Machine Learning in R: Breast Cancer

Morgan Gant

April 24, 2023

**Abstract**

The most common type of cancer among women in the United States is Breast Cancer. Many forget it’s a cancer that occurs often in women but can form in males as well. With technological advancements, Diagnosis and treatment for breast cancer has increased and have been able to increase survival rates. This is due to earlier detection, personalized treatments, and better understanding of disease. Machine learning can play an important part when it comes to being diagnosed. This project aims at being able to identify the most significant variables/measurements of a lobe from the data set and spot potential breast cancer diagnoses. I have obtained access to the Breast Cancer Wisconsin (Diagnostic) data set from Kaggle. The dataset has been reviewed and cleaned. I have also engineered a few features and worked on hyper tuning models to predict if a lobe is cancerous or benign. **(mayoclinic)**

Table of Contents

[Introduction 3](#_Toc132816514)

[Data Description 3](#_Toc132816515)

[Exploratory Analysis 4](#_Toc132816516)

[Predictive Modeling 7](#_Toc132816517)

[Model Performance 8](#_Toc132816518)

[Conclusion 9](#_Toc132816519)

[Implications and Further Questions 9](#_Toc132816520)

# Introduction

Brest Cancer is not foreign to many people. It occurs when healthy cells change and reproduce rapidly/abnormally, then they turn into tumors. The mutated cells can spread through your breasts to your lymph nodes or other parts of the body. Tumors can occur in different areas of the breast: Glands that make milk, ducts that hold the milk, and other tissue areas. Signs to be aware of are lobes or thickening of tissue in the breast, change of size and appearance of breast, changed to the skin over the breast such as: Dimpling, redness, pitting, peeling, crusting, flaking, newly inverted nipple. Doctors think 5-10 percent of breast cancers are linked to gene mutations that are inherited.

As like any other sickness, there are many risk factors associated with Breast Cancer. Some of the risks include being female, older in age, radiation exposure, obesity, personal history of breast conditions/breast cancer, etc. **(mayoclinic)** No medical professionals will ever be able to limit the risk factors, some are determined on the lifestyle of the patient. The focus of this project is to determine the best measurements that classify breast cancer as malignant or benign.

# Data Description

As stated, before in the abstract, this dataset was taken from Kaggle, which is a public website that contains material for data scientists. This data set contains 32 columns and 570 rows and has no missing values. Each row represents a patient, the first two columns are the patient ID and their diagnosis of a benign or malignant lobe, the other columns represent a measurement of a lobe on their breast.

|  |  |  |
| --- | --- | --- |
| **Variable Name** | **Variable Type** | **Description** |
| id | Numerical | Patient ID |
| diagnosis | Categorical | Target:  M- Malignant  B- Benign |
| radius\_mean | Numerical | Radius of the lobes |
| texture\_mean | Numerical | Mean of Surface Texture |
| perimeter\_mean | Numerical | Outer Perimeter of Lobes |
| area\_mean | Numerical | Mean area of Lobes |
| smoothness\_mean | Numerical | Mean of Smoothness Levels |
| compactness\_mean | Numerical | Mean of Compactness |
| concavity\_mean | Numerical | Mean of Concavity |
| concave points\_mean | Numerical | Mean of Concave Points |
| symmetry\_mean | Numerical | Mean of Symmetry |
| fractal\_dimension\_mean | Numerical | Mean of Fractal Dimension |
| radius\_se | Numerical | SE of Radius |
| texture\_se | Numerical | SE of Texture |
| perimeter\_se | Numerical | Perimeter of SE |
| area\_se | Numerical | Area of SE |
| smoothness\_se | Numerical | SE of Smoothness |
| compactness\_se | Numerical | SE of Compactness |
| concavity\_se | Numerical | SE of Concavity |
| concave points\_se | Numerical | SE of Concave Points |
| symmetry\_se | Numerical | SE of symmetry |
| fractal\_dimension\_se | Numerical | SE of Fractal Dimension |
| radisu\_worst | Numerical | Worst Radius |
| texture\_worst | Numerical | Worst Texture |
| perimeter\_worst | Numerical | Worst Perimeter |
| area\_worst | Numerical | Worst Area |
| smoothness\_worst | Numerical | Worst Smoothness |
| compactness\_worst | Numerical | Worst Compactness |
| concavity\_worst | Numerical | Worst Concavity |
| concave points\_worst | Numerical | Worst Concave Points |
| symmetry\_worst | Numerical | Worst Symmetry |
| fractal\_dimension\_worst | Numerical | Worst Fractal Dimension |

