Applications of Machine Learning in Mammography

Dylan Sosa, Kevin Scannell

May 8, 2018

Abstract

Here we describe five machine learning algorithms as applied to the problem of classifying images of samples collected from a fine needle aspiration (FNA) performed on potentially cancerous breast mass. Algorithms presented in this article include: random forest, support vector machine, neural network, naive baye's, and logistic regression. We report overall success in classifying images as malignant or benign with each method. The support vector machine with optimized hyperparameters found through randomized search achieved the highest accuracy score with 99.3%.

1 Introduction

Mammography is the process of examining human breast tissue for diagnosis and screening. The primary application is to detect cancerous tissues. The terms tumor and cancer are sometimes used interchangeably which can be misleading. A tumor is not necessarily a cancer; "tumor" simply refers to a mass. For example, a collection of fluid would meet the definition of a tumor. A cancer is a particularly threatening type of tumor. [2]. In this paper we are attempting to use images of tumors samples obtained via fine needle aspiration to diagnose the samples as malignant or benign where malignant refers to a cancerous growth and benign refers to a non-cancerous mass. In an FNA biopsy (fig. 1), a physician uses a very thin, hollow needle attached to a syringe to withdraw (aspirate) a small amount of tissue or fluid from a suspicious area. The biopsy sample is then checked to see if there are cancer cells in it. [3] The dataset used in this study was collected and curated by Drs. Wolgberg, Street, and Mangasarian of the University of Wisconsin. [1] We hope this study will exemplify the benefit of applying computational techniques to biomedicine.

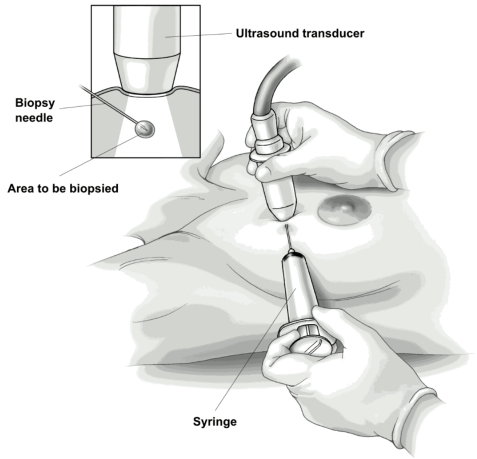

Figure 1: Illustration of an FNA procedure to obtain breast tissue to screen for cancerous cells. Image courtesy of the American Cancer Society.

2 Dataset

The dataset used in this study was obtained from the Machine Learning Repository curated by the University of California, Irvine[1] The data describes characteristics of the cell nuclei present in the FNA image. Ten real-valued features were computed for each cell nucleus in each image: radius (mean of distances from center to points on the perimeter), texture (stanpydard deviation of gray-scale values), perimeter, area, smoothness (local variation in radius lengths), compactness (perimeter^2 / area - 1.0) , concavity (severity of concave portions of the contour), concave points (number of concave portions of the contour), symmetry, and fractal dimension. The mean, standard error, and "worst" or largest (mean of the three largest values) of these features were computed for each image, resulting in 30 features. After preprocessing the dataset, feature columns with maximum predictive ability were kept and others that were too correlated were removed. This resulted in keeping fifteen features (texture, perimeter, smoothness, compactness, and symmetry) to be used for prediction with machine learning algorithms. The dataset was then split into training, development, and test sets.

3 Methods

3.1 Random Forest Classifier

The first algorithm described here is an ensemble method, random forest. This algorithm builds multiple decision trees and makes a classification based on voting with each tree having a vote. In this study we adjusted the hyperparameter of number of estimators as well as different criteria to measure the quality of splits. We used 100, 500, and 1000 trees as well as entropy and gini impurity. The results of each configuration are presented in table 1.

Table 1: Random Forest Model Accuracies

Model	% Accuracy Score
Random Forest n=100	98.24
Random Forest n=100, entropy	97.36
Random Forest n=500	93
Random Forest n=1000	95

3.2 Support Vector Machine

Next, we used multiple support vector machine implementations to approach this classification problem. First was a simple model that surprisingly acheived 94% accuracy. After this we used the "kernel trick" to search for C and gamma parameters in the rbf kernel space which resulted in a 98% classification accuracy. The final support vector machine implementation we considered was a randomized search for optimal C and gamma hyperamaters. This search resulted in C=3.888044829725808 gamma=0.07815467993169875, respectively. With these parameters we achieved a 99.3% classification accuracy. Fig. 2 shows the precision-recall curve for the last SVM implementation and table 2 compares prediction accuracies of the three support vector machines.

