ISSN 1816-6075 (Print), 1818-0523 (Online)
Journal of System and Management Sciences
Vol. 12 (2022) No. 3, pp. 251-266
DOI:10.33168/JSMS.2022.0416

Improving Machine Learning Algorithms for Breast Cancer Prediction

Yoon-Teck Bau, Tishanthini Sasidaran, Chien-Le Goh

Faculty of Computer & Informatics, Multimedia University, Cyberjaya, Malaysia

ytbau@mmu.edu.my, tishanthinisasidaran@gmail.com, clgoh@mmu.edu.my

Abstract. Early prediction of breast cancer can prevent death or receiving late treatment. The purpose of this research is to improve machine learning algorithms in predicting breast cancer that will assist patients and healthcare systems. The machine learning algorithms for the prediction of breast cancer are the methods applied in this research by using these following algorithms which are decision tree, random forest, naive Bayes, and gradient boosting due to their high performance. This research uses data from the breast cancer of Wisconsin (diagnostic) dataset of the general surgery department. The results from this research are that by using the stratified k-fold cross validation as a part of the random forest classifier achieved 100% for all four performance scores which are accuracy, recall, precision and F1. The stratified k-fold also improved two machine learning algorithms. In addition, data visualization was applied to the random forest algorithm for result understanding. The implication from the best method is that it could increase the number of accurate breast cancer detections. The values by selecting the best method from this research could assist doctors in early breast cancer detection and increase the number of breast cancer survival rates by receiving early treatment from accurate prediction.

Keywords: machine learning algorithms, decision tree, random forest, naive bayes, gradient boosting, classifier, breast cancer prediction, stratified k-fold, cross validation.

1. Introduction

Cancer deaths are one of the major issues for humankind. One of the common cancers among women is breast cancer. When the cells in the breast grow in an uncontrolled way, creating a mass tissue called a tumor, breast cancer happens.

Breast cancer is the leading cause of death among women worldwide (Azamjah et al., 2019). With early breast cancer prediction, serious health problems or death can be prevented. Accuracy in prediction is critical in breast cancer treatment. With healthcare innovations and clinical cytology, a patient has a better survival rate through early treatment if there is early discovery, differentiating benign from malignant tumors. Machine learning algorithms can help in this regard.

One of the research gaps of breast cancer research is that breast cancer datasets are not in machine learning readable format. With the help of preprocessing techniques, we can convert breast cancer to machine learning readable in text-based form. These machines readable forms have enabled the use of machine learning algorithms in breast cancer prediction and for further research studies.

Machine learning is about making computers adapt their actions from data so that these actions get more accurate (Marsland, 2015). There are various researches on machine learning algorithms for breast cancer prediction. The most common machine learning algorithms used are decision tree, random forest and naive Bayes and the most common performance score used is accuracy. From the reviewed papers, the implemented machine learning algorithms have not yet achieved a 100% performance score. In addition, there is the question of which is the most suitable machine learning that can produce the highest performance scores in breast cancer prediction.

This research aims to compare four machine learning algorithms that will be used to predict breast cancers. The four performance scores used for comparison of results are accuracy, recall, precision and F1. In addition, data visualizations will also be utilized to communicate the results clearly and effectively.

The remainders of this paper are arranged as follows. Section 2 will describe a literature review on notable machine learning algorithms and their scores on breast cancer prediction. In Section 3, we shall describe the methodologies of the machine learning algorithms compared in this research. The results and discussion about the results will be presented in Sections 4 and 5. The last section will be the conclusion of this research.

2. Literature Review

We reviewed all these papers and the results obtained are in the Table 1 below. All these papers reviewed are using the same dataset which is the Wisconsin breast cancer dataset. Recent years, the breast cancer prediction problem is still actively studied by many researchers using the same dataset with different machine learning algorithms.

Ak (2020) used k-nearest neighbors, logistic regression, naive Bayes, random forest and support vector machine to classify breast cancer in the Wisconsin breast cancer datasets. Logistic regression showed classification with the highest accuracy of 98.1%, followed by k-nearest neighbors at 96.90%, support vector machine at 95.90%, naive Bayes at 95.60%, decision tree at 95.60% and random forest at 95.60%.

Amrane et al. (2018) applied k-nearest neighbors and naive Bayes to build classifiers for the same dataset. The results of the comparison were that the k-nearest neighbors' machine learning algorithm gave the highest accuracy of 97.51% and followed by naive Bayes at 96.19%.