# Exploratory Analysis

The first step taken to engage in the data was to create visualizations to see if there were any significant relationships, distributions, or interesting variables.

*Figure 1:*

Chart

Description automatically generated

We can see by looking at *Figure 1*, that most of the cases in the Breast Cancer data set are Benign (non-cancerous). Since this is unbalanced, when splitting the data into train and test to run models, the distribution should be accounted for.

The first variables that were examined are radius\_mean and diagnosis:

*Figure 2:*

Chart, histogram

Description automatically generated

In *Figure 2*, it shows that lobes with smaller radius means tended to be benign cases where lobes with larger higher radius means were more likely to be malignant, or cancerous. In this data set there looks to be more benign cases with smaller average radii compared to the larger range of average lobe radius’s that were classified as malignant.

*Figure 3:*

Chart, histogram

Description automatically generated

*Figure 3* is looking at variables texture\_mean and diagnosis. Both classifications have similar ranges. However, on average, Benign cases had lobes with lower surface texture means and Malignant had higher surface texture means. There were outliers in the benign and malignant cases as you can see on the tail ends of the graph.

*Figure 4:*

Chart, histogram

Description automatically generated

*Figure 4* is looking at Perimeter Mean. We can see that lobes that are smaller are more likely to be benign where the cancerous lobes make an appearance more frequently as the perimeter mean increases.

*Figure 5:*

Chart, histogram

Description automatically generated

Above in *Figure 5,* most of the lobes with smaller mean areas are benign where when the area size starts to increase, they become more likely to be malignant growths.

Overall, there is a common relationship of benign growths to be smaller where the larger growths are on average, more likely to be malignant.

Based on my analysis, there wasn’t any missing data but to be sure, I started by omitting any data that was missing. I also changed diagnosis into a numeric variable with 0 being benign and 1 being malignant, this way we can use binomial classification in the models.

# Predictive Modeling

We are now able to start modeling the predictive variables to see which ones should be included into the model and which model has the best performance when it comes to predicting on the variable target. There will be three models being tested:

**1.** AdaBoost Classification- AdaBoost, short for Adaptive Boosting, is a machine learning algorithm that can be used for classification problems. The main idea behind AdaBoost is to combine weak classifiers to create a strong classifier.

**2.** Random Forest- Supervised Machine Learning algorithm that builds decision trees based off the data, the trees are then averages and it will return a prediction result

**3.** Gradient Boosting- Machine Learning Booster that tries to build the next best model off the previous models and in hopes, will minimize the error.

When considering models, an ideal model is one that’s not more complex than it needs to be. This being, I ran a Random Forest Feature Importance and chose the ten most important variables when predicting on diagnosis. Below are the variables that will be included in the model:

1. concave.points\_worst
2. area\_worst
3. radius\_worst
4. perimeter\_worst
5. texture\_worst
6. texture\_mean
7. concance.points\_mean
8. area\_se
9. concavity\_worst
10. concavity\_mean

The first step was to split the data into training and test data frames. 80% went to train and 20% into test and the unbalanced data was taken into consideration during the split. Once it was split, I specified which variables would be included in the modeling process. Having too many variables can cause your models to overfit and have poor performance so only the top 10 mentioned above were implemented into the models. The performance metric that will be used is accuracy. I am using accuracy because it is a good measurement when you want to measure how often your model correctly predicts the outcome of a binary classification problem. Accuracy is calculated by dividing the number of correctly classified instances by the total number of instances in the dataset. It is shown as a percentage, with a value of 100% indicating that the model has correctly classified all cases.

