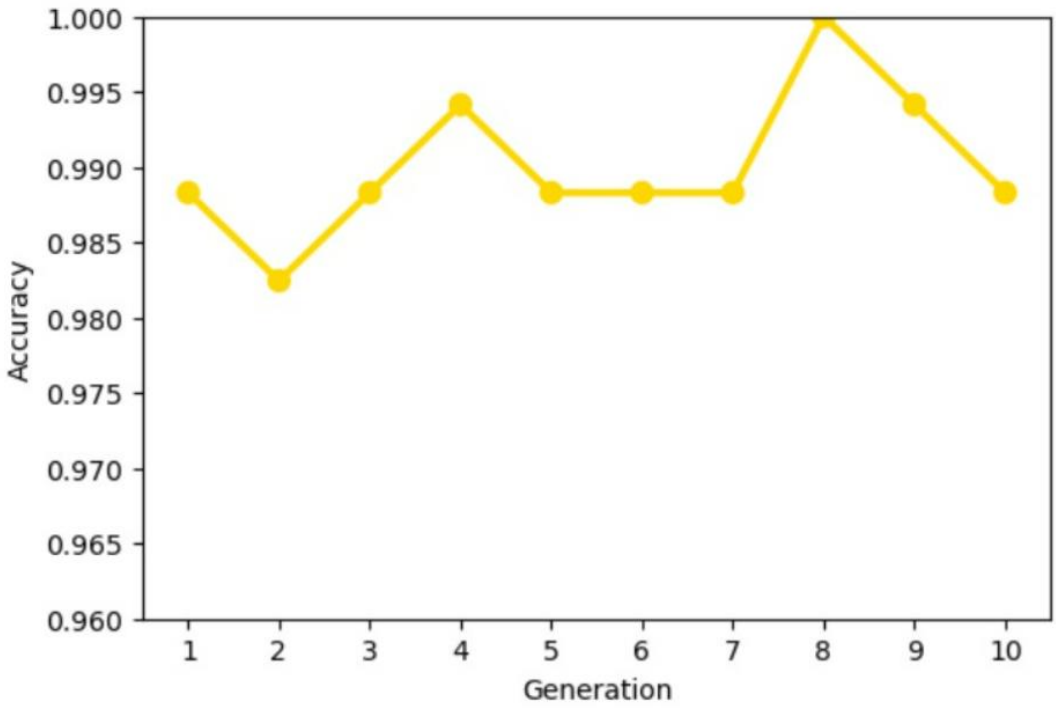

Figure 3. Accuracy v/s Generation Graph.

After obtaining accuracy in all the generation we visualize its graph as in Figure 3. It shows 100 % in eighth generation of GA with XGBoost. For all generations there is accuracy of more than 98 %.