Table 2: Support Vector Machine Model Accuracies

Model	% Accuracy Score
Simple SVM	94
SVM rbf kernel	98
SVM with optimal gamma & C	99.3

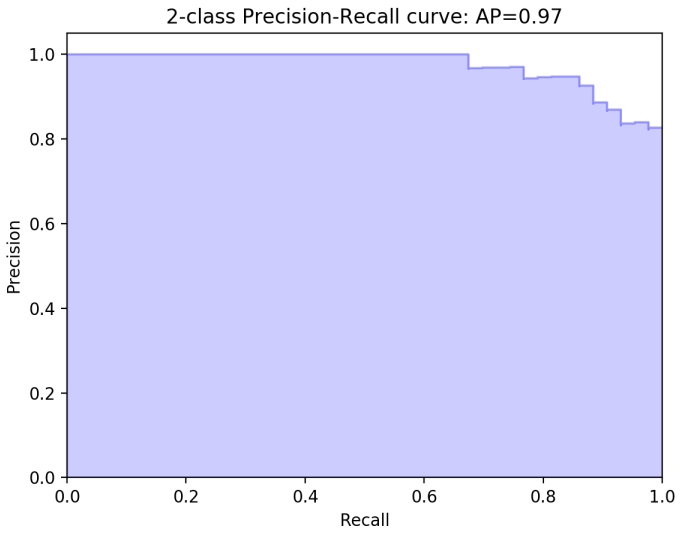

Figure 2: Precision-Recall Curve plot for SVM with optimal parameters found via randomized search. Average precision-recall score of 0.97.

3.3 Neural Network

The third algorithm we implemented to approach this classification problem was a three-layered densely-connected neural network. We used the same architecture in each network implementation, but varied hyperparameters. The general topology (fig. 3) of the networks were: an input layer with 64 neurons and an hyperbolic tangent activation function, a hidden layer with 64 neurons and an hyperbolic tangent activation function, and an output layer with a single neuron (as we are prediction either malignant or benign) which utilized either a softmax or sigmoid activation function. During hyperparameter tuning with our development set we used several different loss functions: hinge, squared hinge, and binary cross entropy. We also used two different optimizers: stochastic gradient descent and the adam optimizer which is an extensions to the typical stochastic gradient descent. In each implementation we utilized batch training with a batch size of 128 and 1000 epochs. The resulting classification accuracies are collected in table 3, and figures 4 and 5 show visualizations of the accuracy and loss at each epoch of the best-performing neural network implemented here.

Table 3: Neural Network Model Accuracies

Model	% Accuracy Score
softmax, hinge loss, SGD	37.14
softmax, squared hinge loss, SGD	37.14
softmax, SGD, binary cross entropy	37.14
sigmoid, SGD, binary cross entropy	82.86
sigmoid, adam optimizer, binary cross entropy	98.4

