

# Supervised Classification using Gradient Boosting Machine: Wisconsin Breast Cancer Dataset

**Samyam Aryal**  
School of Public Health  
SRM Institute of Science and Technology  
(SRMIST)  
Kattankulathur, India

**Bikalpa Paudel**  
Ayurveda Campus  
Institute of Medicine  
Kathmandu, Nepal

**Abstract** – Gradient boosting is rapidly becoming one of the most used methods for shallow learning. Tuned gradient boosting methods are quickly replacing standard shallow learning methods like random forest. In the current study, we have proposed breast cancer classification, using Wisconsin Diagnostic Breast Cancer Dataset, using gradient boosting machines. The performance of the method is evaluated using accuracy, sensitivity, specificity, confusion matrix, positive predictive value, negative predictive value, receiver operating characteristic (ROC) curve, and area under the curve (AUC). Gradient boost is compared in terms of accuracy and ROC curve with tuned random forests and logistic regression method of classification to showcase its effectiveness. The results show classification accuracy of 98.88% for the GB model without parameter tuning, 97.89% when accuracy was used as metric for choosing best fit, and 99.3% when ROC was used as a metric for choosing the best fit.

**Keywords**– Supervised learning, Gradient boosting method, Wisconsin diagnostic breast cancer dataset, ROC.

## I. INTRODUCTION

Machine learning algorithms are a set of statistical, probabilistic and optimization methods used to learn large, unstructured and complex datasets detect useful patterns from them and make predictions within an acceptable range.[1] Supervised machine learning falls under the broad category of machine learning. In this method, the algorithms are designed with the help of labeled training datasets and then the algorithms are used to classify the unlabeled testing dataset into similar groups.

Some of the supervised machine learning algorithms – logistic regression, random forest, and gradient boosting methods – used for the shallow learning are discussed below:

### 1. Logistic regression

Logistic regression is employed to classify a model with binary variables, usually related with the occurrence or non-occurrence of an event, based on the probability value calculated for the given instance. The conditional probability of an event is represented as

$$Pr(Y = 1 | X) = p(X)$$

Where,  $Pr(Y=1|X)$  implies the probability of the response belonging to the category 1 due to predictor X.

Since the probability value should be within the range of 0 and 1, logistic regression utilizes logistic function,

$$p(X) = \frac{e^{\beta_0 + \beta_1 X}}{1 + e^{\beta_0 + \beta_1 X}} \quad (1)$$

This function (1) can be further manipulated into:

$$\frac{p(X)}{1 - p(X)} = e^{\beta_0 + \beta_1 X}$$

The left side of the above equation is called odds, defined as the ratio of probability of occurrence of an event to the probability of non-occurrence of the event. In order to create a linear decision boundary for separation of the categories, logarithm on both sides of the final equation is taken, resulting in:

$$\log\left(\frac{p(X)}{1 - p(X)}\right) = \beta_0 + \beta_1 X$$

For estimating the regression coefficients  $\beta_1$  and  $\beta_2$  maximum likelihood is used given by the following likelihood function:

$$l(\beta_0, \beta_1) = \prod_{i: y_i=1} p(x_i) \times \prod_{i: y_i=0} (1 - p(x_i))$$

The estimates are chosen to maximize the likelihood function. So, for every sample labeled as 1 maximum likelihood tries to estimate the coefficients such that the product of all conditional probabilities of category 1 samples is as close to 1 and similarly for the samples labelled 0 it tries to estimate the coefficients such that the product of the complement of their conditional probabilities is as close to 1 as possible. After the coefficients have been estimated, probability  $p(X)$  is calculated and based upon the calculated value the given sample is classified into either of the category.

## 2. Decision trees and Random forest

Decision tree is one of the primary algorithms which classifies the given data on the basis of outcomes resulting from the corresponding test questions. Based on the number of attributes and, their relation with each other and the response, the decision tree gives a multiple levels of branching. Here the top most node includes the attribute which classifies the given data with least error and the terminal node (leaves) consists of the outcome. Beginning from the top most node the data is repeatedly classified on the basis of test questions associated with remaining attributes, finally revealing the decision outcome.

