Diagnosing Cancer Tumors with Machine Learning

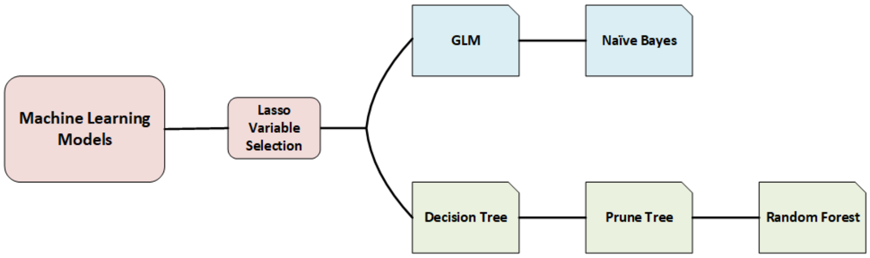
By: Pratik Ghawate

Introduction:

Machine learning is a powerful concept, and tool that can be applied to an unlimited number of applications in the world. In this case, I am using machine learning to work with cancer data to help diagnose whether a specific tumor will be benign, or malignant. A benign tumor is a tumor that is not cancerous and does not spread (NCI Stats, n.d.). Whereas a malignant tumor is cancerous, and could spread throughout the body (NCI Stats, n.d.). Using machine learning for something as important as cancer can yield important results, and insights in order to learn, ideally come up with solutions for cancer.

Cancer research is a good application for machine learning since it is a deadly condition, and it affects individuals worldwide. In the United States, cancer is the second leading cause of death, and in 2023 approximately two million individuals will be diagnosed with some form of cancer this year (NCI, 2023). In this paper, I focus on breast cancer which is one of the most common cancer diagnoses since it approximately makes up 15% of all diagnoses (NCI, 2023).

The following sections will go through the breast cancer tumor data I have prepared, and the machine-learning models I used. The models tested are: GLM Naive Bayes, Decision Tree, Pruned Tree, and Random Forest. Below is the general methodology for testing our data:

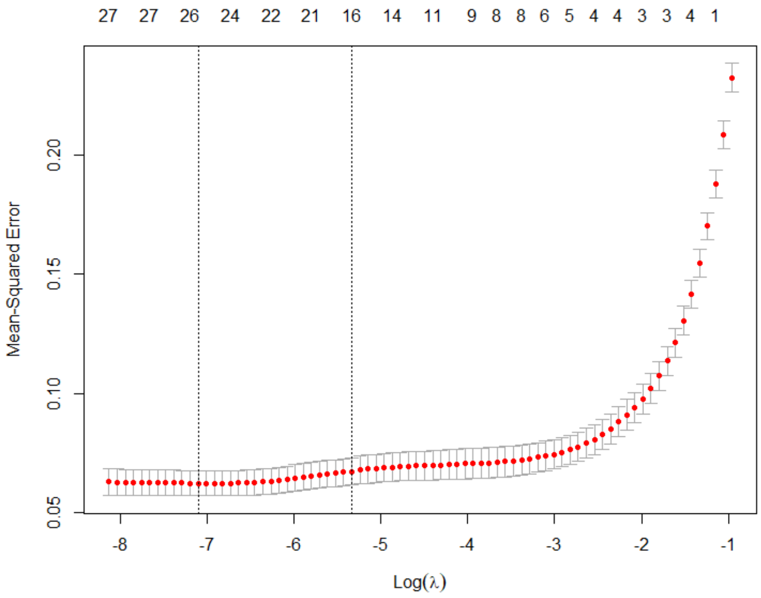


The best model for data will be chosen off of several criteria, and conclusions will be discussed.

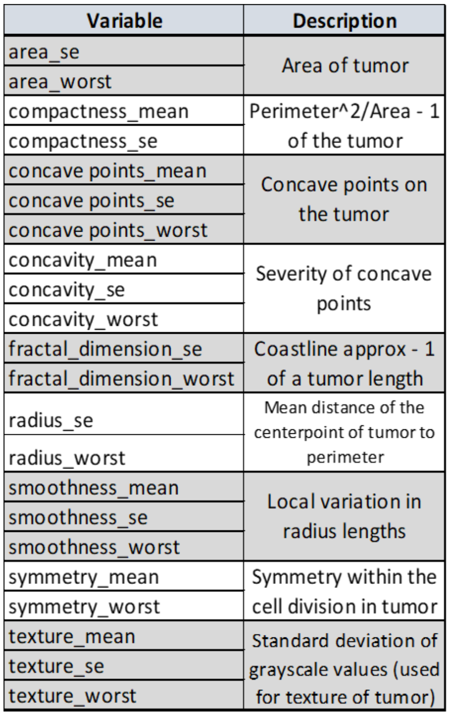
Data & Variables:

The data used for the models was obtained from Kaggle.com, and the data file is called “Cancer Data, Benign and Malignant Cancer Data.” The data set has observations on 570 different tumors. There are 30 independant quantitative variables that describe the tumor, and one qualitative response variable that describes the diagnosis as benign or malignant. Additionally, the data set was easy to work with since there was no missing data.

In terms of data preparation, there were several steps I took before I could run the models. First, since the response variable is qualitative I wanted to change it to quantitative value. Therefore, I created a new variable “response” that takes the place of the original response variable, which was “diagnoses.” The original values were benign, and malignant which were changed to “0” and “1” respectively. The next step was to determine what variables I wanted to include in each model. A preliminary test of a GLM model yielded low accuracy results so it was determined that I need to use variable selection. The data was split 75%/25% (this split was used throughout) and Lasso variable selection was performed to find the most significant variables. Below shows the Mean Squared Error for different numbers of variables from the Lasso selection:



The results from the Lasso selection returned 22 significant variables which will be used for all models. Below are the variables used, and descriptions of each variable:



Variable	Description
area_se	Area of tumor
area_worst	
compactness_mean	Perimeter^2/Area - 1 of the tumor
compactness_se	
concave points_mean	Concave points on the tumor
concave points_se	
concave points_worst	
concavity_mean	Severity of concave points
concavity_se	
concavity_worst	
fractal_dimension_se	Coastline approx - 1 of a tumor length
fractal_dimension_worst	
radius_se	Mean distance of the centerpoint of tumor to perimeter
radius_worst	
smoothness_mean	Local variation in radius lengths
smoothness_se	
smoothness_worst	
symmetry_mean	Symmetry within the cell division in tumor
symmetry_worst	
texture_mean	Standard deviation of grayscale values (used for texture of tumor)
texture_se	
texture_worst	

Notes:

SE= Standard Error

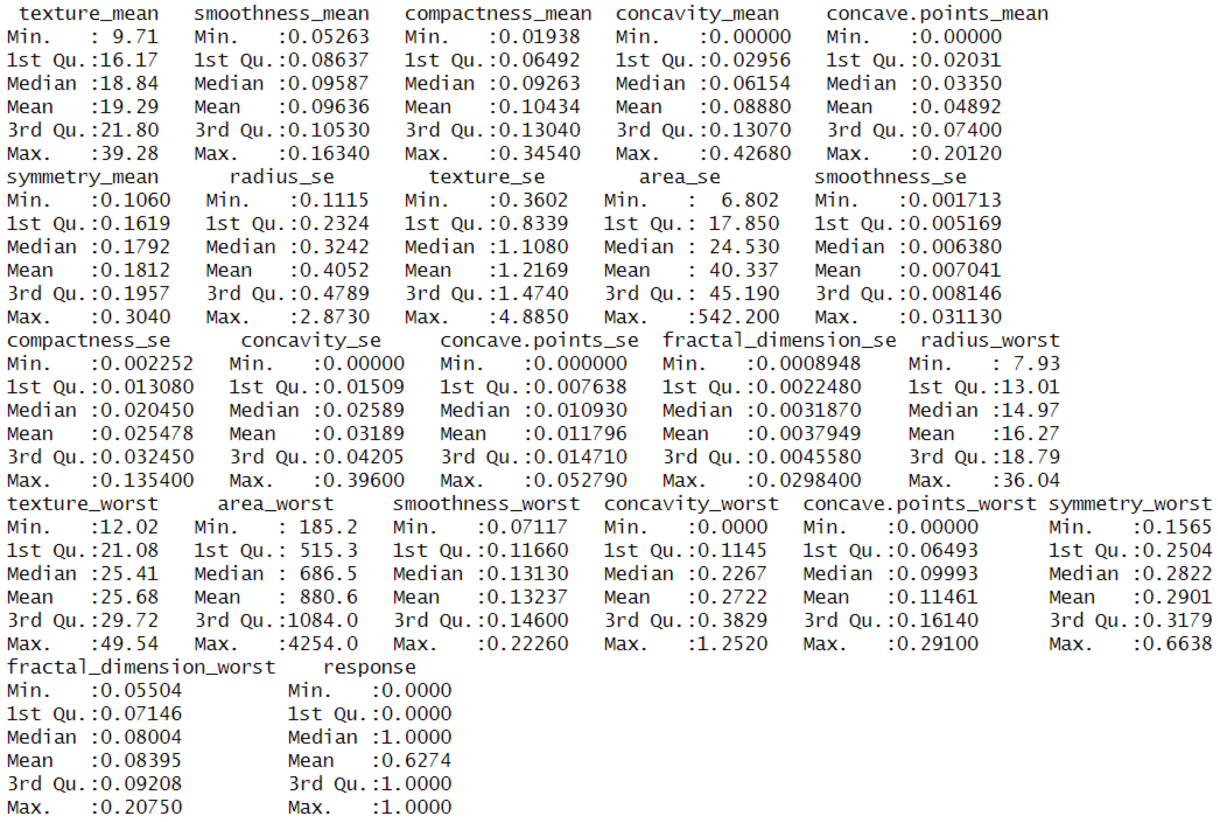
Mean= Mean Value

Worst= Mean of the 3 Largest/Worst Values

Using only 22 of the variables significantly increased the accuracy of the GLM (which will be discussed later). The following section will discuss the summary statistics and data visualization of the variables used.