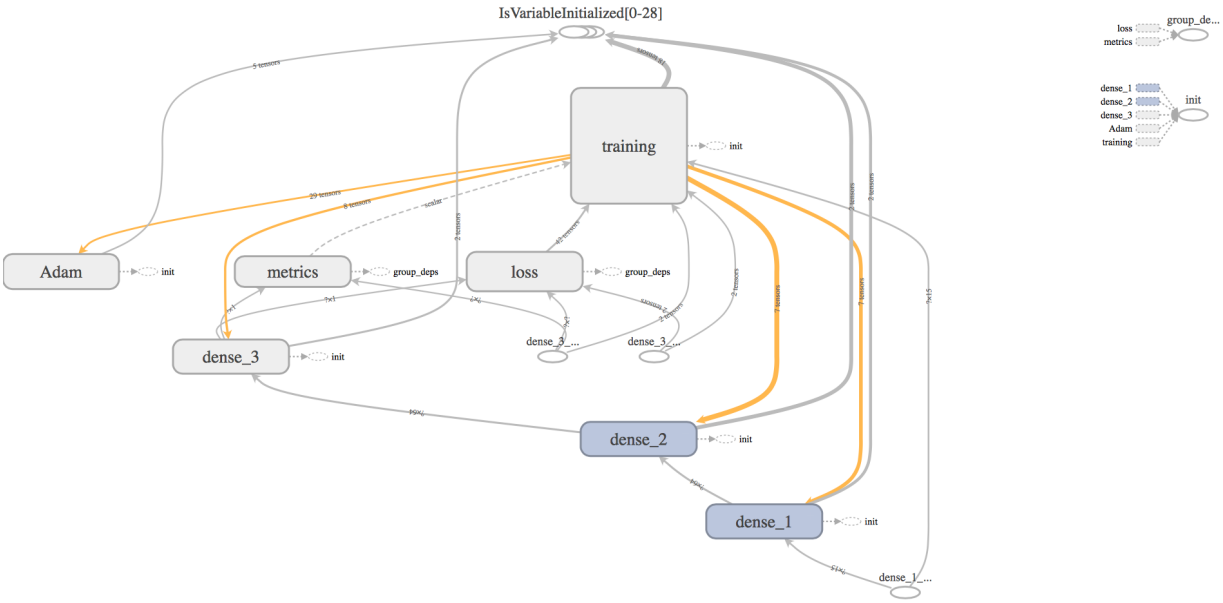

Figure 3: Schematic of the final neural network used in this study which illustrates the densely-connected architecture, adam optimizer, and binary cross-entropy loss. Visualized with tensorboard.

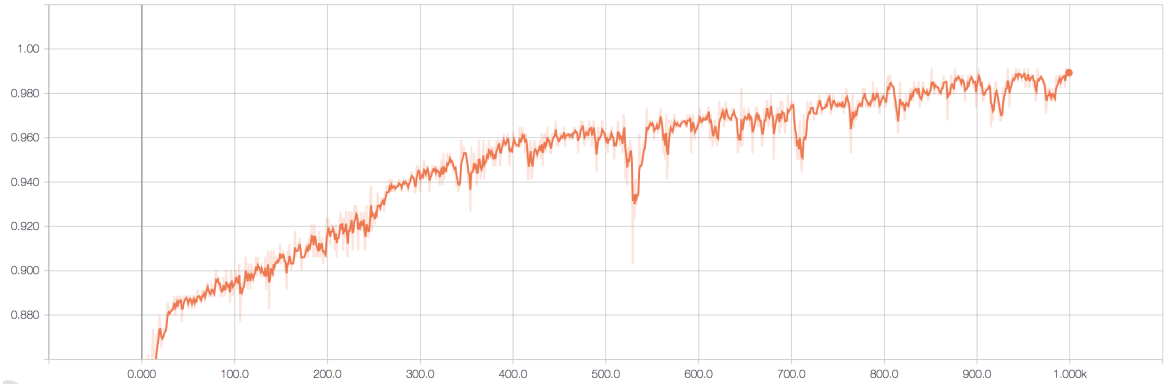

Figure 4: Graph produced by tensorboard which displays the neural network's classification accuracy at each of the one-thousand epochs.

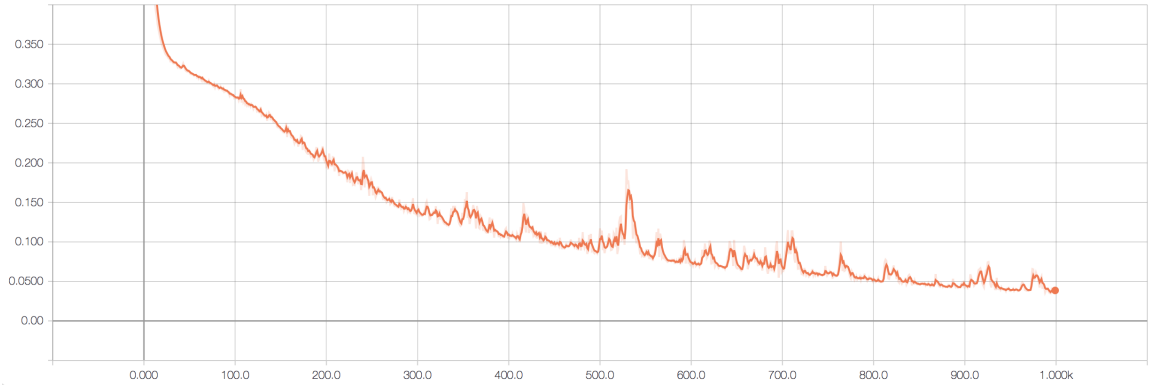

Figure 5: Binary cross-entropy loss graph at each epoch during fitting; produced by tensoboard.