Random forest is a modified and extended form of decision tree. It consists of numerous decision trees constructed from various datasets created by choosing random data from original dataset. In this method, the new instance is run through the decision trees of the random forests. The instance is then classified into the category of the decision outcome which gets the highest number of votes.

From a given training data,  $b= 1$  to  $B$  number of bootstrap sample of size  $N$  are selected. A subset of features is selected randomly whose size ( $m$ ) is usually equivalent to square root of total number of features ( $p$ ) i.e.  $m \approx \sqrt{p}$ .

Each bootstrap sample is then remodeled with the recently selected features. For constructing the decision tree from each sample, the best feature that classifies the data is to be chosen using gini impurity.

To calculate the gini impurity of a given feature, the response of the particular dataset is first classified into  $n$  classes with reference to the binary categories of the given feature. For each category of the given feature variable  $f$ , the gini index is given by:

$$gini(f) = 1 - \sum_{(i=1)}^n p_i \times (1 - p_i)$$

where,  $p_i$  is the probability of picking the data-point with class  $i$

If  $f_1$  with  $N_1$  number of dataset and  $f_2$  with  $N_2$  number of dataset are considered as the binary categories of the given feature then reduction in impurity

is calculated with weighted average of the respective gini impurities, given as:

$$gini_r(f) = \frac{N_1}{N} gini(f_1) + \frac{N_2}{N} gini(f_2)$$

The feature which gives the least value of  $gini_r(f)$  is considered as the best classifier and allocated in the root node of the tree. The root node is further divided into internodes based on the  $gini_r(f)$  value of remaining features until the terminating node is obtained. In this way decision trees are prepared for all  $b=1$  to  $B$  bootstrap samples.

The classification of the new instance is predicted on the basis of maximum votes for a given class of response after running the given instance into all the trees of recently constructed random forest.

## 3. Gradient boost

Gradient boost is another machine learning technique that utilizes decision trees for prediction.[2] The algorithm constructs a new base-learners to be maximally correlated with the negative gradient of the loss function, associated with the whole ensemble.[3] The decision trees in this method are built in series by comparing the attributes and the residual, which is the difference between actual value and predicted value. Each tree will try to correct the residual errors in the predictions from the adjacent previous tree and progress towards accurate predictions.

Predictive analytics play an important role in clinical research.[4] It applies technology and statistical methods to analyze and predict outcomes for individual patients, after going through massive amounts of information. But conventional modelling methods used for predictive analysis face trouble to fit the complex interactions and high-dimensional relationships of the features present in a multi-model biomedical dataset. With the advancement in machine learning, different techniques have been developed to deal with these complications. Gradient boosting is one of these techniques which is able to recursively fit a weak learner to the residual so as to improve model performance with a gradually increasing number of iterations.[4]

For any regression model dataset represented as:

$$Data\{(x_i, y_i)\}_{i=1}^n$$

Where,  $x_i$  vector of predictors for  $i^{th}$  observation and  $y_i$  observed value for  $i^{th}$  observation.

Gradient boost constructs the new base-learners to be maximally correlated with the negative gradient of the loss function, which is actually the difference between the observed and predicted values. [5] The gradient boost progresses by building models minimizing this loss.

The model is initialized with a constant value which is determined by:

$$f_0(x) = \underset{\gamma}{\operatorname{argmin}} \sum_{i=1}^n L(y_i, \gamma)$$

Where,  $L(y_i, \gamma)$  is the loss function,  $\gamma$  refers to predicted values and  $n$  is the total number of observations. Here  $f_0(x)$  is trying to find a value that minimizes the sum of loss function which eventually means to minimize the sum of the squared residuals. Finally  $f_0(x)$  gives:

$$f_0(x) = \frac{\sum_{i=1}^n y_i}{n}$$

The next step consists of constructing decision trees, with considerations of  $M$  being the total number of trees and  $m$  being the index of decision, such that if  $m=1$  we are taking first tree in account and if  $m=M$  then we are taking last tree in account.

