Using Genetic Algorithm for Breast Cancer Feature Selection

Abstract:

Breast cancer has emerged as a highly prevalent kind of cancer among women on a global scale. The timely detection of breast cancer is crucial for effective treatment, since failure to identify the disease in its early stages is associated with one of the highest fatality rates among various types of cancer. Various tools can be employed for the purpose of detection; however, computer-based diagnosis systems have gained popularity due to their cost-effectiveness and efficiency. This also results in inaccurate detections. Therefore, the process of feature selection plays a crucial role in improving the precision of computer-based algorithms. This study use genetic algorithms to perform feature selection in a wrapper methodology for the purpose of breast cancer diagnosis. The efficacy of the suggested model has been assessed by experimentation with a diverse set of 17 classifiers. The utilization of genetic algorithms for feature selection has resulted in a notable improvement in training accuracy. The Extra Trees, MLP, Random Forest, and Logistic models achieved the highest training accuracy, with a reported accuracy of 100%. On the other hand, the GaussianNB model had the lowest training accuracy, reported at 0.925. In addition, the utilization of feature selection resulted in enhancements in many performance metrics, including validation accuracy, sensitivity, specificity, F1-score, Matthews Correlation Coefficient, specificity, and sensitivity.

1. Introduction:

The tremendous progress in technology has facilitated the expansion and accessibility of datasets. Datasets typically contain a large number of features, necessitating the utilization of sophisticated pattern identification techniques by researchers to extract meaningful information from these extensive samples. Datasets are generated across various disciplines, ranging from biology to astrophysics, and may consist of a multitude of attributes, potentially numbering in the thousands. However, it is important to note that certain features inside machine learning and deep learning models may be considered redundant or irrelevant, hence not contributing significantly to the overall performance. Therefore, given the multitude of characteristics

included in datasets, it is imperative to employ techniques that effectively reduce dimensionality and extract salient features [1-3]. The efficiency of the classification model can be significantly enhanced with the implementation of feature reduction techniques. The careful consideration of features is of utmost importance, while ensuring that the unique properties of the dataset are preserved. Selections of features are crucial, but the characteristics of the dataset should not be lost. Hence, it is important to classify the features as weak and strong [4]. Data mining has been employed within the medical field to establish correlations, as the abundance of data might provide challenges for medical professionals when making diagnoses.The user's text is too brief to be rewritten in an academic manner. The application of database analysis is particularly prevalent in the realm of automated diagnostic systems [5].

Breast cancer is the most prevalent invasive cancer among women [6]. Around 2.3 million women worldwide are diagnosed with breast cancer every year.[7] Most of the patients that have brain cancer are over fifty years old [7]. Early detection of breast cancer has been proven to be very effective in reducing the mortality rate of patients. However, the survival rate depends on many factors, especially on both stage and molecular subtypes [7].

Various conventional techniques are employed for the identification of breast cancer, including biopsy, physical examination, breast ultrasound, diagnostic mammography, and breast magnetic resonance imaging (MRI). After the detection of breast cancer, supplementary examinations are performed to ascertain the presence of cancerous cells spreading either within the breast or to other regions of the body. The aforementioned procedure is commonly referred to as staging. The classification of breast cancer stages is contingent upon its localization within the breast, involvement of the axillary lymph nodes, and potential spread beyond the confines of the breast tissue. The therapy required for an individual diagnosed with breast cancer can be determined by medical professionals based on the specific type and stage of the disease [8].

Mammography is one of the most common methods for breast cancer detection, which is utilized by radiologists. However, radiologists may interpret the results differently or inaccurately, thus the accuracy rate of mammography fluctuates between 68% and 79% [9, 10]. Another way is biopsy, which can be expensive, risky, invasive, but accurate. These detection techniques can categorize patients into a 'benign' group without breast cancer or a 'malignant' group showing substantial signs of the disease [10]. Also, it is essential to note that benign tumors are safer than malignant tumors in many cases. Computer aided system may help doctors

to understand the differences between these two categories. As mentioned before feature selection, a preprocessing technique may help doctors in breast cancer detection/classification.

Three approaches to feature selection are Filter, Wrapper, and Embedded [11, 12]. The filter approach scores the selected subset based on the intrinsic properties of the data, without considering the classifier algorithm [13]. The wrapper method finds the best set of features for a specific algorithm and area [14]. In other words, the chosen set of features is determined by training and assessing a classifier using only the variables within the suggested group. During the model-building process in the embedded approach, the optimal subset of features is selected [11].

Numerous studies in the literature have utilized a range of feature selection techniques on breast cancer datasets. These techniques encompass the ant colony algorithm, discrete particle swarm optimization, the wrapper strategy combined with a genetic algorithm, feature selection rooted in support vectors, incorporating fisher's linear discriminate and support vector machine, the rapid correlation-based feature selection (FCBF), its multi-threaded version, the decision-dependent and -independent correlation (DDC- DIC), the Rough set K-Means Clustering method, and the adjusted correlation rough set feature selection approach (MCRSFS) [11, 15-21].

In our study, we adopted a wrapper feature selection method derived from a genetic algorithm. By employing 17 different classifiers, we examined the impact of the genetic algorithm on the accuracy of these classifiers, using breast cancer dataset.

2. Dataset

The dataset was taken from UC Irvine machine learning repository [22]. Characteristics of cell nuclei are derived from a digitized image taken from a fine needle aspirate (FNA) of a breast mass. The data have been collected by Dr. William H. Wolberg between 1989 and 1991. From this dataset, 569 patients' data were used along with 31 different features. There are 357 benign, 212 malignant patients.

Table 1: Possible Features for Feature Selection

Features		
Diagnosis	Fractional Dimension Mean	Radius Worst

Radius Mean	Radius Standard Error	Texture Worst
Texture Mean	Texture Standard Error	Perimeter Worst
Perimeter Mean	Perimeter Standard Error	Area Worst
Area Mean	Area Standard Error	Smoothness Worst
Smoothness Mean	Smoothness Standard Error	Compactness Worst
Compactness Mean	Compactness Standard Error	Concavity Worst
Concavity Mean	Concavity Standard Error	Concave Points Worst
Concave Points Mean	Concave Points Standard Error	Symmetry Worst
Symmetry Mean	Fractional Dimension Standard Error	Fractional Dimension Worst

Ten real-valued features are computed for each cell nucleus as it is seen in table 1.

3. Classifiers

In this study, 21 different machine learning classifiers have been used in this study to compare the effectiveness of genetic algorithms.

3.1.Extra Trees Classifier

There are many tree based algorithms and models exist, however, it is different as instead of using a bootstrap replica, it grows the trees using the entire learning sample and selects cut-points for nodes entirely at random[23]. For each decision tree, a random subset of attributes from the dataset is chosen. The dataset is then divided based on random divisions within those attributes, with the optimal split being selected [24]. The Extra-Trees classifier makes a set of decision trees using the usual top-down way [25]. One of the strengths of this algorithm is computational efficiency [23].

3.2.Adaboost Classifier

Adaboost as from its name, it is a boosting machine learning algorithm. It combines multiple weak learning models along with a weighted linear combination [25]. AdaBoost applies a learning method step by step to adjusted versions of the initial training data

[26].Adaboostoperates iteratively, and when there are misclassified instances, more weight is given other iterations. Weights of incorrectly classified instances are raised/increased but correctly classified instances are diminished. The algorithm consistently uses the base classifier on the training data, altering weights in every cycle [25]. The final model is a linear combination of the models obtained from various cycles.

3.3. Random Forest Classifier

Random Forest classifier is very similar to extra trees classifier. The algorithm forms a group of decision trees in order to improve the accuracy of the decision trees. Also, this classifier uses random selection of features and bagging sample method [25, 27-30]. Using bagging, every decision tree in the ensemble is formed from a resembled version of the training data. Each tree in the ensemble serves as a base estimator to establish the class label for an unlabeled sample, with the final decision made based on the majority of votes/average [25].

3.4. Bagging Classifier

Bagging classifier is one of the meta-estimators, creating models by fitting each base classifier on an arbitrary subsample of the dataset. Afterward, it gathers the outcomes from all the models to make the final decision. The bagging classifiers use two different methods: highest average likelihood from the base classifiers and majority voting, which rules the suspicious nodes in the network to establish the predicted label [25].

3.5.Gaussian Naive Bayes Classifier

Naive Bayes technique is largely applied in machine learning models as it has a computational efficiency. This algorithm has a low variance value and a high cost of bias. It uses incremental learning, which means estimations can be updated. It operates on the premise that each individual parameter independently influences the outcome variable. It uses probabilities so thoughtless to noise [31].

3.6.Category Boosting Classifier

Category Boosting is a machine learning algorithm, which is designed to handle categorical features. It is one of the gradient boosting frameworks, which build an ensemble of decision trees in a sequential manner. Moreover, for reducing over fitting, it has an algorithm to encode categorical features and uses L2 regularization.

3.7.LightGBM Classifier

Light Gradient Boosting (LightGBM) is a gradient boosting framework, using tree-based learning algorithms [32, 33]. It is effective in training large datasets and with high-dimensional features. Gradient-based One-Side Sampling helps to retain data instances with large gradients. Gradient-based One-Side Sampling helps to retain data instances with large gradients and randomly samples a small portion of data instances with small gradients, reducing the data used in each iteration with minimal loss in accuracy. Moreover, it can achieve faster training times compared to other gradient boosting algorithms [32, 33].