Shahare & Giri (2015) applied and made a result comparison between artificial neural network and support vector machine of five different kernel functions which are linear, quadratic, polynomial, radial basis function (RBF) and multilayer perceptron (MLP) for breast cancer prediction on the Wisconsin breast cancer dataset. From the findings of this study, the accuracy of ANN was 96.15%. The highest score in terms of accuracy obtained was support vector machine using linear function at 99.00%, followed by quadratic at 96.00%, polynomial at 95.00%, MLP at 98.50% and RBF at 98.50%.

Gopal et al. (2021) attempted to find ways to predict breast cancer in its early stage. The techniques used were logistic regression, MLP and random forest. The highest accuracy was achieved by MLP at 98.00%, followed by random forest at 95.00% and logistic regression at 79.00%.

Ara et al. (2021) applied decision tree, k-nearest neighbors, logistic regression, naive Bayes, random forest and support vector machine on the Wisconsin breast cancer dataset. From their study, both the support vector machine and random forest gave the highest accuracy at 96.50%, followed by k-nearest neighbors at 95.80%, decision tree at 95.10%, and logistic regression at 94.40% and naive Bayes at 92.30%.

Basunia et al. (2020) applied CART, k-nearest neighbors, logistic regression, random forest, stacking classifier and support vector machine. The stacking classifier obtained the highest accuracy at 97.20%. Both random forest and logistic regression achieved 97.08%. They were followed by the k-nearest neighbors and support vector machine, both at 95.91%. CART achieved 94.74%. The stacking classifier used was an ensemble algorithm which combines multiple classification algorithms.

In a study by Khourdifi & Bahaj (2018), the support vector machine algorithm achieved the highest accuracy at 97.90% followed by k-nearest neighbors at 96.10%, random forest at 96.00% and naïve Bayes at 92.60%.

Pawar et al. (2021) reports the highest accuracy achieved by XGBoost at 98.24% followed by random foresta at 97.36%, support vector machine at 96.49%, Ada-boost and decision tree both at 94.73% followed by k-nearest neighbors at 93.85%.

Bayrak et al. (2019) reported the highest accuracy achieved by support vector machine at 96.99% followed by artificial neural networks at 95.44%.

Mridha (2021) considered seven different machine learning algorithms using the breast cancer Wisconsin dataset. The highest accuracy was achieved by artificial neural network at 99.73% followed by random forest at 98.83%. Both logistic regression and support vector machine both achieved 98.24%, followed by gradient booster at 96.49%, naïve Bayes at 94.73% and k-nearest neighbors at 91.22%.

Erkal & Ayyildiz (2021) reported the highest accuracy achieved by BayesNet at 97.13%, followed by k-nearest neighbors at 96.99%, support vector machine at 96.85%, logistic regression and random forest both at 96.56%. They were followed by naïve Bayes at 95.99% and multilayer perceptron at 95.85%.

The Table 1 below summarizes the studies reviewed above on breast cancer prediction.

Table 1: Recent studies on breast cancer prediction using different machine learning algorithms and their accuracy.

Reference	Dataset	Algorithm	Result
1. Ak (2020)	Wisconsin breast cancer dataset	1. Decision tree 2. K-nearest Neighbors 3. Logistic regression 4. Naive Bayes 5. Random forest 6. Support vector machine	Logistic regression showed classification with the highest accuracy at 98.10%, followed by k-nearest neighbors at 96.90%, support vector machine at 95.90%, naive Bayes at 95.60%, decision tree at 95.60% and random forest at 95.60%
2. Amrane et al. (2018)	Wisconsin breast cancer dataset	1. K-nearest neighbors 2. Naive Bayes	K-nearest neighbors achieved the highest accuracy at 97.51% followed by naive Bayes at 96.19%
3. Shahare & Giri (2015)	Wisconsin breast cancer dataset	1. Artificial neural network 2. Support vector machine (linear, quadratic, polynomial, radial basis function & multilayer perceptron)	The highest accuracy was achieved by support vector machine using linear function at 99.00% followed by radial basis function at 98.50%, multilayer perceptron function at 98.50%, while artificial neural network achieved 96.15%, followed by quadratic function at 96.00% and polynomial function at 95.00%