### Model Performance

|  |  |  |
| --- | --- | --- |
| **Random Forest** | | |
| Trees | Depth | Performance: Accuracy |
| 300 | 3 | .9292 |
| 500 | 3 | .9204 |
| 300 | 5 | .9115 |
| 500 | 5 | .9292 |

|  |  |  |
| --- | --- | --- |
| **Gradient Boost** | | |
| Trees | Depth | Performance: Accuracy |
| 300 | 3 | .9558 |
| 500 | 3 | .9667 |
| 300 | 5 | .9735 |
| 500 | 5 | .9558 |

|  |  |  |
| --- | --- | --- |
| **AdaBoost** | | |
| Trees | Depth | Performance: Accuracy |
| 300 | 3 | .9735 |
| 500 | 3 | .9646 |
| 300 | 5 | .9735 |
| 500 | 5 | .9646 |

As you can see from the above charts, the results are very close. To see if this performance is continuous, I ran 10 iterations of each model, below are the results:

|  |  |  |
| --- | --- | --- |
| **Random Forest** | | |
| Trees | Depth | Average Accuracy |
| 300 | 3 | 0.9256637 |
| 500 | 3 | 0.9265487 |
| 300 | 5 | 0.9309735 |
| 500 | 5 | 0.9274336 |

|  |  |  |
| --- | --- | --- |
| **Gradient Boost** | | |
| Trees | Depth | Average Accuracy |
| 300 | 3 | **0.9743363 (Best)** |
| 500 | 3 | 0.9654867 |
| 300 | 5 | 0.9672566 |
| 500 | 5 | 0.9584071 |

|  |  |  |
| --- | --- | --- |
| **AdaBoost** | | |
| Trees | Depth | Average Accuracy |
| 300 | 3 | 0.9663717 |
| 500 | 3 | 0.9690265 |
| 300 | 5 | 0.9699115 |
| 500 | 5 | 0.9707965 |

The average accuracies of each model were closer than I expected. Majority of the models didn’t change very much from its singular performance I ran initially.

# Conclusion

Twelve different models were built and compared. Overall, the best model with the highest accuracy was a Gradient Boosting model with 300 trees, a depth of 3 and an accuracy score of .9743363. Coming in close at second was an AdaBoost model with 500 trees, a depth of 5 and an accuracy score of .9707965. Having an increase of estimators and depth from the Gradient model makes it more complex.

# Implications and Further Questions

What was most beneficial during this project was learning how to navigate the software and problem solve. I’m dissatisfied with the level of work I completed because of the issues I ran into along the way. A big delay was with R packages. Each method derives from different packages and each package requires a certain version of R. I spent a lot of my time researching packages, trying, and testing and for one of my models I just picked a different one to implement. This is frustrating because I know my time could have been used more efficiently tuning model parameters, engineering features, and making my predictive modeling process more complex. The biggest setback was how long it took me to debug the code. I was unaware of what the error was saying and that I could run an error checking instruction when stepping into a line of code to see where the breakdown was happening at. What was wrong was that my data needed to be in a factor object, and it was stored in a vector object. The error it gave out was that the prediction and test objects were at different levels. I tried to manually change them, reset my data, convert it to different data types, etc. Nothing was working and I finally went in to seek help because I had run out of ideas. One thing we aren’t taught in school is how to navigate software, most of the time you can just adjust your code and push it through.

The next step I would take to make this project more complex is learning hyperparameter tuning methods. In python there are methods like GridSearchCV, RandomizedSearchCV and Optuna. It would make my models more specific, detailed, and accurate.

Sources

Dataset: <https://www.kaggle.com/datasets/yasserh/breast-cancer-dataset>

Information: <https://www.mayoclinic.org/diseases-conditions/breast-cancer/symptoms-causes/syc-20352470>