Cancer Detection at Early Stage Using Algorithm Tuning of Support Vector Machine

[1]Sushama Dhaware, [2]Dr. Surendra Bhosale

[1]M.Tech, [2]Associate Professor
[1]Electrical Engineering Department
[1]VJTI, Mumbai, India.

Abstract: Several types of research have been done for early detection of breast cancer so that patient's life can be saved. Previous study was based upon mammogram images. But, mammogram images sometimes give false detection that may endanger the patient's health. . It is necessary to find an alternative method which is easier to implement and efficient also. The main objective is to assess the correctness in classifying data with as benign cancer and malignant cancer with respect to efficiency of each algorithm in terms of accuracy. The next important objective is to increase the accuracy of SVM, KNN, CART using standardization and tuning step by step.

***Keywords* – Algorithm tuning, SVM, CART, KNN, ML**

I. Introduction

Breast cancer is one of the most common cancers among women in the world, accounting for the majority of new cancer cases and cancer-related deaths according to global statistics, making it a significant public health problem in today's society. We have applied three machine learning algorithms to a breast cancer data set in order to predict the benign and malignant cancer. The aim of this research work is to predict breast cancer with early detection and prevention can reduce the risk of death of cancer patient. Classification And Regression Trees (CART), Support Vector Machines (SVM), and K-Nearest Neighbors (*k-NN*) are most commonly used algorithms. Machine learning algorithm gives better accuracy and efficiency as compare to detection of breast cancer using mammogram images. But validation is necessary. [1]

Exploratory analysis is done using python commands to know the different attributes and data variation in the dataset. Data visualization and pre-processing is done to get sense of data distribution. To prepare a data-frame, initially we have used 50% of total dataset is used as training set and remaining 50% of dataset is used as validation set[6] and then 70%-30% ratio. In the execution, the first part consists of comparing effectiveness of three algorithms in terms of accuracy. For baseline algorithm checking we have used 15 fold cross validation technique. From initial run of code, it looks like *k-NN* and CART perform better than SVM. To improve the performance of each algorithm we have implemented standardization on the data set. The improvement is likely for all the models. We have used pipelines that standardize the data and build the model for each fold in the cross-validation test harness. That way we can get correct detection of how each model with standardized data might perform on testing data. After using scaled data significant improvement is seen in SVM accuracy. The next important step is algorithm tuning. For tuning of CART three parameters are chosen. First one is number of features which are changed in three ways as follows: 1) square root of total number of features 2) log_2 (total number of features) 3) total number of features. The next two are minimum number of samples for splitting and for leaf. For tuning of KNN four parameters are changed which are number of neighbors, three algorithms (KD-tree, ball tree and Brute force), leaf size and weights. Weights are varied in two ways: uniform and distance. For tuning of SVM two parameters are changed -the value of regularization parameter C and the type of kernel. Four different kernels namely linear, sigmoid, polynomial and RBF (Radial Basis Function) are chosen for this purpose. We have used the grid search method using 15 fold cross validation on a standardized dataset. After this step, we got value of C and type of kernel which gives best performance in case of SVM. Using this we have built our model. From the confusion matrix it can be seen that there is only three case of misclassification when implemented on test data in case of SVM.

Section I represents the introduction and motivation behind this research. Section II gives general information about the dataset we have used for this project. Section III illustrates the entire methodology of the project that we had used. It gives the entire flow of the research work. The first part is focused on the baseline algorithm tuning. The second part gives a brief introduction about cross validation technique. The next part is about scaling and standardisation of data. The next part depicts about algorithm tuning. Last part states the performance measure for result we have got. Section IV is the result part. Section V illustrates the conclusion and future scope of this research. Last part states all the references we have used.

II. Dataset

This data set was created by Dr. William H. Wolberg, physician at the University Of Wisconsin Hospital at Madison, Wisconsin, U.S.A. [2, 3]. The paper represents various parameters that are useful in predicting if a tumor is malignant or benign. Features are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. This can be found on UCI Machine Learning Repository. Total Number of Instances is 569 and total number of attributes is 32. The dataset contains feature such as radius mean, texture mean, perimeter mean, area mean, smoothness mean, compactness, fractal dimension mean, concavity mean such as 30 other features.