Statistics & Data Visualization:

Below are the summary statistics for the 22 variables that were used for the different models. Each variable has the minimum, median, mean, and maximum values with the other quartiles shown throughout the summary.



```
  texture_mean   smoothness_mean   compactness_mean  concavity_mean    concave.points_mean
 Min.   : 9.71   Min.   :0.05263   Min.   :0.01938   Min.   :0.00000   Min.   :0.00000    
 1st Qu.:16.17   1st Qu.:0.08637   1st Qu.:0.06492   1st Qu.:0.02956   1st Qu.:0.02031    
 Median :18.84   Median :0.09587   Median :0.09263   Median :0.06154   Median :0.03350    
 Mean   :19.29   Mean   :0.09636   Mean   :0.10434   Mean   :0.08880   Mean   :0.04892    
 3rd Qu.:21.80   3rd Qu.:0.10530   3rd Qu.:0.13040   3rd Qu.:0.13070   3rd Qu.:0.07400    
 Max.   :39.28   Max.   :0.16340   Max.   :0.34540   Max.   :0.42680   Max.   :0.20120    
 symmetry_mean      radius_se        texture_se        area_se         smoothness_se     
 Min.   :0.1060   Min.   :0.1115   Min.   :0.3602   Min.   :  6.802   Min.   :0.001713  
 1st Qu.:0.1619   1st Qu.:0.2324   1st Qu.:0.8339   1st Qu.: 17.850   1st Qu.:0.005169  
 Median :0.1792   Median :0.3242   Median :1.1080   Median : 24.530   Median :0.006380  
 Mean   :0.1812   Mean   :0.4052   Mean   :1.2169   Mean   : 40.337   Mean   :0.007041  
 3rd Qu.:0.1957   3rd Qu.:0.4789   3rd Qu.:1.4740   3rd Qu.: 45.190   3rd Qu.:0.008146  
 Max.   :0.3040   Max.   :2.8730   Max.   :4.8850   Max.   :542.200   Max.   :0.031130  
 compactness_se      concavity_se     concave.points_se  fractal_dimension_se  radius_worst  
 Min.   :0.002252   Min.   :0.00000   Min.   :0.000000   Min.   :0.0008948    Min.   : 7.93  
 1st Qu.:0.013080   1st Qu.:0.01509   1st Qu.:0.007638   1st Qu.:0.0022480    1st Qu.:13.01  
 Median :0.020450   Median :0.02589   Median :0.010930   Median :0.0031870    Median :14.97  
 Mean   :0.025478   Mean   :0.03189   Mean   :0.011796   Mean   :0.0037949    Mean   :16.27  
 3rd Qu.:0.032450   3rd Qu.:0.04205   3rd Qu.:0.014710   3rd Qu.:0.0045580    3rd Qu.:18.79  
 Max.   :0.135400   Max.   :0.39600   Max.   :0.052790   Max.   :0.0298400    Max.   :36.04  
 texture_worst     area_worst      smoothness_worst  concavity_worst  concave.points_worst symmetry_worst  
 Min.   :12.02   Min.   : 185.2   Min.   :0.07117   Min.   :0.0000   Min.   :0.00000      Min.   :0.1565  
 1st Qu.:21.08   1st Qu.: 515.3   1st Qu.:0.11660   1st Qu.:0.1145   1st Qu.:0.06493      1st Qu.:0.2504  
 Median :25.41   Median : 686.5   Median :0.13130   Median :0.2267   Median :0.09993      Median :0.2822  
 Mean   :25.68   Mean   : 880.6   Mean   :0.13237   Mean   :0.2722   Mean   :0.11461      Mean   :0.2901  
 3rd Qu.:29.72   3rd Qu.:1084.0   3rd Qu.:0.14600   3rd Qu.:0.3829   3rd Qu.:0.16140      3rd Qu.:0.3179  
 Max.   :49.54   Max.   :4254.0   Max.   :0.22260   Max.   :1.2520   Max.   :0.29100      Max.   :0.6638  
 fractal_dimension_worst    response     
 Min.   :0.05504         Min.   :0.0000  
 1st Qu.:0.07146         1st Qu.:0.0000  
 Median :0.08004         Median :1.0000  
 Mean   :0.08395         Mean   :0.6274  
 3rd Qu.:0.09208         3rd Qu.:1.0000  
 Max.   :0.20750         Max.   :1.0000  
```

Looking through the summary statistics, it was difficult to visualize the data so I decided to use two different styles of data visualization to be able to interpret the data much better.

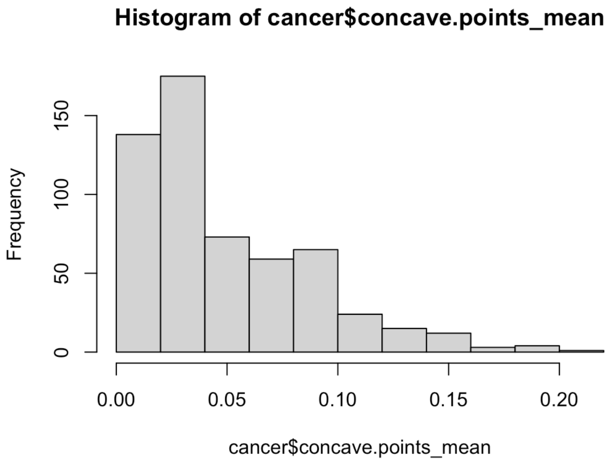
To start off, I decided to use histograms for each variable and plot them to visually see the clusters and frequency of our data.

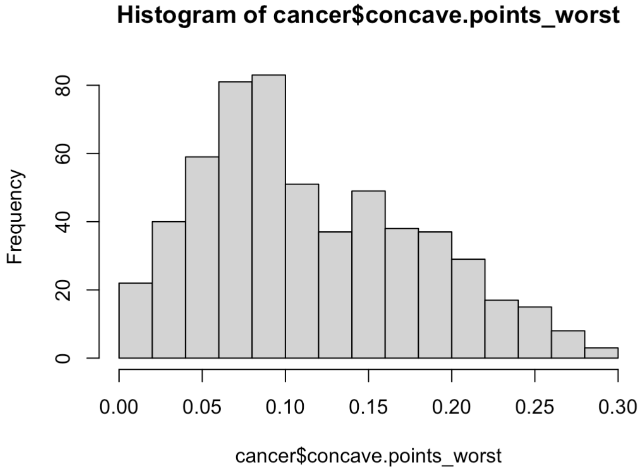
Histogram

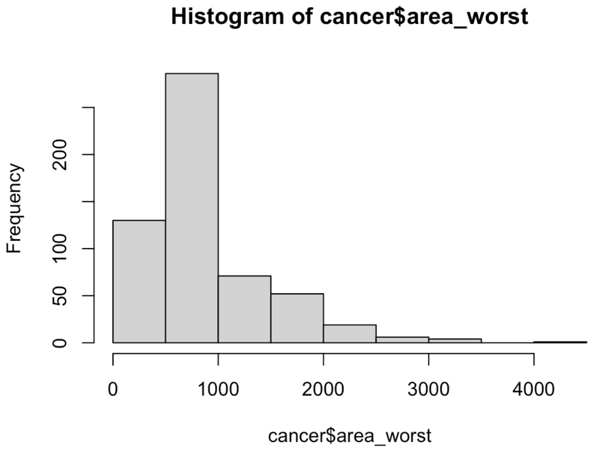
Shown below are the 4 most significant variables out of the 22 variables that I used shown in a histogram. (The rest are shown in the appendix).

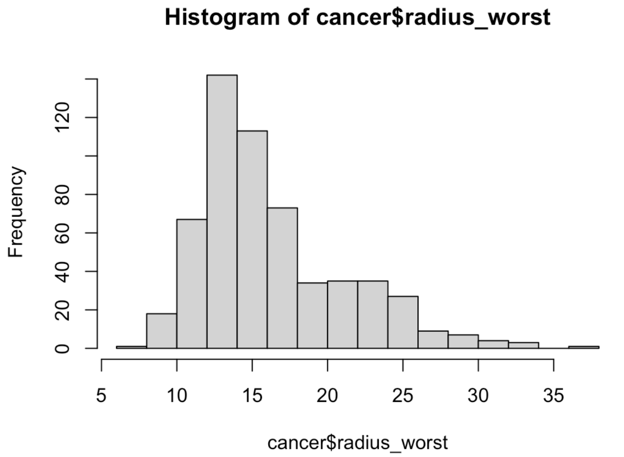
The four variables that are shown below are **concave points mean**, **concave points worst, area worst, and radius worst.** Through these histograms you are able to see the distribution of the data for each variable. For example, you can see that most of the values from concave points mean are between 0.00-0.05 as they are the most frequent. This type of visualization also makes it much easier to see which values from each variable are common and least common as well.

The problem with the histograms is that you can’t differentiate if these cells and values that are retracted from are either benign or malignant. I then decided to investigate box plots to combat that problem.


Histogram of cancer$concave.points_mean
Frequency
0
50
100
150
0.00
0.05
0.10
0.15
0.20
cancer$concave.points_mean


Histogram of cancer$concave.points_worst
Frequency
0
20
40
60
80
0.00
0.05
0.10
0.15
0.20
0.25
0.30
cancer$concave.points_worst


Histogram of cancer$area_worst
Frequency
0
50
100
200
0
1000
2000
3000
4000
cancer$area_worst

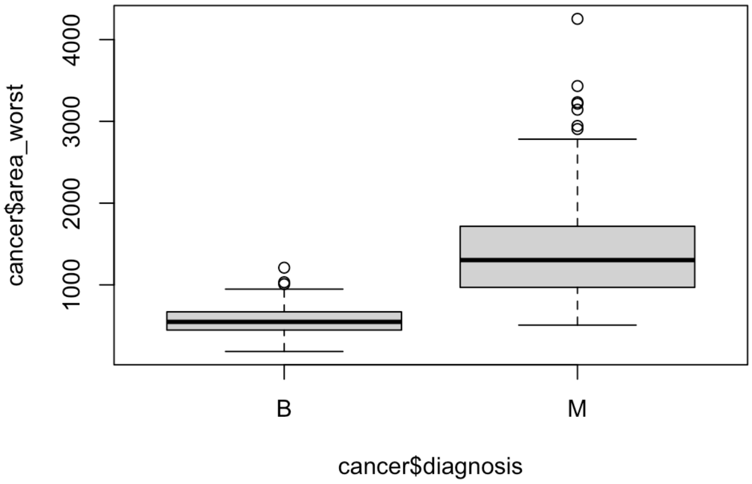

Histogram of cancer$radius_worst
Frequency
0
20
40
60
80
120
5
10
15
20
25
30
35
cancer$radius_worst

