Original Scientific Article Journal of Decision Analytics and Intelligent Computing
Vol. 4 issue 1, (2023) 32-46 https://doi.org/10.31181/jdaic10028012024c

An efficient approach for breast cancer classification using machine learning

Vedatrayee Chatterjee[1,*], Arnab Maitra[1], Soubhik Ghosh[1], Hritik Banerjee[1], Subhadeep Puitandi[1] and Ankita Mukherjee[1]

[1] Department of Computer Science & Engineering, Asansol Engineering College, Asansol, India

* Correspondence: vedatrayeechatterjee@gmail.com

Received 25 June 2023
Accepted for publication 21 January 2024
Published 28 January 2024

Abstract

Breast cancer, a life-threatening disease affecting millions worldwide, poses significant challenges due to its time-consuming manual determination process, potential risks, and human errors. It is a condition where cells of the breast develop unnaturally and uncontrollably, resulting in a mass called a tumor. If lumps in the breast are not addressed, they can spread to other regions of the body, including the bones, liver, and lungs.
Early diagnosis is crucial for effective treatment and improved patient outcomes. In this research paper, we focus on employing machine learning models to achieve quick identification of breast cancer tumors as benign or malignant. The primary objective is to develop a decision-making visualization pattern using swarm plots and heat maps. To accomplish this, we utilized the Light GBM (Gradient Boosting Machine) algorithm and compared its performance against other established machine learning models, namely Logistic Regression, Gradient Boosting Algorithm, Random Forest Algorithm, and XG Boost Algorithm. Ultimately, our study demonstrates that the Light GBM Algorithm exhibits the highest accuracy of 96.98% in distinguishing between benign and malignant breast tumors.

Keywords: Breast Cancer, Dataset, Machine Learning, Gradient Boosting Algorithm, Random Forest Algorithm.

1. Introduction

Today, Breast cancer is affecting individuals, particularly women. According to the World Health Organization (WHO). It's a leading cause of female mortality. Around a million women succumb to breast cancer annually (Simon et al., 2021) with India's fatality rate at 13.92%. The prevalence is higher in Australia, Europe and the US, while Malaysia observes later-stage presentations (Vyas et al., 2022). Regular screening is vital due to asymptomatic cases. Early detection aids treatment and survival. Contributing factors include family history, obesity, radiation exposure, and genetics.

Recently discovered, breast cancer is categorized as malignant or benign. Analyzing tumor characteristics helps differentiate them. Benign tumors are low-risk, while malignant ones spread to neighboring tissues and

the body. Artificial Intelligence (AI) is being employed to classify breast cancer. AI algorithms train on datasets to label tumors as 1 for benign or 0 for malignant (Bardou et al., 2018).

The primary objective of this research is to establish a method for early tumor diagnosis, since the conventional diagnostic approach can yield false positives, leading to unnecessary procedures.

1.1 Problem Definition

Currently, India reports approximately 178,000 cases of breast cancer. However, manually determining cancer in these cases is an arduous and time-consuming process, often leading to delays and the possibility of human errors. To address this issue, we aim to develop a predictive model that can efficiently classify breast tumors as either malignant or benign using Machine Learning techniques. Our approach involves analyzing the correlation between various features, eliminating redundant data, and ultimately creating a highly accurate model. By leveraging these advanced technologies, we strive to enhance the early detection and diagnosis of breast cancer, which can significantly improve patient outcomes.

1.2 Objectives

The initial aim of this study is to examine breast cancer data derived from a diagnostic dataset comprising 30 feature columns and approximately 570 rows. The primary goal is to identify common characteristics in these groups that distinguish benign cases from malignant ones effectively. Subsequently, we plan to generate a heatmap visualization to identify and eliminate redundant features from the dataset. Finally, our ultimate objective is to create a machine learning model that enables users to classify breast cancer cases as either benign or malignant accurately. By accomplishing these objectives, we hope to enhance the diagnostic process and contribute to more efficient and precise breast cancer classification.

1.3 Scope

Our project aims to address challenges and propose solutions to enhance accuracy in breast cancer classification. Accuracy is a critical factor, as an imprecise model can lead to suboptimal outcomes. The research primarily centers around improving the accuracy of various algorithms, namely Logistic Regression, Gradient Boosting Algorithm, Random Forest Algorithm (Octaviani and Rustam, 2019), XG Boost Algorithm, and Light GBM Algorithm. The objective is to achieve the highest possible accuracy for the model by fine-tuning and optimizing these algorithms. By tackling accuracy-related issues, we aspire to provide more reliable and effective breast cancer classification results.

2. Machine Learning Algorithms Used For Breast Cancer Prediction

2.1 Filter Method

The filter method is a prominent approach for feature selection. Filter methods determine feature relevance by employing statistical metrics prior to model training. This process streamlines feature selection, enhancing the efficiency of the machine learning pipeline. It involves steps like computing scores for each feature based on metrics capturing their relationship with the target variable, ranking features according to these scores, and setting a threshold to retain or discard features. Notably, filter methods assess feature relevance independently, making them computationally efficient and suitable for large datasets. These methods are model-agnostic, allowing their application across diverse problems and data types. Common scoring metrics include correlation coefficients, mutual information, and variance thresholding (Asri et al., 2016).