4. Gopal et al. (2021)	Wisconsin breast cancer dataset	1. Logistic regression 2. Multilayer perceptron 3. Random forest	Multilayer perceptron gives the highest accuracy 98.00% followed by random forest 95.00% and logistic regression 79.00%
5. Ara et al. (2021)	Wisconsin breast cancer dataset	1. Decision tree 2. K-nearest neighbors 3. Logistic regression 4. Naive Bayes 5. Random forest 6. Support vector machine	Both the support vector machine and random forest gave the highest accuracy at 96.50% followed by k-nearest neighbors at 95.80%, decision tree at 95.10%, logistic regression at 94.40% and naive Bayes at 92.30%
6. Basunia et al. (2020)	Wisconsin breast cancer dataset	1. CART 2. K-nearest neighbors 3. Logistic regression 4. Random forest 5. Stacking classifier 6. Support vector machine	Stacking classifier achieved the highest accuracy at 97.20% followed by both the random forest and logistic regression at 97.08%. Both the k-nearest neighbors and support vector machine achieved 95.91% and CART achieved 94.74%
7. Khourdifi & Bahaj (2018)	Wisconsin breast cancer dataset	1. K-nearest neighbors 2. Naive Bayes 3. Random forest 4. Support Vector Machine	The support vector machine achieved the highest accuracy at 97.90% followed by k-nearest neighbors at 96.10%, random forest at 96.00% and naïve Bayes at 92.60%
8. Pawar et al. (2021)	Wisconsin breast cancer dataset	1. Ada-boost 2. Decision tree 3. K-nearest neighbors 4. Random forest 5. Support vector machine 6. XGBoost	The highest accuracy was achieved by XGBoost at 98.24% followed by random forest at 97.36%, support vector machine at 96.49%, Ada-boost and decision tree at 94.73% followed by k-nearest neighbors at 93.85%
9. Bayrak et al. (2019)	Wisconsin breast cancer dataset	1. Artificial neural network 2. Support vector machine	Support vector machine achieved the highest accuracy at 96.99% followed by artificial neural network at 95.44%

10. Mridha (2021)	Wisconsin breast cancer dataset	1. Artificial neural network 2. Gradient booster 3. K-nearest neighbors 4. Logistic regression 5. Naïve Bayes 6. Random forest 7. Support vector machine	The highest accuracy achieved by artificial neural network was 99.73% followed by random forest at 98.83%, logistic regression and support vector machine both at 98.24%, followed by gradient booster at 96.49%, naïve Bayes at 94.73% and k-nearest neighbors at 91.22%
11. Erkal & Ayyildiz (2021)	Wisconsin breast cancer dataset	1. BayesNet 2. K-nearest neighbors 3. Logistic regression 4. Multilayer perceptron 5. Naïve Bayes 6. Random forest 7. Support vector machine	BayesNet achieved the highest accuracy at 97.13% followed by k-nearest neighbors at 96.99%, support vector machine at 96.85%, logistic regression and random forest at 96.56%, followed by naïve Bayes at 95.99% and multilayer perceptron at 95.85%

3. Methodology

Four machine learning algorithms were used to predict breast cancer in this research. They were decision tree, random forest, naive Bayes and gradient boosting. The Scikit-learn library (Pedregosa et al., 2011) and Python were used to implement them.

3.1. Dataset

Breast cancer of Wisconsin (diagnostic) dataset of general surgery department from UCI machine learning repository has 569 instances and 32 features. The dataset includes a diagnosis label which states whether the cancer is benign or malignant.

The samples in the dataset are actually images that have been digitized to 32 features in text form. They describe characteristics of the breast tumor for each patient. The dataset features consist of id, diagnosis and ten main image characteristics. The ten main image characteristics of the tumor are the radius, the texture, the perimeter, the area, the smoothness, the compactness, the concavity, the concave points, the symmetry and the fractal dimension. Each characteristic of the tumor images has the mean, the standard error (se) and the worst which resulted in 30 more features. For example, one of the ten main image characteristics becomes texture mean, texture standard error and texture worst.

3.2. Data analysis and data preprocessing

The diagnosis feature in the data frame is an object type. Hence, it was converted from the object type into integer values 0 and 1. Label encoder function from the Python library was used to convert categorical data into numerical data. The converted values of the diagnosis feature were 0 for benign and 1 for malignant. From Fig. 1 below, the dataset contains 357 benign instances labeled 0 and 212 malignant instances labeled 1.

