International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 Original Research Paper

Enhancing Breast Cancer Detection: Exploring Machine Learning Approaches Through AUC-ROC Evaluation

N. Senthil Kumar[1], K. Mohan Kumar[1]

Submitted: 18/04/2024 **Revised:** 5/05/2024 **Accepted:** 10/05/2024

Abstract: Breast cancer is a prominent cause of female mortality worldwide addressed through machine learning techniques using the Wisconsin Breast Cancer Dataset (WBCD) to compare classifier effectiveness. The Random Forest (RF), Logistic Regression (LR), and Decision Tree (DT) models are assessed in this job scenario with an emphasis on metrics related to recall, accuracy, and precision. This research emphasizes the value of early detection and the prospective applications of machine learning in the medical field. According to this research, the use of cutting-edge technology in medical diagnostics has shown encouraging results, providing hope for more effective and prompt identification and treatment of breast cancer.

Keywords: BreastCancer , MachineLearning, Algorithms

1. Introduction

Breast cancer is a prevalent and complex disease characterized by the uncontrolled growth of malignant cells within breast tissue. It is one of the most commonly diagnosed cancers among women globally and represents a significant public health concern. Breast cancer can also occur in men, albeit less frequently. Breast cancer can manifest in a number of ways, from small tumors in one area of the body to metastasized cancer that has spread to other areas. Although survival rates have increased due to early identification and treatment improvements, breast cancer is still the primary cause of cancer-related deaths among women globally. Early detection is greatly aided by screening techniques like self-examination, clinical breast exams, and mammography. Treatment options for breast cancer can include surgery, radiation treatment, chemotherapy, hormone therapy, targeted therapy, or a combination of these techniques, depending on the stage and type of the disease. Despite significant progress in research and treatment, challenges persist in understanding the complexities of breast cancer biology, improving access to care, and reducing disparities in outcomes among different populations. Ongoing efforts in prevention, early detection, and innovative treatment approaches are essential in the fight against breast cancer.

1.1 Machine Learning

Machine Learning comprises three key stages: data cleaning and preprocessing, model training, and model evaluation. Data cleaning involves various processes, such as eliminating missing or corrupted data and removing irrelevant information.

[1]*PG and Research Department of Computer Science, Rajah Serfoji Government College, Thanjavur 613005*
(Affiliated to Bharathidasan University, Trichirappalli -620 024)

Preprocessing transforms data into an understandable format. Real-world data often exhibits a wide range of values for each attribute, which can be narrowed using scaling techniques. In cases of extensive feature sets, dimensionality reduction techniques are employed to simplify the learning process.

1.2 Machine Learning Algorithms

Logistic Regression (LR): The observed data must be fitted with a logistic function that converts the input variables to the output variable using a sigmoidal curve in order for LR to work. Random Forest (RF): The RF ensemble learning technique handles both regression and classification tasks. Decision Trees (DT): DT divides the dataset into subsets recursively, with each node's most important feature serving as the basis. With the use of data, machine learning (ML) is revolutionizing the medical field by enhancing patient outcomes, diagnosis, treatment planning, and operational efficiency.

2. Literature Survey

Varsha Nemade et al. (2023) explained the significance of utilizing machine learning (ML) algorithms to analyze breast cancer datasets and presents a model for breast cancer data classification, employing six ML techniques: Decision Tree (DT), K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Random Forest (RF), Naïve Bayes (NB), and Logistic Regression (LR), as well as ensemble techniques, on the WDBC dataset. Results achieved the highest accuracy at 97%, while LR attained the highest Area Under the Curve (AUC) at 0.996. Additionally, XGBoost ensemble technique demonstrated high accuracy

(97%) and AUC (0.99).

