Using Machine Learning Algorithms in R for Breast Cancer Predictions

Introduction

Breast cancer is something that has been around since ancient greece, and it continues to be a risk for women if they don't get regularly tested.

According to the breascancer.org website and UCdavis university the survival rate when found early is 99%.

Roughly 13% or about one in eight U.S women will develop invasive breast cancer in the course of their life.

This is why its important that there is a preventative way to seek help and have accurate screenings for breast cancer.

BREAST CANCER — BY THE NUMBERS

270K
Number of breast cancer cases diagnosed annually

99%
Survival rate for stage 1 breast cancer

45
Age most women should begin annual screenings

62%
Percentage diagnosed at an early stage

3.5M
Number of breast cancer survivors in the U.S.

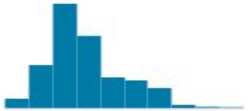
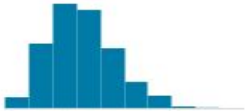
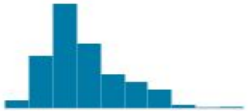
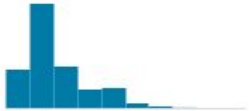
SOURCE: AMERICAN CANCER SOCIETY

The data

In order to find a way to view this problem I found a data set that was modeled on the wisconsin breast cancer data. Found here https://archive.ics.uci.edu/dataset/17/breast+cancer+wisconsin+diagnostic.

This dataset from kaggle can be found here https://www.kaggle.com/datasets/yasserh/breast-cancer-dataset

Here is a snippet of the data- As you can see in the data there are 63% benign and 37% malignant. There are also 32 features, with 569 observations.

A diagnosis Target: M - Malignant B - Benign	# radius_mean Radius of Lobes	# texture_mean Mean of Surface Texture	# perimeter_mean Outer Perimeter of Lobes	# area_mean Mean Area of Lobes
B 63% M 37%	 6.98 28.1	 9.71 39.3	 43.8 189	 144 2.5k
M	17.99	10.38	122.8	1001
M	20.57	17.77	132.9	1326
M	19.69	21.25	130	1203
M	11.42	20.38	77.58	386.1
M	20.29	14.34	135.1	1297
M	12.45	15.7	82.57	477.1
M	18.25	19.98	119.6	1040
M	13.71	20.83	90.2	577.9
M	13	21.82	87.5	519.8

The models

Now the purpose of this model is to take the feature diagnosis that has the data points of malignant and benign and use them in order to determine whether the tumor is one of those two categories.

In order to do this I will be using 4 models. Logistic regression, Random Forest, GLM net, and Support Vector Machine.

Each of these models has their own advantages and disadvantages, so we can find which of these is the best model overall for the dataset.

GLM model

This is a curvegain plot used to check how similar the actual and predicted values are.

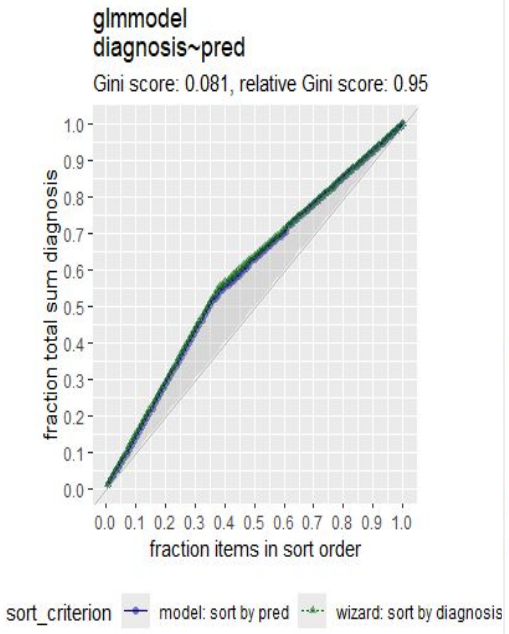
With a gini score that low its fairly similar.

I made the glm model in Rstudio three different ways in order to make sure that the data was correct and all the same regardless of the way you did it.

I did it with tidymodels and tidyverse, using the recipe and workflow functions.

Next I did it with the glm function and making my own model

Lastly I used the caret package to make one as well.




```
          Reference
Prediction  B  M
         B 69  3
         M  2 40

               Accuracy : 0.9561
                 95% CI : (0.9006, 0.9856)
    No Information Rate : 0.6228
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.9062

 Mcnemar's Test P-Value : 1

              Precision : 0.9583
                 Recall : 0.9718
                     F1 : 0.9650
             Prevalence : 0.6228
         Detection Rate : 0.6053
   Detection Prevalence : 0.6316
      Balanced Accuracy : 0.9510

       'Positive' Class : B

 Accuracy     Kappa
0.9561404 0.9062192
Generalized Linear Model

455 samples
 30 predictor
  2 classes: 'B', 'M'

Pre-processing: centered (30), scaled (30)
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 409, 410, 409, 411, 409, 410, ...
Resampling results:

  ROC        Sens       Spec
  0.9335618  0.9578818  0.8988971
```

As you can see from the data its accuracy is 95.6 %

SVM (support vector machine

A support vector machine is a model in which is uses a hyperplane to identify data as either red or blue. This classifies each of them into a category to see which side they fit on.