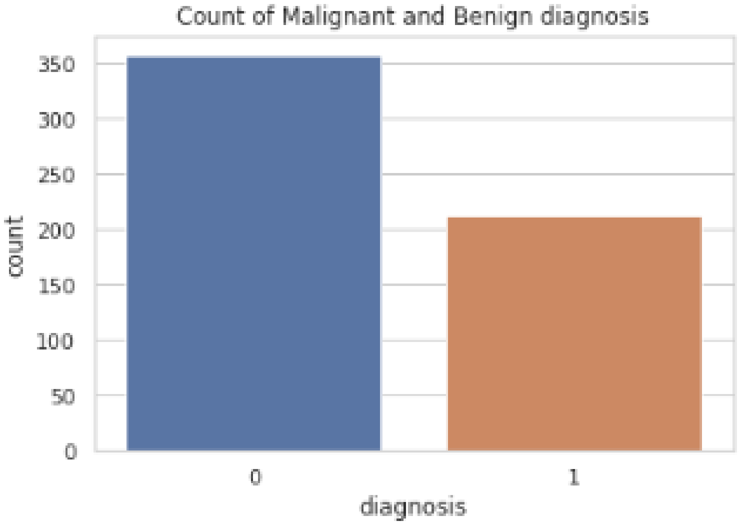

Fig. 1: Count of benign and malignant diagnosis.

The next data preprocessing step was the dropping of the id feature from the data frame. This is it was not needed when machine learning algorithms were applied.

3.3. Machine learning algorithm: Decision tree

Decision tree algorithm is a supervised machine learning algorithm. The decision tree algorithm branches the decision nodes into smaller numbers of samples. The tree building process is repeated for each tree node recursively until there are no more samples, all the remaining samples belong to the same class or until there are no more features remaining.

The Gini index was used for this decision tree machine learning algorithm to calculate the quality of a parent node splitting into child nodes. It is calculated by the following equation (1) where p_i is the probability of a class *i* being classified (Dai et al., 2018):

$$Gini\ index\ =\ 1-\sum_{i=1}^{n}(p_i)^2 \tag{1}$$

The decision tree algorithm is as follows (Marsland, 2015).

Algorithm: Decision tree

```
if all samples have the same class:
    return a leaf with that class
else if there are no features left to test:
    return a leaf with the most common class
else:
    select the feature $F$' of the set of samples $S$ that minimizes the Gini index
    to split the current parent node into child nodes
    add a branch from the parent node for each possible value $f$ in feature $F$'
    for each branch:
        calculate $S_f$ by removing $F$' from the set of features
        recursively call this algorithm with $S_f$ to compute the Gini index relative
to the current set of samples considering only features never selected before
```

3.4. Machine learning algorithm: Random forest

Random forest algorithm is another machine learning algorithm that can be used to predict breast cancer. It is a supervised machine learning algorithm. It eradicates the limitations of the decision tree algorithm by reducing the overfitting effect. Higher accuracy can be obtained using multiple decision trees generated by the random forest when compared with the accuracy obtained with a single decision tree (Hosni et al., 2019).

For each decision tree, the random forest creates a new bootstrap sample by repeatedly taking small samples, calculating the statistic and taking the average of the calculated statistics. At each node of the decision tree, multiple features are randomly selected and the Gini impurity index of the tree node is computed to split the current parent node into child nodes. Splitting of the tree nodes is repeated until the tree is complete. Instead of relying on one decision tree, the random forest classifier combines the prediction results of all decision trees for a sample during the testing phase and then predicts the final decision based on the majority votes.

The basic random forest algorithm is as follows (Marsland, 2015).

Algorithm: Random forest

```
for each of N decision trees where N is set to 100:
    create a new bootstrap sample set of the training set
```

```
3    use this bootstrap sample set to train a decision tree
4    at each node of the decision tree, randomly select m multiple features and
     compute the Gini index then selecting the minimum Gini index for the tree
     node splitting point
5    repeats until the tree is complete
-----------------------------------------------------------------------------------------------
```

3.5. Machine learning algorithm: Gaussian naive Bayes

The Gaussian naive Bayes algorithm is a supervised machine learning algorithm based on the Gaussian probability density function and the Bayes theorem.

All the feature values are calculated to obtain the Gaussian probability density function (*PDF*) for each class as shown by equation (2). X_i is a predictor feature and x_i is the value of X_i. Y is the target feature and y_j is the value of Y. μ is the mean of all predictor features in a certain class. σ is the standard deviation (Saputra et al., 2018):

$$PDF(X_i = x_i\,, Y_j = y_j\,, \mu\,, \sigma) = \frac{1}{\sigma_{ij}\sqrt{2\pi}}\ e^{-\frac{1}{2}(\frac{x_i - \mu_{ij}}{\sigma_{ij}})^2} \qquad (2)$$

The Bayes theorem is used to determine the probability of a likelihood of the conditional probability of a class C given features F_1 *to* F_n. Given by the equation (3) below, P (F_1 ... F_n | C) is the probability of F_1 *to* F_n given C, $P(C)$ is the probability of C occurring and P (F_1 ... $F_{n)}$ is the probability of F_1 *to* F_n occurring (Saputra et al., 2018).