3.4 Naive Baye's Classifier

This algorithm is a conditional probability model. Given a problem instance to be classified, represented by a vector with n features, it assigns to this probabilities for each of K possible outcomes or classes. This model seemed like an ideal choice for the tumor classification problem under consideration here. This algorithm acheived a 94.7% (table 4) classification accuracy. Further tests were done to analyze precision, recall, and F1 score to measure the robustness of the algorithm. (table 6) Moreover, a confusion matrix was built to illustrate the algorithms effectiveness. (table 5)

Table 4: Naive Baye's Model Accuracy

Model	% Accuracy Score
Naive Bayes	94.7

Table 5: Naive Baye's Test Set Confusion Matrix

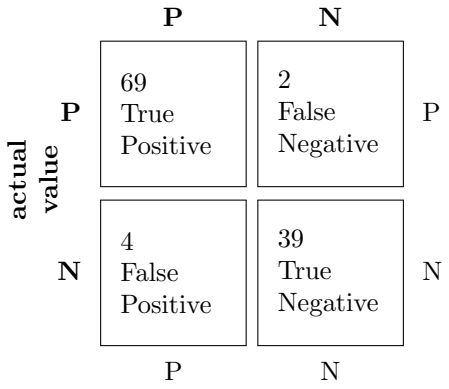

Table 6: Naive Baye's Classification Report

	Development Set			Test Set		
	Precision	Recall	F1	Precision	Recall	F1
Malignant	0.94	0.94	0.94	0.95	0.97	0.96
Benign	0.88	0.88	0.88	0.95	0.91	0.93

3.5 Logistic Regression

The final algorithm implemented for this study was a logistic regression model. Similarly to the Naive Baye's model, we implemented the model once but then more closely analyze its accuracy. This model did not perform as well as the other models described in this paper (table 7), but outperformed some that did not have hyperparameters tuned. The model's performance on both the development and test sets are shown in table 6 which are generally quite high. Again, a confusion matrix was built to analyze the model's effectiveness and as seen in table 8, it did not classify as well as the Naive Baye's model.

Table 7: Logistic Regression Model Accuracy

Model	% Accuracy Score
Logistic Regression	92.9

Table 8: Logistic Regression Test Set Confusion Matrix

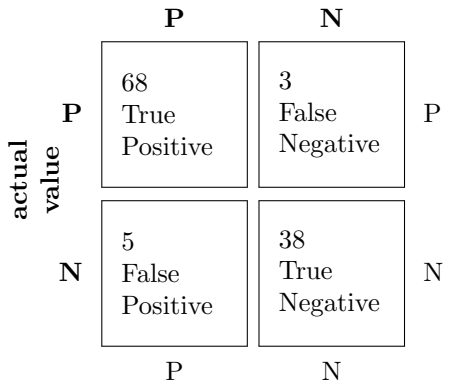

Table 9: Logistic Regression Classification Report

	Development Set			Test Set		
	Precision	Recall	F1	Precision	Recall	F1
Malignant	0.91	0.95	0.93	0.93	0.96	0.94
Benign	0.91	0.83	0.87	0.93	0.88	0.9

3.6 Results & Discussion

Five classification algorithms were implemented to approach the problem of classifying tumor images as either malignant or benign. Table 10 summarizes the best result from each model used in this paper. The support vector machine model obtained a 99.3% accuracy. We are confident in the scores summarized in table 10 because several precautions were taken to avoid over-fitting, including hyperparameter tuning with a development and feature selection. The neural network model had been expected to perform the best, but it may not be so surprising after all that that utilizing a kernel trick with an SVM maximized prediction accuracy.

Table 10: Best Model from each Method

Model	% Accuracy Score
Random Forest n=100	98.24
SVM with optimal gamma & c	99.3
NN with adam, binary cross entropy, sigmoid	98.4
Naive Baye's	94.7
Logistic Regression	92.9

3.7 Conclusions & Future Work

As shown in this study, machine learning can be an effective tool in biomedical scenarios. The support vector machine model had the highest classification accuracy (table 10), but the others were not far behind. This study not only exemplified the potency of machine learning as applied to classification problems, but also the potential for further applications of these techniques in other areas of biomedical practice and research. In future studies like this one, it would be interesting to see how more advanced neural networks (deep learning) compare with kernelized support vector machines. Regardless, it has been demonstrated here that computational methods are highly valuable techniques not just for classical computer science problems.

References

[1] Breast cancer wisconsin (diagnostic) data set. `https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic)`.

[2] Fine needle aspiration biopsy of the breast. `http://pathology.jhu.edu/pc/BasicTypes1.php`.

[3] Fine needle aspiration biopsy of the breast. `https://www.cancer.org/cancer/breast-cancer/screening-tests-and-early-detection/breast-biopsy/fine-needle-aspiration-biopsy-of-the-breast.html`.