Breast Cancer Detection using Machine Learning Methods

Sarah Baalbaki MSAS Program sbaalbak
Ethan Gaskin MSCB Program egaskin
William Hsu MSAS Program tzuhuanh
Madison Stulir MSCB Program mstulir

1 Introduction

Breast cancer is the most common cancer in women in the United States, after skin cancer. It constitutes about 30% of all new female cancers each year [1]. Cancer pathologists commonly use Fine Needle Aspiration Biopsies to investigate suspicious masses in patients [2]. This procedure is a preferred initial method for identifying cancer due to its minimally invasive nature and low cost. But, due to the small sample of cells it collects, sometimes misdiagnoses can happen, either from misinterpreting the images or not puncturing the mass properly [3].

Our goal is to use machine learning to help pathologists interpret digitized images of fine needle aspirates to help improve diagnostic accuracy, which we obtain from the Diagnostic Wisconsin Breast Cancer Database [4, 5].

This dataset is composed of labeled cancerous and non-cancerous cell nuclei features, which we will use as input parameters to train our binary classification models.

Using the provided dataset, we train three different machine learning models to classify breast cancer tumors as malignant or benign. Given the non-linear, mostly gaussian distributions of the data across features, we chose to implement a Naive Bayes Classifier, a Random Forest Classifier, and linear and kernel Support Vector Machine to perform binary classification of malignant versus benign samples.

2 Data

Our dataset is obtained from the Diagnostic Wisconsin Breast Cancer Database. It provides us with a labeled dataset classifying cancerous and non-cancerous cell nuclei based on the analysis of digitized images of fine needle aspirates sampled from breast masses [4].

This dataset describes how cancer-like cell nuclei are using ten different features/attributes of the nuclei: radius (mean of distances from center to points on the perimeter), texture (standard 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 (complexity of the boundary pattern).

Thus, we have a total of 10 real-valued continuous features computed from the cell nuclei during the digitization of the aspirates. We also have 569 samples, and for each sample, each of the 10 features were computed for all cells present in an image then the mean and standard error were calculated and used for that feature. The dataset is slightly imbalanced with 63% benign samples and 37% malignant samples. Moreover, the dataset was complete and there didn't appear to be any

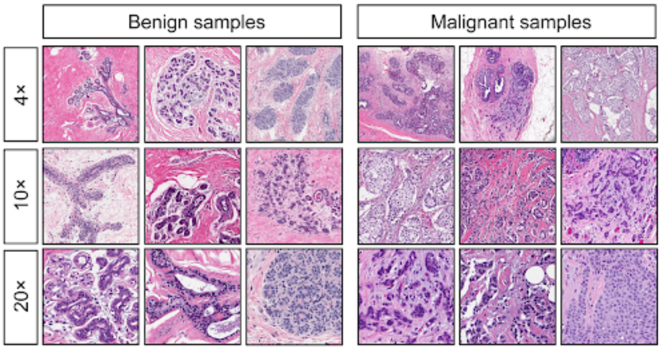

Figure 1: Image of Samples Before Filtering [4]

extreme outliers therefore we chose to not do any pre-processing to prepare the dataset for our models.

To explore the independence of the features, we created a pairwise plot using seaborn.