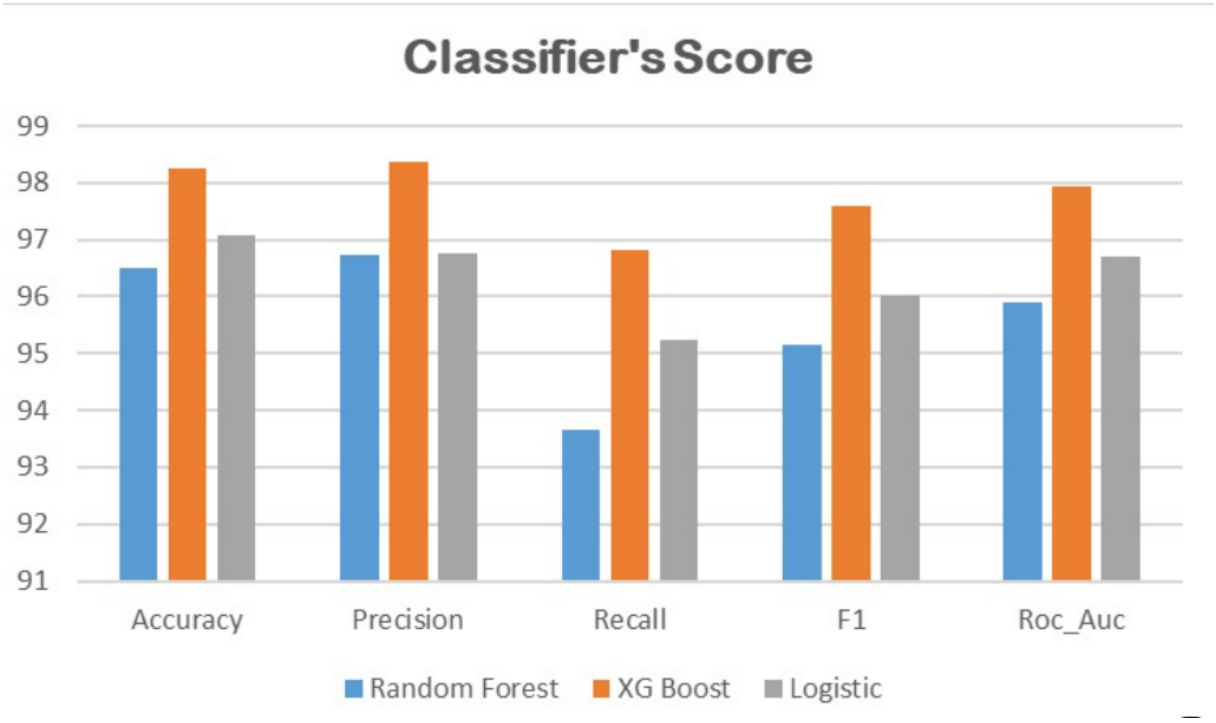

Figure 4. Classifier's Score.

The exact scores of all the classifiers are tabulated in Table 4. We get the highest recall value, F1 score and ROC for XGBoost.

Sr. No.	Classifier	Recall	F1 Score	ROC_AUC
1	XgBoost	0.968254	0.976000	0.979497
2	AdaBoost	0.968254	0.968254	0.974868
3	Logistic	0.952381	0.960000	0.966931
4	Linear SVM	0.936508	0.951613	0.958995
5	Random Forest	0.936508	0.951613	0.958995
6	KNN	0.904762	0.942149	0.947751
7	Gradient Boosting	0.936508	0.944000	0.954365
8	Radial SVM	0.825397	0.904348	0.912698
9	Decision Tree	0.952381	0.902256	0.929894

Table 4.

Below is the confusion matrix of our model (figure 5). These are identified in 4 categories:

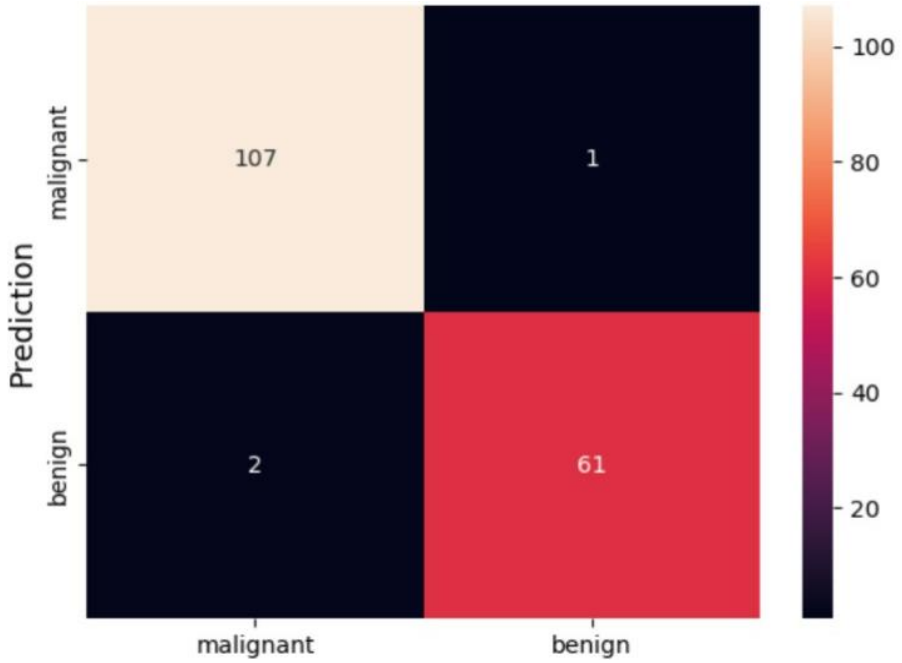

Figure 5. Confusion Matrix.