$$P(C|F_1..F_n) = \frac{P(C)\ \ P(F_1..F_n \mid C)}{P(F_1..F_n)} \qquad (3)$$

The Gaussian naive Bayes algorithm is as follows (Saputra et al., 2018).

```
-----------------------------------------------------------------------------------------------
    Algorithm: Gaussian naive Bayes
-----------------------------------------------------------------------------------------------
1 read the training data
2 calculate the mean and standard deviation of values for each predictor feature
  $F = (f_1, f_2, f_3, f_i, ..., f_n)$ in each class
3 repeats:
4    calculate the probability of $f_i$ based on the Gaussian probability density
     function in each class
5 calculate the likelihood in each class based on the Bayes theorem
6 predict using the greatest likelihood
-----------------------------------------------------------------------------------------------
```

3.6. Machine learning algorithm: Gradient boosting

Gradient boosting is a boosting method that creates an ensemble of decision trees. It works by creating a model initially for the input data. Then, it incrementally improves the accuracy by building the next model based on the current models. The combination of the multiple models is generally better than a single model. The final model will attempt to correct the shortcoming by combining boosted ensembles of all the previous models that minimize the overall prediction errors. Binomial deviance was the loss function chosen for this gradient boosting machine learning algorithm.

The gradient boosting algorithm is as follows (Hastie et al., 2017).

```
-----------------------------------------------------------------------------------------------
Algorithm: Gradient boosting
-----------------------------------------------------------------------------------------------
initialize the first iteration model $f_0(x)$ with the minimum value
  based on the loss function, loss function is set to binomial deviance
for $m$ = 1 to $M$ where $M$ is the number of iterations,
  iteration will be terminated when validation score is not improving during the
  training phase
    calculate new residuals for each sample based on the loss function
    construct a new regression tree that fits the samples to the residuals
    calculate the multiplier $\gamma$ for each tree leaf based on the loss function
    update the current $m$ model $f_m(x)$ using the calculated multiplier $\gamma$ value
output the last iteration model $f_M(x)$
make a final class prediction for a sample that uses all the trees in the ensemble
-----------------------------------------------------------------------------------------------
```

3.7. Cross validations

Conventional research methodologies often split the dataset into the training data and the test data randomly. The training data is used for training the machine learning model and the test data is kept independently for validating the performance of the model. If the number of data samples is 100 and the train ratio is 0.7, 70 samples will be randomly selected for training and 30 samples will be randomly selected for testing.

The implemented train-test-split cross validation algorithm is as follows (Marsland, 2015). The train-test-split cross validation algorithm reduces computation time since the model can be trained just one time instead of the model being trained for multiple different subsets of training data.

```
-----------------------------------------------------------------------------------------------
Algorithm: Train-test-split (train ratio = 0.7)
```

```
1 the dataset is randomly partitioned into two subsets
2 the first subset of default value 70% is used as a training
3 the second subset of default value 30% is used for the model as a testing
4 finally, the model that is produced is tested on just one subset of data and
  training on all of the rest.
```

Stratified k-fold was applied for all the four implemented machine learning algorithms to select the training samples and the testing samples according to class proportions where *k* is set to 10. The split of the data is stratified for each class fold. This ensures that each class has a balanced number of samples for each target class label during the validation.

For example, if the number of samples is 100, the number of benign samples is 60, the number of malignant samples is 40, and the number of folds or *k* is set to 10, 54 (9 / 10 * 60) benign samples and 36 (9 / 10 * 40) malignant samples will be selected evenly for each class for training and 6 (1 / 10 * 60) benign samples 4 (1 / 10 * 40) malignant samples will be selected evenly for each class for testing.

The implemented stratified k-fold cross validation algorithm is as follows (Marsland, 2015).

Algorithm: Stratified *k*-fold

```
1 the dataset is evenly partitioned into $k$ subsets for each class
2 one subset is used as a testing
3 $k$-1 subsets are used for the model as a training
4 repeat the same process for all of the different subsets for $k$-1 times where
  the process is to select a different testing subset and a new model is trained on
  the other remaining $k$-1 training subsets
5 finally, the model that produced the lowest validation error is tested and used
```

3.7. Performance scores

A performance score is a measurement to evaluate a machine learning classifier. The four conventional performance scores are accuracy, precision, recall and F1 score. The definitions of the scores are shown in equations (4) to (7). In equations, TP is the True Positive, TN is the True Negative, FP is the False Positive and FN is the False Negative.