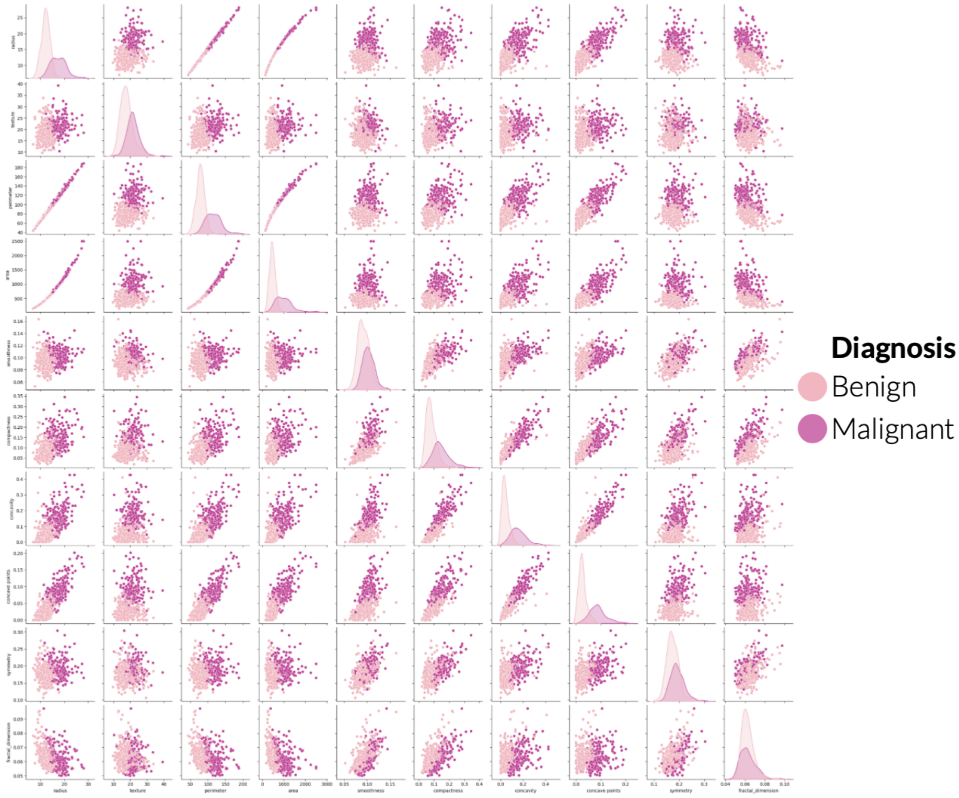

Figure 2: Feature Independence Plot

The plot reveals that the samples are approximately normally distributed in each feature under both classes, which is shown along the main diagonal.

The pair plots, which are off the main diagonal, show bivariate distribution of each feature and labels the samples. Ideally, we want to see that for each pair of features that there isn't too much overlap between the two classes so that we know that the features contribute useful information in distinguishing the classes in a novel way which is independent of other features. That's mostly what we see here.

Also, this allows us to explore the linear separability of the classes as well. If the classes generally appear linearly separable here, then they are likely separable in the complete feature space, 10 dimensions. This is true, and is confirmed later by our different SVM implementations.

3 Methods

Given the non-linear, mostly Gaussian distributions of the data across features, we chose to implement a Naive Bayes Classifier, a Random Forest Classifier, and linear and kernel Support Vector Machine to perform binary classification of malignant versus benign samples. All methods were evaluated using a 5-fold cross-validation.

3.1 Naive Bayes

In our project, a Naive Bayes model was implemented. This choice agrees with previous methods used on the dataset [6] Naive Bayes is modeled using Bayes Theorem. For k classes, this gives:

$$P(c=k|x) = \frac{P(x|c=k)P(c=k)}{P(x)}$$

By assuming each feature to be independent of each other and to be normally distributed, we can obtain the conditional probability distribution function as a multivariate Gaussian distribution:

$$P(x|c=k) = \frac{1}{(2\pi^{\frac{d}{2}}|\Sigma_k|^{\frac{1}{2}})}\exp\{-(x-\mu_k)^T\Sigma_k^{-1}(x-\mu_k)\}$$

for k number of classes and d number of features.

The parameters of the distribution are as follows:

$$\mathbf{x} = \begin{bmatrix} x_{11} & \dots & x_{1d} \\ \vdots & \ddots & \vdots \\ x_{N1} & \dots & x_{Nd} \end{bmatrix}$$

$$\mu = [\mu_1, \quad \dots, \quad \mu_d]$$

$$\mathbf{\Sigma} = \begin{bmatrix} \sigma_1^2 & \dots & \sigma_{1d} \\ \vdots & \ddots & \vdots \\ \sigma_{d1} & \dots & \sigma_d^2 \end{bmatrix}$$

In order to obtain the posterior, we also obtain the parameters for our prior $P(c=k)$ which is a Bernoulli distribution, with the following parameter:

$$\phi = \frac{1}{N}\sum_{n=1}^{N}\iota(c^{(n)}=1)$$

The parameters are calculated using the training data. Then, we predict on the unobserved set and evaluate the performance with the parameters acquired.