2.2 Wrapper Method

The wrapper method stands as an advanced and dynamic technique for feature selection within the realm of machine learning. Unlike filter methods that employ statistical measures, the wrapper method takes a more comprehensive approach by iteratively training and evaluating machine learning models using various feature subsets. This technique involves generating subsets, training models, and evaluating performance for each subset, guided by a chosen performance metric. The subset yielding the best model performance is then selected. Unlike filter methods, the wrapper method considers feature interactions, enabling it to capture complex relationships in the data. Its model-centric approach aligns with the ultimate goal of achieving superior predictive accuracy. However, it does come with computational costs due to its repeated model training and evaluation, and careful handling is required to avoid overfitting. The wrapper method's ability to fine-tune feature selection for optimal model performance makes it a valuable asset in situations where precision is paramount (Asri et al., 2016).

2.3 Embedded Method

The embedded method represents a dynamic and sophisticated approach to feature selection in the realm of machine learning. Unlike filter methods that analyze features prior to model training or wrapper methods that evaluate features independently, the embedded method seamlessly integrates feature selection within the model building process. This method takes advantage of algorithms that inherently assess feature importance while learning from the data. As the model iteratively refines its parameters, it simultaneously adjusts the relevance of features, automatically assigning higher importance to those that significantly contribute to its performance. Algorithms like Lasso Regression, Decision Trees, Random Forests, Gradient Boosting, and certain Neural Networks exemplify embedded methods by either penalizing irrelevant features or calculating feature importance scores. This approach leads to efficient feature selection, insights into feature impact, and often prevents overfitting by penalizing unnecessary attributes. However, it's crucial to note that the applicability of embedded methods is tied to the specific algorithm chosen and its inherent feature selection capabilities. In essence, embedded methods strike a balance between filter methods' efficiency and wrapper methods' performance optimization, yielding both accurate and interpretable models (Joshi and Mehta, 2017).

2.4 Recursive Feature Elimination (RFE)

Recursive Feature Elimination (RFE) operates as a wrapper-style feature selection technique that incorporates a distinct machine learning algorithm at its core. Unlike filter-based methods that score individual features, RFE iteratively refines the feature set. It commences with all features from the training dataset and progressively prunes them until the desired count is achieved. This iterative procedure involves leveraging the designated machine learning algorithm to assess feature importance, discarding the least relevant features, and subsequently retraining the model.

The RFE process continues iteratively, with features being removed step by step, until the specified target number of features remains in the selected subset. This interplay between the wrapper-style approach and the internal utilization of filter-based feature ranking enables RFE to effectively identify a subset of features that optimally contribute to the model's performance (Joshi and Mehta, 2017; Nahid and Kong, 2017).

2.5 Segmentation

The process of segmenting images into patches of varying dimensions, such as 2x2, 3x3, up to 10x10, is referred to as image segmentation. Within this segmentation process, the goal is to train the system to recognize contiguous regions of interest that hold significance for detecting abnormalities like breast cancer. By

eliminating extraneous information from the image, the identification of tumors at an early stage becomes more feasible.

The K-means clustering algorithm plays a pivotal role in this segmentation endeavor. Operating by grouping similar objects together, K-means clustering aids in the creation of distinct clusters. In the context of image segmentation, it assists in aggregating pixels or patches that exhibit likeness. This approach yields promising outcomes, especially when comparable objects are present within a single cluster. Notably, the algorithm's efficiency shines through when dealing with data that is densely located as opposed to being scattered.

Ultimately, this segmentation process, empowered by K-means clustering, facilitates the rapid identification of important regions within images, enhancing the accuracy of early tumor detection (Joshi and Mehta, 2017; Nahid and Kong, 2017).

2.6 Support Vector Machine (SVM)

The primary goal of the support vector machine (SVM) algorithm is to identify a hyperplane within an N-dimensional space, where N represents the number of features, that effectively segregates data points into distinct classes. Given a set of data points with different classes, numerous potential hyperplanes can be considered to separate them. The key objective is to identify a hyperplane that exhibits the maximum margin, indicating the greatest distance between data points of the two classes.

Hyperplanes function as decision boundaries that aid in classifying data points. By categorizing data points on either side of the hyperplane, distinct classes can be assigned. Furthermore, the dimensionality of the hyperplane corresponds to the number of features present.

Support vectors denote data points positioned in close proximity to the hyperplane, significantly influencing its position and orientation. Leveraging these support vectors, the algorithm strives to maximize the margin of separation between the two classes. Notably, altering or removing support vectors would lead to a shift in the hyperplane's position.

SVM stands out as a powerful classifier, particularly when there exists a well-defined separation margin and the data features are high-dimensional. However, its suitability diminishes when handling large datasets due to the extended training time required. Additionally, SVM's performance deteriorates in scenarios where the dataset is tainted with significant levels of noise (Joshi and Mehta, 2017).

2.7 K-Nearest Neighbor (KNN)

K-Nearest Neighbor (KNN) stands out as one of the elementary Machine Learning algorithms, grounded in the principles of Supervised Learning. The KNN algorithm operates on the premise of likening the new, incoming data to the pre-existing instances and then assigning the new data point to the category most akin to the established categories. In essence, KNN taps into the reservoir of stored data and uses similarity to guide its classification process. This method works particularly well when the data is structured and categorically organized.

The operational logic of KNN revolves around identifying data points within the dataset that closely align with the new data point under consideration. The algorithm evaluates the distances between these points, sorting them based on proximity to the target point. The measurement of distance is typically conducted using various methodologies, with the Euclidean distance being a widely favored choice among experts.

The subsequent step involves selecting a specific count of neighboring points whose distances are minimal in relation to other points. These chosen points, often referred to as "neighbours," play a pivotal role in the classification process. Notably, the selection of these neighbours is typically based on an odd number, which aligns with the number of classes present in the problem. For instance, in a binary classification task, the highest count of points from one class will be taken as the basis for classification.