3.8. Quadratic Discriminant Analysis(GDA)

Quadratic Discriminant Analysis (QDA) is a classification method used in both statistics, probabilistic learning, and machine learning. QDA is a generalization of Linear Discriminant Analysis in order to handle where each class has its own covariance matrix, rather than assuming a common covariance matrix for all classes [34, 35]. GDA is used for classifying data into multiple classes based on the maximization of the posterior class probability. For each class, it computes the likelihood of a data point belonging to that class based on its Gaussian distribution [34, 35]. Afterwards, it multiplies this likelihood by the prior probability of that class. The class with the highest posterior probability is the predicted class for the data point. The decision boundary is derived by setting the posterior probabilities of two classes to be equal.

3.9. Support Vector Machine Classifier

The Support Vector Machine (SVM) is a small-sample learning method. Also, it is a supervised machine learning algorithm that is used for classification tasks and regression [36]. It finds the hyper plane, decision boundaries, that best divides a dataset into classes. Thus, it is mostly accurate in separable and non-separable problems [37]. However, when the number of input features is more than 3, it has more than two-dimensional planes.

3.10. Linear Regression

Linear regression is a type of supervised machine-learning algorithm. When there is only one independent feature, it is termed Univariate Linear Regression. However, if there are multiple features, it is referred to as Multivariate Linear Regression. The algorithm aims to identify the optimal linear equation in order to estimate the value of the dependent variable using the independent variables.

3.11. Gradient Boosting Classifier

The Gradient Boosting Classifier is a popular machine learning algorithm used for classification and regression. It is an ensemble learning method. It converts weak learners into strong learners and it was built in a stage-wise fashion. It uses the decision trees as base learners. The algorithm is effective for classifying complex datasets. It is based on probability approximately correct learning. The objective is to minimize the loss between the actual and predicted class values

3.12. K-nearest neighbors Classifier

K-nearest neighbors is one of the most common classifiers that is used in machine learning. It is very simple, however, an effective algorithm for both classification and regression. However, it just memorizes the training set and uses it directly during the test set. Also, it is non-parametric, which means it does not make any assumptions about the underlying data distribution. There is a distance metric, where it measures the distance between data points, using Euclidean distance. After this phase, the number of nearest neighbors to consider when makes a classification decision. It classifies an unknown point based on the majority class among its k nearest neighbors.

3.13. Logistic Classifier

Logistic classifier is an algorithm that is used in machine learning and statistics. It is used for predicting a categorical outcome variable and the outcome comes in the form of a binary outcome variable. It uses the sigmoid function in order to squeeze a linear equation between 0 and 1. Logistic regression creates a linear decision boundary in the feature space.

3.14. Passive Aggressive

The Passive-Aggressive Classifier ıs used for large-scale learning, text classification tasks, and multiple classes. It updates the model incrementally. it can quickly adapt to new data, and it does not need to store the dataset, in other words, it is memory-efficient. It uses weight vector, which gives an idea of the importance of different features and it creates a linear model.

3.15. Multinomial Naive Bayes

Multinomial Naive Bayes algorithm is built on top of Naive Bayes and is useful for classification. It counts each occurrence of each class in the training data and calculates the probability of each class and the possible state of each given class. For additional data points, the algorithm computes each class, calculates the posterior probability and assigns the class the

highest probability. Furthermore, it does not require high memory and is still not suitable for continuous features.

3.16. Decision Tree

Decision Tree is a hierarchical decision-support framework that uses a tree-structured representation of choices and their potential outcomes. It is a supervised machine learning algorithm that can be used for regression and classification problems. One of the most interpretable machine learning algorithms, but it has a possibility to over fit, especially when the tree is deep. It uses a tree-like model for decision-making and splits the dataset into two or more homogeneous sets based on the most significant attribute at each level.

3.17. Multilayer perceptron

Multilayer perceptron is a multi-layer neural network, which has three hidden layers/neurons. Neurons utilize a nonlinear activation function and are uniquely capable of classifying data that isn't linearly separable.

3.18. Stochastic Gradient Descent

Stochastic Gradient Descent is an optimization technique frequently employed in machine learning to identify the model parameters that yield the closest match between predicted and observed outcomes. For classification tasks, the algorithm utilizes a straightforward Stochastic Gradient Descent (SGD) learning process, which accommodates multiple loss functions and penalties.

4. Genetic Algorithm

Originating in the 1960s and 1970s by John Holland and his team, the genetic algorithm (GA) serves as an abstraction of biological evolution, rooted in Charles Darwin's theory of natural selection. It is likely that Holland pioneered the application of crossover and recombination, mutation, and selection in the exploration of adaptive and artificial systems, which will be discussed in this research paper [38]. As a problem-solving strategy, these genetic operators are pivotal to the genetic algorithm. In the time since, numerous genetic algorithm variants have emerged, addressing a broad spectrum of optimization challenges, ranging from graph coloring and pattern recognition to both discrete and continuous systems [38].

In the evolutionary algorithms, genetic algorithms stand out due to the vast range of their applications. Using an iterative process, a Genetic Algorithm is employed for Search and Optimization to determine the best solution among multiple options. A Genetic Algorithm is

essential in identifying the optimal hyper parameters and their values to enhance a deep learning model's performance and can be used to find the most suitable number of features when constructing machine learning models.

Some of the important terminology for genetic algorithms is population, phenotype, chromosome, and fitness score. Below you may see the general structure of the genetic algorithms.

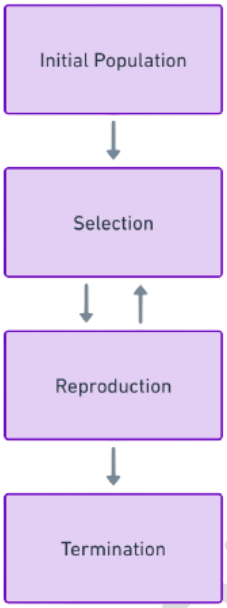


Figure 1: General structure of genetic algorithms

4.1. Initial Population

The Genetic Algorithm Process begins with Population Initialization. Within the current generation, the population represents a subset of solutions. Each individual possesses a gene sequence, often referred to as DNA. An individual's DNA signifies a potential solution to the targeted problem, and it must be structured appropriately. Hence, it is essential to initialize each individual to guarantee they possess some form of DNA. In the context of genetic algorithms, it is crucial to preserve the population's diversity to avoid an issue called premature convergence. This term in evolutionary algorithms refers to the algorithm settling before achieving the best possible solution. There are two ways of population initialization. First one is the random initialization, which initializes the population with completely random gene values. When gene values are randomly assigned, possible genetic-diversity increases within the population. Second method is heuristic initialization, which uses heuristics for solving in a complex issue.

4.2. Selection

Selection process is very essential in genetic algorithms. Each individual has the fitness value of their corresponding DNA. The fitness value of an individual indicates its optimality, showing how close it is to the best solution compared to others. When the fitness function doesn't produce superior fitness values, the genetic algorithm might struggle to generate top-notch solutions. Once a proper fitness function is established, the fitness of each individual is determined. The population is then organized based on these fitness levels, and a portion of those with the lowest fitness is removed. However, a few with lower fitness remain to maintain genetic variety within the group [39].

4.3. Reproduction: Crossover and Mutation

After the selection process, reproduction takes place. Reproduction happens through cross-over and mutation. Cross-over which is simply the mating. Crossing over occurs when genes from the two most fit parents are mixed randomly to create a new solution or genotype. Depending on the segments of genes swapped from the parents, this can be a one-point or multi-Point crossover. The primary goal of crossover is to produce new descendants from individuals with high fitness, thereby enhancing the overall fitness of the population. Once a new population emerges from selection and crossover, it undergoes random alterations via mutation. Mutation serves as a random method to modify a genotype, fostering diversity within the population and aiding in the discovery of enhanced and more efficient solutions.The algorithm's search space broadens when enough new genes are introduced by randomly modifying the genes of the next generation's offspring.

4.4. Termination

This part is the last step of the genetic algorithm. When the genes of the next generation's offspring are randomly modified, the algorithm's search space expands due to the introduction of sufficient new genes. If the termination conditions are met then the evolutionary algorithm can be terminated and output can be seen.

5. Methodology

The hardware used for this experiment has a 2,4 GHz Quad-Core Intel Core i5, 8 GB RAM 2133 MHz LPDDR3, and Tesla P100-PCIE GPU. Experiment has been performed using Python code. The breast cancer dataset was independently fed into 17 different classifiers.The proposed model (Figure 2) has been applied to the Wisconsin breast cancer dataset.

Flowchart of the working genetic algorithm and proposed model that is used in this study can be seen below:

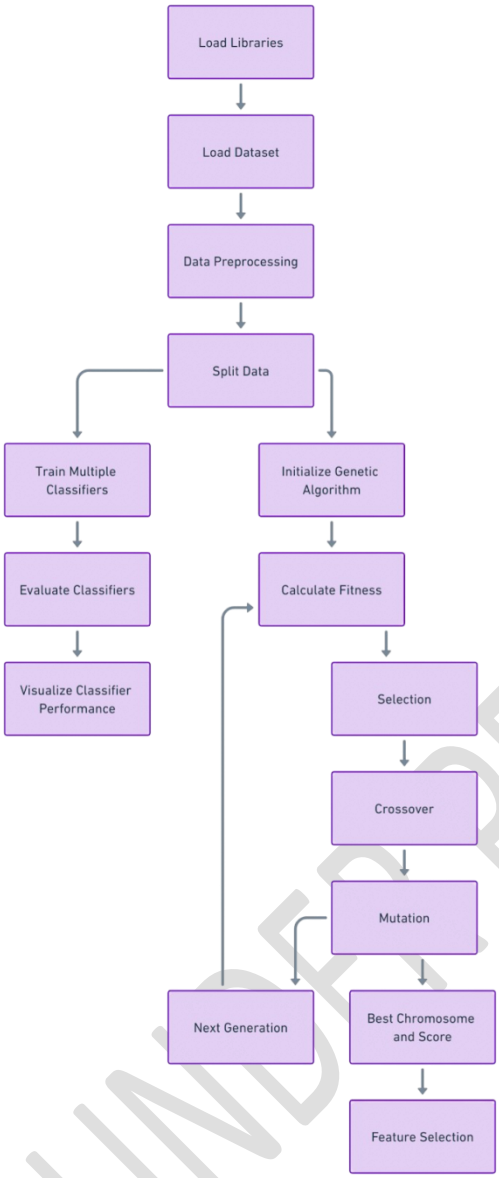


Figure 2: Flowchart and proposed model for this study

In the proposed model, there are 80 chromosomes in each population. Mutation rate has been set to 0.15. There is a single-point crossover at the midpoint of the chromosome. The number of generations has been set to 10.