Attribute information:

1. ID number
2. Diagnosis ((M = malignant, B = benign)

III. METHODOLOGY

3.1 Baseline Algorithm Checking

Baseline testing is basically validation of the documents and specifications on which test cases are meant to checked. Baseline, in general, indicates to a benchmark that forms the base of any new algorithm that is developed or enhanced. Baseline testing is a type of testing which is generally performed by testing engineers. As this is equivalent to benchmarking, it is also called as benchmark testing. This test forms the guideline for other testing to compare the efficiency of a new algorithm or unknown algorithm with a known standard frame of reference. This is a binary classification problem. Since we do not know which one will perform the best initially, we had done a quick test on the few appropriate algorithms with to check how each of them will perform on testing dataset. We had used 15 fold cross validation for each testing. We have used three algorithms namely CART, *k-NN*, SVM. [4]

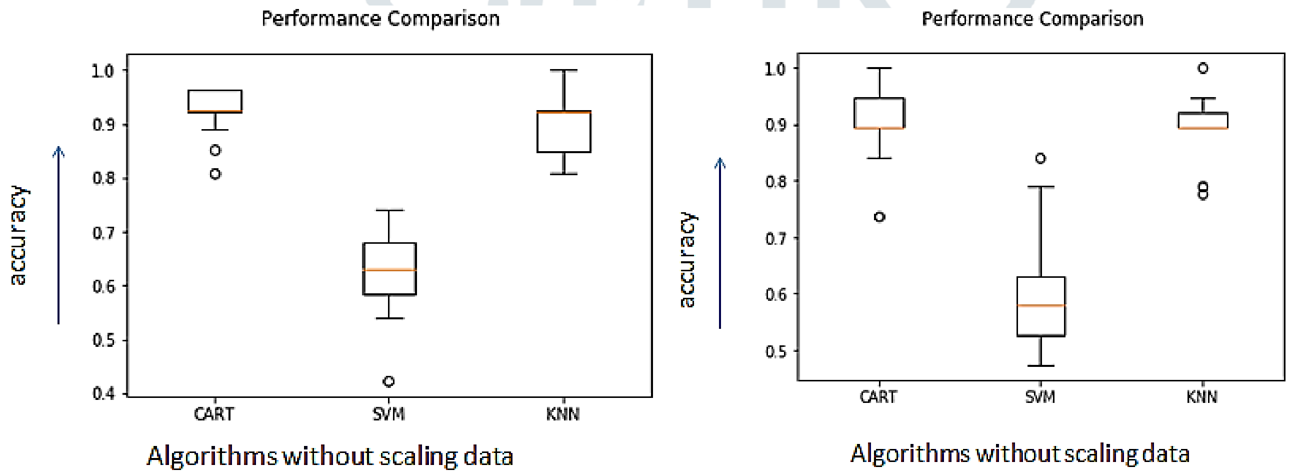
3.2 Cross Validation Technique

There is always a need to verify the efficiency of machine learning model. This process of deciding whether the results are qualified enough to detect the cancer, is known as process of validation. In this process, a numerical estimate difference in predicted and original responses is done, also called the training error. Now it's possible that the model is underfitting or overfitting the data [5]. So, the problem with this evaluation technique is that it does not give an indication of how well the learner will generalize to an unseen data set. This entire process is known as Cross Validation.

In K Fold cross validation what happen is the dataset is divided into k subsets. Now the holdout method is repeated for k times, such that each time, one of the k subsets is used as the test set or validation set and the other k-1 subsets are put together to form a training set[6]. Every data point in training dataset needs to be validated at exactly one time. For k-1 time it should be trained. We have used k=15.