Amit Bhanushali et al. (2023) discussed to achieving high breast cancer survival rates through early diagnosis with a 70% training and 30% testing split, the study employs comparative analysis using five distinct classifiers to simplify the classification process. Results reveal that Random Forest (RF) performed exceptionally well acrossall metrics, including F1 score (93.5%), Precision (93.5%), Recall (93.6%), and Accuracy (93.5%). The findings suggest that machine learning methodologies are more suitable for breast cancer prediction compared to deep learning methods. **Cordova Calle et al. (2022)** assessed a method for breast cancer prediction using data science processes and machine learning. It highlights the increasing prevalence of breast cancer globally and outlines a three-phase analysis method. Machine learning techniques like KNN, gradient boosting classifier, and random forest are used, and evaluation metrics such as accuracy, precision, recall, and F1-Score are discussed.

Sandhu et al. (2022) explored the application of machine learning algorithms for early-stage breast cancer analysis and also examines various datasets and evaluates supervised and unsupervised machine learning algorithms for early breast cancer prediction.

Singh and Agrawal (2022) scrutinized the need for accurate breast cancer detection methods and proposes a study using machine learning classifiers. It discusses the rising prevalence of breast cancer and the potential of computer technology integration in healthcare. Various machine learning techniques, including logistic regression, decision tree, and k-nearest neighbor, are compared based on their accuracy in detecting malignant and benign tumors.

Barwal and Raheja (2022) developed a model that focuses on developing a classification system for breast cancer prediction using the SVOF-KNN method. It discusses the importance of accurate cancer diagnosis and the challenges associated with traditional diagnostic methods. The study utilizes the BC Coimbra dataset and evaluates the SVOF-KNN model's performance in classifying breast cancer risk factors. The results demonstrate improved accuracy compared to existing models, highlighting the potential of the proposed approach in enhancing breast cancer diagnosis.

Akhil et al.(2022) investigate on breast cancer prognosis using machine learning techniques. It discusses the global impact of breast cancer and the need for early detection methods. The study compares five machine learning algorithms using the Wisconsin Cancer Diagnostic dataset and identifies Random Forest as the most accurate algorithm for breast cancer prediction. The findings underscore the potential of machine learning in improving breast cancer prognosis and treatment outcomes.

3. Methodology

This section covers breast cancer diagnosis techniques as well as our suggested plan of action (Figure 1). The performance of any machine learning model varies depending on the factors considered. Therefore, selecting the best qualities is essential to achieving the best results. In order to generate the proper AUC-ROC curves for this study, apply machine learning techniques such as Decision Tree (DT), Random Forest (RF), and Logistic Regression (LR). The selected characteristics are then fed into the suggested classifiers to achieve the greatest identification accuracy for breast cancer. Regarding breast cancer prediction, our testing of various machine learning classifiers has produced great results. For clarity, Algorithm 1 from our research is presented below.

Algorithm 1: Working procedure of Breast Cancer prediction

Input: Breast Cancer Dataset (BCD)

Output: Predicted value of cancer (Malignant or Benign)

1. Load the Breast Cancer Dataset (BCD).
2. Check for missing or NaN values in the dataset.
   2.1 If missing or NaN values found, replace them.
3. Pre-process the dataset:
   3.1 Convert the target variable into numerical values.
      3.1.1 Replace 'M' with 0 and 'B' with 1.
   3.2 Split the dataset into training and testing sets:
      3.2.1 Denote training set as x1, y1 and testing set as x2, y2.
4. Train a model:
   4.1 Train a model using the training data (x1, y1).
   4.2 Test the trained model's performance on the testing data (x2, y2).
5. Feature Scaling:
   5.1 Scale the input data (x) to normalize it.
6. Dimensionality Reduction
7. Classifier Training:

