Comparative Study of Machine Learning Algorithms for Breast Cancer Detection

Shiva Kumar Dande
Electrical and Computer Engineering
Northeastern University
Boston, USA
dande.sh@northeastern.edu

Sai Prasasth Koundinya Gandrakota
Electrical and Computer Engineering
Northeastern University
Boston, USA
gandrakota.s@northeastern.edu

***Abstract* - In this research paper, we investigate the performance of three popular machine learning algorithms, namely Decision Tree, Random Forest and Naive Bayes, on a given dataset in order to predict the risk of Breast Cancer for a patient given various demographic and biological information. The primary objective of this study is to evaluate and compare the accuracy of these algorithms when applied to the dataset.**

I. Introduction

Breast cancer is a severe disease that affects a large portion of the global female population. Early detection is crucial for successful treatment, making the development of accurate and efficient diagnostic tools a crucial area of research. In this paper, we present a comparative study of three popular machine learning (ML) algorithms, namely Decision Tree, Random Forest, and Naive Bayes, for detecting breast cancer in a patient. We randomly split the dataset into training and testing sets to train and evaluate the algorithms. We first trained the Decision Tree algorithm on the training set and evaluated its accuracy on the test set. Next, we trained the Random Forest algorithm by creating an ensemble of Decision Trees and evaluated its performance. We also applied the Naive Bayes algorithm to the dataset and compared its accuracy with the Decision Tree and Random Forest algorithms. This study provides valuable insights into the comparative performance of machine learning algorithms for breast cancer detection. Our findings can serve as a guideline for medical practitioners and researchers who seek to develop efficient and accurate diagnostic tools for breast cancer detection using machine learning techniques

II. Methodology

The dataset consists of 569 breast cancer patient records, containing 30 various clinical and demographic features such as age, tumor size, and lymph node status. The target variable is binary indicating whether the tumor is Benign (B) or Malignant (M).

K-fold cross validation is then performed on the dataset, where in order to determine the optimal hyperparameters, we first split the data into training and testing folds. Taking each fold as a validation fold and the remaining as the training folds, the model is trained using the training folds and validated using the testing fold in order to determine to best hyperparameters for the model based on the mean accuracy of the predictions for all folds for each tested value of the hyperparameters.

Once the optimal hyperparameters have been identified, we then apply each algorithm again to the dataset, using the obtained hyperparameters and compute the accuracy of the predictions. The dataset is split into training and testing data again and the model is trained before validating the predicted targets of the test split of the dataset.

The accuracy of the predictions for each model is compared in order to determine the best approach for the problem statement. Below are the descriptions of the three algorithms used :

A. Decision Tree

Decision tree is a non-parametric algorithm that builds a hierarchical tree structure to represent the relationships between input features and target variables. The tree structure consists of internal nodes, which represent input features, and leaf nodes, which represent target variables or classes.

The algorithm recursively partitions the input space into smaller regions based on the values of input features. It uses a top-down approach called recursive binary splitting to find the best feature and split point at each internal node that maximizes the information gain or minimizes the impurity measure. The information gain measures the reduction in entropy or Gini index of the target variable after splitting, whereas the impurity measure quantifies the degree of class impurity or uncertainty at a node.

Decision Tree is preferred for its interpretability, scalability, and ability to handle missing values and non-linear relationships. The downside of the algorithm is that it is prone to overfitting and instability, especially when the tree becomes too deep and complex. To overcome these issues, various techniques, such as pruning, ensemble methods, and regularization, have been proposed.

B. Random Forest

Random Forest is a machine learning algorithm that combines multiple decision trees to improve the accuracy and robustness of predictions. It is a type of ensemble learning method that uses bagging, also known as bootstrap aggregating, to create multiple decision trees from different subsets of the training data.

The algorithm builds multiple decision trees independently, where each tree is trained on a random subset of the training data and a random subset of the input features. The randomness in feature selection and data sampling helps to reduce the variance and correlation between the trees, which leads to better generalization and robustness. The final prediction is then made by aggregating the predictions of all the trees, either by taking a majority vote or a weighted average.