For constructing first decision tree we will calculate the pseudo-residual with the help of negative gradient of loss function with respect to the predicted value [6]:

$$r_{i,m} = - \left[ \frac{\partial L(y_i, f(x_i))}{\partial f(x_i)} \right]_{f(x)=f_{m-1}(x)} \quad \text{for } i = 1, \dots, n$$

After computing pseudo-residuals for all  $n$  observations, a regression tree fit to the  $r_{i,m}$  values is constructed and terminal regions or leaf nodes  $R_{jm}$  are created where  $j$  refers to the index of the leaf node 1 to  $J_m$  in a given decision tree. To optimize the values obtained in the leaf nodes, we have to compute output value given by:

$$\gamma_{jm} = \underset{\gamma}{\operatorname{argmin}} \sum_{x_i \in R_{jm}} L(y_i, f_{m-1}(x_i) + \gamma)$$

Where,  $f_{m-1}(x_i)$  is the previously predicted value.

Then a new prediction for each sample is made through the equation given below:

$$f_m(x) = f_{m-1}(x) + \nu \sum_{j=1}^{J_m} \gamma_{jm} I(x \in R_{jm})$$

Where,  $\nu$  is the learning rate that is used to reduce the effect of each tree has on the final prediction. The learning rate is multiplied actually with the total summation of the output values calculated for all previous leaf nodes  $R_{jm}$  and then added with previously predicted value to obtain the new prediction for each observation.

This procedure is iterated from  $m=1$  to  $M$  and final prediction value is obtained from  $f_M$ .

In case of classification dataset similar principles of procedures are applied, only the formulating portions differ. The key component in any classification dataset is the probability values of occurrence or non-occurrence of an event. So most of the formulating portion in gradient

boosting of classification is going to involve the probability values and log of the odds values.

The initial prediction is made by:

$$\begin{aligned} f_0(x) &= \underset{\gamma}{\operatorname{argmin}} \sum_{i=1}^n L(y_i, \gamma) \\ &= \\ &= \underset{\gamma}{\operatorname{argmin}} \sum_{i=1}^n -y_i \cdot \log(p) + \log(1-p) \cdot (1-y_i) \\ &= \underset{\gamma}{\operatorname{argmin}} \sum_{i=1}^n -y_i \cdot \log(\text{odds}) + \log(1 + e^{\log(\text{odds})}) \end{aligned}$$

Where,  $p$  is the predicted probability

Then the residual is calculated with the following equation and the decision tree is constructed

$$\begin{aligned} r_{i,m} &= - \left[ \frac{\partial L(y_i, f(x_i))}{\partial f(x_i)} \right]_{f(x)=f_{m-1}(x)} \quad \text{for } i = 1, \dots, n \\ &= y - \frac{e^{\log(\text{odds})}}{1 + e^{\log(\text{odds})}} \\ &= y - p \end{aligned}$$

The next step after constructing the decision tree is to calculate the output values for each leaf node  $R_{jm}$  given by:

$$\begin{aligned} \gamma_{jm} &= \underset{\gamma}{\operatorname{argmin}} \sum_{x_i \in R_{jm}} L(y_i, f_{m-1}(x_i) + \gamma) \\ &= \underset{\gamma}{\operatorname{argmin}} \sum_{x_i \in R_{jm}} -y_i \cdot [f_{m-1}(x_i) + \gamma] + \log(1 + e^{f_{m-1}(x_i) + \gamma}) \end{aligned}$$

Using second order Taylor polynomial, the loss function is approximated as the ratio of derivative of loss function to second derivative of loss function and the output value is given by:

$$\gamma_{jm} = \frac{\sum (y - p)}{\sum p(1 - p)}$$

Then new prediction is made in similar manner:

$$f_m(x) = f_{m-1}(x) + \nu \sum_{j=1}^{J_m} \gamma_{jm} I(x \in R_{jm})$$

The process is iterated for numerous times to find  $f_M$ .