7.1 Train a classifier using the scaled and dimensionality-reduced data.

8. Cross-Validation:

8.1 Perform cross-validation with the voting classifier to assess its performance.

9. Performance Evaluation:

9.1 Compute performance evaluation metrics to assess the model's effectiveness.

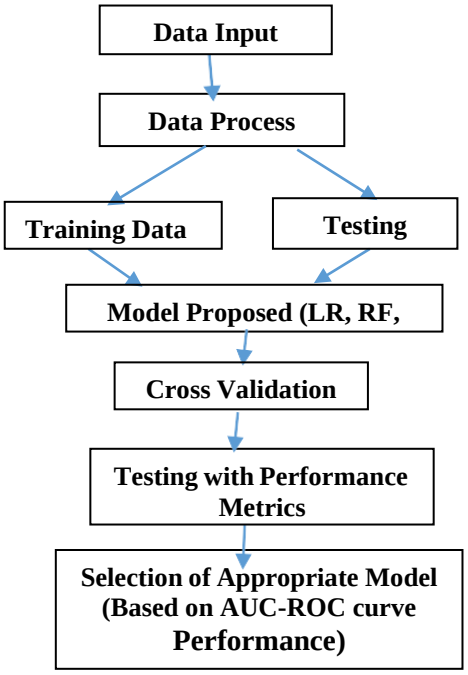

Fig 1: Working procedure of proposed methodology

3.1 DATASET

The Wisconsin Breast Cancer Dataset, developed by Dr. William H. Wolberg from the University of Wisconsin Hospital, is commonly utilized for research works. This dataset, which was gathered from the UCI repository, has 30 features and 569 instances with no missing values. As seen in Figure, 357 of these cases are benign (non-cancerous) breast cancer, and 212 are malignant (cancerous) cases. For the purpose of model training and testing, the total dataset is split randomly into training and testing sets in a 3:1 ratio. To avoid redundancy, a column named "id" is removed from consideration, as it does not impact the outcome. Each factor in the table 1 below has a direct correlation with the result and is considered a predictive variable for breast cancer. The degree of correlation between each segment and the outcome is shown by the priority ratio given to it in the table.

3.2 Data Preparatory Processes

To maximize the dataset's performance, data preparation is essential. Commonly found in databases, noise, missing numbers, and data imbalances can all have a negative effect on the accuracy of the results. Before applying machine learning techniques, the dataset needs to be cleaned up of these undesirable elements. 'M' and 'B' stand for malignant and benign diagnoses, respectively, in the dataset. These category values are translated into their numerical equivalents, where 1 denotes benign and 0 indicates malignant. The scale of every feature is then standardized using Max-Min Normalization, guaranteeing consistency. Further ensuring that every feature has an equal impact on the distance metric is data standardization. Once pre-processing is completed, the dataset has several unique features.The statistical analysis of the dataset is detailed in Table 1.

Table 1-Statistical data analysis

Statistical analysis of the dataset.

Features	Mean	Standard Division	Max	Min
radius_mean	14.127	3.524	6.981	28.11
texture_mean	19.289	4.3	9.71	39.28
perimeter_mean	91.969	24.29	43.79	188.50
area_mean	654.889	351.91	143.5	2501.00
smoothness_mean	0.096	0.014	0.052	0.16
compactness_mean	0.104	0.0528	0.019	0.34
concavity_mean	0.088	0.079	0	0.42
concave points_mean	0.048	0.038	0	0.20
symmetry_mean	0.181	0.027	0.106	0.30
fractal_dimension_mean	0.062	0.007	0.049	0.09
radius_se	0.405	0.277	0.11	2.87
texture_se	1.2168	0.55	0.36	4.88
perimeter_se	2.866	2.021	0.757	21.98
area_se	40.337	45.49	6.802	542.20
smoothness_se	0.007	0.003	0.0017	0.03
compactness_se	0.025	0.017	0.002	0.13
concavity_se	0.031	0.030	0	0.39
concave points_se	0.011	0.006	0	0.05
symmetry_se	0.020	0.008	0.0078	0.07
fractal_dimension_se	0.003	0.002	0.00089	0.029
radius_worst	16.269	4.833	7.93	36.04
texture_worst	25.677	6.146	12.02	49.54
perimeter_worst	107.26	33.602	50.41	251.20
area_worst	880.58	569.356	185.2	4254.00
smoothness_worst	0.132	0.022	0.071	0.22
compactness_worst	0.254	0.157	0.027	1.058
concavity_worst	0.272	0.208	0	1.252
concave points_worst	0.114	0.065	0	0.29
symmetry_worst	0.290	0.0618	0.156	0.66
fractal_dimension_worst	0.083	0.018	0.055	0.207

3.3 Data Exploration Phase

A thorough analysis of the dataset based on its distinctive features is necessary before launching into the machine learning workflow. Different ways to treating different forms of data—numerical, categorical, ordinal, and nominal—are needed. The general quality of the data can be shown by a perceptive examination of the major predictor columns. Examining the relationships between various data points is necessary for this, as the accompanying Figure 2 illustrates.