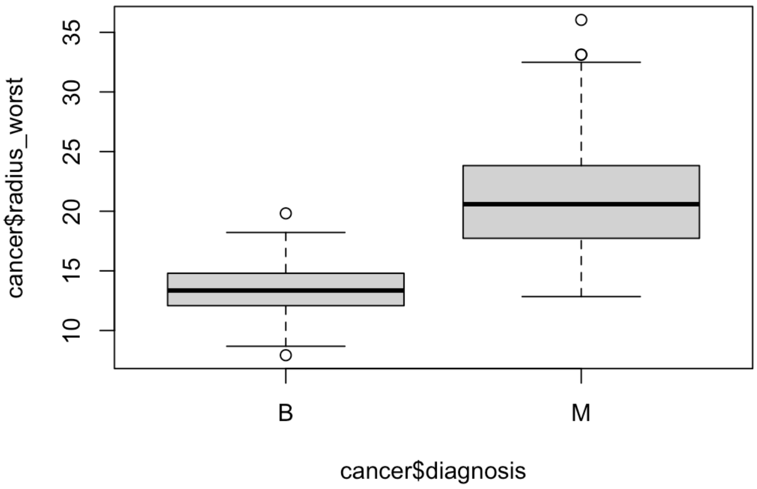
Box Plots

Shown below are the same 4 significant variables (concave points mean, concave points worst, area worst, and radius worst) in box plots (the rest are shown in the appendix).

Unlike the histogram, the box plots can differentiate between the benign and malignant cells shown on the X-axis of the box plots. I can extract the median, maximum, minimum, different quartiles and the outliers from each variable while sorting them out based on if the cells are cancerous or not.

I are able to see that the values for each variable are higher in general when the cells are malignant compared to the benign non cancerous cells.





Models:

GLM Model

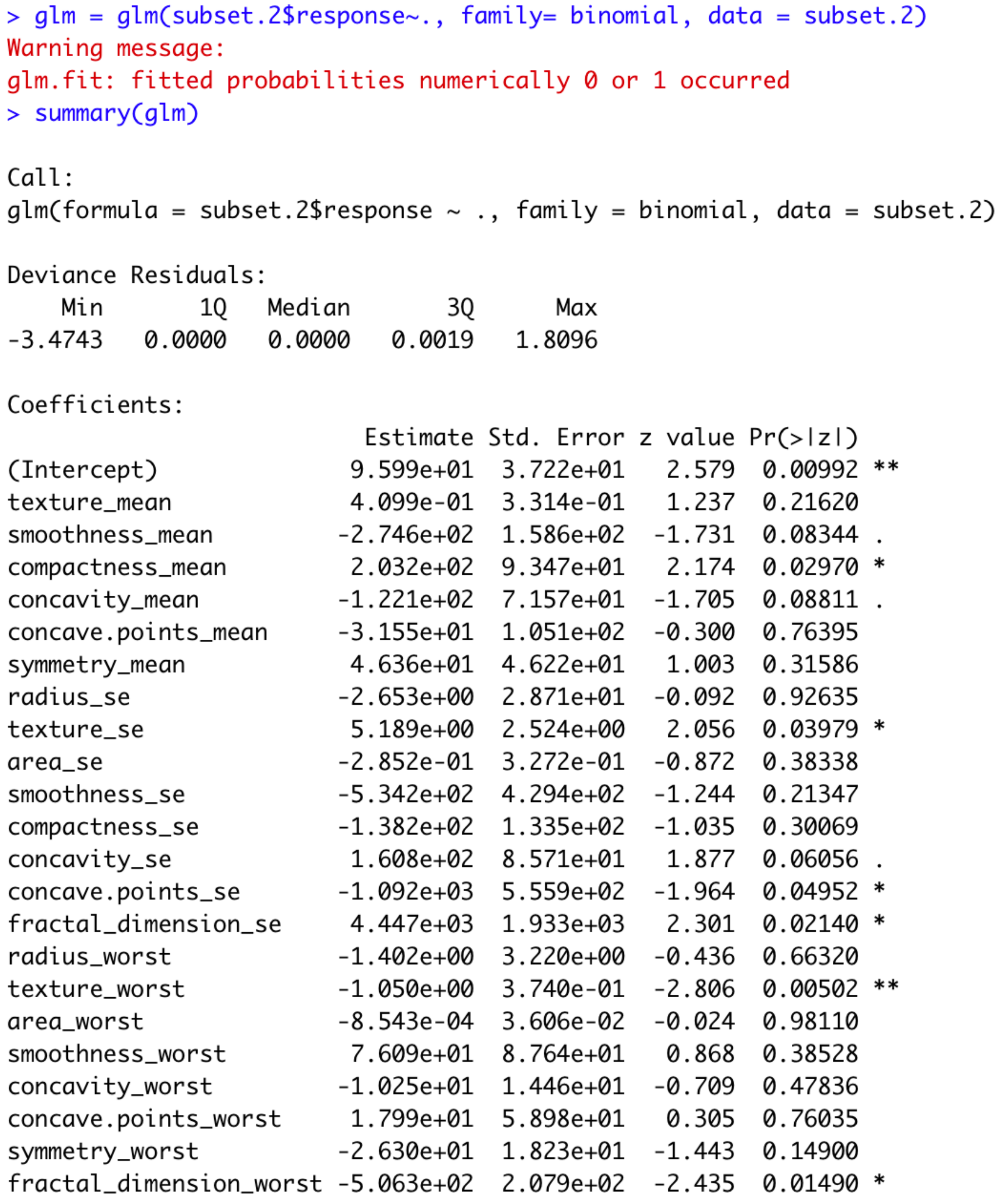
Advantages:

1. Flexibility: GLMs can be used to model a wide range of response variables, including binary, count, and continuous variables.
2. Interpretable: GLMs provide interpretable model coefficients that can be used to understand the relationship between the predictor variables and the response variable.
3. Can handle non-normal distributions: GLMs are able to handle non-normal response variables, such as count data, binary data, or skewed continuous data, by using appropriate link functions.

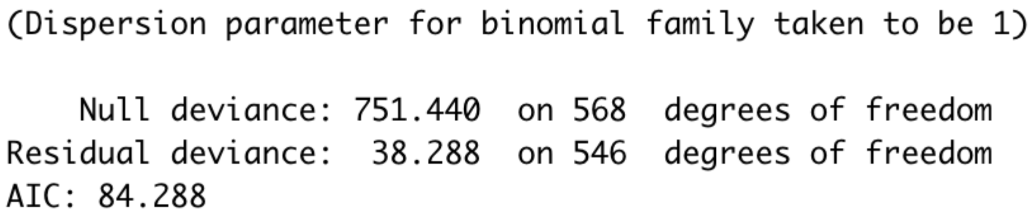
4. Efficient: GLMs are computationally efficient and can handle large datasets with many predictor variables.

Disadvantages:

1. Limited application: GLMs are not suitable for modeling data where the relationship between the response and predictor variables is complex or non-linear.
2. Assumes independence: GLMs assume that the observations are independent, which may not be true in some cases where the data is clustered or correlated.
3. Requires correct specification: GLMs require correct specification of the link function and distribution of the response variable, which can be challenging in some cases.
4. Not always robust: GLMs are sensitive to outliers and influential observations, which can affect the accuracy and reliability of the model.



```
> glm = glm(subset.2$response~., family= binomial, data = subset.2)
Warning message:
glm.fit: fitted probabilities numerically 0 or 1 occurred 
> summary(glm)

Call:
glm(formula = subset.2$response ~ ., family = binomial, data = subset.2)

Deviance Residuals: 
    Min       1Q   Median       3Q      Max  
-3.4743   0.0000   0.0000   0.0019   1.8096  

Coefficients:
                          Estimate Std. Error z value Pr(>|z|)   
(Intercept)              9.599e+01  3.722e+01   2.579  0.00992 **
texture_mean             4.099e-01  3.314e-01   1.237  0.21620   
smoothness_mean         -2.746e+02  1.586e+02  -1.731  0.08344 . 
compactness_mean         2.032e+02  9.347e+01   2.174  0.02970 * 
concavity_mean          -1.221e+02  7.157e+01  -1.705  0.08811 . 
concave.points_mean     -3.155e+01  1.051e+02  -0.300  0.76395   
symmetry_mean            4.636e+01  4.622e+01   1.003  0.31586   
radius_se               -2.653e+00  2.871e+01  -0.092  0.92635   
texture_se               5.189e+00  2.524e+00   2.056  0.03979 * 
area_se                 -2.852e-01  3.272e-01  -0.872  0.38338   
smoothness_se           -5.342e+02  4.294e+02  -1.244  0.21347   
compactness_se          -1.382e+02  1.335e+02  -1.035  0.30069   
concavity_se             1.608e+02  8.571e+01   1.877  0.06056 . 
concave.points_se       -1.092e+03  5.559e+02  -1.964  0.04952 * 
fractal_dimension_se     4.447e+03  1.933e+03   2.301  0.02140 * 
radius_worst            -1.402e+00  3.220e+00  -0.436  0.66320   
texture_worst           -1.050e+00  3.740e-01  -2.806  0.00502 **
area_worst              -8.543e-04  3.606e-02  -0.024  0.98110   
smoothness_worst         7.609e+01  8.764e+01   0.868  0.38528   
concavity_worst         -1.025e+01  1.446e+01  -0.709  0.47836   
concave.points_worst     1.799e+01  5.898e+01   0.305  0.76035   
symmetry_worst          -2.630e+01  1.823e+01  -1.443  0.14900   
fractal_dimension_worst -5.063e+02  2.079e+02  -2.435  0.01490 * 
```