Applying new dataset in  $f_M$  gives the log(odds) value which is to be converted into probability value. The probability value will be used to classify the given data into either of the category.

• **Summarizing the parameters that need to be included:**

Learning rate ( $\nu$ ): Between 0-1, a very high learning rate results missing out on global minima while a very low learning rate can increase computational costs and fixation on local minima. This can be avoided with selection of large number of trees which inconveniently adds further costs to computation.

Number of trees ( $nr_{i,m}$ ): Higher the number of trees greater is the prediction capability. However, with higher number of trees, the overfitting should always be accounted for.

Number of leaves in terminal node: This combined with number of trees can lead to shallowness or the depth of the tree. This can result in early stopping of the tree. Number of leaves equal to one can be good for classification problem but leads to overfitting for a regression problem.

Choosing estimators and feature selection: Feature selection is the choosing of subsets of relevant features for the use in the model. The features were ranked according to the importance and based on that the performance of the model is assessed and the top predictors are used to fit the final model.[7]

$$F = \{x_1, x_2, \dots, x_N\}$$

$$F' \subseteq F = \{x'_1, x'_2, \dots, x'_M\}$$

Find a subset  $F' \subseteq F$  that maximizes the model's capability for classification.

The feature importance relies on 3 parameters for classification problem. Gain which provides the relative contribution of the feature towards the calculated accuracy of the model. Cover provides the relative number of observation – in the tree – related to the feature. Frequency provides the number of time the feature occurs in the whole model.

• **Cancer Classification**

Cancer is defined as uncontrolled cellular proliferation which arises through a series of alterations in DNA. The rapidly proliferating cells forms mass of extra tissue which are referred as tumors. The tumors are classified as benign and malignant. Benign tumors are those that remain localized and do not spread to other sites. While, malignant are those which invade and destroy adjacent tissues and spread to distant sites. Malignant tumors are collectively referred to as cancers.

Breast cancer refers to malignant tumors originating from breast tissue.[8] The incidence of breast cancer is only second to that of lung cancer and the disease represents the leading cause of cancer-related deaths among women.[9] Early detection and accurate diagnosis of the disease have been playing great role in the prevention of fatality. Recently, supervised machine learning is being seen with keen interest in biomedical field as they promise to offer better sensitivity and/or specificity of detection and diagnosis of disease. Various studies have been done using machine learning techniques for diagnosing breast cancer.

**II. METHODOLOGY**

We have used the data from Wisconsin Diagnostic Breast Cancer dataset which is available at UCI Machine Learning Repository. It is one of the most commonly used dataset for machine learning problems. The WDBC has 569 observations with 32 features (patient id, diagnosis and 30 real-value input features). The features were manufactured from images of fine needle aspiration (FNA) of breast lump. These describe the characteristics of cell nuclei present in the image. The real-valued features are radius, texture, perimeter, area, smoothness, compactness, concavity, concave points, symmetry and fractal dimension for each cell nucleus.[10] The mean, standard error and largest of this features were recorded thus resulting in 30 features. The caret package in R 4.0.1 was used to analyze and cross-validate (10fold, 10times) the model.

**1. Performance evaluation:**

Several measures are used for the assessment of the model. These methods include accuracy, sensitivity, specificity, confusion matrix (table 1), positive predictive value, negative predictive value and ROC curve.