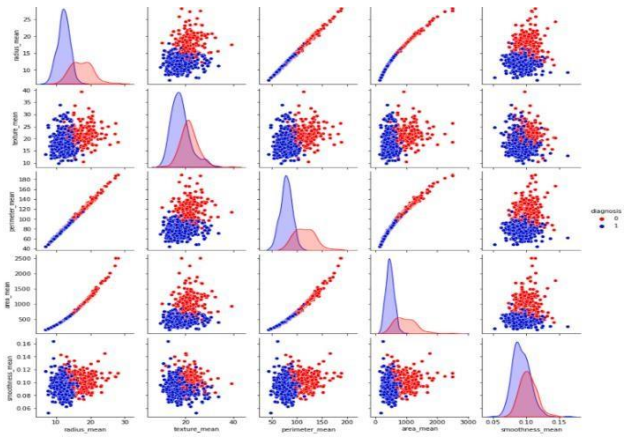

Figure 2 : Pair plot showcasing the features [radius_mean, texture_mean, perimeter_mean, area_mean, smoothness_mean].

Jupiter Notebook was used to examine the dataset once it was imported into the Anaconda distribution. After analyzing the properties of the dataset, machine learning models were created and verified. To assess the performance of each ML algorithm, confusion matrices were created. (Figure 3 to 8).

```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

# keeps the plots in one place. calls image as static pngs
%matplotlib inline
import matplotlib.pyplot as plt # side-stepping mpl backend
import matplotlib.gridspec as gridspec # subplots
import seaborn as sns

#Import models from scikit learn module:
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import KFold
from sklearn.ensemble import RandomForestClassifier
from sklearn.tree import DecisionTreeClassifier, export_graphviz
from sklearn import metrics
from sklearn.metrics import confusion_matrix



# import warnings filter
from warnings import simplefilter
# ignore all future warnings
simplefilter(action='ignore', category=FutureWarning)

#Seaborn
import seaborn as sns


df = pd.read_csv('C:\\Users\\user\\Desktop\\DOCUMENTS\\base papers\\cancer_data.csv');

df.info()
```

Figure 3: Importing data set

```
df = pd.read_csv('C:\\Users\\user\\Desktop\\DOCUMENTS\\base papers\\cancer_data.
df.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 569 entries, 0 to 568
Data columns (total 33 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   id                       569 non-null    int64
 1   diagnosis                569 non-null    object
 2   radius_mean              569 non-null    float64
 3   texture_mean             569 non-null    float64
 4   perimeter_mean           569 non-null    float64
 5   area_mean                569 non-null    float64
 6   smoothness_mean          569 non-null    float64
 7   compactness_mean         569 non-null    float64
 8   concavity_mean           569 non-null    float64
 9   concave points_mean      569 non-null    float64
 10  symmetry_mean            569 non-null    float64
 11  fractal_dimension_mean   569 non-null    float64
 12  radius_se                569 non-null    float64
 13  texture_se               569 non-null    float64
 14  perimeter_se             569 non-null    float64
 15  area_se                  569 non-null    float64
 16  smoothness_se            569 non-null    float64
 17  compactness_se           569 non-null    float64
 18  concavity_se             569 non-null    float64
 19  concave points_se        569 non-null    float64
 20  symmetry_se              569 non-null    float64
 21  fractal_dimension_se     569 non-null    float64
 22  radius_worst             569 non-null    float64
 23  texture_worst            569 non-null    float64
 24  perimeter_worst          569 non-null    float64
 25  area_worst               569 non-null    float64
 26  smoothness_worst         569 non-null    float64
 27  compactness_worst        569 non-null    float64
 28  concavity_worst          569 non-null    float64
 29  concave points_worst     569 non-null    float64
 30  symmetry_worst           569 non-null    float64
```