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4}$$

$$precision = \frac{TP}{TP + FP} \tag{5}$$

$$recall = \frac{TP}{TP + FN} \tag{6}$$

$$F1\ score = 2 * \frac{precision * recall}{precision + recall} \tag{7}$$

4. Results

Tables 2 and 3 show the four different performance scores for the four different machine learning models used in this research. Table 2 shows the scores when train-test-split cross validation was performed and Table 3 shows the scores when stratified k-fold cross validation was performed.

Table 2: Four different performance scores for four different machine learning models using train-test-split cross validation (best score is highlighted in **bold**).

Machine Learning	Accuracy	Recall	Precision	F1
Random Forest	97.07	93.65	98.33	95.93
Gradient Boosting	95.90	93.65	95.16	94.40
Decision Tree	94.15	95.23	89.55	92.30
Naive Bayes	94.15	90.47	93.44	91.93

Table 3: Four different performance scores for four different machine learning models using stratified k-fold cross validation (best score is highlighted in **bold**).

Machine Learning	Accuracy	Recall	Precision	F1
Random Forest	100.00	100.00	100.00	100.00
Gradient Boosting	92.85	90.47	90.47	90.47
Decision Tree	92.85	90.47	90.47	90.47
Naive Bayes	98.21	95.23	100.00	97.56

Fig. 2 shows one out of the hundred trees generated by the random forest machine learning algorithm. Each node from the random forest tree has five pieces of information. They are the decision rule with a feature name, the Gini index, the remaining samples on the node (labeled as samples), the number of remaining samples in each class (labeled as value) and the class (either the benign class or the malignant class). The Gini index was used as a deciding factor on how best to split a parent node to child nodes based on the conventional method (Dai et al., 2018). Leave nodes do not have decision rules.