Prediction of the model

Table -I: Confusion matrix representation

		Prediction of the model	
		Positive	Negative
Actual	Positive	True positive (TP)	False negative (FN)
	Negative	False positive (FP)	True negative (TN)

$$\text{Classification accuracy (\%)} = \frac{TP + TN}{TP + FP + FN + TN} \times 100$$

$$\text{Sensitivity (\%)} = \frac{TP}{TP + FN} \times 100$$

$$\text{Specificity (\%)} = \frac{TN}{FP + TN} \times 100$$

$$\text{Positive predictive value} = \frac{TP}{TP + FP} \times 100$$

$$\text{Negative predictive value} = \frac{TN}{FN + TN} \times 100$$

ROC curve or Recursive Operating Characteristic curve is a probability curve plotted with sensitivity against (1 – Specificity) to measure the capability of model in classifying the data by varying the discrimination threshold.[11] Furthermore, the area under the curve (AUC) is used as a convenient measure to compare the ROC curve plotted for different classifying methods. The classifying model with greater AUC value is the better one. In nutshell, ROC curve helps to identify the best threshold for classifying the instances and AUC helps to decide which classifying method is better.

### III. RESULTS AND DISCUSSIONS

75%-25% splitting of the 569 variables was done then cross-validated and checked for accuracy. Various metrics and parameters were used to evaluate and select the best model (table 2, table 3). Gradient boosting was compared with random forest and logistic regression (table 4, figure 1, figure 2).

Table -II: Accuracy, sensitivity, specificity, positive predictive value and negative predictive value for variously tuned GBM

Model/ Measures	Non-tuned GBM	Tuned GBM (Metric:accuracy)	Tuned GBM (Metric:AUC)
Accuracy	98.59%	97.89%	99.3%
Sensitivity	98.88%	98.88%	100%
Specificity	98.11%	96.23%	98.11%
PPV	98.88%	97.78%	98.89%
NPV	98.11%	98.08%	100%

Table -III: Confusion matrix for the present model

		Prediction of the model according to GBM (Metric – AUC)	
		Benign	Malignant
Actual	Benign	89	0
	Malignant	1	52

Table -IV: Comparison of classification accuracies between the tuned GBM, random forest and logistic regression

Model/ Measures	Tuned GBM (Metric:AUC)	Tuned Random Forest	Tuned Logistic Regression
Accuracy	99.3%	97.89%	90.14%
Sensitivity	100%	97.75%	86.52%
Specificity	98.11%	98.11%	96.23%
PPV	98.89%	98.86%	97.47%
NPV	100%	96.30%	80.95%
AUC	0.990566	0.9793301	0.9137163

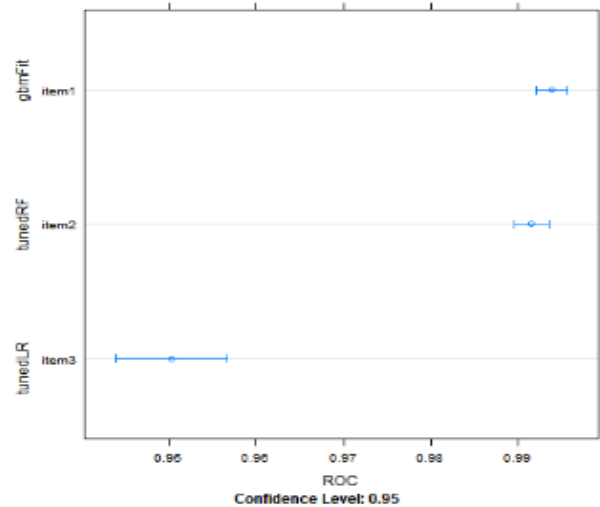


Fig. 1. Comparing the Accuracy of Gradient Boosting Machines, Random forest, and Logistic regression from resamples

The accuracy, sensitivity and specificity for tuned GBM was obtained. The accuracy with gradient boosting when AUC was used as metric was found to be 98.3%. the sensitivity was 100% and the specificity was 98.11%. Significant improvement over logistic regression and moderate improvement over tuned random forest in terms of accuracy and AUC was obtained.