Each feature selection is different for different classifiers and algorithms. For instance, for AdaBoost Classifier, selected features are radius mean, area mean, smoothness mean,

concavity mean, texture standard error, area standard error, smoothness standard error, compactness standard error, fractal dimension standard error, texture worst, perimeter worst, area worst, smoothness worst, compactness worst, concavity worst, concave points worst. However, for decision tree classifier, the selected features are radius mean, texture mean, perimeter mean, smoothness mean, compactness mean, concave points mean, fractal dimension mean, radius standard error, texture standard error, smoothness standard error, concave points standard error, symmetry standard error, fractal dimension standard error, radius worst, texture worst, smoothness worst, compactness worst, concavity worst, symmetry worst, fractal dimension worst. Thus, genetic algorithm have helped to make predictions by reducing the features in the dataset.

The dataset has been splitted to 50/50, 60/40, 70/30, 80/20, 90/10 for the training and test set, but the ratio of 80/20 has been kept because it has led to the highest accuracy among other ratios.

5.1. Performance Metrics

"Performance metrics of the model were calculated to ascertain the reliability of the study."40 Some of the metrics that have been used in this study can be seen below:Sensitivity (Sens), Specificity (Spec), F1-score (F1), Matthews Correlation Coefficient (MCC)

Accuracy=TP+TNTP+TN+FP+FN

Sensitivity=TPTP+FN

Precision=TPTP+FP

F1-score=2TP(2TP+FP+FN)

Specificity=TNFP+TN

Matthews Correlation Coefficient =(TPxTN)-(FPxFN)(TP+FP)(TP+FN)(TN+FP)(TN+FN)

6. Results

Before employing a genetic algorithm for feature selection, the accuracy of various classifiers and algorithms can be observed below.

Table 2: Accuracies of Different Classifiers

Classifier	Training Accuracy
Extra Trees	0.973684

AdaBoost	0.973684
GaussianNB	0.973684
MLP	0.973684
LGBM	0.973684
CatBoost	0.964912
Random Forest	0.964912
SGD	0.964912
Bagging	0.964912
QDA	0.956140
Gradient Boosting	0.956140
KNeighbors	0.956140
Logistic	0.956140
RadialSVM	0.947368

Passive Aggressive	0.947368
PolySVM	0.947368
MultinomialNB	0.938596
Decision Tree	0.938596

After employing genetic algorithm for feature selection, there was an increase in accuracy for most of the classifiers although some of the classifiers did not have any improvement. Below table shows the extracted features using genetic algorithm

Table 3: Classifiers and Extracted Features

Classifier	Extracted Features
ExtraTrees	radius mean, area mean, smoothness mean, concavity mean, texture standard error, area standard error, smoothness standard error, compactness standard error, fractal dimension standard error, texture worst, perimeter worst, area worst, smoothness worst, compactness worst, concavity worst, concave points worst
AdaBoost	radius mean, texture mean, perimeter mean, area mean, smoothness mean, concavity mean, texture standard error, area standard error, compactness standard error, concave points standard error, symmetry standard error, fractal dimension standard error, radius worst, texture worst, smoothness worst, compactness worst, symmetry worst

GaussianNB	radius mean, area mean, compactness mean, concavity mean, concave points mean, symmetry mean, perimeter standard error, concavity standard error, concave points standard error, texture worst, perimeter worst, concave points worst
MLP	radius mean, perimeter mean, compactness mean, concave points mean, symmetry mean, radius standard error, area standard error, compactness standard error, symmetry standard error, fractal dimension standard error, radius worst, texture worst, concavity worst
LGBM	texture mean, area mean, smoothness mean, concave points mean, symmetry mean, fractal dimension mean, area standard error, smoothness standard error, compactness standard error, concave points standard error, fractal dimension standard error, texture worst, smoothness worst, concavity worst
CatBoost	area mean, smoothness mean, compactness mean, concavity mean, radius standard error, texture standard error, area standard error, smoothness standard error, compactness standard error, fractal dimension standard error, perimeter worst, area worst, smoothness worst
RandomForest	area mean, smoothness mean, compactness mean, concavity mean, radius standard error, texture standard error, area standard error, smoothness standard error, compactness standard error, fractal dimension standard error, perimeter worst, area worst, smoothness

	worst
SGD	texture mean, area mean, smoothness mean, symmetry mean, texture standard error, perimeter standard error, compactness standard error, concavity standard error, fractal dimension standard error, texture worst, perimeter worst, area worst, concavity worst, fractal dimension worst
Bagging	area mean, smoothness mean, compactness mean, concavity mean, radius standard error, texture standard error, area standard error, smoothness standard error, compactness standard error, fractal dimension standard error, perimeter worst, area worst, smoothness worst
QDA	texture mean, area mean, smoothness mean, symmetry mean, texture standard error, perimeter standard error, compactness standard error, concavity standard error, fractal dimension standard error, texture worst, perimeter worst, area worst, concavity worst, fractal dimension worst
GradientBoosting	radius mean, texture mean, smoothness mean, concavity mean, symmetry mean, fractal dimension mean, perimeter standard error, smoothness standard error, concavity standard error, fractal dimension standard error, perimeter worst, compactness worst, concavity worst, symmetry worst, fractal dimension worst

KNeighbors	texture mean, perimeter mean, smoothness mean, concavity mean, concave points mean, symmetry mean, area standard error, smoothness standard error, concave points standard error, symmetry standard error, fractal dimension standard error, perimeter worst, smoothness worst, concavity worst, concave points worst, fractal dimension worst
Logistic	texture mean, perimeter mean, smoothness mean, concavity mean, concave points mean, symmetry mean, area standard error, smoothness standard error, concave points standard error, symmetry standard error, fractal dimension standard error, perimeter worst, smoothness worst, concavity worst, concave points worst, fractal dimension worst
RadialSVM	radius mean, texture mean, smoothness mean, concavity mean, area standard error, smoothness standard error, compactness standard error, concavity standard error, radius worst, texture worst, smoothness worst, concavity worst, concave points worst, symmetry worst, fractal dimension worst
PolySVM	radius mean, perimeter mean, area mean, concavity mean, concave points mean, fractal dimension mean, perimeter standard error, area standard error, concavity standard error, concave points standard error, symmetry standard error, texture worst, perimeter worst, smoothness worst, concavity worst, fractal dimension worst
MultinomialNB	perimeter mean, area mean, smoothness mean, compactness mean, concavity mean, concave points mean, fractal dimension mean, perimeter standard error, area standard error, fractal dimension standard error, radius worst, texture worst, perimeter worst, smoothness worst, concavity worst, symmetry worst

DecisionTree	radius mean, texture mean, perimeter mean, smoothness mean, compactness mean, concave points mean, fractal dimension mean, radius standard error, texture standard error, smoothness standard error, concave points standard error, symmetry standard error, fractal dimension standard error, radius worst, texture worst, smoothness worst, compactness worst, concavity worst, symmetry worst, fractal dimension worst

Table 4 shows the true positive (TP), True Negative (TN), False Positive (FP), Training Accuracy (TA), Validation Accuracy (VA), Sensitivity (Sens), Specificity (Spec), F1-score (F1), Matthews Correlation Coefficient (MCC), and Time of each classifier, more information can be found in the appendix section.