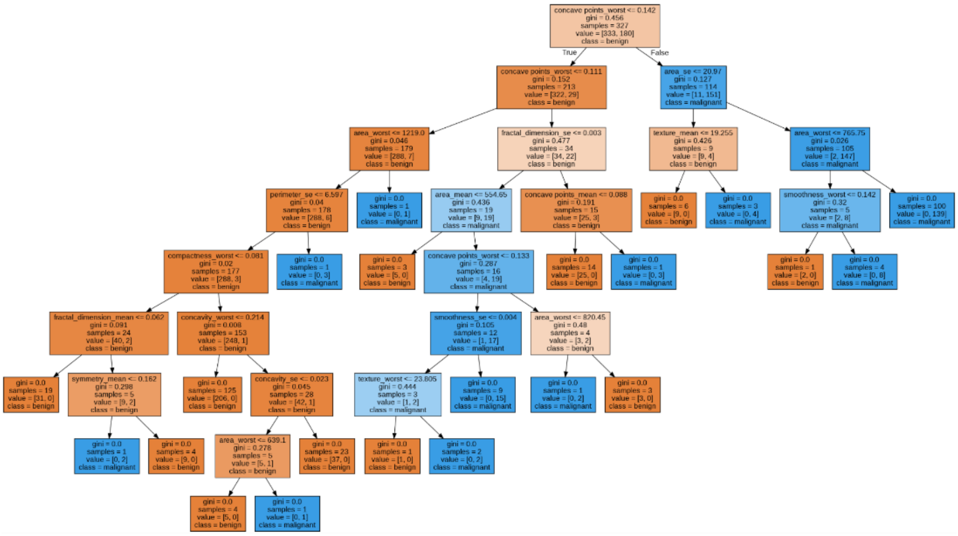
concave points_worst <= 0.142 gini = 0.456 samples = 327 value = [333, 180] class = benign
True
False
concave points_worst <= 0.111 gini = 0.152 samples = 213 value = [322, 29] class = benign
area_se <= 20.97 gini = 0.127 samples = 114 value = [11, 151] class = malignant
area_worst <= 1219.0 gini = 0.046 samples = 179 value = [288, 7] class = benign
fractal_dimension_se <= 0.003 gini = 0.477 samples = 34 value = [34, 22] class = benign
texture_mean <= 19.255 gini = 0.426 samples = 9 value = [9, 4] class = benign
area_worst <= 765.75 gini = 0.026 samples = 105 value = [2, 147] class = malignant
perimeter_se <= 6.597 gini = 0.04 samples = 178 value = [288, 6] class = benign
gini = 0.0 samples = 1 value = [0, 1] class = malignant
area_mean <= 554.65 gini = 0.436 samples = 19 value = [9, 19] class = malignant
concave points_mean <= 0.088 gini = 0.191 samples = 15 value = [25, 3] class = benign
gini = 0.0 samples = 6 value = [9, 0] class = benign
gini = 0.0 samples = 3 value = [0, 4] class = malignant
smoothness_worst <= 0.142 gini = 0.32 samples = 5 value = [2, 8] class = malignant
gini = 0.0 samples = 100 value = [0, 139] class = malignant
compactness_worst <= 0.081 gini = 0.02 samples = 177 value = [288, 3] class = benign
gini = 0.0 samples = 1 value = [0, 3] class = malignant
gini = 0.0 samples = 3 value = [5, 0] class = benign
concave points_worst <= 0.133 gini = 0.287 samples = 16 value = [4, 19] class = malignant
gini = 0.0 samples = 14 value = [25, 0] class = benign
gini = 0.0 samples = 1 value = [0, 3] class = malignant
gini = 0.0 samples = 1 value = [2, 0] class = benign
gini = 0.0 samples = 4 value = [0, 8] class = malignant
fractal_dimension_mean <= 0.062 gini = 0.091 samples = 24 value = [40, 2] class = benign
concavity_worst <= 0.214 gini = 0.008 samples = 153 value = [248, 1] class = benign
smoothness_se <= 0.004 gini = 0.105 samples = 12 value = [1, 17] class = malignant
area_worst <= 820.45 gini = 0.48 samples = 4 value = [3, 2] class = benign
gini = 0.0 samples = 19 value = [31, 0] class = benign
symmetry_mean <= 0.162 gini = 0.298 samples = 5 value = [9, 2] class = benign
gini = 0.0 samples = 125 value = [206, 0] class = benign
concavity_se <= 0.023 gini = 0.045 samples = 28 value = [42, 1] class = benign
texture_worst <= 23.805 gini = 0.444 samples = 3 value = [1, 2] class = malignant
gini = 0.0 samples = 9 value = [0, 15] class = malignant
gini = 0.0 samples = 1 value = [0, 2] class = malignant
gini = 0.0 samples = 3 value = [3, 0] class = benign
gini = 0.0 samples = 1 value = [0, 2] class = malignant
gini = 0.0 samples = 4 value = [9, 0] class = benign
area_worst <= 639.1 gini = 0.278 samples = 5 value = [5, 1] class = benign
gini = 0.0 samples = 23 value = [37, 0] class = benign
gini = 0.0 samples = 1 value = [1, 0] class = benign
gini = 0.0 samples = 2 value = [0, 2] class = malignant
gini = 0.0 samples = 4 value = [5, 0] class = benign
gini = 0.0 samples = 1 value = [0, 1] class = malignant

Fig. 2: Random forest tree visualization.

It is simple to understand and interpret results from the above tree. We start from the tree root. From a non-leaf node, if it passes to meet the rule condition then we proceed to the left child node and if it fails to meet the rule condition then we proceed to the right child node. This process continues. Finally, we stop at the leaf node of the tree which predicts whether it is benign or malignant.

5. Discussion

Two factors have contributed to the improvement in accuracy, recall, precision and F1. One factor is the techniques which were for pre-processing and another factor is the use of stratified k-fold cross validation.

Pre-processing techniques have been applied to the diagnosis feature and the id feature. The diagnosis feature in the data, whether benign or malignant, was converted into 0 or 1. The id feature, this feature was dropped from the data as it was not needed in building the machine learning models.

The data utilized was an imbalanced dataset because the data contains more benign instances than malignant instances. Out of all instances, the percentage of benign instances is 63% but the percentage of malignant instances is 37%. The k-fold technique does not consider the imbalanced dataset but stratified k-fold cross validation takes the imbalanced data into account.

From the experiments using the stratified k-fold cross validation, the random forest machine learning algorithm was able to achieve 100% for all the four performance scores. In this research, the stratified k-fold algorithm was also able to improve the random forest algorithm and the naive Bayes machine learning algorithm. The remaining two machine learning algorithms were not improved but they were still able to achieve scores of more than 90% for all the performance scores. Both gradient boosting algorithm and decision tree algorithm have the overfitting disadvantage and the stratified k-fold applied together with them try to overfit the algorithms hence the algorithms were not improved. The random forest machine learning algorithm does not have an overfitting disadvantage thus it is able to improve. Naive Bayes is able to improve because of the larger number of benign instances to get better predictions.