Figure 4: Extracting Features of Data set

```
In [5]: ### Loading the data file
        df = pd.read_csv("C:\\Users\\user\\Desktop\\DOCUMENTS\\base papers\\cancer_data.csv",header = 0)
        col = ['radius_mean','perimeter_mean','area_mean','compactness_mean','concave points_mean','diagnosis']
        df[col]
        # df.head(10)
```

Out[5]:

	radius_mean	perimeter_mean	area_mean	compactness_mean	concave points_mean	diagnosis
0	17.99	122.80	1001.0	0.27760	0.14710	M
1	20.57	132.90	1326.0	0.07864	0.07017	M
2	19.69	130.00	1203.0	0.15990	0.12790	M
3	11.42	77.58	386.1	0.28390	0.10520	M
4	20.29	135.10	1297.0	0.13280	0.10430	M
...	...	...	...	...	...	...
564	21.56	142.00	1479.0	0.11590	0.13890	M
565	20.13	131.20	1261.0	0.10340	0.09791	M
566	16.60	108.30	858.1	0.10230	0.05302	M
567	20.60	140.10	1265.0	0.27700	0.15200	M
568	7.76	47.92	181.0	0.04362	0.00000	B

569 rows × 6 columns

Figure 5: Displaying the Recording

```
In [15]: df[col].describe()
```

Out[15]:

	radius_mean	perimeter_mean	area_mean	compactness_mean	concave points_mean	diagnosis
count	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000
mean	14.127292	91.969033	654.889104	0.104341	0.048919	0.372583
std	3.524049	24.298981	351.914129	0.052813	0.038803	0.483918
min	6.981000	43.790000	143.500000	0.019380	0.000000	0.000000
25%	11.700000	75.170000	420.300000	0.064920	0.020310	0.000000
50%	13.370000	86.240000	551.100000	0.092630	0.033500	0.000000
75%	15.780000	104.100000	782.700000	0.130400	0.074000	1.000000
max	28.110000	188.500000	2501.000000	0.345400	0.201200	1.000000

Figure 6: Extracting the Description of the dataset.

```
In [7]: #### Total count of Malignant(M) and Benign (B)
        df['diagnosis'].value_counts()
```

```
Out[7]: diagnosis
        B    357
        M    212
        Name: count, dtype: int64
```

Figure 7: displaying the Result of diagnosis

```
In [18]: def classification_model(model, data, predictors, outcome):

           #Fit the model:
           model.fit(data[predictors],data[outcome])

           #Make predictions on training set:
           predictions = model.predict(data[predictors])
           print(type(predictions))
         #   print("Precitions - {}".format(predictions == 1))
           #Print accuracy
           accuracy = metrics.accuracy_score(predictions,data[outcome])
           print("Accuracy : %s" % "{0:.3%}".format(accuracy))

           #Perform k-fold cross-validation with 5 folds
           kf = KFold(n_splits=5)
           kf.get_n_splits(data)
           error = []
           for train, test in kf.split(data):
             # Filter training data
             train_predictors = (data[predictors].iloc[train,:])

             # The target we're using to train the algorithm.
             train_target = data[outcome].iloc[train]
```

Figure 8: Model Development

4. Model Evaluation

4.1 Performance Metrics

4.1.1 Confusion Matrix

A confusion matrix provides a tabular framework for evaluating the prediction accuracy of a classification task. In this study, three classifiers with different performance outcomes were employed to forecast breast cancer. By utilizing the confusion matrix to compare each class to the others, we were able to determine which samples were improperly classified and so assess the classification accuracy. The functionality of these models is demonstrated by the confusion matrix in Figures 9 to 11.

Furthermore, a variety of assessment criteria were employed to ascertain the machine learning classifiers'

efficacy. The method for computing accuracy (AC), precision (P), and recall (R), among other metrics derived from the analysis of the confusion matrix, is presented in Table 2. The samples used to generate these metrics.