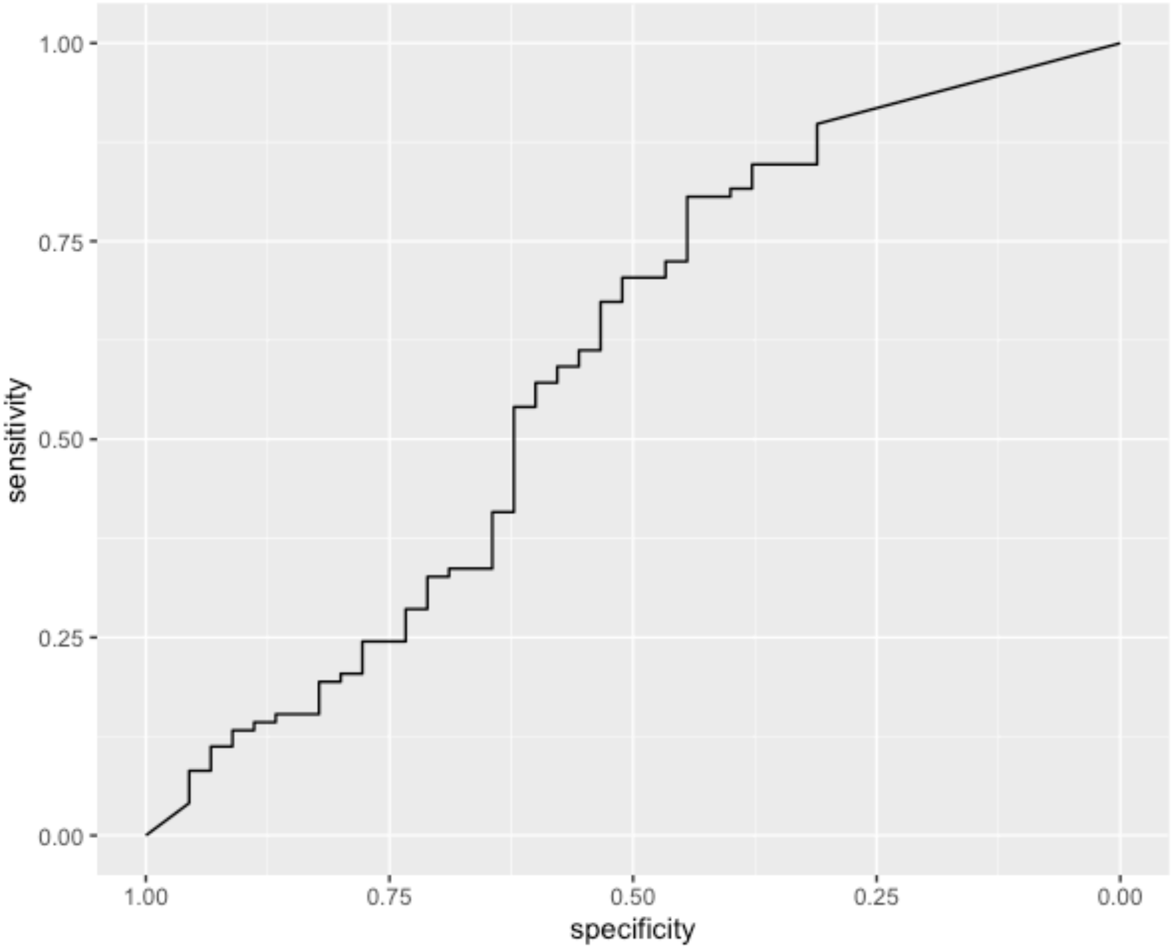
```
(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 751.440  on 568  degrees of freedom
Residual deviance:  38.288  on 546  degrees of freedom
AIC: 84.288
```

Confusion matrix:

```
> glm.sum= confusionMatrix(data= as.factor(glm.class), reference=as.factor(test1$response), positive="1")
> glm.sum
Confusion Matrix and Statistics

          Reference
Prediction  0  1
         0 24 33
         1 21 65
                                          
               Accuracy : 0.6224          
                 95% CI : (0.5375, 0.702) 
    No Information Rate : 0.6853          
    P-Value [Acc > NIR] : 0.9547          
                                          
                  Kappa : 0.1834          
                                          
 Mcnemar's Test P-Value : 0.1344          
                                          
            Sensitivity : 0.6633          
            Specificity : 0.5333          
         Pos Pred Value : 0.7558          
         Neg Pred Value : 0.4211          
             Prevalence : 0.6853          
         Detection Rate : 0.4545          
   Detection Prevalence : 0.6014          
      Balanced Accuracy : 0.5983          
                                          
       'Positive' Class : 1               
                                          
>
```

For GLM, I fit the model using the variables I selected from Lasso Variable selection, I get AIC of 84.288 and using variables elected by Lasso variable selection confusion matrix gave an accuracy of 62.24 % with sensitivity and specificity of 0.6633 and 0.5333 respectively.

I get Area Under the Curve of ROC as 0.595 and the ROC curve is shown in below figure.



Naive Bayesian Model

Advantages:

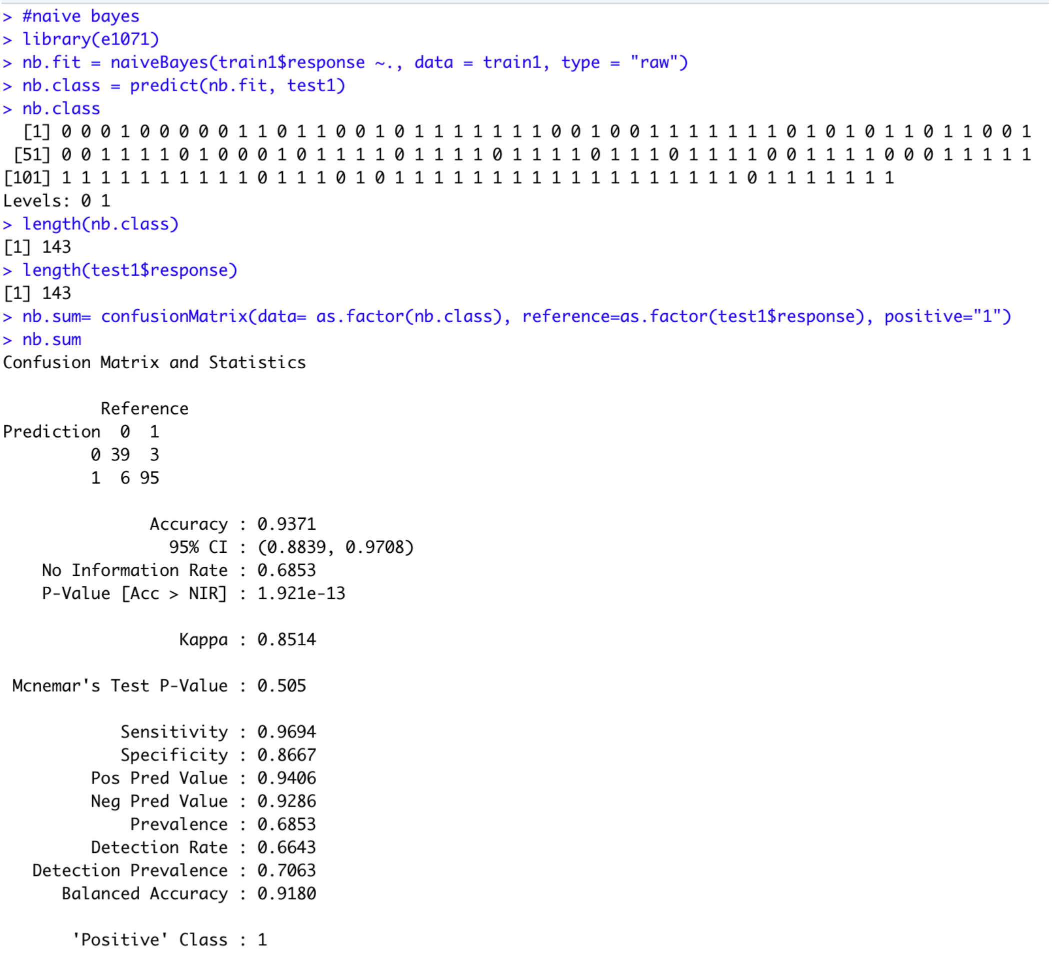
1. Simplicity: Naive Bayes is easy to understand and implement. It is based on simple probabilistic principles and makes strong assumptions about the independence of features, which simplifies the modeling process.
2. Efficiency: Naive Bayes is computationally efficient and can train models quickly, even on large datasets. It requires a small amount of training data to estimate the parameters accurately.
3. Scalability: Due to its simplicity, Naive Bayes performs well in high-dimensional spaces, making it suitable for problems with a large number of features. It can handle a large number of predictors efficiently.
4. Interpretability: The model's decision-making process is based on simple probabilities, which makes it highly interpretable. It can provide insights into how each feature contributes to the classification.

Disadvantages:

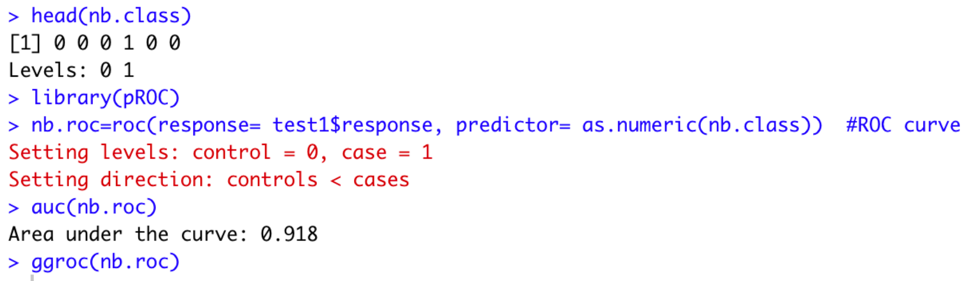
1. Independence assumption: The Naive Bayes algorithm assumes that all features are independent, which is often an oversimplified assumption. In real-world scenarios, features are often correlated, and this assumption may not hold true, leading to suboptimal performance.
2. Limited expressiveness: Due to its simplicity, Naive Bayes may not capture complex relationships in the data as well as more advanced models like neural

networks or decision trees. It may struggle with capturing interactions between features.

3. Data scarcity: Naive Bayes relies on the availability of sufficient training data to estimate the probabilities accurately. If the dataset is small or certain classes are underrepresented, it may lead to biased or unreliable probability estimates.