Table 4: Classifiers and Performance Evaluation

Classifiers	TP	FP	TN	FP	TA	VA	Sens	Spec	Prec	NPV	FPR	FDR	FNR	F1	MCC	Time
Extra Trees	65	2	44	3	1.000	0.9561	0.9559	0.9565	0.9701	0.9362	0.0435	0.0299	0.0441	0.9630	0.9094	14.597
AdaBoost	69	2	41	2	0.9912	0.9649	0.9718	0.9535	0.9718	0.9535	0.0465	0.0282	0.0282	0.9718	0.9253	197.39
Gaussian NB	71	0	39	4	0.925	0.9649	0.9467	1.000	1.000	0.9070	0	0	0.0533	0.9726	0.9266	7.82
MLP	70	1	44	39	1.000	0.9561	0.9459	0.9750	0.9859	0.9070	0.0250	0.0141	0.0541	0.9655	0.9068	1298.99
LGBM	69	2	41	2	0.9912	0.9649	0.9718	95.35	97.18	95.35	0.0465	0.0282	0.0282	0.9718	0.9253	224.51
CatBoos	7	0	4	2	0.9	0.9	0.9	1.0	1.0	0.9	0	0	0.0	0.9	0.9	864.

t	1		1		912	825	726	000	000	535			274	861	630	42
Random Forest	71	0	42	1	1.000	0.912	0.861	1.000	1.000	0.767	0	0	0.0139	99.30	98.14	608.55
SGD	71	0	35	8	0.737	92.98	0.987	1.000	1.000	0.140	0	0	0.1013	0.9467	0.8553	12.73
Decision Tree	70	1	39	4	0.737	0.649	0.467	1.000	1.000	0.070	0	0	0.0533	0.9726	0.9266	23.06
QDA	67	4	41	2	0.825	0.391	0.571	0.111	0.437	0.318	0.889	0.563	0.0429	0.9504	0.8719	10.75
Gradient Boosting	71	0	41	2	0.912	0.825	0.726	1.000	1.000	0.535	0	0	0.0274	0.9861	0.9630	452.80
KNeighbors	71	0	40	3	0.825	0.737	95.95	1.000	1.000	0.302	0	0	0.0405	0.9793	0.9447	28.50
Logistic	70	1	41	2	1.000	0.737	97.22	0.762	0.859	0.535	0.238	0.141	0.0278	0.9790	0.9439	277.44
Radial SVM	71	0	39	4	0.649	0.649	0.467	1.000	1.000	0.070	0	0	0.0533	0.9726	0.9266	48.33
Poly SVM	71	0	39	4	0.649	0.649	0.467	1.000	1.000	0.070	0	0	0.0533	0.9726	0.9266	31.30
MultinomialNB	71	0	37	6	0.561	0.474	0.221	1.000	1.000	0.605	0	0	0.0779	0.9595	0.8907	9.40

Bagging	70	1	40	3	0.9825	0.9649	0.9589	0.9756	0.9859	0.9302	0.0244	0.0141	0.0411	0.9722	0.9253	97.31

After feature selection, there has been an increase in training accuracy and also performance metrics. All the models had a training accuracy higher than 0.9500, however without feature selection, nine of the classifiers fell below this threshold. The time required for training varied widely, ranging from 7.82 seconds for GaussianNB to 1298.99 seconds for MLP. Below you may find the summary of the table and maximum/minimum values for each performance metrics.

Table 5: Maximum/Minimum values for each performance metrics.

Metric	**Maximum Values**	**Minimum Values**
TP (True Positive)	71 (Multiple Classifiers)	65 (Extra Trees)
FP (False Positive)	4 (QDA)	0 (Multiple Classifiers)
TN (True Negative)	42 (Random Forest)	4 (MLP)
TA (Training Accuracy)	1.0000 (Extra Trees, MLP, Random Forest, Logistic)	0.925 (GaussianNB)

VA (Validation Accuracy)	0.9912 (Random Forest)	0.9298 (SGD)
Sens (Sensitivity)	1.0000 (Multiple Classifiers)	0.8987 (SGD)
Spec (Specificity)	1.0000 (Multiple Classifiers)	0.9070 (Multiple Classifiers)
Prec (Precision)	1.0000 (Multiple Classifiers)	0.9070 (Multiple Classifiers)
NPV (Negative Predictive Value)	0.9767 (Random Forest)	0.8140 (SGD)
FPR (False Positive Rate)	0.0889 (QDA)	0 (Multiple Classifiers)
FDR (False Discovery Rate)	0.0563 (QDA)	0 (Multiple Classifiers)

FNR (False Negative Rate)	0.1013 (SGD)	0.0139 (Random Forest)
F1 (F1 Score)	99.30 (Random Forest)	0.9467 (SGD)
MCC (Matthews Correlation Coefficient)	98.14 (Random Forest)	0.8553 (SGD)
Time	1298.99 (MLP)	7.82 (GaussianNB)

7. Discussions

In various studies, the advantages of using Genetic Algorithms for feature selection have been well-documented [40,41]. This study designs a feature selection model that employs Genetic Algorithms to pinpoint relevant features, particularly useful when dealing with problems that have a lot of features.

When it is compared with 17 classifiers without any feature selection, the results clearly indicate that feature selection has led to an improvement in performance metrics. Table 6 below presents a comparative analysis of classification accuracies from various other studies that employed different methods of feature selection for the same dataset.

Table 6: Comparison of other proposed methods

Classifier	This Study (Random Forest)	ANN	SVM	Graph-Based	PS-Classifier

Test Accuracy	99.12%	96.70%	96.50%	96.40%	96.90%

This table indicates that Random Forest is the most effective classifier for feature selection using genetic algorithms. Additionally, this study has outperformed many other methods in the literature as seen in Table 7. The use of genetic algorithms has yielded a significant improvement in accuracy compared to traditional methods like Threshold Variance and Pearson Correlation for feature selection

Conclusion:

The present study employs a genetic algorithm for the purpose of feature selection in the context of breast cancer detection. The model underwent evaluation using a total of 17 distinct classifiers, and it was determined that the Random Forest Classifier produced the best level of test accuracy, specifically 99.12%. The findings indicate that the act of choosing suitable features has the potential to enhance the performance of classification. The present study provides evidence of the efficacy of a genetic algorithm in the context of feature selection. Furthermore, it proposes that future investigations should investigate the potential of applying this algorithm to various forms of cancer, including but not limited to brain, prostate, and kidney cancer.

References:

1. Robbins, K. R., et al. "The ant colony algorithm for feature selection in high-dimension gene expression data for disease classification." *Mathematical Medicine and Biology: a Journal of the IMA* 24.4 (2007): 413-426.

2. Moradi, Parham, and Mehrdad Rostami. "A graph theoretic approach for unsupervised feature selection." *Engineering Applications of Artificial Intelligence* 44 (2015): 33-45.

3. https://journalofbigdata.springeropen.com/articles/10.1186/s40537-020-00398-3

4. Kohavi, R., John, G.H., “Wrappers for feature subset selection”, Artificial Intelligence, 97(1-2): 273-324, (1997).

5 Aalaei, Shokoufeh et al. “Feature selection using genetic algorithm for breast cancer diagnosis: experiment on three different datasets.” *Iranian journal of basic medical sciences* vol. 19,5 (2016): 476-82.

6 Łukasiewicz, Sergiusz et al. “Breast Cancer-Epidemiology, Risk Factors, Classification, Prognostic Markers, and Current Treatment Strategies-An Updated Review.” *Cancers* vol. 13,17 4287. 25 Aug. 2021, doi:10.3390/cancers13174287

7 Basha, S. Saheb, and K. Satya Prasad. "AUTOMATIC DETECTION OF BREAST CANCER MASS IN MAMMOGRAMS USING MORPHOLOGICAL OPERATORS AND FUZZY C--MEANS CLUSTERING." *Journal of Theoretical & Applied Information Technology* 5.6 (2009).

8 Kuhl, Christiane K et al. “Mammography, breast ultrasound, and magnetic resonance imaging for surveillance of women at high familial risk for breast cancer.” *Journal of clinical oncology : official journal of the American Society of Clinical Oncology* vol. 23,33 (2005): 8469-76. doi:10.1200/JCO.2004.00.4960

9 Elmore, Joann G., et al. "Variability in radiologists' interpretations of mammograms." *New England Journal of Medicine* 331.22 (1994): 1493-1499.

10 Fletcher, Suzanne W., et al. "Report of the international workshop on screening for breast cancer." *JNCI: Journal of the National Cancer Institute* 85.20 (1993): 1644-1656.

11 Aalaei, Shokoufeh et al. “Feature selection using genetic algorithm for breast cancer diagnosis: experiment on three different datasets.” *Iranian journal of basic medical sciences* vol. 19,5 (2016): 476-82.

12 Guyon, Isabelle, and André Elisseeff. "An introduction to variable and feature selection." *Journal of machine learning research*3.Mar (2003): 1157-1182.

13 Bermejo, Pablo, Jose A. Gámez, and Jose M. Puerta. "A GRASP algorithm for fast hybrid (filter-wrapper) feature subset selection in high-dimensional datasets." *Pattern Recognition Letters* 32.5 (2011): 701-711.

14https://link.springer.com/chapter/10.1007/978-1-4615-5725-8_3#:~:text=The%20wrapper%20method%20searches%20for,approach%20to%20feature%20subset%20selection.

15 Aghdam, Mehdi Hosseinzadeh, Nasser Ghasem-Aghaee, and Mohammad Ehsan Basiri. "Application of ant colony optimization for feature selection in text categorization." *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*. IEEE, 2008.

16 Unler, Alper, and Alper Murat. "A discrete particle swarm optimization method for feature selection in binary classification problems." *European Journal of Operational Research* 206.3 (2010): 528-539.

17 Karegowda, Asha Gowda, M. A. Jayaram, and A. S. Manjunath. "Feature subset selection problem using wrapper approach in supervised learning." *International journal of Computer applications* 1.7 (2010): 13-17.

18 Youn, Eunseog, et al. "Support vector-based feature selection using Fisher's linear discriminant and Support Vector Machine." *Expert Systems with Applications* 37.9 (2010): 6148-6156.

19 Deisy, C., et al. "Efficient dimensionality reduction approaches for feature selection." *International Conference on Computational Intelligence and Multimedia Applications (ICCIMA 2007)*. Vol. 2. IEEE, 2007.

20 Sridevi, T., and A. Murugan. "An intelligent classifier for breast cancer diagnosis based on K-Means clustering and rough set." *International Journal of Computer Applications* 85.11 (2014).

21 Sridevi, T., and A. Murugan. "A novel feature selection method for effective breast cancer diagnosis and prognosis." *International Journal of Computer Applications* 88.11 (2014).