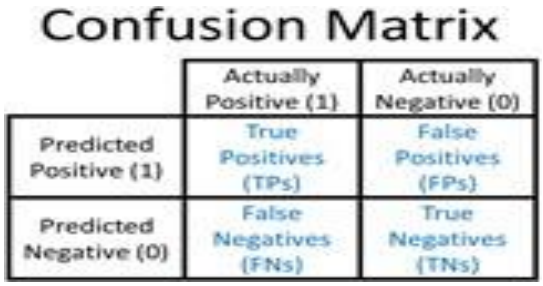

Fig 9: Confusion Matrix

Table 2: Performance Metrics

Accuracy (%)	**(TP + TN) / (TP + TN + FP + FN) × 100**
Precision (%)	**TP / (TP + FP) × 100**
Recall (%)	**TP / (TP + FN) × 100**
Error (%)	**FP + FN) / (TP + TN + FP + FN) × 100**

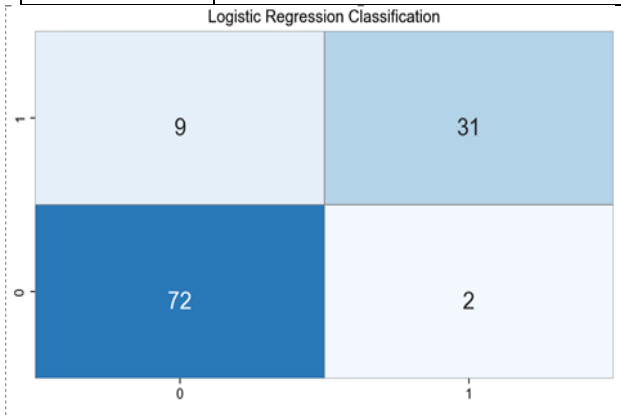

Figure 10. Confusion Matrix of LR

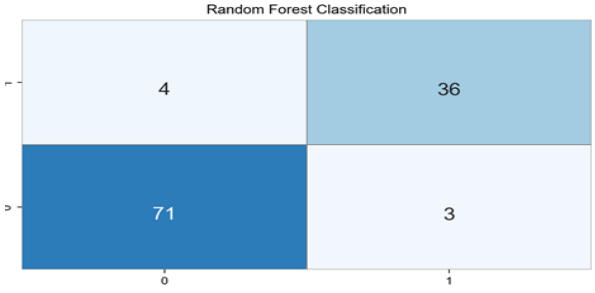

Figure 11. Confusion Matrix of RF Classifier

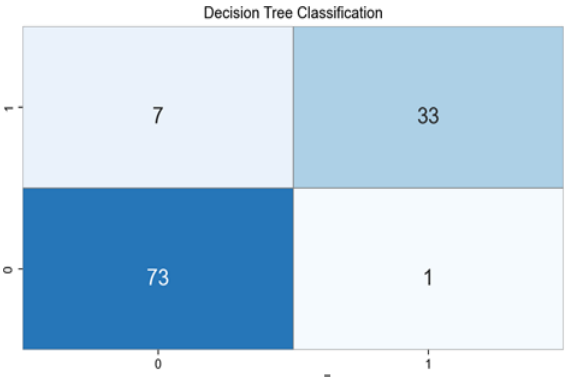

Figure 12: Confusion Matrix of Decision Tree

```
In [21]: # print(type(false_positive_rate1))
         # print(true_positive_rate1.size)
         roc_array = np.array([false_positive_rate1, true_positive_rate1])
         df_roc = pd.DataFrame(roc_array)
         df_roc_tr = df_roc.transpose()
         df_roc_tr.columns = ['False Positive Rate','True Positive Rate']
         df_roc_tr


         plt.subplots(1, figsize=(10,10))
         plt.title('ROC - Logistics Regression Model')
         plt.plot(false_positive_rate1,true_positive_rate1)
         plt.plot([0,1], ls="--")
         # plt.plot([0,0],[1,0], c=".7"),plt.plot([1,1], c=".7")
         plt.ylabel('True Positive Rate')
         plt.xlabel('False Positive Rate')
         plt.savefig("ROC-Logist.png")
         plt.show()
```