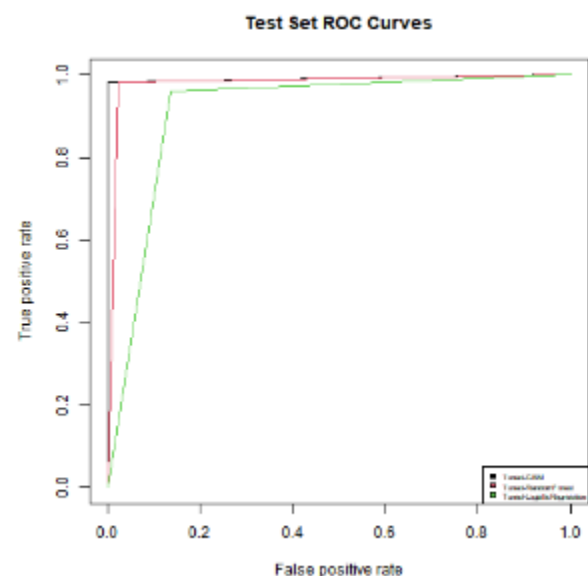


Fig.2.ROC curve for tuned GBM, Random Forest, and Logistic Regression.

### IV. CONCLUSION

A clinical decision making system based on Gradient-Boosting done for diagnosing breast cancer using datasets of Wisconsin Diagnostic Breast Cancer. It is observed that the present model when AUC is taken as a metric

yields a classification accuracy of 99.3%. Along with the additional performance measures such as sensitivity, specificity, positive predictive value, negative predictive value, confusion matrix and ROC curve, the current GB-based model provides great degree of accuracy in addition to faster classification using lesser computing power – compared to deep learning models. Also, the accuracy yielded by the model is much improved over traditionally used models like random forest and logistic regression. Further parameters and the feature selection need to be applied for further boost in performance while lowering the computing costs.

## REFERENCE

- [1]. Uddin S, Khan A, Hossain ME, Moni MA. Comparing different supervised machine learning algorithms for disease prediction. *BMC Medical Informatics and Decision Making*. 2019;19(1).
- [2]. Natekin A, Knoll A. Gradient boosting machines, a tutorial. *Frontiers in Neurobotics*. 2013;7.
- [3]. Ullah E, Mall R, Rawi R, Moustaid-Moussa N, Butt AA, Bensmail H. Harnessing Qatar Biobank to understand type 2 diabetes and obesity in adult Qataris from the First Qatar Biobank Project. *Journal of Translational Medicine*. 2018;16(1).
- [4]. Zhang Z, Zhao Y, Canes A, Steinberg D, Lyashevskaya O. Predictive analytics with gradient boosting in clinical medicine. *Annals of Translational Medicine*. 2019;7(7):152–.
- [5]. Bogner K, Pappenberger F, Zappa M. Machine Learning Techniques for Predicting the Energy Consumption/Production and Its Uncertainties Driven by Meteorological Observations and Forecasts. *Sustainability*. 2019;11(12):3328.
- [6]. Friedman JH. Greedy Function Approximation : A Gradient Boosting Machine. *The Annals of Statistics*. 2001;29(5):1189–232.
- [7]. Akay MF. Support vector machines combined with feature selection for breast cancer diagnosis. *Expert Systems with Applications*. 2009;36(2):3240–7.
- [8]. Akram M, Iqbal M, Daniyal M, Khan AU. Awareness and current knowledge of breast cancer. *Biological Research*. 2017;50(1).
- [9]. Quoirin E. Advanced non-small cell lung cancer in elderly patients. *Breathe*. 2012;9(1):26–34.
- [10]. Khairunnahar L, Hasib MA, Rezanur RHB, Islam MR, Hosain MK. Classification of malignant and benign tissue with logistic regression. *Informatics in Medicine Unlocked*. 2019;16:100189.
- [11]. Narkhede S. Understanding AUC - ROC Curve [Internet]. *Medium*. Towards Data Science; 2019 [cited 2020Jul2]. Available from: <https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5>