22 10.24432/C5DW2B

https://archive.ics.uci.edu/dataset/17/breast+cancer+wisconsin+diagnostic

23 DOI 10.1007/s10994-006-6226-1

24https://www.tacoma.uw.edu/sites/default/files/2021-08/melanson_david_senior_thesis_2020.pdf

25 https://www.mdpi.com/2078-2489/11/6/332

26 Freund, Yoav, and Robert E. Schapire. "Experiments with a new boosting algorithm." *icml*. Vol. 96. 1996.

27 Breiman, Leo. "Bagging predictors." *Machine learning* 24 (1996): 123-140.

28 Amit, Yali, and Donald Geman. "Shape quantization and recognition with randomized trees." *Neural computation* 9.7 (1997): 1545-1588.

29 Ho, Tin Kam. "The random subspace method for constructing decision forests." *IEEE transactions on pattern analysis and machine intelligence* 20.8 (1998): 832-844.

30 Ho, Tin Kam. "Random decision forests." *Proceedings of 3rd international conference on document analysis and recognition*. Vol. 1. IEEE, 1995.

31https://www.researchgate.net/publication/361392986_Gaussian_Naive_Bayes_Algorithm_A_Reliable_Technique_Involved_in_the_Assortment_of_the_Segregation_in_Cancer

32 https://www.mdpi.com/2077-1312/9/5/496

33https://proceedings.neurips.cc/paper_files/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf

34 https://arxiv.org/pdf/1906.02590.pdf

35 https://academicworks.cuny.edu/cgi/viewcontent.cgi?article=1535&context=gc_etds

36 https://link.springer.com/chapter/10.1007/978-3-642-34041-3_27

37 https://eudl.eu/pdf/10.4108/eai.13-7-2017.2270596

38 https://doi.org/10.1016/B978-0-12-416743-8.00005-1

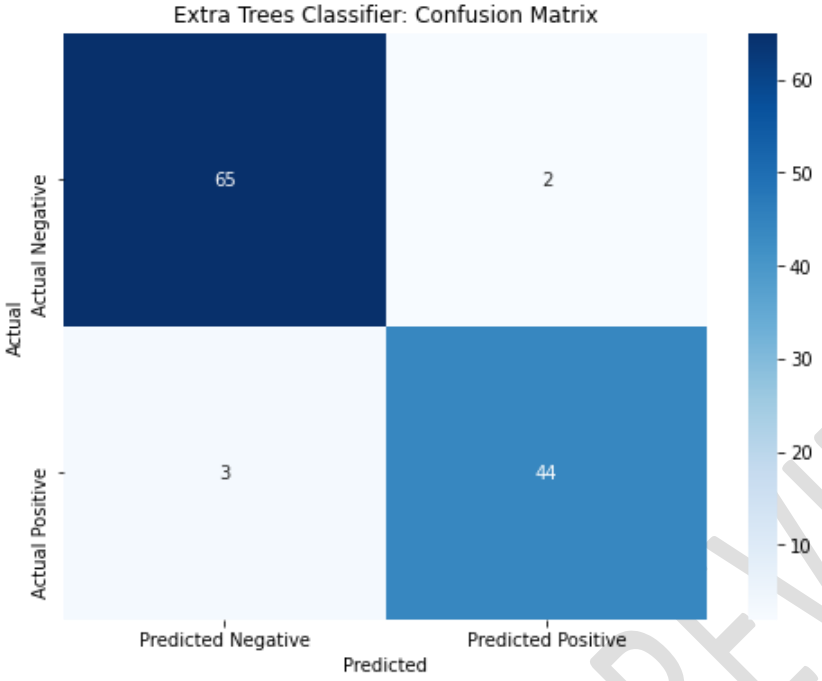
39 https://inis.iaea.org/collection/NCLCollectionStore/_Public/38/027/38027911.pdf

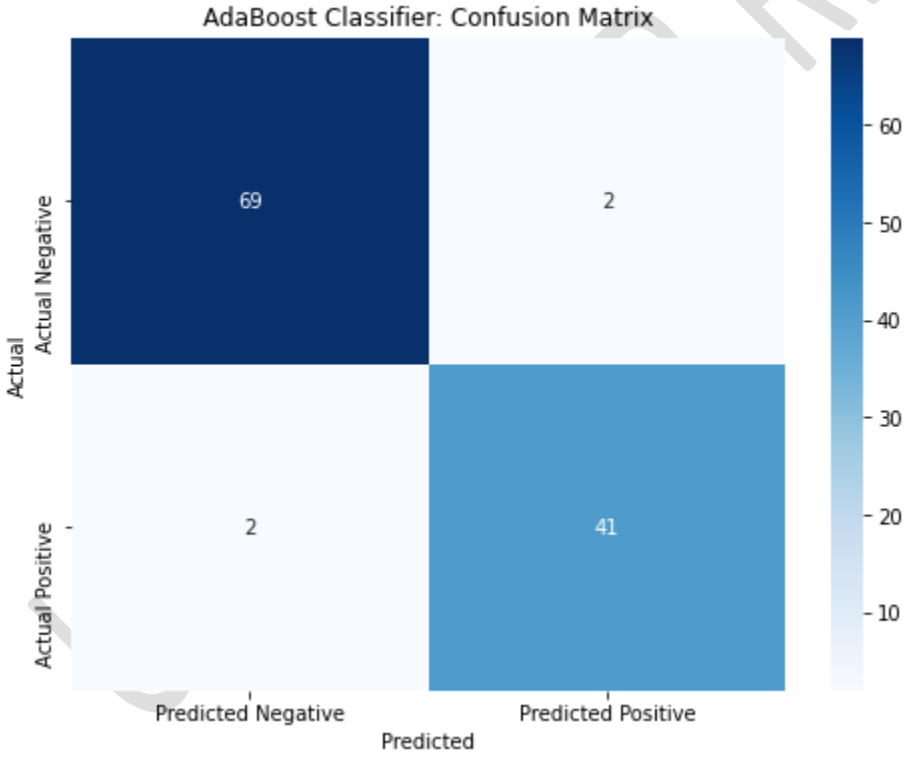
40 Eroltu, Kaan. "Comparing different Convolutional Neural Networks for the classification of Alzheimer's Disease." *Journal of High School Science* 7.3 (2023).

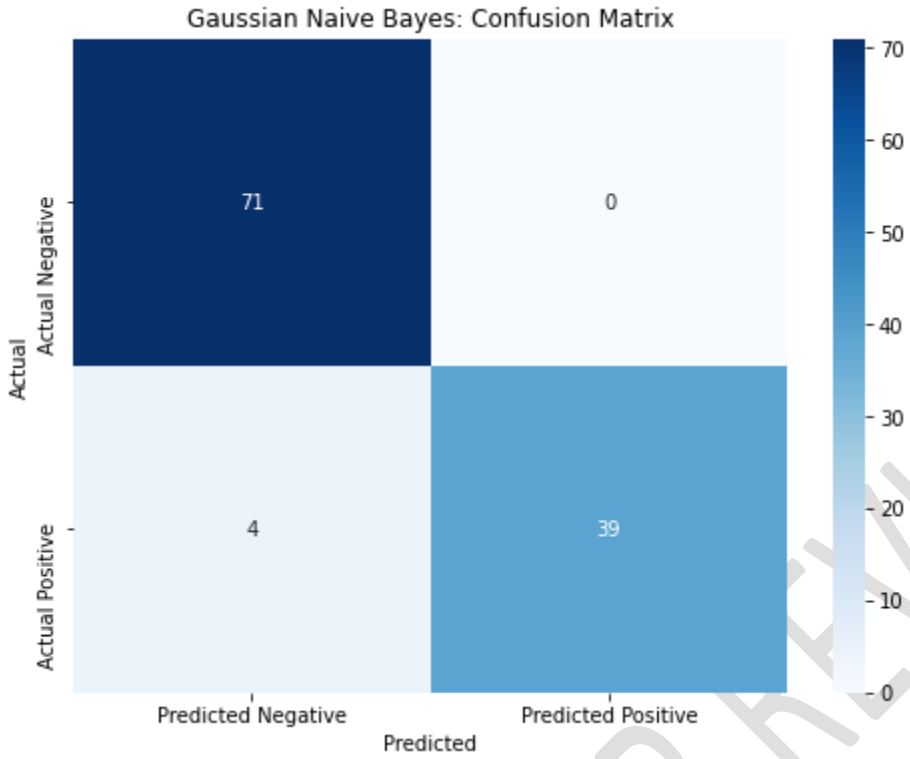
41 Oh, Il-Seok, Jin-Seon Lee, and Byung-Ro Moon. "Hybrid genetic algorithms for feature selection." *IEEE Transactions on pattern analysis and machine intelligence* 26.11 (2004): 1424-1437.

42 Hadizadeh, Farzin, Saadat Vahdani, and Mehrnaz Jafarpour. "Quantitative structure-activity relationship studies of 4-imidazolyl-1, 4-dihydropyridines as calcium channel blockers." Iranian journal of basic medical sciences 16.8 (2013): 910.