KNN is admired for its simplicity in implementation and its ability to manage substantial datasets. However, it comes with a trade-off: the computational cost can be high due to the need to calculate distances across the entire training set. Additionally, the choice of the parameter 'K' – the number of neighbors to consider – can influence the algorithm's complexity and, subsequently, its performance (Joshi and Mehta, 2017).

2.8 Random Forest

Random Forest is a type of supervised learning algorithm that operates by amalgamating multiple Decision Trees. A Decision Tree is structured hierarchically, with nodes representing specific conditions based on a set of features. The branches within the tree steer the decision-making process towards the leaf nodes, which ultimately determine the class labels of the data instances. Decision Trees can be generated using either Recursive Partitioning or Conditional Inference Tree methods.

Recursive Partitioning involves constructing a Decision Tree incrementally by deliberating whether to split each node further. This process entails partitioning the source dataset into subsets through attribute value tests. The recursion continues until a node's subset contains only instances with identical target variable values.

In contrast, the Conditional Inference Tree approach is grounded in statistical principles. It employs non-parametric tests as criteria for splitting nodes, with corrections for multiple testing to mitigate the risk of overfitting.

Random Forest is particularly well-suited for modeling high-dimensional data due to its capacity to handle diverse data types such as missing values, continuous variables, as well as categorical and binary data. However, it's worth noting that when working with very large datasets, the memory usage can increase due to the size of the trees generated. One common challenge is the potential for overfitting, which emphasizes the need to fine-tune the algorithm's hyper parameters to achieve optimal performance (Joshi and Mehta, 2017; Asri et al., 2016).

2.9 Logistic Regression

In the realm of linear regression, the derived hyperplane is insufficient for predicting the dependent variable through the independent variables. This limitation becomes pronounced, especially when grappling with categorical data. This is where logistic regression steps in. Logistic regression shines when categorical data is in play, diverging from linear regression's approach of predicting continuous outcomes. Instead, logistic regression tackles the task of discerning the truth or falsehood of an assertion, effectively delving into classification problems.

Unlike its linear counterpart, logistic regression leverages the sigmoid function to transform the independent variable into a probability expression bounded within the range of 0 to 1 concerning the dependent variable. This probabilistic nature equips logistic regression to not only offer probabilities but also to adeptly categorize new samples based on a combination of continuous and discrete measurements. This versatility renders logistic regression a sought-after algorithm in the realm of Machine Learning.

However, it's worth noting that logistic regression does carry a limitation in the form of its assumption regarding the linearity between the dependent and independent variables. This assumption can pose challenges when dealing with datasets that don't conform to linear relationships, potentially affecting the algorithm's predictive accuracy (Joshi and Mehta, 2017).

2.10 Decision Tree

Decision Trees (DT) serve a dual purpose, finding utility in both classification and regression tasks. This versatile algorithm employs a tree structure characterized by two fundamental node types: the decision node

and the leaf node. The decision node embodies a test that guides the traversal of the tree, while the ultimate classification or prediction occurs within the leaf node.

In essence, Decision Trees excel at capturing intricate decision-making processes in a highly interpretable format. This makes them valuable tools not only for classifying data points but also for predicting continuous outcomes through regression. By effectively compartmentalizing decisions into decision nodes and outcomes into leaf nodes, Decision Trees offer an intuitive approach to solving a wide array of predictive tasks (Vyas et al., 2022).

2.11 Ensemble Methods

Ensemble learning stands as a potent Machine Learning technique where multiple distinct models are trained to collectively address a shared problem, ultimately yielding enhanced outcomes. The core principle underlying ensemble learning is the belief that by effectively amalgamating weak models, we can achieve heightened accuracy and robustness in our predictions.

The underlying concept is grounded in the notion that the amalgamation of individual models can lead to a final model that not only outperforms the individual constituents but also demonstrates a greater capacity to withstand variations and uncertainties in the data. This collaborative approach capitalizes on the strengths of each individual model, creating a formidable ensemble that contributes to more accurate and reliable predictions (Vyas et al., 2022; Simon et al., 2021).

2.12 Bagging

During the process of model preparation, whether dealing with a classification or regression task, we construct a function that takes input data and produces an output result. This function is devised based on the characteristics of the training dataset. It's imperative to acknowledge that due to the inherent variability present within the training dataset (reflecting observed instances drawn from an underlying, unknown distribution), the resultant fitted model is also subject to this variability. If an entirely different dataset had been observed, the model generated would indeed diverge, showcasing a distinct configuration and behavior (Vyas et al., 2022).

2.13 Naïve Bayes

The Naïve Bayes classifier stands as a prominent supervised learning algorithm designed for classification tasks. Its foundation rests upon the Bayes theorem, a probabilistic formula that determines the probability of an event occurring given that another event has already taken place. As a versatile tool, Naïve Bayes finds extensive use across industries due to its simplicity and efficacy in various Machine Learning applications.

Naïve Bayes operates on the assumption of feature independence, a premise that, while simplifying computations, often deviates from the intricate interdependencies present in real-world scenarios. This oversimplification can restrict the algorithm's practicality across complex use cases.

An inherent challenge faced by this algorithm is the 'zero-frequency problem,' wherein it assigns zero probability to a categorical variable category that wasn't encountered during training. This limitation can be addressed by employing smoothing techniques that introduce minimal probabilities to unseen categories, enhancing the algorithm's robustness.

Another noteworthy limitation of Naïve Bayes is its reliance on sizable datasets to achieve peak accuracy. The algorithm's performance tends to flourish with an ample volume of data, allowing it to draw more reliable conclusions and make accurate predictions.