3.3 Results after Baseline Algorithm Checking

(1) Training and testing dataset are in ratio 70%-30%



(a)70%-30% (b) 50%-50%

Fig.1: Results after baseline algorithm checking

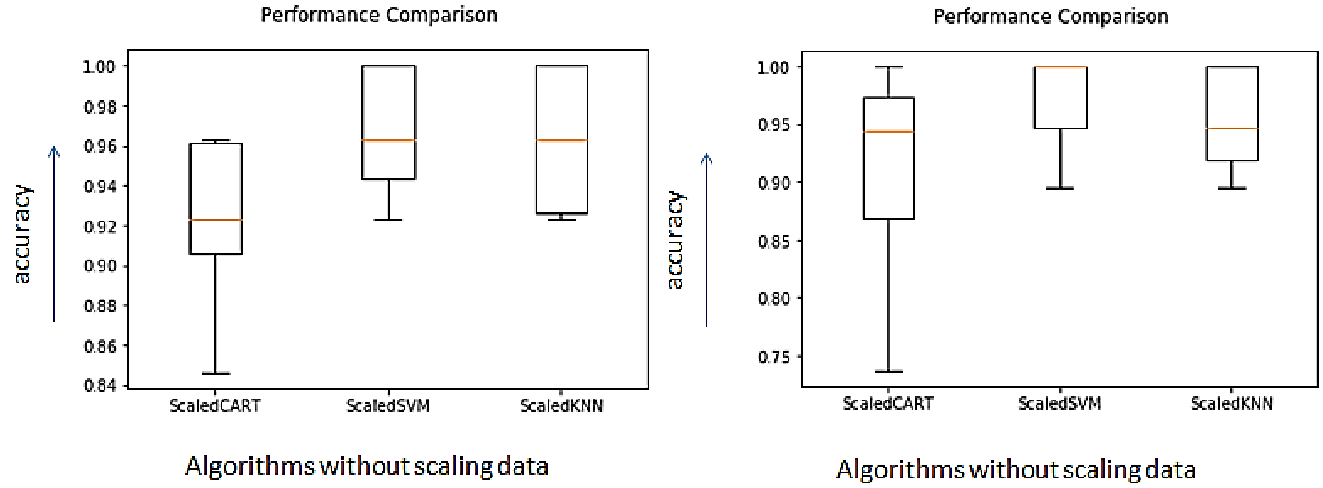
From the initial run it looks like KNN and CART performs better than SVM on given dataset. For CART efficiency is 92%, KNN efficiency is 90% and for SVM efficiency is 62% in case of training and testing dataset in ratio 70%-30%. For CART efficiency is 90%, for KNN efficiency is 89% and for SVM efficiency is 59%.

3.4 Evaluation of algorithm on standardized data

The efficiency of the few machines learning algorithm could be improved in terms of accuracy if a standardized dataset is being used. [2].Standardization of datasets is a common necessity for many machine learning estimators implemented. Algorithm might give lower efficiency if features in dataset are not normally distributed i.e. Gaussian with zero mean and unit variance.

Centering and scaling of dataset perform independently on each feature by computing the relevant statistics on the samples in the training set. Mean and standard deviation are then stored.Mean and standard deviation can later use in the transform method. Transformers are usually combined with classifiers, repressors or other estimators to build a one composite estimator. For this purpose pipelining method is commonly used. Pipeline is frequently used with features which connects the output of transformers into a composite feature space. This is useful as there is often a fixed sequence of steps in processing the data. [7]

3.5 Results after standardization



(a) 70%-30%

(b) 50%-50%

Fig. 2: Results after standardization

After standardization and scaling there is significant improvement in the efficiency of SVM. For scaled CART efficiency is 92%, for scaled KNN efficiency is 96.22% and for scaled SVM efficiency is 96.98% in case of training and testing dataset in 70%-30% ratio. In case of 50%-50% splitting, scaled CART efficiency is 90.85%, for scaled KNN efficiency is 95.77% and for scaled SVM efficiency is 96.98%.

3.6 Algorithm tuning

When tuning algorithms we should have a high confidence in the results given by our test harness. There are two methods available for tuning. First one is random search and second is grid search of parameters. Grid search technique evaluates the testing cases for specific parameters mentioned in a grid. The exhaustive grid search algorithm method separates a closed and bounded interval into *n* subintervals with disjoint interiors and evaluates the function at each endpoint of the subintervals. The interval which consists of the union of the two subintervals which contain the extreme value is retained and the process is repeated. The process terminates either when the length of the interval of uncertainty is less than the pre-assigned tolerance if the magnitude of the midpoint of the interval of uncertainty is less than or equal to one or when the relative length of the interval of uncertainty to the magnitude of its midpoint is less than the pre-assigned tolerance if the magnitude of the midpoint is greater than one.

(a) Tuning of CART

(1) Number of features

Number of features can be used in three different ways:

First when number of features used as square root of total number of features. Second one is when number of features are equal to log_2 of toal number of features. Last one total number of features are exactly used as it is.

(2) Minimum number of samples for splitting

The minimum number of samples required to split an internal node. It can be integer or float.

(3) Minimum number of samples for leaf

These are the minimum number of samples required to be at a leaf node. It can be integer or float.
We got most accurate configuration with minimum samples leaf 3, with the accuracy of 94.47% in case of 70%-30% splitting. We got the most accurate configuration with minimum samples leaf 4 and maximum features are square root of total number of features with the accuracy of 93.66% in case of 50%-50% splitting.

(b) Tuning of KNN

(1) Algorithm

We have used KD-tree and ball tree algorithm for tuning of KNN.

(2) Number of neighbors

The default number of neighbors is 5.