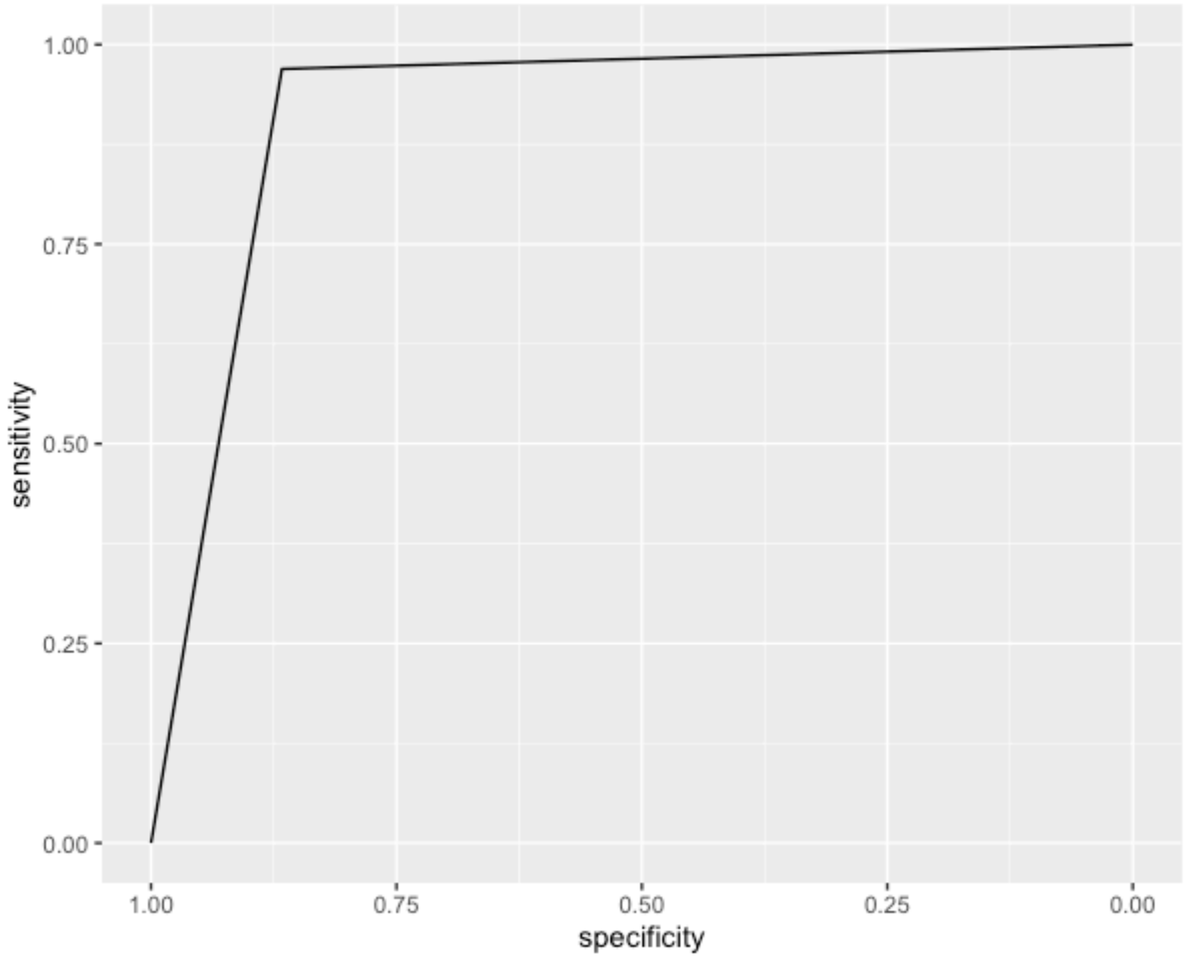


```
> #naive bayes
> library(e1071)
> nb.fit = naiveBayes(train1$response ~., data = train1, type = "raw")
> nb.class = predict(nb.fit, test1)
> nb.class
  [1] 0 0 0 1 0 0 0 0 0 1 1 0 1 1 0 0 1 0 1 1 1 1 1 1 1 0 0 1 0 0 1 1 1 1 1 1 1 0 1 0 1 0 1 1 0 1 1 0 0 1
 [51] 0 0 1 1 1 1 0 1 0 0 0 1 0 1 1 1 1 0 1 1 1 1 0 1 1 1 1 0 1 1 1 0 1 1 1 1 0 0 1 1 1 1 0 0 0 1 1 1 1 1
[101] 1 1 1 1 1 1 1 1 1 1 0 1 1 1 0 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1
Levels: 0 1
> length(nb.class)
[1] 143
> length(test1$response)
[1] 143
> nb.sum= confusionMatrix(data= as.factor(nb.class), reference=as.factor(test1$response), positive="1")
> nb.sum
Confusion Matrix and Statistics

          Reference
Prediction  0  1
         0 39  3
         1  6 95

               Accuracy : 0.9371
                 95% CI : (0.8839, 0.9708)
    No Information Rate : 0.6853
    P-Value [Acc > NIR] : 1.921e-13

                  Kappa : 0.8514

 Mcnemar's Test P-Value : 0.505

            Sensitivity : 0.9694
            Specificity : 0.8667
         Pos Pred Value : 0.9406
         Neg Pred Value : 0.9286
             Prevalence : 0.6853
         Detection Rate : 0.6643
   Detection Prevalence : 0.7063
      Balanced Accuracy : 0.9180

       'Positive' Class : 1
```



```
> head(nb.class)
[1] 0 0 0 1 0 0
Levels: 0 1
> library(pROC)
> nb.roc=roc(response= test1$response, predictor= as.numeric(nb.class))  #ROC curve
Setting levels: control = 0, case = 1
Setting direction: controls < cases
> auc(nb.roc)
Area under the curve: 0.918
> ggroc(nb.roc)
```

Naive Bayesian model gave us an accuracy of 93.71% with AUC of 91.8 % and ROC curve as follows:



Decision Tree

Advantages:

1. Simple and Easy to Understand: Decision trees may be viewed and understood by non-experts, making them simple and easy to understand. An understandable and simple-to-follow tree-like representation of the decisions and rules is used.

2. Feature Selection: Decision Trees, which divide the data according to the value and relevance of the characteristics, can aid in identifying the most crucial features for prediction and feature selection.

3. Decision Trees are non-parametric, which means that no assumptions regarding the distribution of the underlying data or the association between the features and the goal variable are necessary.

4. Handles Both category and Continuous Data: Decision Trees are adaptable and useful for a variety of data types since they can handle both category and continuous data.

Disadvantages:

1. Overfitting: If the tree is too deep or the data is noisy, decision trees may overfit the data, which results in poor generalization performance on fresh data.

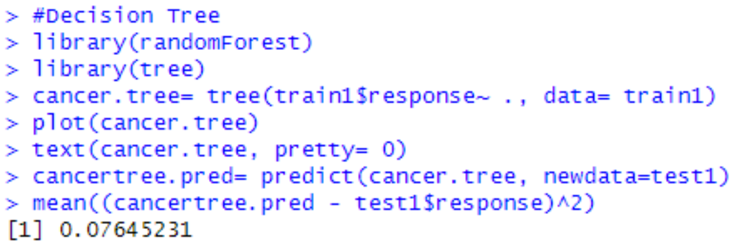
2. Instability: Because the tree structure can alter depending on the training data, Decision Trees might be unstable and sensitive to tiny changes in the data.

3. Bias: Because decision trees tend to divide the data based on those features more frequently, they may be biased in favor of features with more levels or categories.

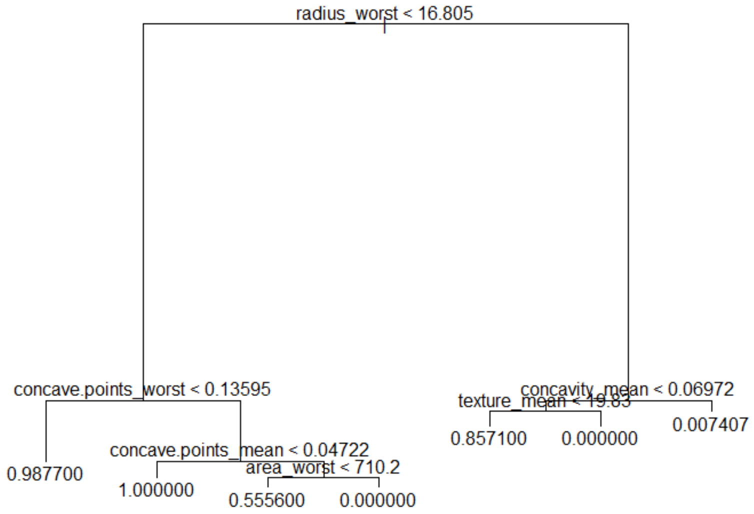
4. Limited Predictive Power: Because decision trees rely on straightforward rules and splits, they might be ineffective when there is a complex or non-linear relationship between the features and the target variable.