In summation, while Naïve Bayes offers a straightforward and potent approach to classification, its success hinges on acknowledging its assumptions and understanding its limitations. By addressing challenges such as the

'zero-frequency problem' and considering its applicability to specific data scenarios, practitioners can harness its capabilities effectively (Nahid and Kong, 2017).

2.14 Boosting

Boosting strategies operate with a similar underlying principle as ensemble techniques: they assemble an ensemble of models, aimed at collectively achieving the performance of a strong, superior learner. However, in contrast to the focus of ensembling on variance reduction, boosting adopts a distinct approach. It dynamically binds multiple weak learners together in an adaptive fashion, where each individual model in the ensemble receives more attention in areas where previous models were lacking efficacy.

Each newly introduced model concentrates its efforts on challenging instances, progressively refining the overall ensemble's performance. This iterative process culminates in the creation of a robust learner with reduced bias, even though boosting can also incidentally contribute to variance reduction.

Boosting is not only applicable to classification but also extends to regression tasks. While primarily geared towards mitigating bias, boosting favors base models that possess low variance and high bias. Adaboost and Gradient Boosting stand as two significant algorithms within the realm of boosting. These methodologies diverge in their approach to creating and combining the weak learners throughout the iterative process (Asri et al., 2016).

2.14.1 Adaptive / Adaboost Boosting

Adaptive boosting (Adaboost) updates the weights attached to each training data point, whereas Gradient Boosting modifies the values of these points. This fundamental distinction arises from their respective approaches to addressing the optimization problem of approximating a composite model with weighted combinations of weak learners (Vyas et al., 2022; Asri et al., 2016; Derangula et al., 2021).

2.14.2 Gradient Boosting

Gradient boosting represents a state-of-the-art predictive approach that iteratively tackles a complex optimization challenge, yielding a model expressed as a linear combination of fundamental predictors—typically in the form of decision trees. This innovative methodology orchestrates the assembly of a predictive model and subsequently refines its generality by facilitating optimization for a wide range of differentiable loss functions. Employing a gradient descent algorithm, this technique systematically reduces the loss by introducing new decision trees. Notably, gradient boosting proves its efficacy in resolving predictive modeling intricacies across both regression and classification tasks.

The foundation of Gradient Boosting (GB) lies in the notion that the most promising subsequent model, when integrated with existing models, serves to minimize overall predictive errors. This approach leverages the insight that integrating prior model outcomes into the construction of new models contributes to error reduction. For instance, the GB framework accommodates optimization across diverse loss functions and offers a spectrum of hyperparameter adjustments, endowing the fitting process with a high degree of adaptability and flexibility (Bazazeh and Shubair, 2016; Hassan et al., 2023).

2.14.2.1 Extreme Gradient Boosting (XGBoost / XGB)

XGBoost (XGB) stands as a gradient boosting framework rooted in ensemble machine learning strategies that leverage decision trees. This technique's swiftness in processing and capacity for scalability suggest its potential in yielding highly valuable insights. XGB finds utility across both regression and classification tasks. At its core, this methodology aims to achieve precise classification by progressively identifying weak algorithms. It employs the gradient descent technique to craft individualized decision trees, initially establishing sets of threshold values that undergo iterative refinement via the minimization of residuals during tree construction. Within the

context of gradient boosting, weak learners manifest as regression trees, with each tree mapping an input dataset to specific leaves bearing continuous markers. The process entails minimizing a regularized function (encompassing L1 and L2 norms), characterized by a convex loss function rooted in the disparity between predictions and target outputs. The training process incrementally introduces new trees to predict the residuals or errors of preceding trees. These predictions are then amalgamated with the outputs of prior trees to ultimately generate the final prediction (Bazazeh and Shubair, 2016; Hassan et al., 2023).

2.14.2.2 Light Gradient Boosting (Light GBM)

A novel approach in breast cancer detection has been introduced utilizing the Light Gradient Boost machine learning technique. This innovative method aims to transform initially weak learners into robust ones, thereby achieving enhanced accuracy in breast cancer detection.

Unlike the conventional employment of weak learners as standalone classifiers, this technique leverages a boosting ensemble to achieve heightened classification accuracy.

In this approach, the weak learners are harnessed as classifiers, which alone may not yield optimal classification accuracy. However, the concept of a strong learner emerges through the ensemble of these weak classifiers. This ensemble-based boosting technique is rooted in tree-based classification. Notably, the Light Gradient Boost machine learning technique molds the decision tree classifier into a unique weak learner structure, characterized by a vertical orientation. This innovative design, termed the "Leaf-wise Decision Tree Algorithm," showcases its distinctiveness in minimizing training loss compared to alternative algorithms.

Through these advancements, the Light Gradient Boost technique demonstrates its potential to significantly improve breast cancer detection accuracy, thus offering promising avenues for enhanced medical diagnostics.

3. Methodology

The research methodology aims to discern the disparities between benign and malignant breast cancer cases. Initially, breast cancer data is gathered from a diagnostic dataset. The dataset is then preprocessed, and any missing values are handled by removal. Next, we utilize swarm plots to visualize and compare the features, assessing if there are distinct patterns between benign and malignant cases. Outliers in the features are identified and removed to ensure data integrity.

Following the outlier removal, the preprocessed data is split into training and testing datasets. We proceed to train the data using various machine learning models such as Logistic Regression (Sultana and Jilani, 2018), Random Forest Algorithm, XG Boost Algorithm, and Light GBM Algorithm. The objective is to identify the model that yields the highest accuracy. Finally, based on the best-performing model, we construct a predictive system to effectively classify breast cancer cases as either benign or malignant. This methodology allows us to gain valuable insights into the characteristics that differentiate these types of cancer and create a robust predictive tool to aid in accurate diagnosis.