3.2 Random Forest

For our random forest implementation, we formed a code hierarchy as seen below. This choice agrees with previous methods used on the dataset [7]

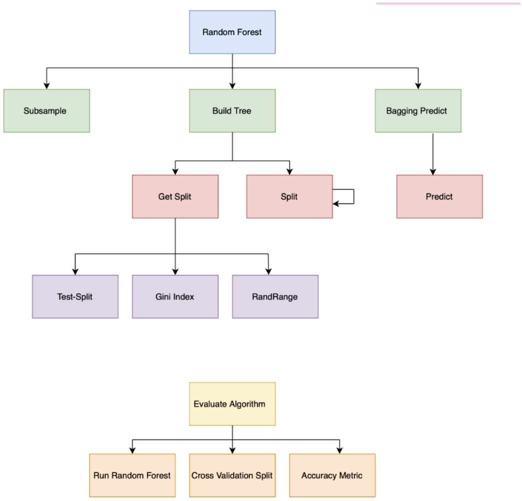

Figure 3: Random Forest Code Architecture

We make sure to incorporate bagging, by randomly selecting a subset of the data with replacement to form multiple datasets to train our forests on. We also make sure to select a subset of features to calculate the the highest purity from at each level to ensure that we have variance between the trees we generate, and this number is taken as 3 features out of the 10 possible ones, which we choose randomly. This also keeps best feature from being over represented in the trees and having very similar splits. For each feature, we obtain the weighted gini index for each group, which treats each group as a category, and choose the one that results in the greatest purity (lowest gini index) as the feature to further split on. We then call a function to separate the remaining data points into the left and right sub-trees of this chosen node, and then further split the sub-trees in a similar fashion.

3.3 SVM

A support vector machine (SVM) is a supervised classification method that is effective for data in high dimensional space. It aims to create the best decision boundary hyperplane to segregate n-dimensional space into classes to be used to classify future input data points. The decision boundary is formed using support vectors, or datapoints that are closest to the boundary point, to best determine the hyperplane that maximizes the decision margin. By using only these datapoints, the hyperplane is not swayed by outlier datapoints.

Support Vector Machines can be trained using linear or non-linear decision boundaries. If the data between classes is linearly separable, a linear SVM is ideal. If the data between classes cannot be separated by a straight line, a non-linear support vector machine can be used. A non-linear support vector machine uses an radial basis function (rbf) kernel, with parameters C and gamma. Gamma determines the width of the kernel function, and C serves as a regularization parameter, controlling whether the model emphasizes a good fit for the data and the simplicity of the decision boundary output. A large gamma can lead to over-fitting of the data, while a small gamma causes excess constraint on the model. Larger C emphasizes a smaller margin, while a lower C emphasizes a larger margin, and therefore a simpler decision boundary. C and gamma must be optimized to the dataset in order to achieve the highest performance non-linear SVM model for the dataset.

4 Results

4.1 Naive Bayes

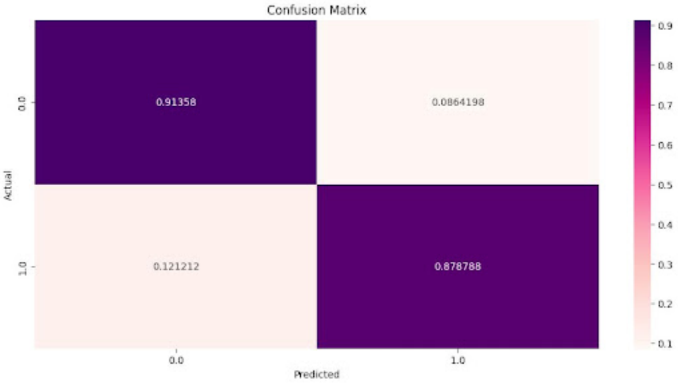

Figure 4: Confusion Matrix For Naive Bayes Results after 5-fold Cross Validation

We generate a confusion matrix to observe the true positive (TP), true negative (TN), false positive (FP), and false negative (FN) rates, which we will discuss below.

4.2 SVM

We implemented both linear and non-linear SVM, and evaluated the results to determine the optimal method for our dataset. These methods were implemented using sklearn. For non-linear SVM, we used grid search to determine the optimal parameters for C and gamma, by trying combinations of C and gamma in a range of generally used values. In Figure 7, the results for each parameter combination was plotted, with the color indicating the accuracy of the parameters. The optimal parameters were determined to be C=1000 and gamma=0.0001. These parameters were then used to train a model with 5-fold cross validation. Results are seen in table 1.