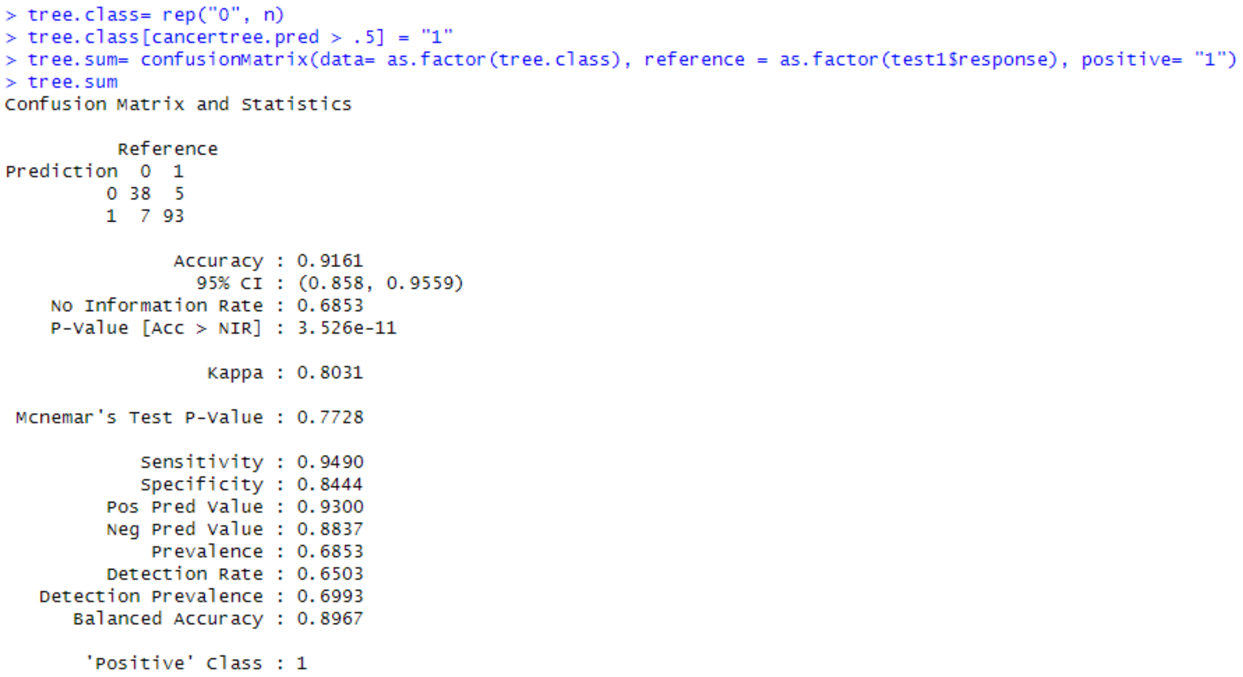
Model selection:

I fit the decision tree using the train dataset. The train dataset which I used in fitting the decision tree has the variables which I selected from Lasso regression. The decision tree has 7 terminal nodes. After the decision tree, I did the prediction using a test dataset. And finally measured the accuracy using the confusion matrix and I got the accuracy of 91.61% with sensitivity and specificity of 0.9490 and 0.8444 respectively.

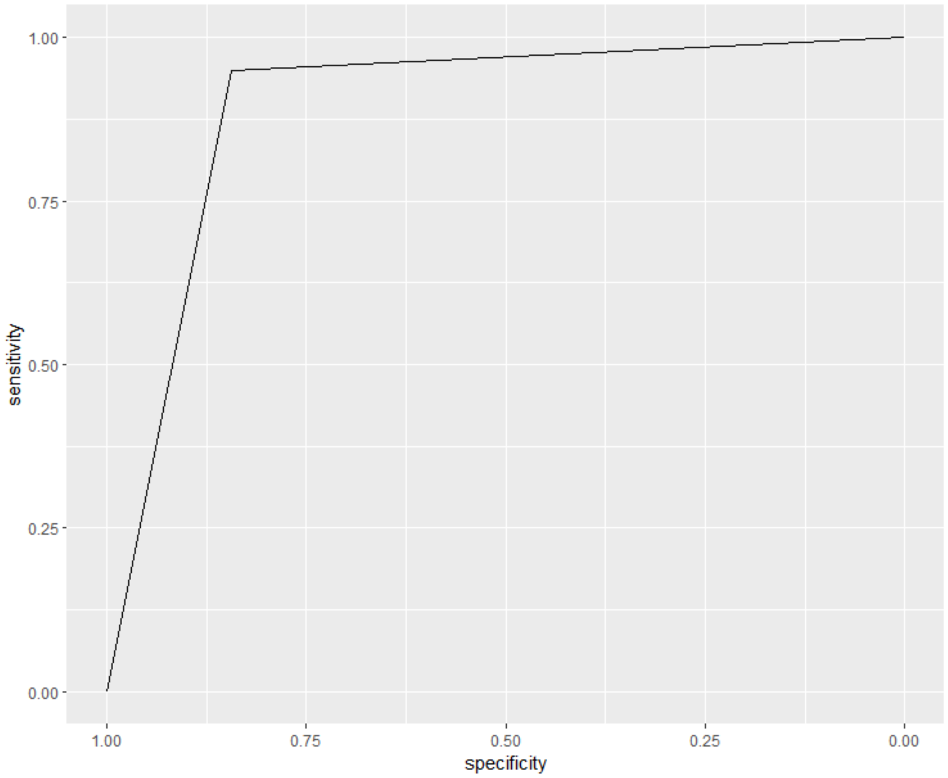


```
> #Decision Tree
> library(randomForest)
> library(tree)
> cancer.tree= tree(train1$response~ ., data= train1)
> plot(cancer.tree)
> text(cancer.tree, pretty= 0)
> cancertree.pred= predict(cancer.tree, newdata=test1)
> mean((cancertree.pred - test1$response)^2)
[1] 0.07645231
```





```
> tree.class= rep("0", n)
> tree.class[cancertree.pred > .5] = "1"
> tree.sum= confusionMatrix(data= as.factor(tree.class), reference = as.factor(test1$response), positive= "1")
> tree.sum
Confusion Matrix and Statistics

          Reference
Prediction  0  1
         0 38  5
         1  7 93

               Accuracy : 0.9161
                 95% CI : (0.858, 0.9559)
    No Information Rate : 0.6853
    P-Value [Acc > NIR] : 3.526e-11

                  Kappa : 0.8031

 Mcnemar's Test P-Value : 0.7728

            Sensitivity : 0.9490
            Specificity : 0.8444
         Pos Pred Value : 0.9300
         Neg Pred Value : 0.8837
             Prevalence : 0.6853
         Detection Rate : 0.6503
   Detection Prevalence : 0.6993
      Balanced Accuracy : 0.8967

       'Positive' Class : 1
```



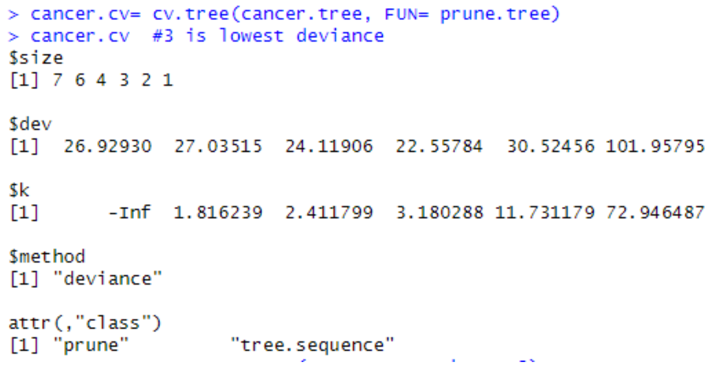
The ROC curve represents the performance of the Decision tree. The area under the curve in ROC is 0.8967.

Pruned Tree

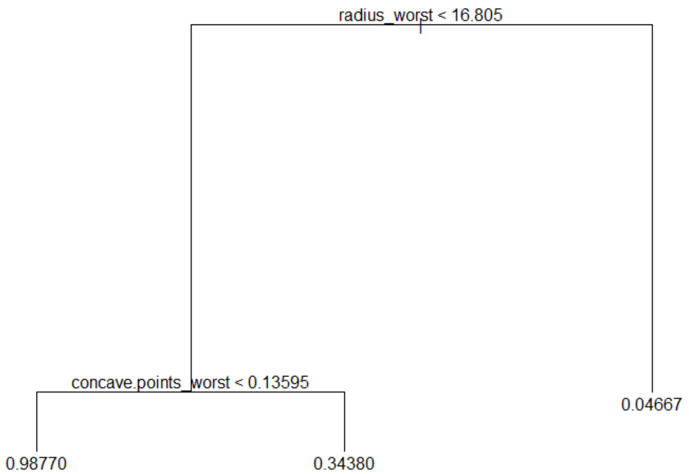
A base decision tree can be very useful in visualizing the data, as well as being simple. Yet, decision trees often run into the problem of overfitting the data that leads to poor test set performance. Therefore I took the decision tree one step further, and took steps to prune the tree. Pruning takes a tuning parameter that is gained from cross validation to get a better set of splits. In general, less spits in a tree means lower variance.

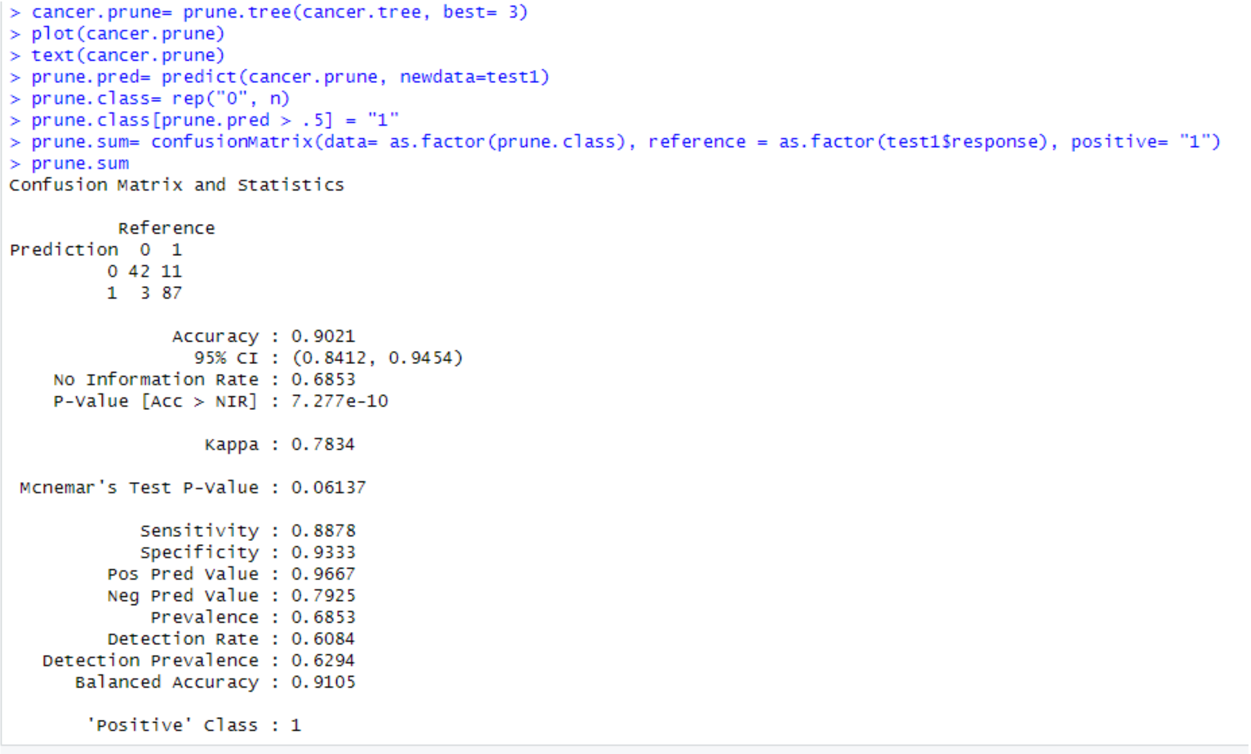
Model Selection-

Pruning a decision tree has similar steps to making a decision tree except a slightly different function is used, “cv.tree” and “prune.tree.” The first function allows us to find the number of terminal nodes that will have the lowest deviance, which returns the lowest deviance for 3 terminal nodes. The value of 3 is put into the “prune.tree” function in order to return a pruned decision tree. The steps to return the accuracy and ROC are the same as the previous decision tree. The accuracy of the prune tree is .9021 at 95% confidence interval which is actually lower than the base decision tree. Therefore, the base decision tree should be used over the pruned tree. This shows that 3 nodes are sufficient to have an accurate model but 7 nodes give a better model, and does not overfit the data.