1) **True Positive (Malignant-Malignant) –** It represents the number of patients who have the disease and classified correctly.
2) **True Negative (Benign-Benign) –** It represents the number of patients who are perfectly healthy and classified correctly.
3) **False Positive (Benign-Malignant) –** It represents the misclassified patients who are healthy.
4) **False Negative (Malignant-Benign) –** It represents the misclassified healthy people who are actually having disease.

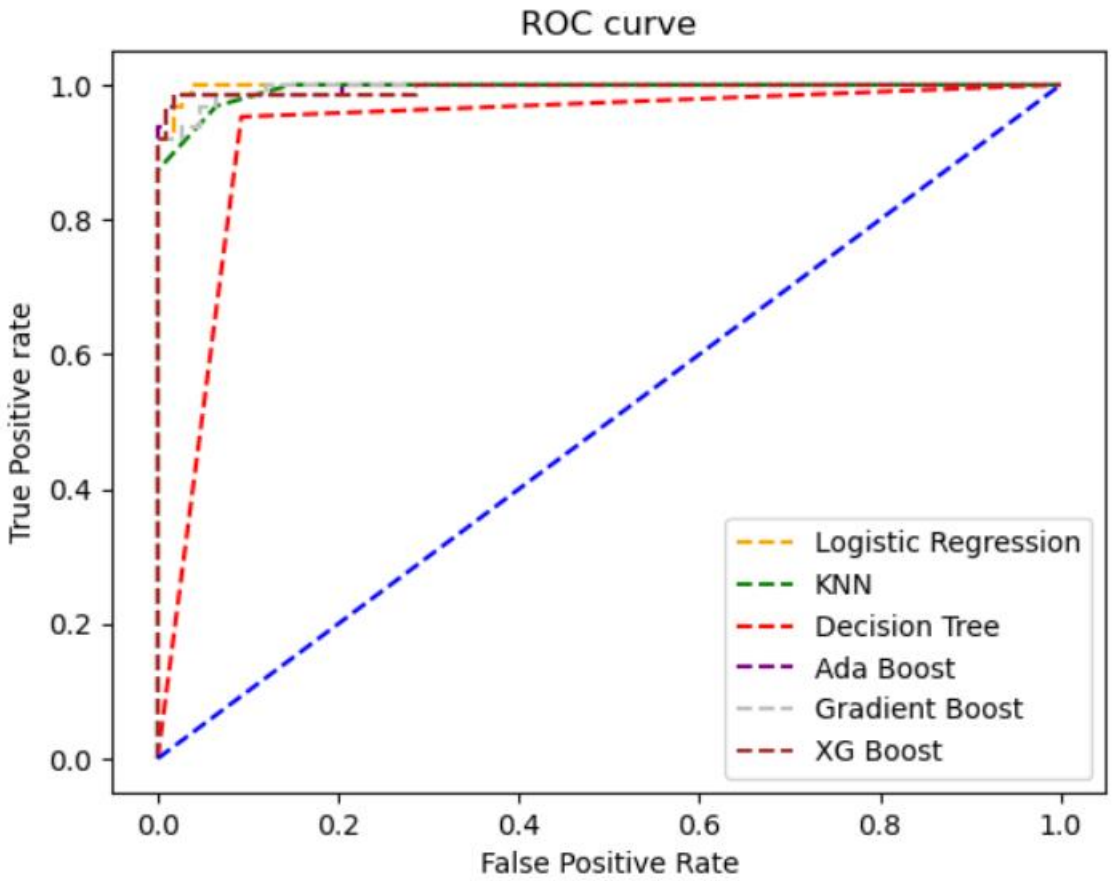

Figure 6. ROC Curve.