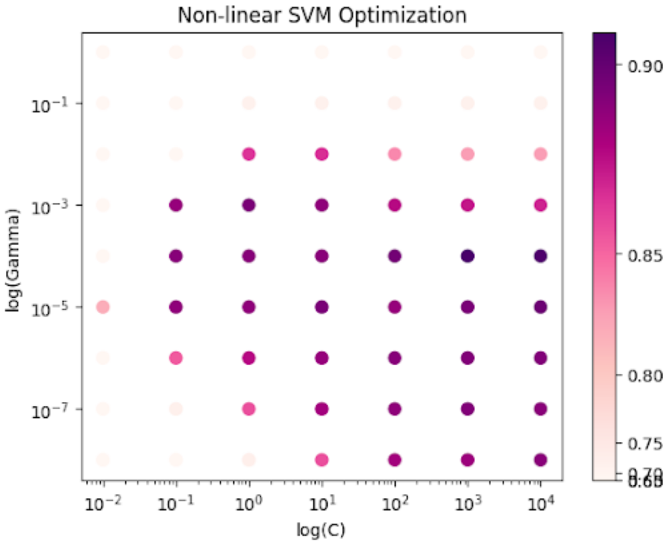

Figure 5: Optimization of Non-Linear SVM Parameters Using Grid Search

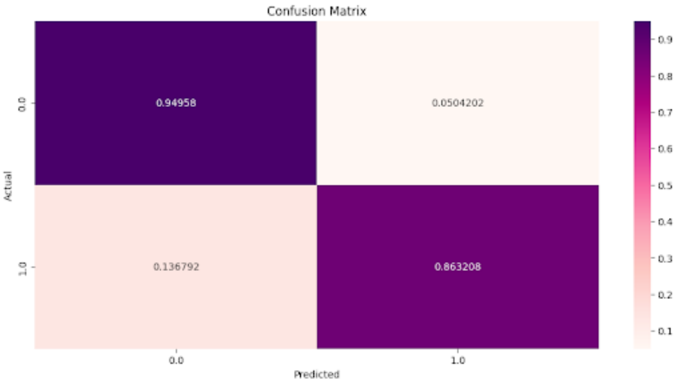

Figure 6: Confusion Matrix For Linear Support Vector Machine Results after 5-fold Cross Validation

Additionally, a linear SVM model was used to classify the dataset using 5-fold cross validation. The results were found to be higher for linear SVM than non-linear SVM, available in Table 1. We can interpret these results to mean that our dataset was linearly separable, and therefore using a non-linear kernel is not necessary. This is consistent with the pairwise plot of the data generated in Figure 2. The confusion matrix of the linear SVM model was plotted and is seen in Figure 8.

4.3 Random Forest

We tried constructing different random forests of different sizes, specifically ranging from 5-10 trees, and found that using a random forest of 9 trees gives us the best testing accuracy on unobserved data.

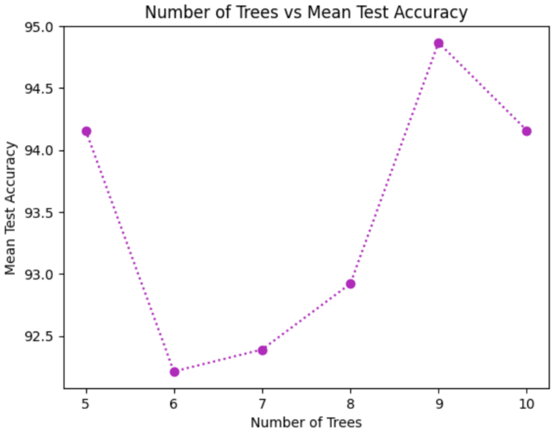

Figure 7: Random Forest Model Test Accuracy versus Number of Trees

We also generate a confusion matrix to observe the TP, TN, FP, and FN rates.

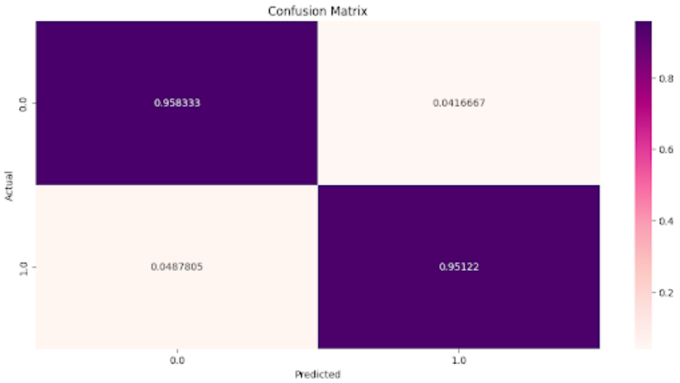

Figure 8: Confusion Matrix For Random Forest Results after 5-fold Cross Validation

4.4 Overall Results

Since we are dealing with breast cancer diagnosis, we need to pay attention not only to overall model performance, but also to the false positive and false negative rates.