3.1 Breast Cancer Dataset

For this research, we utilized a diagnostic dataset containing 569 rows and 30 columns. These 30 parameters were chosen as the basis for our analysis. These attributes play a vital role in producing visualization patterns, making it easier to generate heat maps for feature visualization.

3.2 Data Cleaning Procedure

Once the dataset is imported using the Panda’s library, it becomes crucial to check for the presence of any missing values. The data cleaning process involves eliminating entire rows that contain any missing values. This

step ensures that subsequent tasks, such as visualization, can be carried out effectively with high accuracy. Heat maps are then employed to identify and remove outliers, further enhancing the accuracy of the analysis.

4. Results and Discussion

The results demonstrate that the Light GBM algorithm is the most suitable for classifying breast cancer as either benign or malignant, achieving an impressive accuracy score of 96.98%. To determine the best accuracy for the model, we used various machine learning models, including Logistic Regression, Gradient Boosting, Random Forest, XG Boost, and Light GBM Algorithm (Derangula et al., 2021).

Figure 1 presents a count plot illustrating the distribution of benign and malignant cases in the dataset.

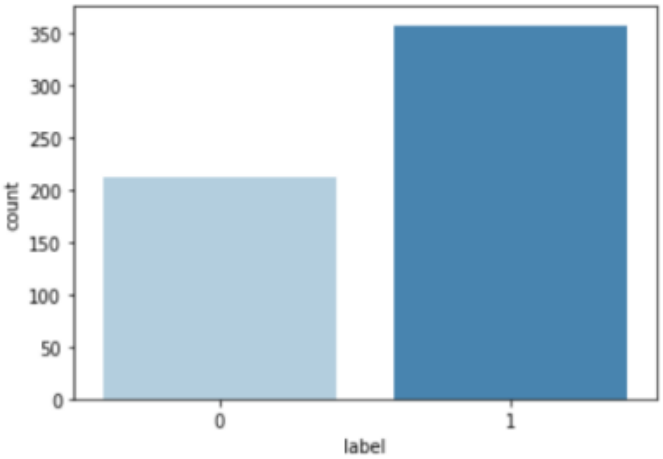

Figure 1. Count chart for benign (1) and malignant (0)

Swarm plot graphs were generated to visualize the relationship between the first 5 features (mean area, mean radius, mean texture, mean perimeter, and mean smoothness) out of the total 30 features, assessing their correlations (Figure 2).

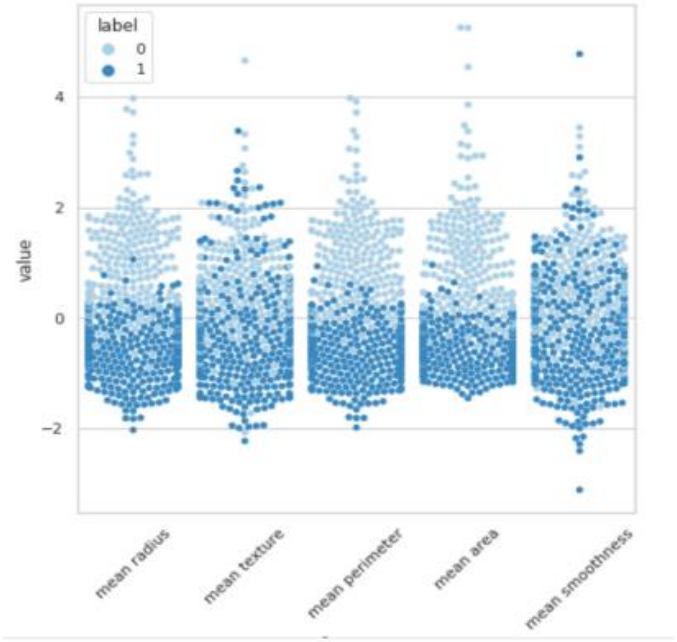

Figure 2. Swarm plot for 5 features of dataset

Similarly, swarm plot graphs visualize features 5 to 10, including mean fractal dimension, mean compactness, mean concavity, mean concave points, and mean symmetry, to explore their relationships (Figure 3).

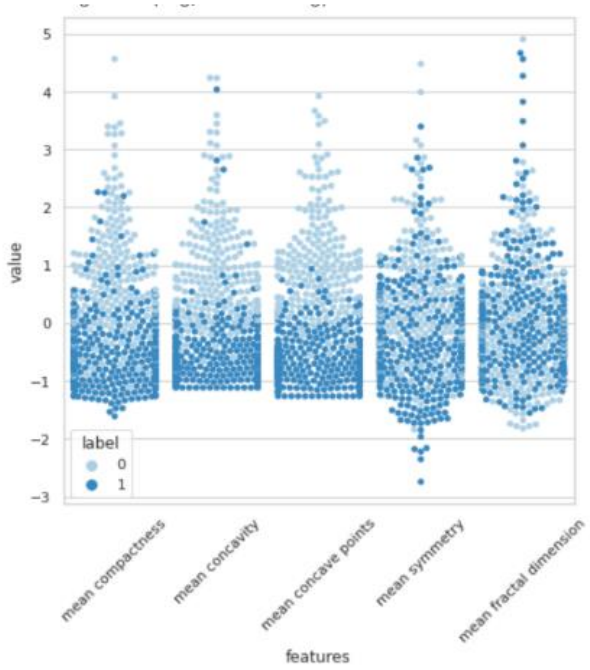

Figure 3. Swarm plot for 5 - 10 features of dataset

Furthermore, swarm plot graphs visualize features 10 to 15, namely texture error, radius error, smoothness error, perimeter error, and area error, to assess their correlations (Figure 4).