(3) Leaf Size

Leaf size passed to Ball Tree or KD Tree. The optimal value or efficient value certainly depends on total number of features used for classification

(4) Weights

Weight function is mostly used for prediction purpose. It can be uniform where all the pointes in neighborhood are having the same weight. Second type is weight can be inverse of their distance.

We got most accurate configuration with number of neighbours 9, ball tree algorithm with the accuracy of 96.73% in case of 70%-30% splitting. We got most accurate configuration with number of neighbours 5, ball tree algorithm with the accuracy of 96.12% in case of 50%-50% splitting.

(C) Tuning of SVM

For algorithm tuning we have chosen different kernels as follows:

Linear Kernel: A linear kernel can be used as normal dot product any two given observations. The product between two vectors is the sum of the multiplication of each pair of input values [8].

Polynomial Kernel: A polynomial kernel is a more generalized form of the linear kernel. The polynomial kernel can distinguish curved or nonlinear input space.

Radial Basis Function Kernel: The Radial basis function kernel is a popular kernel function commonly used in support vector machine classification. RBF can map feature space in infinite dimensional space.

Value of C: The C parameter is also known as regularization parameter. It basically trades off correct classification of training examples against maximization of the decision function’s margin. For larger values of C, a smaller margin will be accepted if the decision function is better at classifying all training points correctly.

We got the most accurate configuration of SVM with RBF kernel and C=1.7, with the accuracy of 97.23% in case of 70%-30% splitting. We got the most accurate configuration of SVM with Linear kernel and C=0.1**,** with the accuracy of 97.18% in case of 50%-50% splitting.

3.7 Measure for Performance Evaluation

In order to evaluate the prediction performance of SVM classifier, we define and compute the classification accuracy, precision, recall, f1-score, support respectively. The formulations are as follows:

- **Accuracy:** This is the number of correct predictions divided by the total number of instances in the dataset.

$$Accuracy = \left(\frac{TP + TN}{TP + TN + FP + FN}\right) \qquad (3.1)$$

Where, TP = True Positive, TN = True Negative, FP = False Positive, FN = False Negative

- **Precision**: Percentage of correctly classified elements for a given class.

$$Precision = \frac{TP}{TP + FP} \qquad (3.2)$$

- **Recall:** The Recall or TP-Rate is the proportion of the correctly identified positive cases.

$$Recall = \frac{TP}{TP + FN} \qquad (3.3)$$

- **F1-score**: It is a harmonic mean_that combines precision and recall.

$$F = 2 \times \frac{precision \times recall}{precision + recall} \qquad (3.4)$$

- **Support**: It is a number of actual occurrences of the class in a specified dataset

IV RESULTS AND DISCUSSION

Table 1: Accuracy score after Tuning

Algorithm (70%-30%)	Testing Dataset Accuracy	Run time	Algorithm (50%-50%)	Testing Dataset Accuracy	Run time
CART	0.918129	0.004001	CART	0.908772	0.002001
KNN	0.970760	0.003002	KNN	0.961404	0.002001
SVM	0.982456	0.007998	SVM	0.975439	0.003000

Final accuracy after tuning of CART, KNN and SVM is 94%, 96% and 97% on training dataset respectively. In case of training dataset, accuracy after tuning of CART, KNN and SVM is 91.81%, 97% and 98.24% respectively. Accuracy of SVM is better than these two algorithms. In case of breast cancer detection this accuracy is important.

Table 2: Classification Report after Tuning

Algorithm (70%-30%)	Class	Precision	Recall	F1-score	Algorithm (50%-50%)	Class	Precision	Recall	F1-score
CART	Benign	0.96	0.91	0.93	CART	Benign	0.94	0.91	0.93
	Malignant	0.86	0.94	0.89		Malignant	0.85	0.90	0.87
	Average	0.92	0.92	0.92		Average	0.91	0.91	0.91
KNN	Benign	0.97	0.98	0.98	KNN	Benign	0.97	0.97	0.97
	Malignant	0.97	0.95	0.96		Malignant	0.95	0.94	0.94
	Average	0.97	0.97	0.97		Average	0.96	0.96	0.96
SVM	Benign	0.98	0.99	0.99	SVM	Benign	0.98	0.98	0.98
	Malignant	0.98	0.97	0.98		Malignant	0.96	0.97	0.96
	Average	0.98	0.98	0.98		Average	0.98	0.98	0.98