Using this I found that I got an accuracy of 95.6% which is the same as the glm function.

```
          Reference
Prediction  B  M
         B 69  3
         M  2 40

               Accuracy : 0.9561
                 95% CI : (0.9006, 0.9856)
    No Information Rate : 0.6228
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.9062

 Mcnemar's Test P-Value : 1

              Precision : 0.9583
                 Recall : 0.9718
                     F1 : 0.9650
             Prevalence : 0.6228
         Detection Rate : 0.6053
   Detection Prevalence : 0.6316
      Balanced Accuracy : 0.9510

       'Positive' Class : B

 Accuracy     Kappa
0.9561404 0.9062192
Support Vector Machines with Linear Kernel

455 samples
 30 predictor
  2 classes: 'B', 'M'

Pre-processing: centered (30), scaled (30)
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 410, 409, 409, 409, 410, 409, ...
Resampling results:

  ROC        Sens       Spec
  0.9912018  0.9858374  0.9231618
```

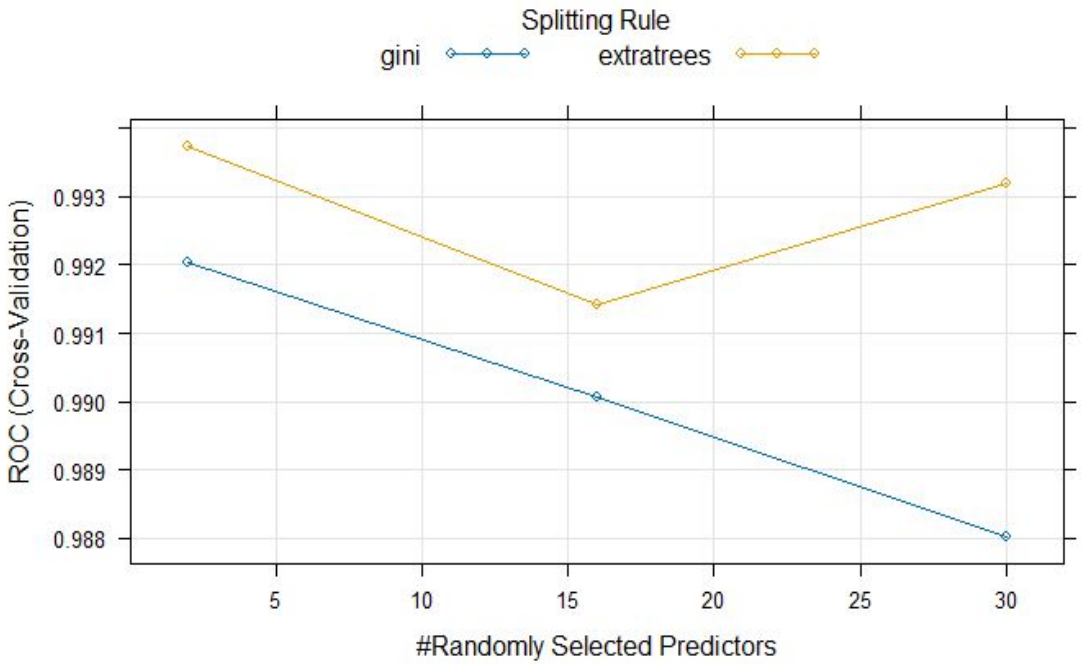
Random Forest model

What this does is it takes in the dataset, puts them into different groups and has each “expert” vote on a final decision. These subsets of data each contribute to the overall accuracy at the end along with its predictions.

These experts help figure out the model and the accuracy.

As you can see from the model it has an accuracy of 99%.

One thing to note is that I have the tuning meter set to 3 and each run of the forest will give a different accuracy. It averages out between 98-99% and rarely ever has went under those.