False positives may lead to further testing and potentially early detection of cancer, but in other cases they can also have negative consequences and can cause significant psychological distress, anxiety, and unnecessary medical procedures, such as biopsies, which can be invasive and painful. Moreover, false positives can lead to unnecessary healthcare costs, and may even result in harm to patients due to over diagnosis and unnecessary treatment. On the other hand, false negatives are also an important factor in breast cancer diagnosis, and happens when breast cancer is present but fails to be detected. It may lead to a delayed diagnosis and worse outcomes for the patient, where in some cases it might also be too late.

From our three methods above, the ones with the lowest false positive rate is random forest and the one with the lowest false negative rate is random forest as well.

We also form a table comparing the performance of our methods reflected by the prediction accuracy on our training set (using cross validation), the accuracy on the test set (using the unobserved testing data), and the F1 score.

The F1 score is used to quantify a classification model's accuracy. It considers both precision and recall, and is calculated as the harmonic mean of precision and recall, where precision is the proportion of true positive classifications out of all positive classifications, and recall is the proportion of true positive classifications out of all actual positive cases. The F1 score ranges from 0 to 1, with higher scores reflecting better model performance.

Table 1: Overall Results Table

Method	Naive Bayes	Random Forest (9 Trees)	SVM (linear)	SVM (non-linear)
Model Accuracy on Training set	91.3%	99.558%	91.9%	92.9%
Model Accuracy on Test set	90.35%	94.8672%	91.7%	91.0%
F1 Score	81.2%	90.1%	89%	87%

This table of results shows us that the model with highest training accuracy is random forest (99.5%), the model with highest test accuracy is also random forest(94.8%), and that with the highest F1 score is also random forest (90%), followed by linear SVM, non-linear SVM, and finally Naive Bayes.

5 Conclusions

From our model results, we can see that Linear SVM performs better than non-linear SVM, as reflected by the numerical results and the plots we did to show that the data and features are linearly separable.

Although Naive Bayes assumes all the features to be independent, it performed surprisingly well. This suggests orthogonality between features conditioned on the classes, which seems to be consistent with the linear separability of the binary classes seen between each pair of features in the pair plot.

We also find that Random forest is the model that performs the best with our dataset, achieving a 99% training accuracy, and a 94.8% testing accuracy. The high training accuracy would suggest over-fitting of the data, but the testing accuracy demonstrates that the average accuracy (from averaging the testing on the last fold) was high which suggests the model is relatively robust when classifying new points.

We can see that the methods we chose do a good job at classifying between benign and malignant cases based on cell characteristics, but we still do not achieve 100% accuracy in any. Since we are dealing with breast cancer, we would like to have as high an accuracy as poosible, since we are dealing with diagnosing people whose lives are at risk.

5.1 Future Work

To further improve our results, we can try to implement a mixture model for the features instead of assuming normal distributions and independence and assess performance on a combined model.
Also, we can train our models with more data (569 samples in our dataset) and ensure an equal class distribution among our samples, with the hopes of obtaining higher model performance scores and accuracies. By testing our model on a larger dataset, we can see if the accuracies still hold up for diagnostic settings (determine if the model is robust enough to actually improve prediction of breast cancer from fine needle aspirate biopsies for new patients).

References

[1] Centers for Disease Control and Prevention. Basic information about breast cancer. https://www.cdc.gov/cancer/breast/basic_info/index.html, n.d. Accessed: May 4, 2023.

[2] American Cancer Society. How common is breast cancer? https://www.cancer.org/cancer/types/breast-cancer/about/how-common-is-breast-cancer.html, n.d. Accessed: May 4, 2023.

[3] American Cancer Society. 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, n.d. Accessed: May 4, 2023.

[4] Dheeru Dua and Casey Graff. UCI machine learning repository. http://archive.ics.uci.edu/ml, 2017.

[5] Street, W. N., Wolberg, W. H., Mangasarian, O. L. Nuclear feature extraction for breast tumor diagnosis. in biomedical image processing and biomedical visualization. Biomedical image processing and biomedical visualization (Vol. 1905, pp. 861-870). SPIE., 1993. Accessed: May 4, 2023.

[6] Chotirat Ann and Dimitrios Gunopulos. Scaling up the naive bayesian classifier: Using decision trees for feature selection. http://rexa.info/paper/4695569c53cd581fcc193415a8a94a1f92abf607, n.d. Accessed: May 4, 2023.

[7] Krzysztof Grabczewski and Włodzisław Duch. Heterogeneous forests of decision trees. http://rexa.info/paper/b19579eae108f0efb0d9adf97e480280f8e4f7a8, 2002. Accessed: May 4, 2023.