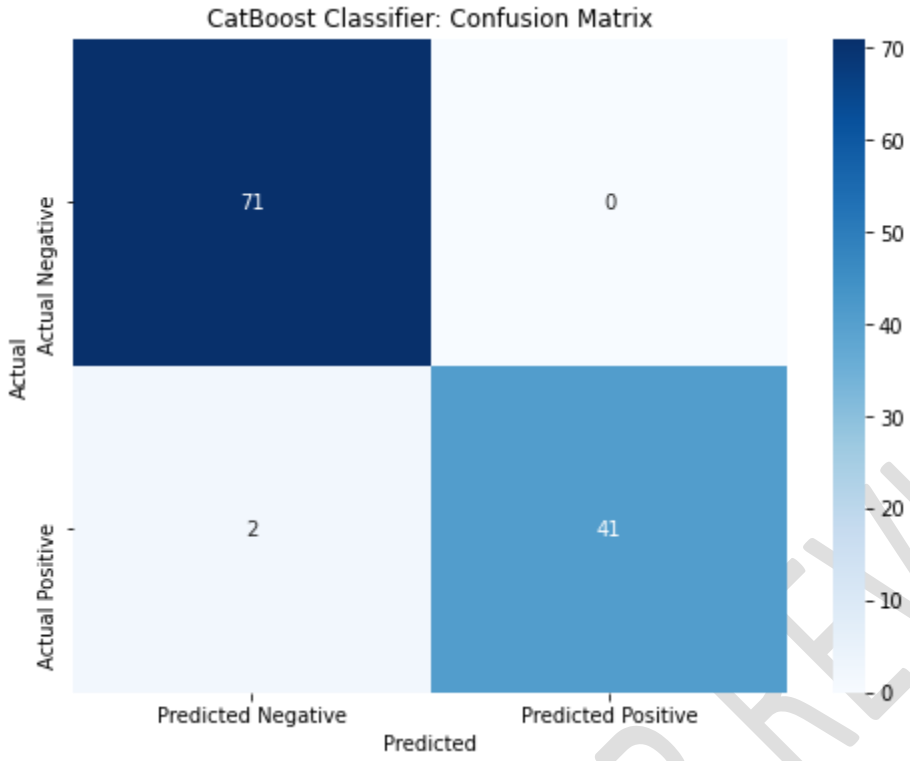
Appendix

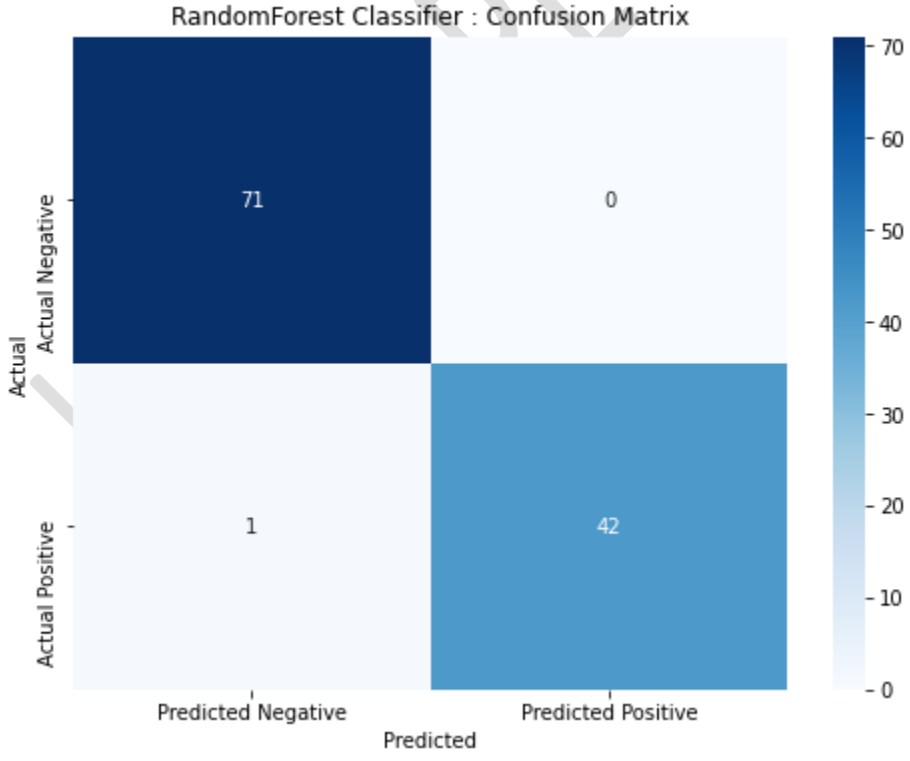

Extra Trees Classifier: Confusion Matrix
65
2
3
44
Actual Negative
Actual Positive
Actual
Predicted Negative
Predicted Positive
Predicted
60
50
40
30
20
10

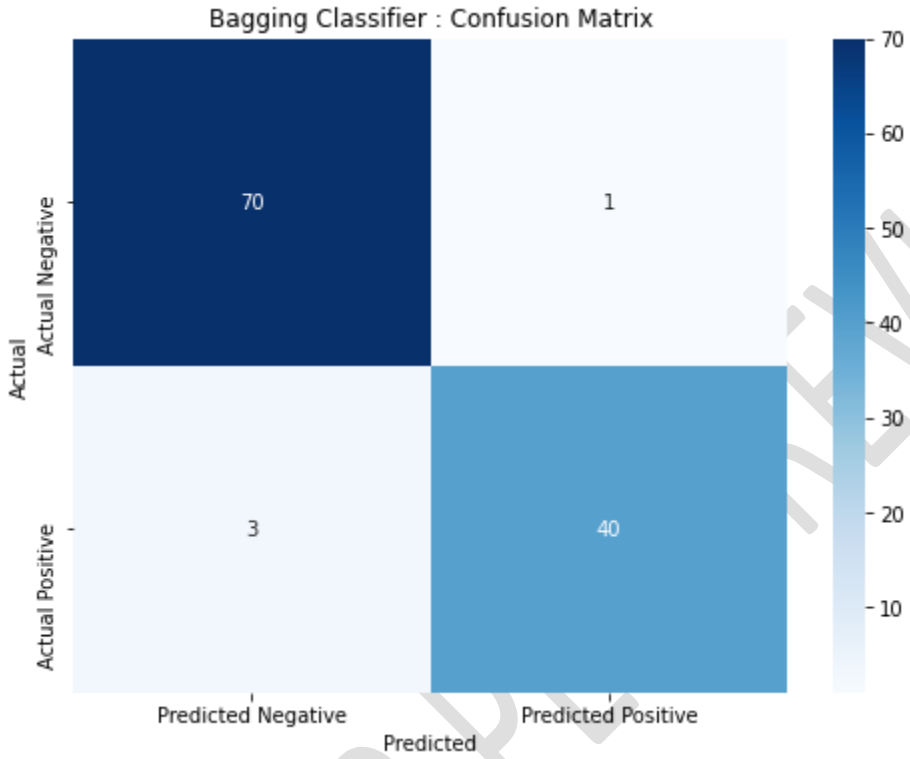

AdaBoost Classifier: Confusion Matrix
69
2
2
41
Actual Negative
Actual Positive
Actual
Predicted Negative
Predicted Positive
Predicted
60
50
40
30
20
10