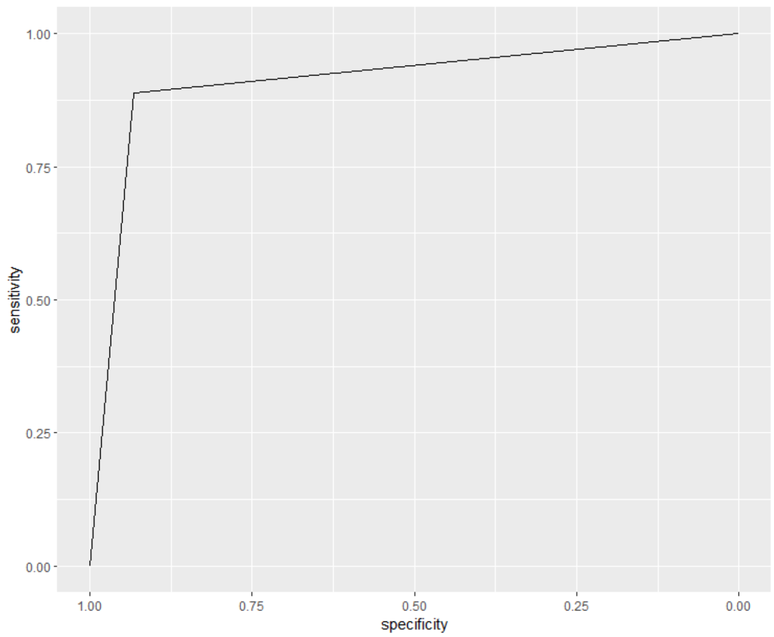
```
> cancer.cv= cv.tree(cancer.tree, FUN= prune.tree)
> cancer.cv  #3 is lowest deviance
$size
[1] 7 6 4 3 2 1

$dev
[1]  26.92930  27.03515  24.11906  22.55784  30.52456 101.95795

$k
[1]      -Inf  1.816239  2.411799  3.180288 11.731179 72.946487

$method
[1] "deviance"

attr(,"class")
[1] "prune"         "tree.sequence"
```





```
> cancer.prune= prune.tree(cancer.tree, best= 3)
> plot(cancer.prune)
> text(cancer.prune)
> prune.pred= predict(cancer.prune, newdata=test1)
> prune.class= rep("0", n)
> prune.class[prune.pred > .5] = "1"
> prune.sum= confusionMatrix(data= as.factor(prune.class), reference = as.factor(test1$response), positive= "1")
> prune.sum
Confusion Matrix and Statistics

          Reference
Prediction  0  1
         0 42 11
         1  3 87

               Accuracy : 0.9021
                 95% CI : (0.8412, 0.9454)
    No Information Rate : 0.6853
    P-Value [Acc > NIR] : 7.277e-10

                  Kappa : 0.7834

 Mcnemar's Test P-Value : 0.06137

            Sensitivity : 0.8878
            Specificity : 0.9333
         Pos Pred Value : 0.9667
         Neg Pred Value : 0.7925
             Prevalence : 0.6853
         Detection Rate : 0.6084
   Detection Prevalence : 0.6294
      Balanced Accuracy : 0.9105

       'Positive' Class : 1
```

The ROC curve represents the performance of the pruned tree. The area under the curve in ROC is 0.9105, which is higher than the base decision tree.



Random Forest

Advantages:

1. High Accuracy: Random Forest is a well-liked option for predictive modeling since, on average, it offers high accuracy in comparison to other algorithms.

2. resilient to Noise and Outliers: Random Forest generates many decision trees and averages their output to produce a more stable and dependable prediction, making it resilient to noise and outliers in the data.

3. Handles Large Datasets: Random Forest is scalable and effective for data analysis since it can handle datasets with a lot of attributes and observations.

4. Feature Importance: Random Forest offers a metric for feature importance that can be used to pinpoint the most crucial elements for feature selection and prediction.

Disadvantages:

1. Overfitting: Random Forest can overfit the data if there are too many trees or if the data is too noisy, which has a negative impact on how well it generalizes to new data.

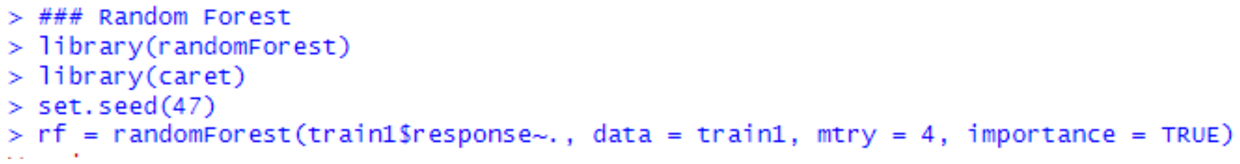
2. Interpretability: Because the prediction is based on a group of decision trees rather than a single model, Random Forest might be challenging to interpret.

3. Computationally Expensive: Random Forest can be computationally costly, particularly for huge datasets or when the number of trees is quite high, necessitating a significant amount of computer time and resources.

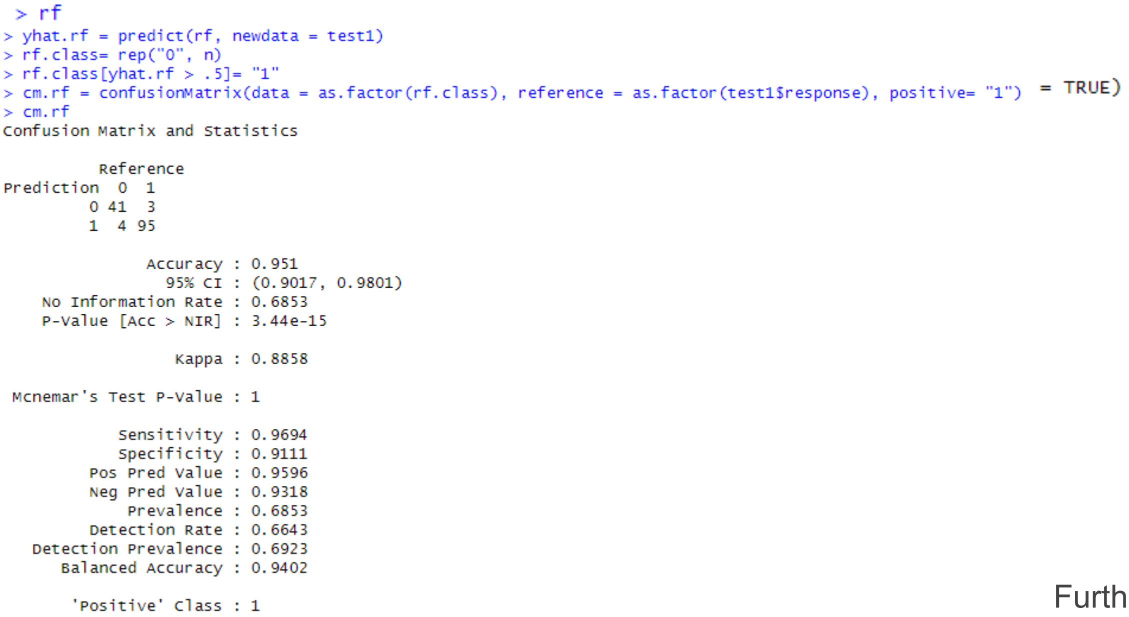
4. Unbalanced Data: In unbalanced datasets, Random Forest can be biased in favor of the majority class, which results in subpar prediction accuracy for the minority class.

Model Selection:

I used Random Forest to see how it handles our response variable i.e., “Diagnosis”, it turns out with highest accuracy among all the models. I fit the random forest model using four variables for each split. Random forest generated 500 trees with mean squared residuals of 0.0297. I also did the prediction using a test dataset using the Random Forest model and reviewed the confusion matrix. Confusion matrix for the random forest comes with the accuracy of 95.10 % at 95% confidence interval. Sensitivity and Specificity for Random Forest are 0.9694 and 0.9111 respectively.