Figure 13: Drawing ROC Curve

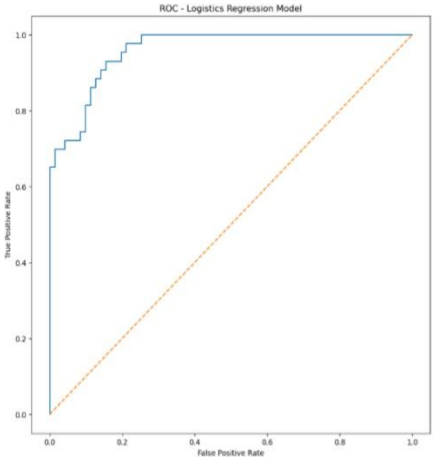

Figure 14: ROC curve for LR Model

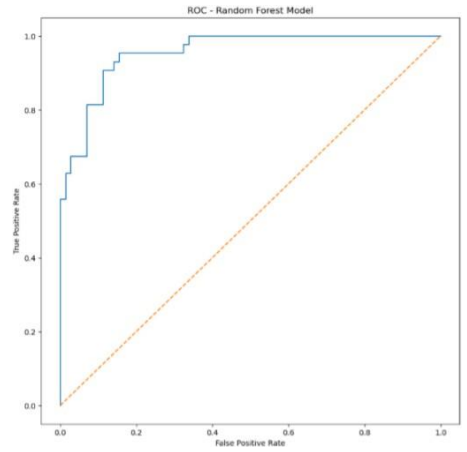

Figure 15: ROC Curve for RF Model

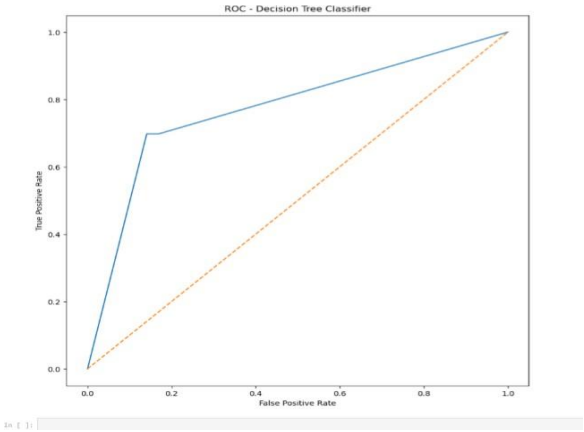

Figure 16: ROC Curve for Decision Tree model

5. Result And Discussion

Table 3: Performance of ML classifiers

S. No	ML Classifiers	Accuracy	Cross-Validation Score	Roc-Auc_Score
1	Logistic Regression	90.3 %	92.3 %	95.90 %
2	Random Forest	94.5 %	93.4 %	95.51 %
3	Decision Tree	97.14 %	89.01	77.41 %

Logistic Regression achieves an accuracy of 90.3%, a cross-validation score of 92.3%, and a ROC-AUC score of 95.90%. It performs consistently well across all metrics.

Random Forest demonstrates a higher accuracy of 94.5%, slightly lower than Decision Tree, but with a higher cross-validation score of 93.4% and a slightly lower ROC-AUC score of 95.51%. It showcases robust performance across all metrics.

Decision Tree shows the highest accuracy among the three at 97.14%. However, it has a lower cross-validation score of 89.01% and the lowest ROC-AUC score of 77.41%. While it excels in accuracy, its performance in generalization and distinguishing between classes might not be as strong as the other classifiers. In summary, while Decision Tree performs the best in terms of accuracy, Random Forest emerges as the most balanced classifier with consistently high performance across accuracy, cross-validation, and ROC-AUC scores. The AUC-ROC curve, which provides information about the effectiveness of our machine learning classifier, is shown in Figures 14, 15, and 16. In essence, ROC analysis evaluates a binary classifier's sensitivity and specificity. AUC (Area Under the Curve) is a statistic used to measure the overall performance of a binary classification model based on its ROC curve. The ROC curve is a representation of a probability curve. Greater AUC values show that the model is more effective at differentiating between classes. True Positive Rate (TPR) versus False Positive Rate (FPR) is plotted on the ROC curve, where an AUC of 1 denotes perfect discrimination and an AUC of 0 denotes no discrimination. Positive and negative class values can be reliably distinguished using a classifier with an AUC between 0.5 and 1.