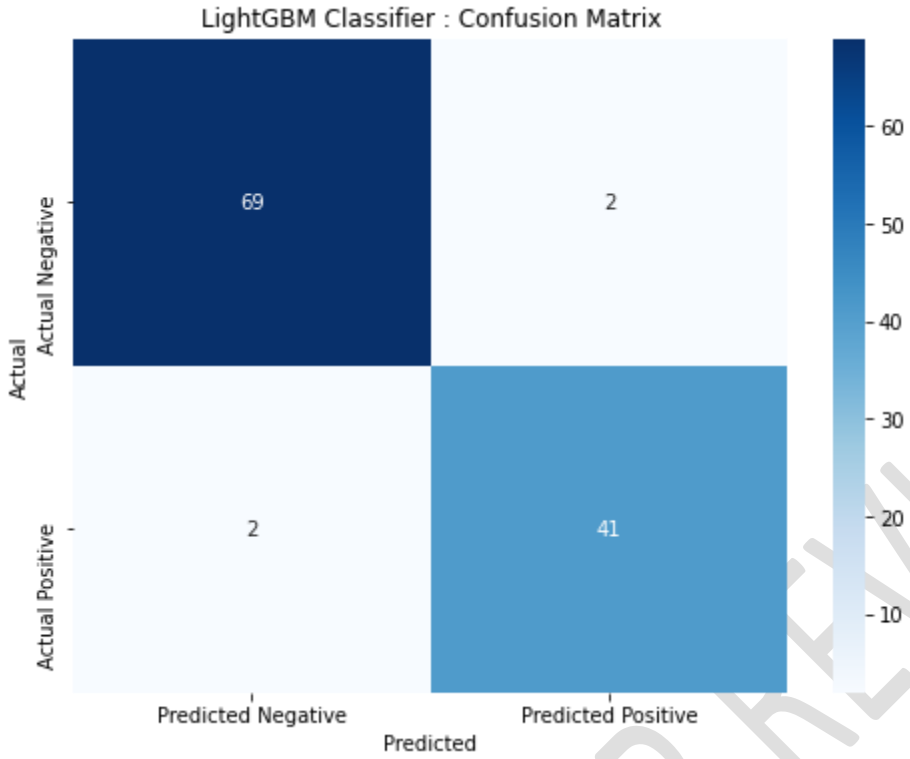


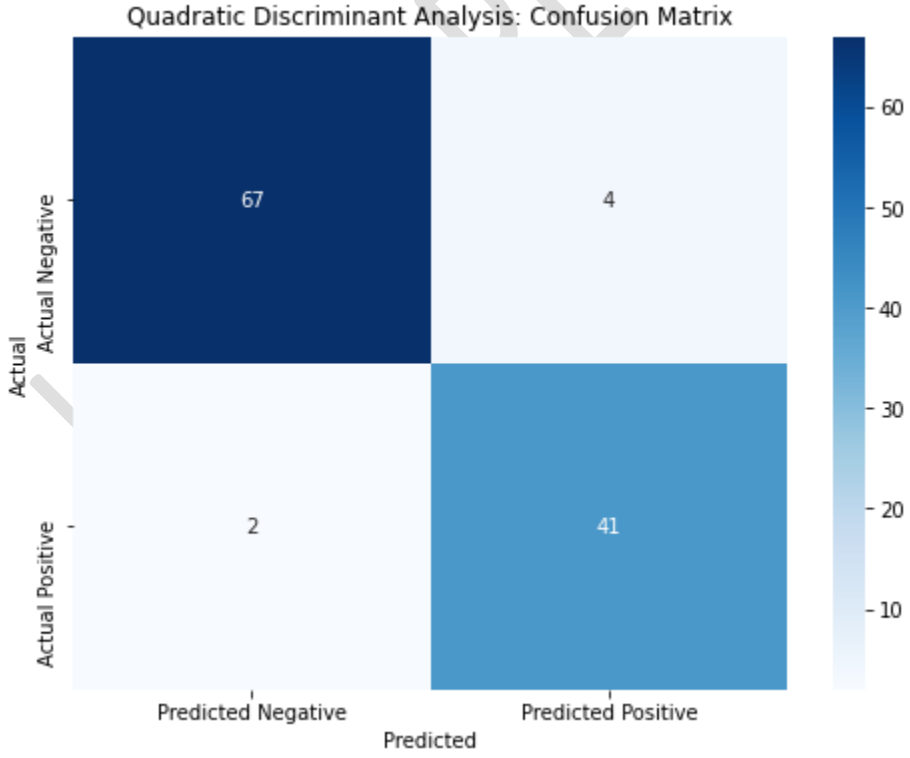
\

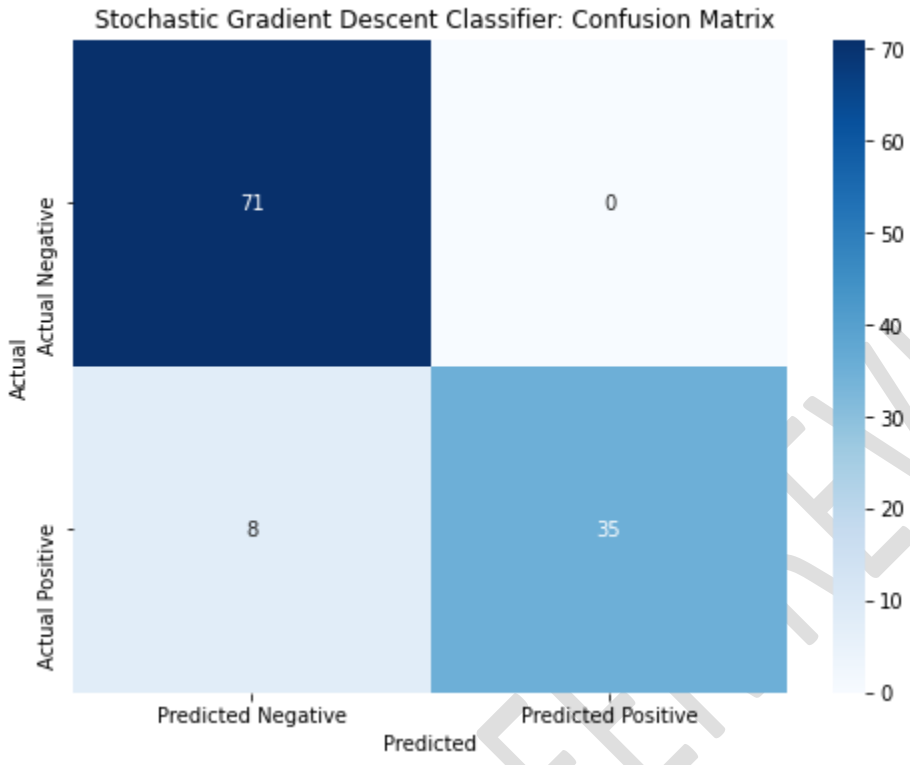


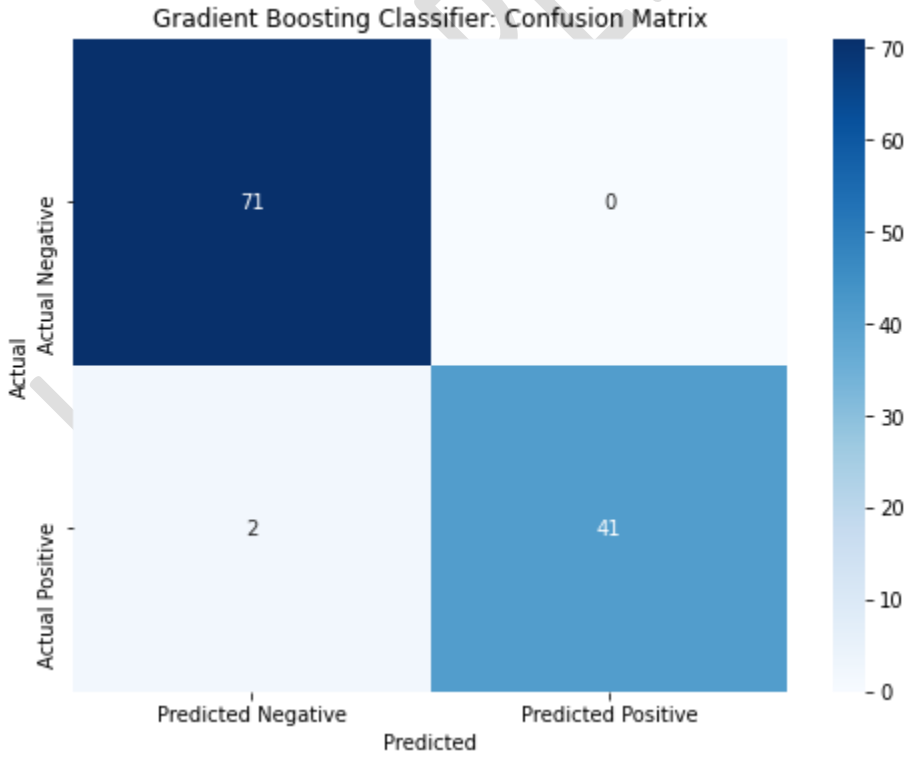


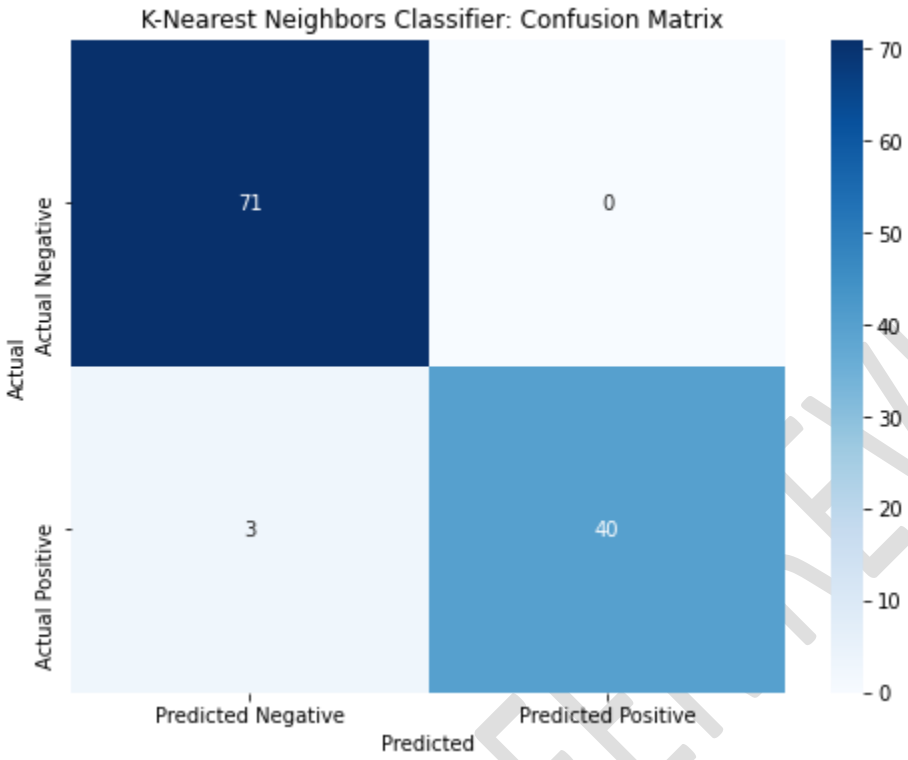


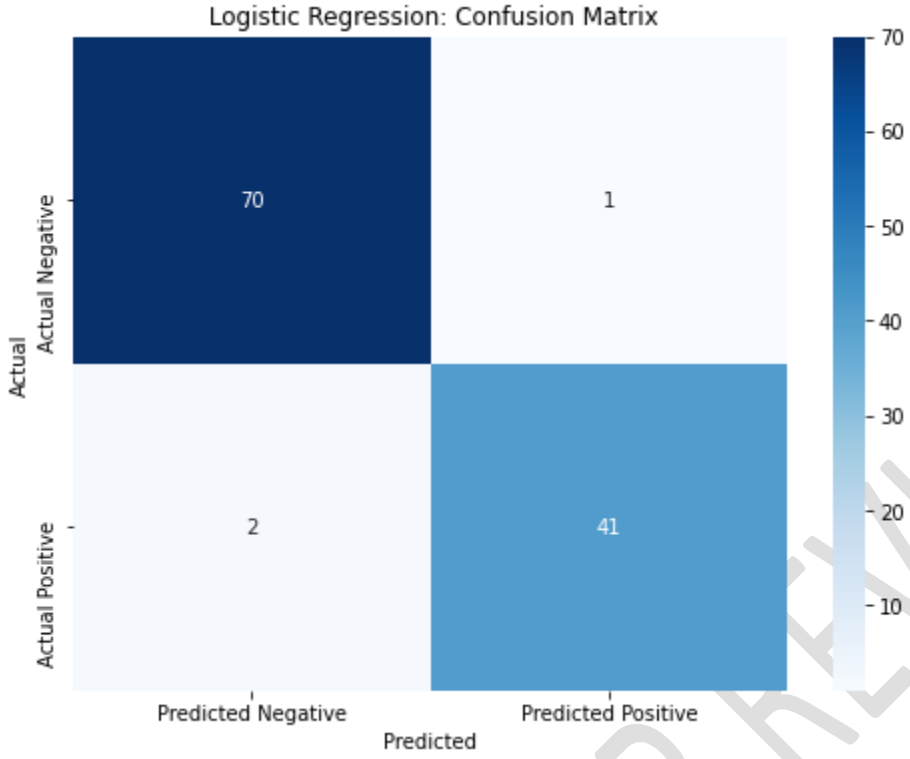

LightGBM Classifier : Confusion Matrix
69
2
2
41
Actual
Actual Negative
Actual Positive
Predicted Negative
Predicted Positive
Predicted
60
50
40
30
20
10