6. Conclusion

In this research, four machine learning algorithms were implemented to predict breast cancer with two different cross validation methods. The four machine learning algorithms were decision tree, random forest, naive Bayes and gradient boosting. The two different cross validation algorithms were train-test-split and stratified k-fold. Python programming language and the Scikit-learn library were used to implement the algorithms and to record the experimental results.

From the results, it can be observed that when the stratified k-fold validation method was used to validate the four machine learning models, the model generated by the random forest algorithm achieved the highest accuracy at 100.00% for all the four performance scores. The stratified k-fold algorithm also improved two machine learning algorithms. The two improved machine learning algorithms using the stratified k-fold algorithm are random forest and naive Bayes.

This research has limitations since only been implemented in one of cancers which is breast cancer. In the future, it is proposed to apply the designed and implemented improved machine learning algorithms in other cancer types. The contribution discovers that with the implemented random forest machine learning algorithm is able to achieve 100% performance scores as it could increase the number of accurate breast cancer detection. The best machine learning algorithm from this research could assist doctors in accurate breast cancer detection in a short time using a computer and be able to assist patients to seek early treatment from the accurate diagnosis.

References

Amrane, M., Oukid, S., Gagaoua, I., & Ensari, T. (2018). Breast cancer classification using machine learning. 2018 *Electric Electronics, Computer Science, Biomedical Engineerings Meeting (EBBT).*

Ara, S., Das, A., & Dey, A. (2021). Malignant and benign breast cancer classification using machine learning algorithms. *International Conference on Artificial Intelligence (ICAI)*.

Azamjah, N., Soltan-Zadeh, Y., & Zayeri, F. (2019), Global trend of breast cancer mortality rate: A 25-year study. *Asian Pacific journal of cancer prevention: APJCP*, 20(7).

Basunia, M. R., Basunia, M. R., Pervin, I. A., Pervin, I. A., Mahmud, M. A., Mahmud, M. A., & Arifuzzaman, M. (2020). On predicting and analyzing breast cancer using data mining approach. *IEEE Region 10 Symposium*.

Bayrak, A., E., Kirci, P., & Ensari, T. (2019). Comparison of machine learning methods for breast cancer diagnosis. *Institute of Electrical and Electronics Engineers (IEEE)*.

Dai, B., Chen, R., Zhu, S., & Zhang, W. (2018). Using random forest algorithm for breast cancer diagnosis. *International Symposium on Computer, Consumer and Control (IS3C)*.

Erkal, B. & Ayyildiz, E., T. (2021). Using machine learning methods in early diagnosis of breast cancer. *Medical Technologies Congress (TIPTEKNO)*.

Fatih, A. M. (2020). A comparative analysis of breast cancer detection and diagnosis using data visualization and machine learning applications. *Healthcare*, 8(2), 1-23.

Hastie, T., Tibshirani, R., & Friedman, J. (2017). The elements of statistical learning: data mining, inference, and prediction. Springer.

Gopal, V., Al-Turjman, F., Kumar, R., Anand, L., & Rajesh, M. (2021). Feature selection and classification in breast cancer prediction using IoT and machine learning. *Measurement*, 1-11.

Hosni, M., Abnane, I., Idri, A., Gea, J. M. C., & Alemán, J. L. F. (2019). Reviewing ensemble classification methods in breast cancer. *Computer Methods and Programs in Biomedicine*, 177, 89-112.

Khourdifi, Y. & Bahaj, M. (2048). Applying best machine learning algorithms for breast cancer prediction and classification. *Institute of Electrical and Electronics Engineers (IEEE)*.

Marsland, S., (2015). Machine learning: An algorithmic perspective. CRC Press.

Mridha, K. (2021). Early prediction of breast cancer by using artificial neural network and machine learning techniques. *10th IEEE International Conference on Communication Systems and Network Technologies*.

Pawar, S., Bagal, P., Shukla, P., & Dawkhar, A. (2021). Detection of breast cancer using machine learning classifier. *Asian Conference on Innovation in Technology (ASIANCON)*.

Pedregosa et al. (2011), Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825-2830.

Saputra M. F. A., Widiyaningtyas T., & Wibawa A. P. (2018). Illiteracy classification using K means - Naïve Bayes algorithm. *International Journal on Informatics Visualization (JOIV)*, 3(2), 153-158.

Shahare P. D. & Giri R. N. (2015). Comparative analysis of artificial neural network and support vector machine classification for breast cancer detection, *International Research Journal of Engineering and Technology (IRJET)*, 2, 1-6.

UCI machine learning repository: Breast cancer Wisconsin (diagnostic) dataset.