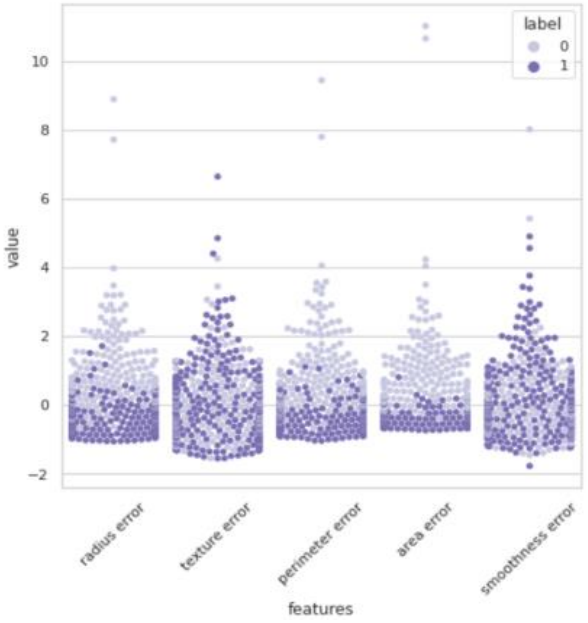

Figure 4. Swarm plot for 10 - 15 features of dataset

Additionally, swarm plot graphs visualize features 15 to 20, such as symmetry error, compactness error, concavity error, fractal dimension error, and concave points error, to understand their relationships (Figure 5).

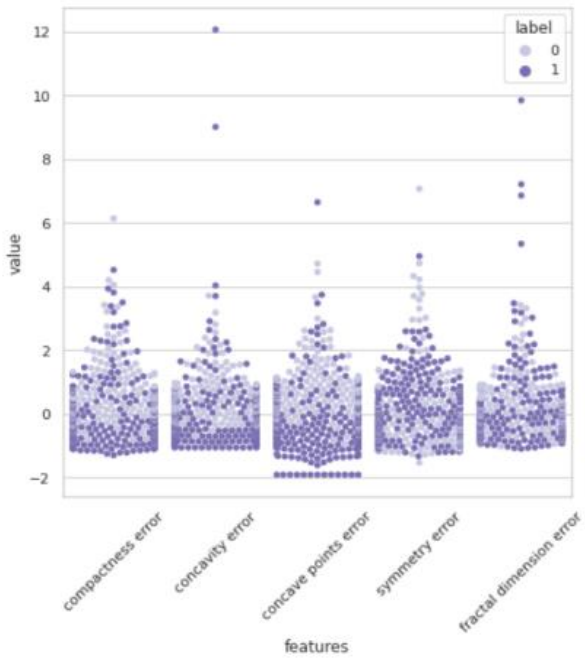

Figure 5. Swarm plot for 15 - 20 features of dataset

Moreover, swarm plot graphs visualize features 20 to 25, including worst radius, worst texture, worst perimeter, worst area, and worst smoothness, to explore their correlations (Figure 6).

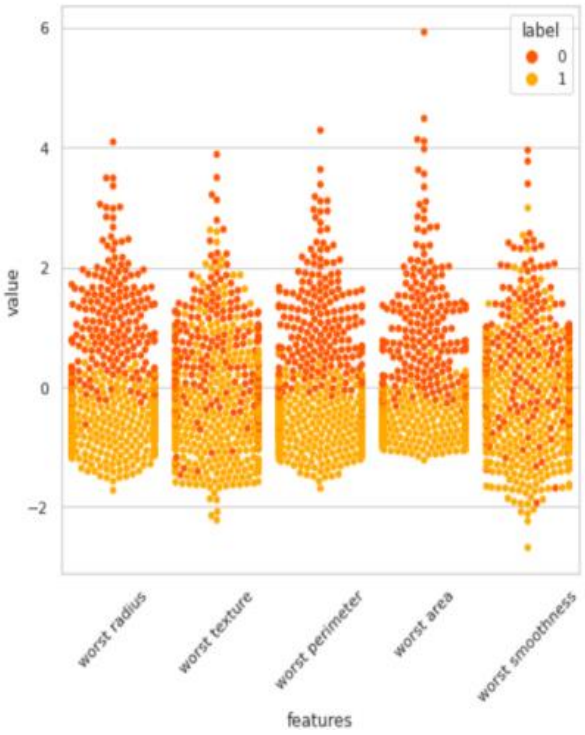

Figure 6. Swarm plot 20 - 25 features of dataset

Lastly, swarm plot graphs visualize features 25 to 30, consisting of worst compactness, worst concavity, worst symmetry, worst fractal dimension, and worst concave points, to assess their relationships (Figure 7).

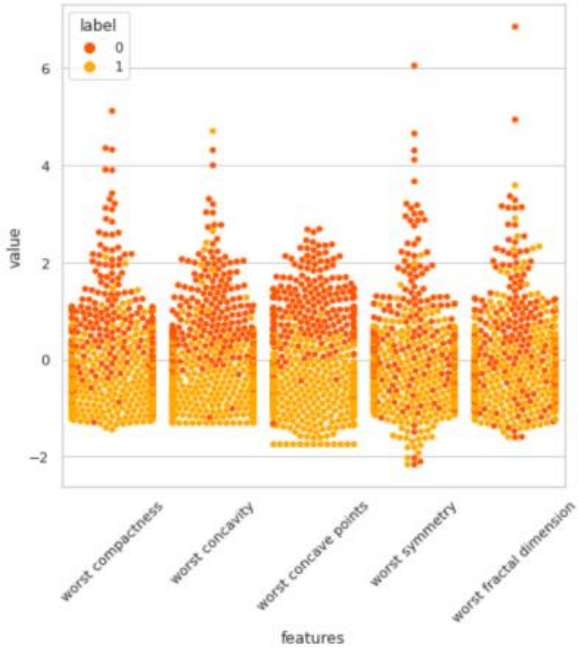

Figure 7. Swarm plot for 25 - 30 features of dataset

Figure 8 displays a heatmap showing all feature names and their correlations.