Table 3: Accuracy score after Tuning

Algorithm (70%-30%)	Benign	Malignant	Algorithm (50%-50%)	Benign	Malignant	Class
CART	98	10	CART	171	16	Benign
	4	59		10	88	Malignant
KNN	106	2	KNN	182	5	Benign
	3	60		6	92	Malignant
SVM	107	1	SVM	183	4	Benign
	2	61		3	95	Malignant

The number of misclassification cases in case of CART when training and testing dataset were in ratio 70%-30% was 14 and in case of 50%-50% ratio number of misclassification cases were 26. The number of misclassification cases in case of KNN when training and testing dataset were in ratio 70%-30% was 5 and in case of 50%-50% ratio number of misclassification cases were 11. The number of misclassification cases in case of SVM when training and testing dataset were in ratio 70%-30% was 3 and in case of 50%-50% ratio number of misclassification cases were 7. The number indicates that accuracy is better in case of 70%-30% ratio.

V CONCLUSION

In this project we have successfully utilized method of baseline algorithm checking to understand how different classifier gives different result. Though it shows low accuracy of SVM, we have improved accuracy level using scaled version of dataset and k-fold cross validation technique. Centering and scaling helped in improving the accuracy of machine learning algorithm.

Using scaled data SVM’s accuracy increased from 62% to 96%. After tuning we got maximum accuracy for training and testing dataset in 70%-30% ratio at value of c=1.7 and RBF kernel which is 97.23%. On applying these parameters on SVC classifier we got maximum accuracy of 98.24% with only 3 cases of misclassification. Using scaled data CART’s accuracy increased from 92.71% to 92.21%. CART did not show any significant increase after scaling. After tuning we got maximum accuracy for training and testing dataset in 70%-30% ratio at minimum samples leaf 3, with the accuracy of 94.47%. On applying these parameters on CART classifier we got accuracy of 91.81% with 14 cases of misclassification.
Using scaled data KNN’s accuracy increased from 90.17% to 96.22%. . After tuning we got maximum accuracy for training and testing dataset in 70%-30% ratio at 9 neighbours of 96.73%. On applying these parameters on KNN classifier we got accuracy of 97.07% with 5 cases of misclassification.

VI ACKNOWLEDGMENT

I would like to express gratitude to all those people whose support and cooperation has been an invaluable asset during this project. I would also like to thank our guide DR. Surendra Bhosale for guiding me throughout this project. I would also like to thank all other teaching and non-teaching staff members of the Electrical Engineering Department, VJTI for directly or indirectly helping me for the completion of the project and the resources provided

REFERENCES

[1] Youness Khourdifi, Mohamed Bahaj,”Applying Best Machine Learning Algorithms for Breast Cancer Prediction and Classification”, International Conference on Electronics, Control, Optimization and Computer Science (ICECOCS), 2018.

[2] Abien Fred M. Agarap,"On Breast Cancer Detection: An Application of Machine Learning Algorithms on the Wisconsin Diagnostic Dataset", ICMLSC , February 2–4, 2018, Phu Quoc Island, Viet Nam, 2018.

[3] Dana Bazazeh and Raed Shubair ,"Comparative Study of Machine Learning Algorithms for Breast Cancer Detection and Diagnosis", 5th International Conference on Electronic Devices, Systems and Applications (ICEDSA), 2016.

[4] Fatemeh Shirazi , Esmat Rashedi"Detection of cancer tumors in mammography images using support vector machine and mixed gravitational search algorithm", 1st Conference on Swarm Intelligence and Evolutionary Computation (CSIEC2016), Higher Education Complex of Bam, Iran, 2016.

[5] Ahmed M. Abdel-Zaher, Ayman M. Eldeib ,"Breast Cancer Classification Using Deep Belief Networks" , Expert Systems With Applications , doi:10.1016/j.eswa.2015.10.015,2015.

[6] Hui-Ling Chen , Bo Yang, Jie Liu , Da-You Liu,"A support vector machine classifier with rough set-based feature selection for breast cancer diagnosis,H.-L. Chen et al. / Expert Systems with Applications 38 9014–9022, 2015

[7] Soumadip Ghosh, Sujoy Mondal, Bhaskar Ghosh,"A Comparative Study of Breast Cancer Detection based on SVM and MLP BPN Classifier", First International Conference on Automation, Control, Energy and Systems (ACES), 2014.

[8] Muhammad Hussain, Summrina Kanwal Wajid, Ali Elzaar, Mohammed Berbar,"A Comparison of SVM Kernel Functions for Breast Cancer Detection", Eighth International Conference Computer Graphics, Imaging and Visualization, 2014.