Random Forest is known for its high accuracy, scalability, robustness to noise and outliers, and ability to handle high-

dimensional and non-linear data. It also provides estimates of feature importance, which can be useful for feature selection and interpretation.

Random Forest also has limitations, such as increased computational and storage requirements, reduced interpretability compared to a single decision tree, and the possibility of overfitting when the number of trees is too large or the data is too noisy or imbalanced.

C. Naive Bayes

Naive Bayes is based on Bayes' theorem, which is a probabilistic approach that calculates the probability of a hypothesis or event given some evidence or observations.

The algorithm assumes that the input features are conditionally independent given the class label, which allows for the efficient and accurate estimation of the joint probability distribution of the input features and the class labels.

The algorithm first calculates the prior probability of each class label and then calculates the conditional probability of each input feature given each class label.

To make a prediction for a new input instance, the algorithm calculates the posterior probability of each class label given the input features, using Bayes' theorem and the estimated probabilities. It then outputs the class label with the highest posterior probability as the prediction.

The advantages of the Naive Bayes algorithm are its simplicity, efficiency, and robustness to irrelevant features and missing data. It also provides interpretable probabilities and can handle both discrete and continuous input features.

However, Naive Bayes also has some limitations, such as the assumption of conditional independence, which may not hold in many real-world problems, and the sensitivity to the choice of prior probabilities and the quality of the training data.

III. Experiments And Results

Each model was trained and tested using the same dataset i.e., the breast cancer records dataset. For the K-fold cross validation K = 10 folds is used where a 90%-10% split of the dataset into training and testing folds respectively.

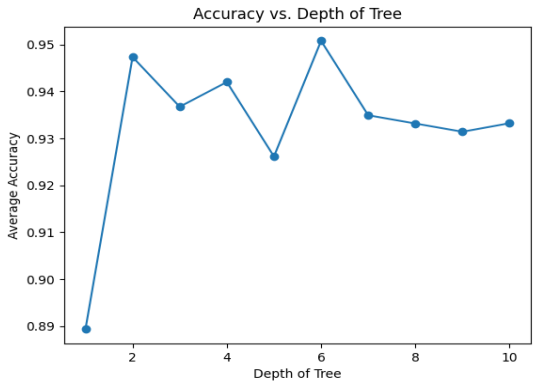
For all algorithms, after the optimal hyperparameters are obtained, the data is split on a 80%-20% basis (80% for training and 20% for testing). This split gives us 455 training samples and 114 validation samples.

A. Decision Tree

The maximum depth of the decision tree was varied between 1 - 10, and after 10-fold cross validation was performed, the average accuracy across all folds for each depth value was obtained and the value corresponding to the highest accuracy was selected as the optimal value for maximum depth.

As seen below, the optimum maximum depth selected was 6 as it pertained to the highest accuracy obtained after 10-fold cross validation.

The accuracy is seen to be increasing initially as the depth of the tree increases, however as we cross the optimal value the accuracy starts to decrease again.



Using the maximum depth as 6, the model is then tested on the entire dataset to obtain the final accuracy and confusion matrix:

ACCURACY

89.47 %

CONFUSION MATRIX

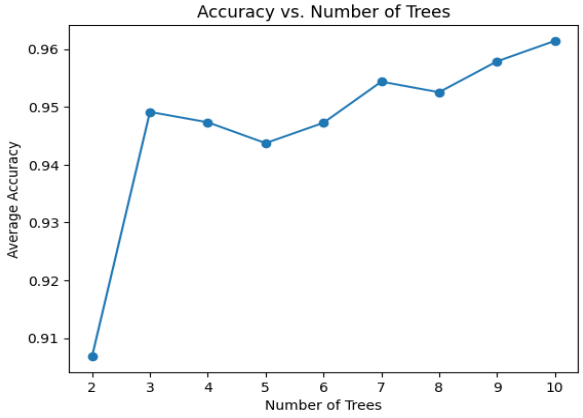
36	**4**
8	**66**

B. Random Forest

The number of decision tress was varied between 2 - 10, and using the optimal maximum depth obtained previously (6), 10-fold cross validation was performed for each number of trees and the average accuracy across all folds for each value was obtained and the value corresponding to the highest accuracy was selected as the optimal value for number of trees.