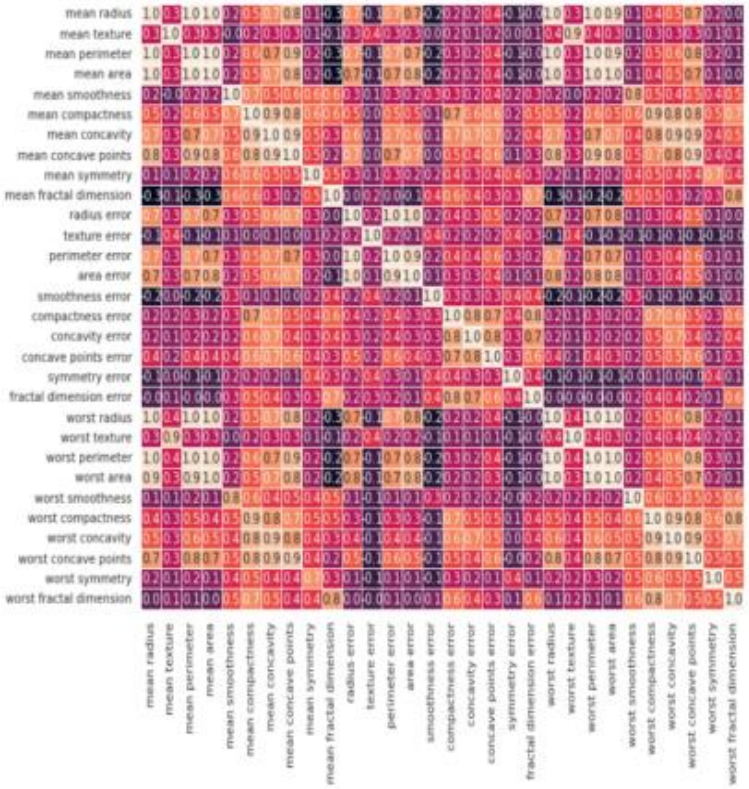

Figure 8. Heat map for 30 features of dataset

Figure 9 illustrates a heatmap with redundant feature names removed and their correlations.

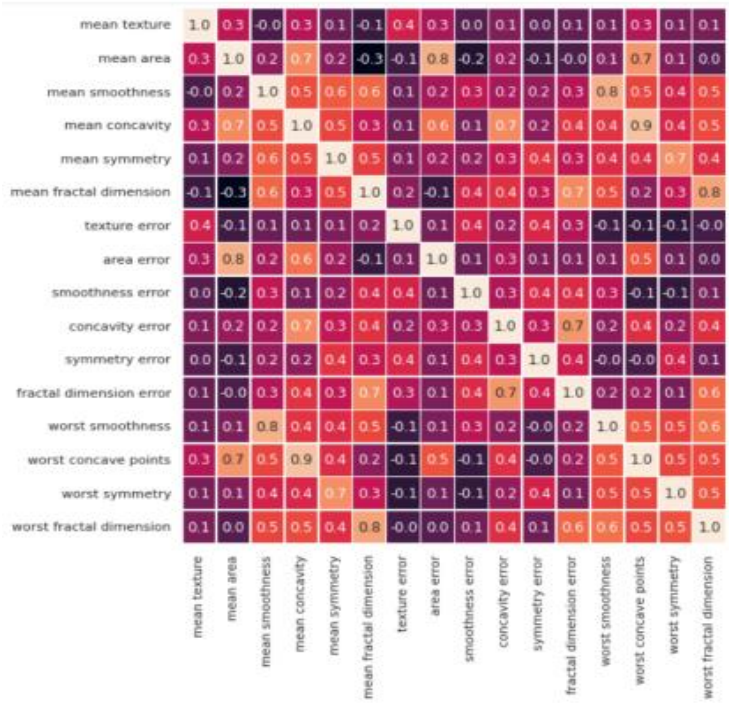

Figure 9. Heat map for 16 features of dataset

The features selected for training the data include mean fractal dimension, worst symmetry, mean texture, mean area, mean smoothness, mean concavity, worst fractal dimension, mean symmetry area error, smoothness error, concavity error, symmetry error, fractal dimension error, worst smoothness, worst concave points, and texture error. These features are considered for training the algorithm and building the model to achieve the highest possible accuracy.

5. Calculations

Accuracy of an algorithm serves as a metric that gauges the proficiency in assigning cases to their appropriate categories. This metric quantifies the ratio of accurate predictions to the total instances present in the dataset. It's crucial to recognize that accuracy's performance is intricately tied to the threshold selected by the classifier, potentially leading to variations across different testing datasets. As a result, accuracy might not be the optimal yardstick for contrasting distinct classifiers, but it can offer a broad perspective on class prediction (Table 1).