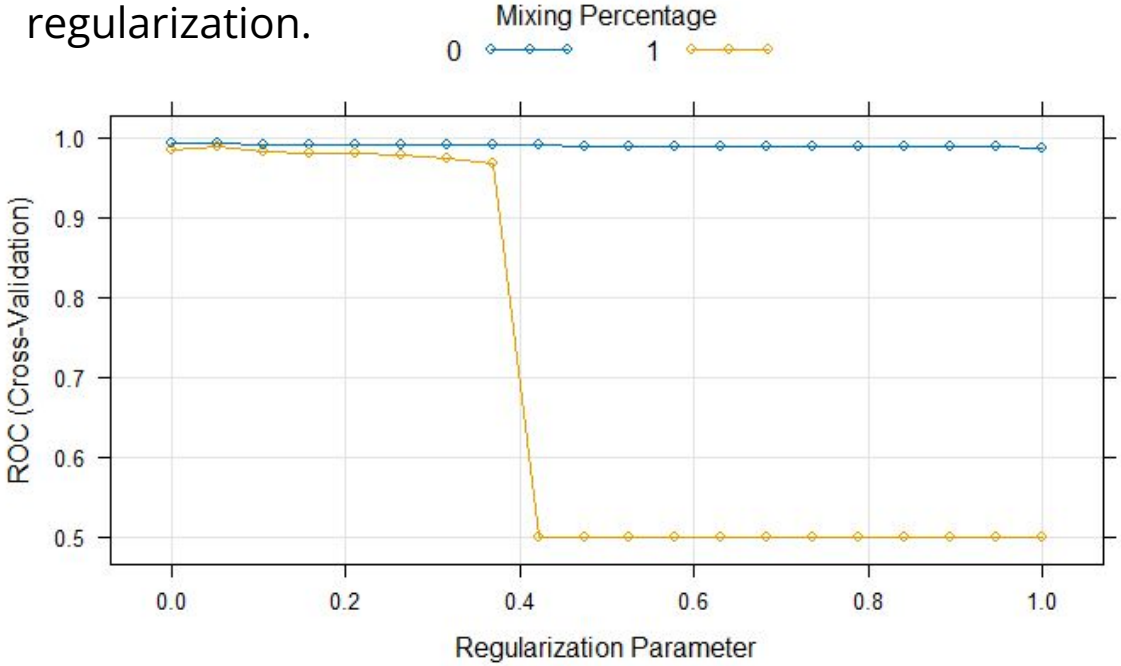

```
          Reference
Prediction  B  M
         B 71  1
         M  0 42

               Accuracy : 0.9912
                 95% CI : (0.9521, 0.9998)
    No Information Rate : 0.6228
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.9812

 Mcnemar's Test P-Value : 1

              Precision : 0.9861
                 Recall : 1.0000
                     F1 : 0.9930
             Prevalence : 0.6228
         Detection Rate : 0.6228
   Detection Prevalence : 0.6316
      Balanced Accuracy : 0.9884

       'Positive' Class : B

  Accuracy     Kappa
0.9912281 0.9812438
```

GLMnet model

This is a package that fits generalized linear and similar models via a penalized maximum likelihood. It uses regularization as a method to prevent overfitting which also uses the lasso, ridge, and elastic net regression to do this.

The accuracy for this model is 98.3%

This graph shows the differences in how the lasso and ridge regularization.




```
          Reference
Prediction  B  M
         B 71  2
         M  0 41

               Accuracy : 0.9825
                 95% CI : (0.9381, 0.9979)
    No Information Rate : 0.6228
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.9623

 Mcnemar's Test P-Value : 0.4795

              Precision : 0.9726
                 Recall : 1.0000
                     F1 : 0.9861
             Prevalence : 0.6228
         Detection Rate : 0.6228
   Detection Prevalence : 0.6404
      Balanced Accuracy : 0.9767

       'Positive' Class : B

 Accuracy     Kappa
0.9824561 0.9623140
[1] 0.9929531
```

Conclusion

Looking a the comparisons you can see that a lot of models have a high accuracy. The one who ends up having the highest overall is the random forest, but the glmnet is the most consistent while being only slightly worse.

```
Models: glmnet, randomforest, glm, SVM
Number of resamples: 10

ROC
                Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
glmnet       0.9768908 0.9893826 0.9948928 0.9929531 1.0000000 1.000000    0
randomforest 0.9787018 0.9887714 0.9958798 0.9937283 1.0000000 1.000000    0
glm          0.8526786 0.9234280 0.9469538 0.9335618 0.9652094 0.979716    0
SVM          0.9391481 0.9957983 0.9958708 0.9912018 0.9974102 1.000000    0

Sens
                Min.   1st Qu.    Median      Mean   3rd Qu. Max. NA's
glmnet       0.9655172 1.0000000 1.0000000 0.9931034 1.0000000    1    0
randomforest 0.9285714 0.9655172 0.9827586 0.9753695 1.0000000    1    0
glm          0.8928571 0.9310345 0.9649015 0.9578818 0.9913793    1    0
SVM          0.9285714 0.9741379 1.0000000 0.9858374 1.0000000    1    0

Spec
                Min.   1st Qu.    Median      Mean   3rd Qu. Max. NA's
glmnet       0.8235294 0.8823529 0.9393382 0.9231618 0.9411765    1    0
randomforest 0.7647059 0.8823529 0.8823529 0.9055147 0.9402574    1    0
glm          0.7058824 0.8823529 0.9117647 0.8988971 0.9411765    1    0
SVM          0.8235294 0.8823529 0.9393382 0.9231618 0.9411765    1    0
```