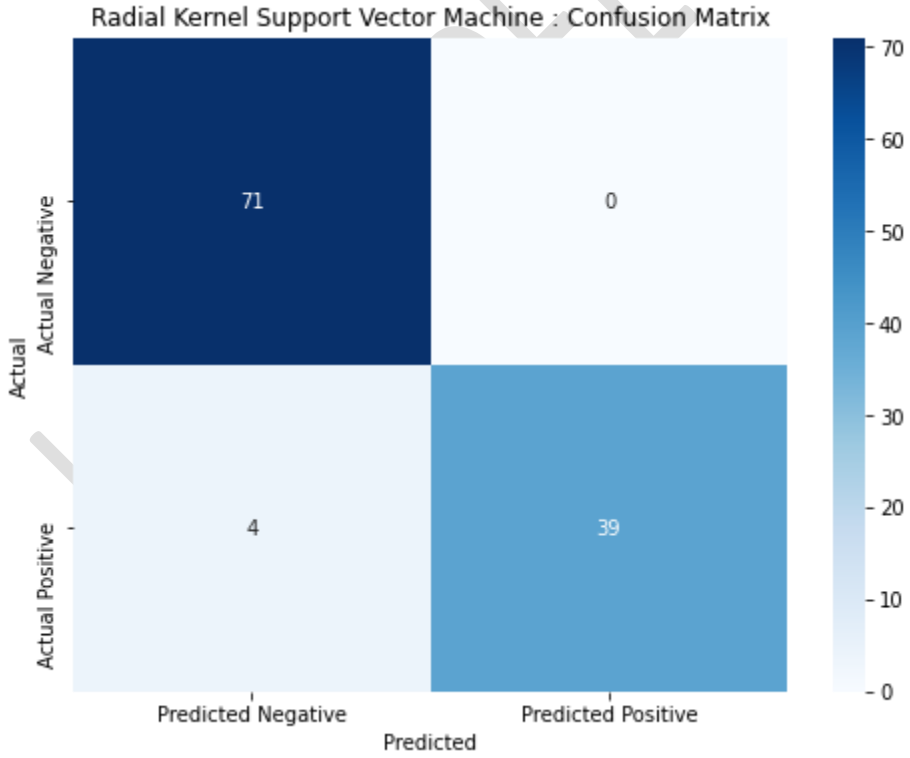

Quadratic Discriminant Analysis: Confusion Matrix
67
4
2
41
Actual
Actual Negative
Actual Positive
Predicted Negative
Predicted Positive
Predicted
60
50
40
30
20
10

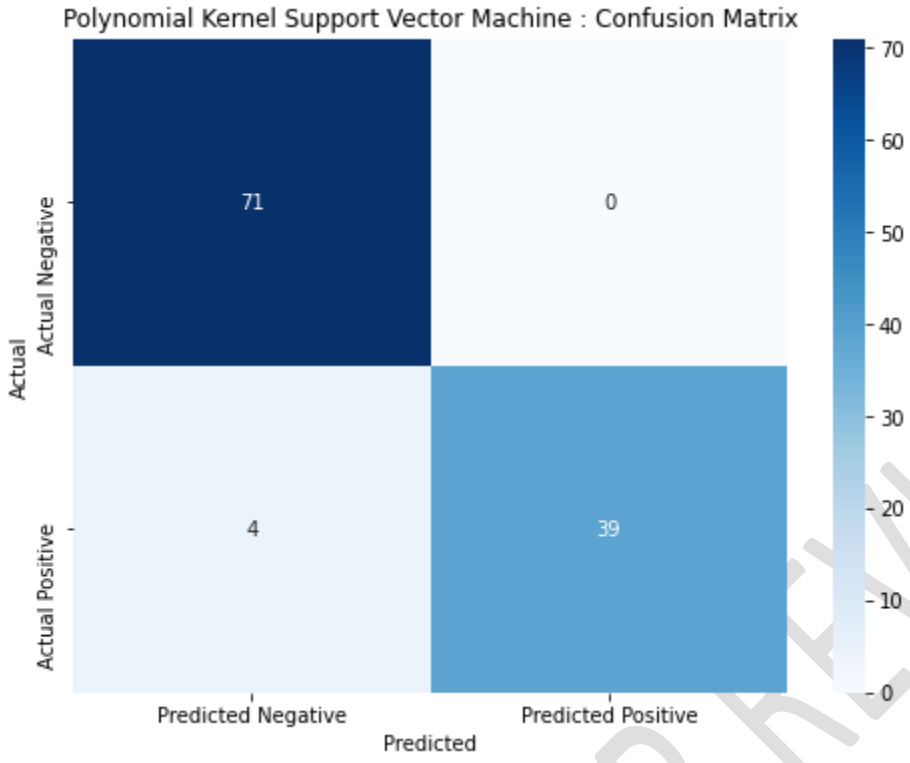

Stochastic Gradient Descent Classifier: Confusion Matrix
71
0
8
35
Actual Negative
Actual Positive
Actual
Predicted Negative
Predicted Positive
Predicted
70
60
50
40
30
20
10
0

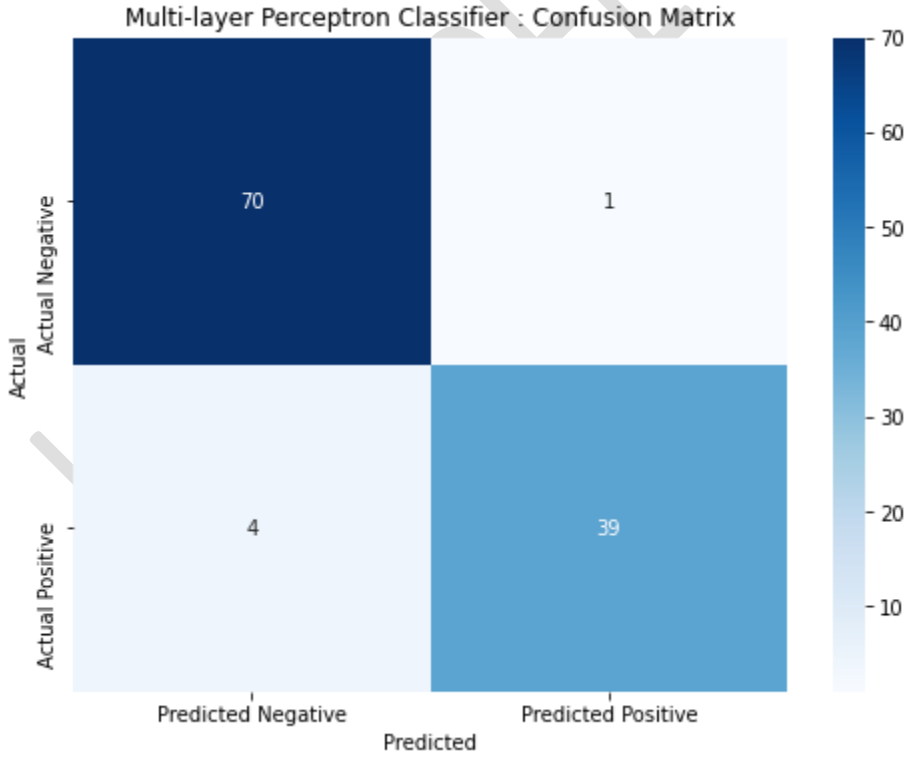

Gradient Boosting Classifier: Confusion Matrix
71
0
2
41
Actual Negative
Actual Positive
Actual
Predicted Negative
Predicted Positive
Predicted
70
60
50
40
30
20
10
0




Logistic Regression: Confusion Matrix
70
1
2
41
Actual Negative
Actual Positive
Actual
Predicted Negative
Predicted Positive
Predicted
70
60
50
40
30
20
10


Radial Kernel Support Vector Machine : Confusion Matrix
71
0
4
39
Actual Negative
Actual Positive
Actual
Predicted Negative
Predicted Positive
Predicted
70
60
50
40
30
20
10
0


Polynomial Kernel Support Vector Machine : Confusion Matrix
71
0
4
39
Actual Negative
Actual Positive
Actual
Predicted Negative
Predicted Positive
Predicted
70
60
50
40
30
20
10
0


Multi-layer Perceptron Classifier : Confusion Matrix
70
1
4
39
Actual Negative
Actual Positive
Actual
Predicted Negative
Predicted Positive
Predicted
70
60
50
40
30
20
10