Table 1. Accuracy Matrix Calculation for Breast Cancer Prediction

		True Class	
		Positive	Negative
Predicted Class	Positive	True Positive (TP)	False Positive (FP)
	Negative	False Negative (FN)	True Negative (TN)

Thus, the calculation of accuracy can be expressed through the subsequent equation:

$$Accuracy = \frac{K(TP) + K(TN)}{K(P) + K(N)} \quad (1)$$

where:

- K is a constant factor used for breast cancer accuracy measurement. It adjusts the significance of True Positives (TP), True Negatives (TN), Positives (P), and Negatives (N).
- K(TP) represents Weighted True Positives - acknowledges the importance of correct cancer identifications.
- K(TN) represents Weighted True Negatives - acknowledges the importance of correct non-cancer identifications.
- K(P) = Weighted True Positives K(TP) + Weighted False Positives K(FP).
- K(N) = Weighted True Negatives K(TN) +False Negatives K(FN).
- K(FP) represents Weighted False Positives - False positives occur when the ML algorithm wrongly predicts a positive outcome.
- K(FN) represents Weighted False Negatives - False negatives occur when the ML algorithm incorrectly predicts a negative outcome.

Utilizing this formula (1), we have subjected four distinct algorithms to testing, Table 2.

Table 2. Accuracy (in Percentage) of the four tested algorithms

Algorithms	Accuracy (in %)
Light GBM	96.98%
Logistic Regression	91.62%
XG Boost	82.16%
Random Forest	76.72%

6. Conclusion

The results of our study reveal that the Light GBM algorithm proves to be highly efficient and straightforward to implement when working with a diagnostic dataset. After removing the outliers, we found that 16 features remained, significantly contributing to the overall accuracy of the model. Among the algorithms tested, Light GBM achieved the highest accuracy of 96.98%. Logistic Regression yielded an accuracy of 91.62%, Random Forest Algorithm achieved 76.72% accuracy, and XG Boost Algorithm attained 82.16% accuracy.

Additionally, we compared our findings with the study of the survival of breast cancer patients using a dataset containing 856 rows and 15 columns with machine learning models. The accuracy obtained in that study was 84% (Lotfnezhad Afshar et al., 2021).

Overall, our research demonstrates that the Light GBM algorithm excels in breast cancer classification on a diagnostic dataset, surpassing other algorithms and achieving higher accuracy.

Hopefully, this will aid individuals in receiving early cancer treatment and proactively manage their lives.

References

Asri, H., Mousannif, H., Al Moatassime, H., & Noel, T. (2016). Using Machine Learning Algorithms for Breast Cancer Risk Prediction and Diagnosis. Procedia Computer Science, 83, 1064-1069.

Bardou, D., Zhang, K., & Ahmad, S. M. (2018). Classification of Breast Cancer Based on Histology Images Using Convolutional Neural Networks. IEEE Access, 6, 24680-24693.

Bazazeh, D. & Shubair, R. (2016). Comparative study of machine learning algorithms for breast cancer detection and diagnosis. Proceedings of the 5th International Conference on Electronic Devices, Systems and Applications (ICEDSA) (pp. 1-4). Ras Al Khaimah, United Arab Emirates: IEEE.

Derangula, A., Karri, P. K., & Edara, S. R. (2021). Feature Selection of Breast Cancer Data Using Gradient Boosting Techniques of Machine Learning. International Journal of Scientific Research in Computer Science and Engineering, 9(3), 7-15.

Hassan, M., Hassan, M, Yasmin, F., Khan, A. R., Zaman, S., Galibuzzaman, Islam, K.K., & Bairagi, A. K. (2023). A comparative assessment of machine learning algorithms with the Least Absolute Shrinkage and Selection Operator for breast cancer detection and prediction. Decision Analytics Journal, 7, 100245.

Joshi, A., & Mehta, A. (2017). Comparative Analysis of Various Machine Learning Techniques for Diagnosis of Breast Cancer. International Journal on Emerging Technologies, 8(1), 522-526.

Lotfnezhad Afshar, H., Jabbari, N., Khalkhali, H.R., & Esnaashari O. (2021). Prediction of Breast Cancer Survival by Machine Learning Methods: An Application of Multiple Imputation. Iranian Journal of Public Health, 50(3), 598-605.

Nahid, A-A., & Kong, Y. (2017). Involvement of Machine Learning for Breast Cancer Image Classification: A Survey. Computational and Mathematical Methods in Medicine, 2017, 3781951.

Octaviani, T. L., & Rustam, Z. (2019). Random Forest for Breast Cancer Prediction. Proceedings of the 4th international symposium on current progress in mathematics and sciences (ISCPMS2018), AIP Conference Proceedings, 2168, 020050-1–020050-6. Depok, Indonesia: AIP Publishing.

Simon, M. S., Hastert, T. A., Barac, A., Banack, H.R., Caan, B.J., Chlebowski, R.T., Foraker, R., Hovsepyan, G., Liu, S., Luo, J., Manson, J.E., Neuhouser, M. L., Okwuosa, T. M., Pan, K., Qi, L., Ruterbusch, J. J., Shadyab, A. H., Thomson, C. A., Wactawski-Wende, J., Waheed, N., & Beebe-Dimmer, J. L. (2021). Cardiometabolic risk factors and survival after cancer in the Women's Health Initiative. Cancer, 127(4), 598-608.

Sultana, J., & Jilani, A. K. (2018). Predicting Breast Cancer Using Logistic Regression and Multi-Class Classifiers. International Journal of Engineering & Technology, 7, 22–22.

Vyas, S., Chauhan, A., Rana, D., & Ansari, N. (2022). Breast Cancer Detection Using Machine Learning Techniques. International Journal for Research in Applied Science and Engineering Technology, 10(5), 3232-3237.