As seen below, the optimum number of trees selected was 10 as it pertained to the highest accuracy obtained after 10-fold cross validation.

The accuracy is seen to be increasing initially as the number of trees increases, especially between 2 and 3, however as we further increase the number of trees the accuracy starts to fluctuate based on the data set. The number of trees with the highest accuracy is selected.



Using the number of trees as 10, the model is then tested on the entire dataset to obtain the final accuracy and confusion matrix:

ACCURACY

95.61 %

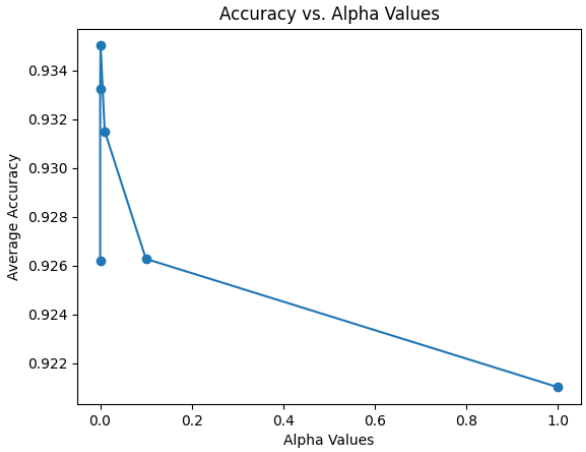
CONFUSION MATRIX

41	**3**
2	**68**

C. *Naive Bayes*

In this project a Gaussian Naive Bayes classifier was implemented, which includes a smoothing parameter (alpha) in order to avoid zero probabilities (which it achieves by adding a small value to each feature's variance).

The values of alpha were created using the numpy.linspace() function and the following values were obtained to test : 0.00001, 0.0001, 0.001, 0.01, 0.1, and 1. After performing 10-fold cross validation, we obtain the optimal value for alpha as 0.001.



Using the optimum alpha value, the model is then tested on the entire dataset to obtain the final accuracy and confusion matrix:

ACCURACY

96.47 %

CONFUSION MATRIX

39	**4**
0	**71**

IV. Conclusion

Our results demonstrate that all three algorithms achieve high accuracy rates for breast cancer detection. The Naive Bayes algorithm outperformed the others, achieving an accuracy rate of 96.5%. The Random Forest algorithm was slightly behind at 95.6% while the Decision Tree algorithm achieved an accuracy rate of 89.5%. The size of the dataset affects this comparison as the Random Forest algorithm would perform better when applied to larger, more complex datasets.

In conclusion, this study provides valuable insights into the comparative performance of machine learning algorithms for breast cancer detection. Our findings can serve as a guideline for medical practitioners and researchers who seek to develop efficient and accurate diagnostic tools for breast cancer detection using machine learning algorithms.

References

[1] Dana Bazazeh; Raed Shubair, *Comparative study of machine learning algorithms for breast cancer detection and diagnosis*, 2016.

[2] Prateek P. Sengar; Mihir J. Gaikwad; Ashlesha S. Nagdive, *Comparative Study of Machine Learning Algorithms for Breast Cancer Prediction*, 2020.

[3] Kevin P. Murphy, *Machine Learning: A Probabilistic Perspective*, MIT Press 2012.

[4] Christopher M. Bishop, *Pattern Recognition and Machine Learning*, Springer 2006.

[5] R. O. Duda, P. E. Hart, D. Stork, *Pattern Classification, 2nd Ed*, Wiley and Sons, 2001

[6] T. Hastie, R. Tibshirani, J. H. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer, 2001.

Appendix

https://github.com/ShivaKumarDande/MachineLearning_Algorithms