6. Conclusion

Decision Tree achieves the highest accuracy at 97.14%, its lower cross-validation score of 89.01% and ROC-AUC score of 77.41% imply potential challenges with generalization and class separation. On the other hand, Logistic Regression demonstrates commendable overall performance, striking a balance between accuracy and generalization with an accuracy of 90.3%, a cross-validation score of 92.3%, and a ROC-AUC score of 95.90%. However, Random Forest surpasses both models with an accuracy of 94.5%, a cross-validation score of 93.4%, and a ROC-AUC score of 95.51%, indicating robust generalization and class separation capabilities. Thus, Random Forest emerges as the preferred choice due to its superior performance across all metrics, rendering it the most reliable classifier for the given dataset.

7. References

1. J. J. Cordova Calle, J. X. Farez Villa and R. I. Hurtado Ortiz, "An analysis method for predicting breast cancer using data science processes and machine learning,", IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC), Ixtapa, Mexico, 2022, pp. 1-4, doi: 10.1109/ROPEC55836.2022.10018755, 2022.
2. J. K. Sandhu, A. Kaur and C. Kaushal, "Analysis of Breast Cancer in Early Stage by UsingMachine LearningAlgorithms: A Review,", IEEE International Conference on Current Development in Engineering and Technology (CCET), Bhopal, India, 2022, pp. 1-7, doi: 10.1109/CCET56606.2022.10080757, 2022.
3. P. Singh and S. Agrawal, "Accuracy Prediction on Detection of Breast Cancer Using Machine Learning Classifiers," 14th International Conference on

Computational Intelligence and Communication Networks (CICN), Al-Khobar, Saudi Arabia,pp. 401-405, doi: 10.1109/CICN56167.2022.10008345, 2022.

4. R. K. Barwal and N. Raheja, "A Classification System for Breast Cancer Prediction using SVOF-KNN method,", International Conference on Augmented Intelligence and Sustainable Systems (ICAISS), Trichy, India, 2022, pp. 765-770, doi: 10.1109/ICAISS55157.2022.10010736, 2022.
5. M. Akhil, P. V. S. Kumar and K. Akhil, "Breast Cancer Prognosis using Machine Learning Techniques: A Literature Survey," ,4th International Conference on Advances in Computing, Communication Control and Networking (ICAC3N), Greater Noida, India, 2022, pp. 389-394, doi: 10.1109/ICAC3N56670.2022.10074163, 2022.
6. Ferlay, J., Colombet, M., Soerjomataram, I., Parkin, D.
7. M., Piñeros, M., Znaor, A., & Bray, F.,"Cancer statistics for the year 2020: An overview. International Journal of Cancer", 149(4), 778-789., 2021.
8. Habib Benlahmard , Rachida Ait Abdelouhahi, "Machine Learning Algorithms For Breast Cancer Prediction And Diagnosis",International Workshop on Edge IA-IoT for Smart Agriculture (SA2IOT) August 9-12, 2021.
9. Disha Harshadbhai Parekh, Vishal Dahiya, "Predicting Breast Cancer using Machine Learning Classifiers and Enhancing the Output by Combining the Predictions to Generate Optimal F1-Score", Biomedical and Biotechnology Research Journal (BBRJ), Volume 5, Issue 3, July-September 2021.
10. Dr. K. Mohan Kumar and N. Senthil Kumar, "Novel Approach to Find the Effective Supervised Machine Learning Algorithm Using Cancer Data Set", INTERNATIONAL JOURNAL OF SPECIAL EDUCATION Vol.37, No.3, 2022.