The Receiver Operating Characteristic Curve is a graph (figure 6) showing the performance of a classification model at all classification thresholds. This curve plots two parameters: True Positive Rate and False Positive Rate. The above curve shows how efficient our technique was to give the highest accuracy among all the related works (Table 1).

2) Comparison

We have used NumPy, pandas and Scikit-learn which are open-source machine learning libraries in Python. An open-source web application named as Jupyter Notebook is used to run the program. We were successfully able to implement the proposed algorithms which gives highest accuracy of 98.24 % (Table 2) in comparison to 97 % with KNN [2], 97.13 % with SVM [6] and 94 % with KNN and k cross validation technique [8]. The testing data we had taken was 20 %, because higher the number of factors and elements in the datasets lower should be the ratio of training and testing which helps in determining getting better results. Hence, the proposed model gives better accuracy than other models.

CONCLUSIONS

The most frequently occurring type of across cancer is breast cancer. There is a chance of twelve percent for a women picked randomly to be diagnosed with the disease [10]. Thus, early detection of breast cancer can save a lot of valuable life. The proposed model in this paper presents a comparative study of different machine learning algorithms and classifiers, for the detection of breast cancer. Performance comparison of the machine learning algorithms techniques has been conducted using the Wisconsin Diagnosis Breast Cancer data set. It has been observed that each of the algorithm had an accuracy of more than 98.2 %, to determine benign or malignant. From Table 2, it is found that GA with a combination of XGBoost is the most effective in detection of breast cancer as it had the best accuracy 99.41 %, precision and F1 score over the other algorithms. Thus, supervised machine learning techniques will be incredibly supportive in early diagnosis and prognosis of a cancer type in cancer research.

Declarations: Work done by,

Jitesh Ranjan Prusty (209205027):
1) Synopsis.
2) Literature Review (5 papers).
3) Paper Writing.
4) PPT.
5) Collection of Dataset.
6) Code Implementation.
7) Result.
8) Report.

Raj Chourasiya (209303370):
1) Synopsis.
2) Literature Review (5 papers).
3) Paper Writing.
4) PPT.
5) Collection of Dataset.
6) Code Implementation.
7) Result.
8) Report.

BIBLIOGRAPHY

1. Dhal, Pradip, and Chandrashekhar Azad. "A comprehensive survey on feature selection in the various fields of machine learning." Applied Intelligence (2022).
2. Leardi, Riccardo, Riccardo Boggia, and M. Terrile. "Genetic algorithms as a strategy for feature selection." Journal of chemometrics 6.5 (1992).
3. Rawal, Ramik. "Breast cancer prediction using machine learning." *Journal of Emerging Technologies and Innovative Research (JETIR)* 13.24 (2020):
4. Aghdam, Mehdi Hosseinzadeh, Nasser Ghasem-Aghaee, and Mohammad Ehsan Basiri. "Text feature selection using ant colony optimization." Expert systems with applications 36.3 (2009)
5. Ghamisi, Pedram, and Jon Atli Benediktsson. "Feature selection based on hybridization of genetic algorithm and particle swarm optimization." IEEE Geoscience and remote sensing letters 12.2 (2014)
6. Asri, Hiba, et al. "Using machine learning algorithms for breast cancer risk prediction and diagnosis." *Procedia Computer Science* 83 (2016): 1064-1069.
7. Venkatesh, B., and J. Anuradha. "A review of feature selection and its methods." Cybernetics and information technologies 19.1 (2019).
8. Sharma, Shubham, Archit Aggarwal, and Tanupriya Choudhury. "Breast cancer detection using machine learning algorithms." *2018 International conference on computational techniques, electronics, and mechanical systems (CTEMS)*. IEEE, 2018.
9. Ghosh, Manosij, et al. "A wrapper-filter feature selection technique based on ant colony optimization." Neural Computing and Applications 32 (2020).
10. Hancer, Emrah. "Differential evolution for feature selection: a fuzzy wrapper–filter approach." Soft Computing 23 (2019).