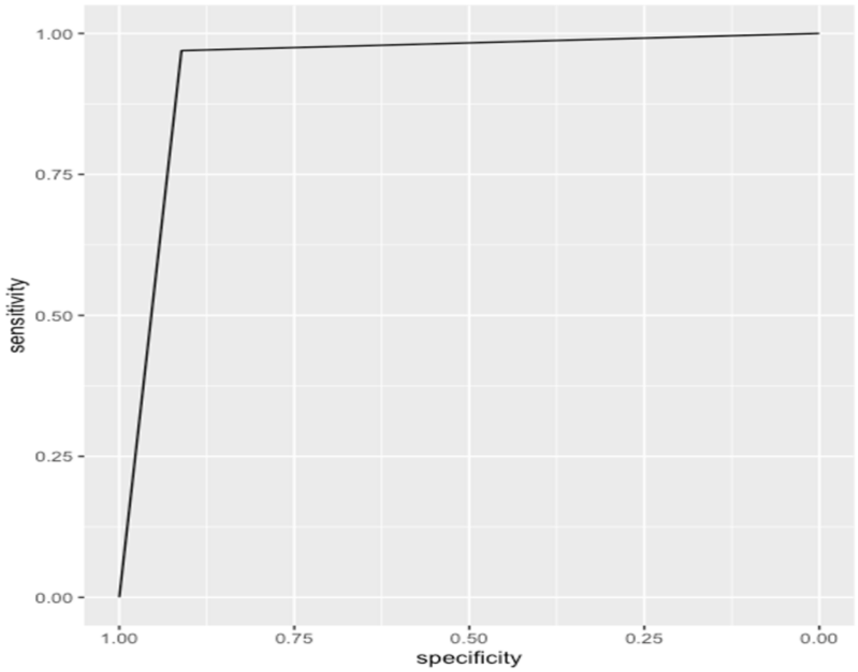


```
> ### Random Forest
> library(randomForest)
> library(caret)
> set.seed(47)
> rf = randomForest(train1$response~., data = train1, mtry = 4, importance = TRUE)
```

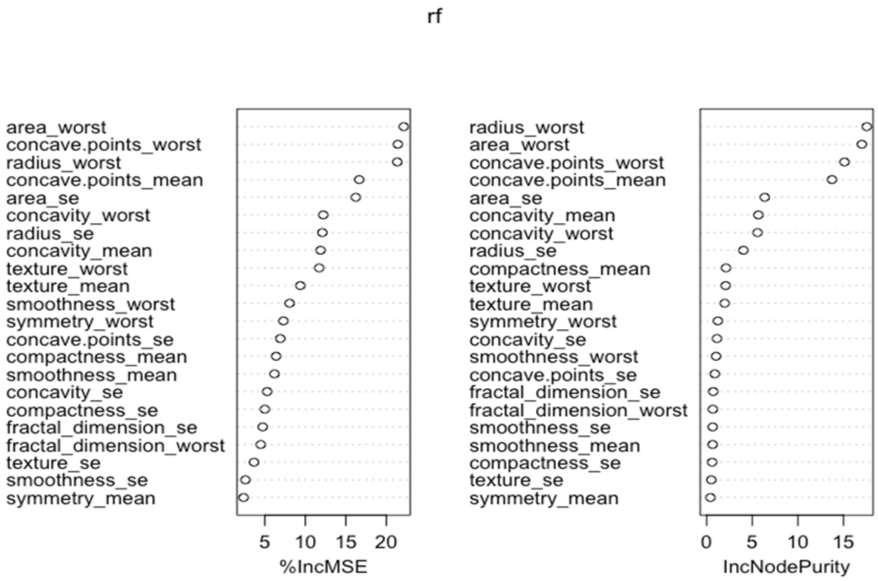


```
 > rf
> yhat.rf = predict(rf, newdata = test1)
> rf.class= rep("0", n)
> rf.class[yhat.rf > .5]= "1"
> cm.rf = confusionMatrix(data = as.factor(rf.class), reference = as.factor(test1$response), positive= "1")  = TRUE)
> cm.rf
Confusion Matrix and Statistics

          Reference
Prediction  0  1
         0 41  3
         1  4 95

               Accuracy : 0.951
                 95% CI : (0.9017, 0.9801)
    No Information Rate : 0.6853
    P-Value [Acc > NIR] : 3.44e-15

                  Kappa : 0.8858

 Mcnemar's Test P-Value : 1

            Sensitivity : 0.9694
            Specificity : 0.9111
         Pos Pred Value : 0.9596
         Neg Pred Value : 0.9318
             Prevalence : 0.6853
         Detection Rate : 0.6643
   Detection Prevalence : 0.6923
      Balanced Accuracy : 0.9402

       'Positive' Class : 1
```

Further I plotted the ROC (Receiver Operating Characteristic) curve to see the performance of random forest in classifying the response variable. ROC turns out with Area Under the Curve of 0.9402.



For more analysis using the Random forest, I take the variable importance plot to see which variables are more significant/ important for classification of the response variable. From the graph I found out that variables area_worst, concanve.points_worst, radius_worst and concave.points_mean were the four most important variables. And variables symmetry_mean, smoothness_se and texture mean were the least important in the graph.

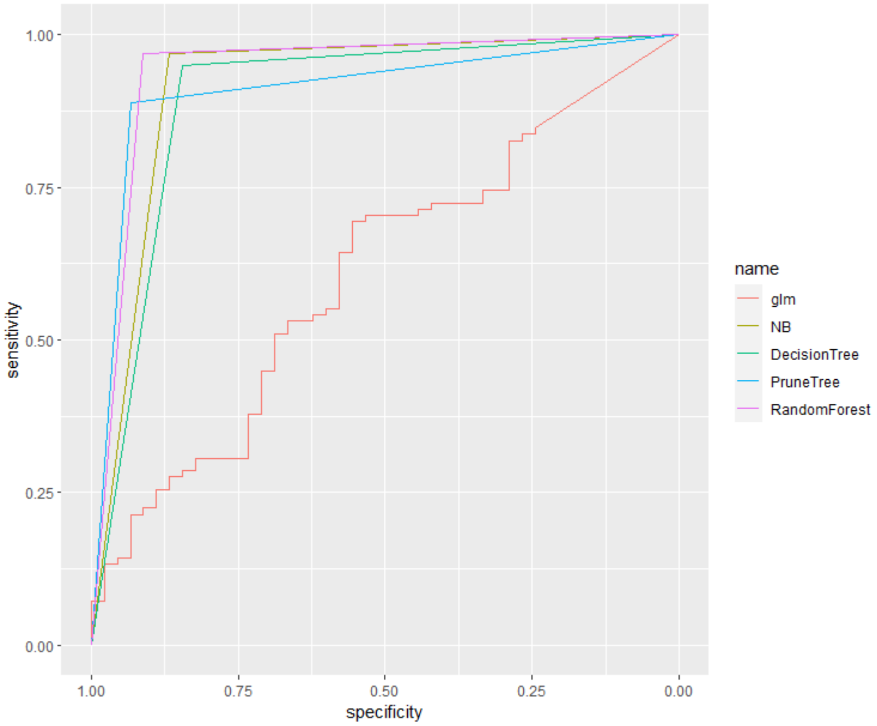


Conclusion:

Overall, mostly every model tested could be used with this data set except the GLM model since its accuracy is significantly lower than the rest. The best model to use is the Random Forest model since it has the highest accuracy, and highest AUC. Below is a summary of the results of each model:

Model	Accuracy	AUC
GLM	62.24%	59.5%
Naive Bayes	93.71%	91.80%
Decision Tree	91.61%	89.67%
Pruned Tree	90.21%	91.05%
Random Forest	95.10%	94.02%

In addition, below is a graph with all the ROC curves plotted together:



With the Random Forest model, I were able to attain 4 variables that are best at predicting whether a breast cancer tumor will be benign, or malignant. The 4 variables are: Area_worst, concave.points_worst, radius_worst, and concave.points_mean. In practice, this can allow cancer researchers to focus on these attributes of a tumor, and use it as a jumping off point to potentially learn how to prevent cancerous tumors. Furthermore, this report shows how useful machine learning is, and how it can be utilized in practical ways.

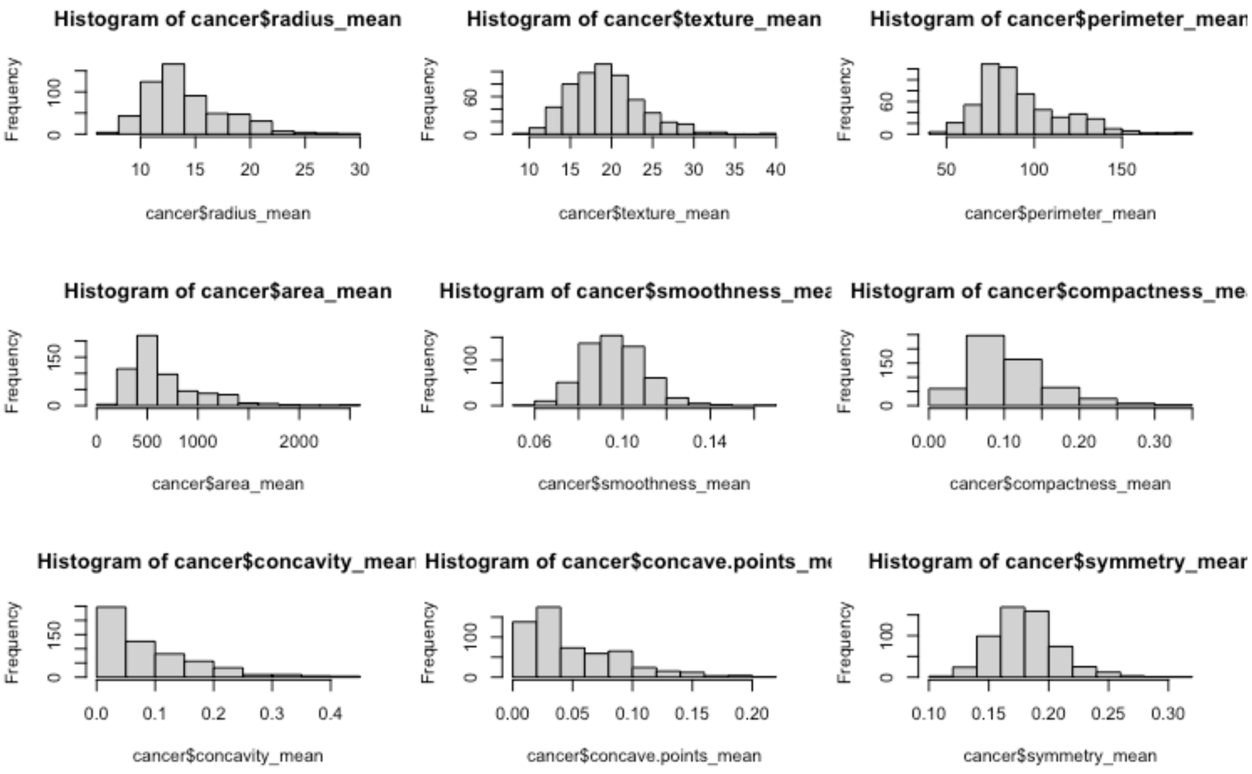
References:

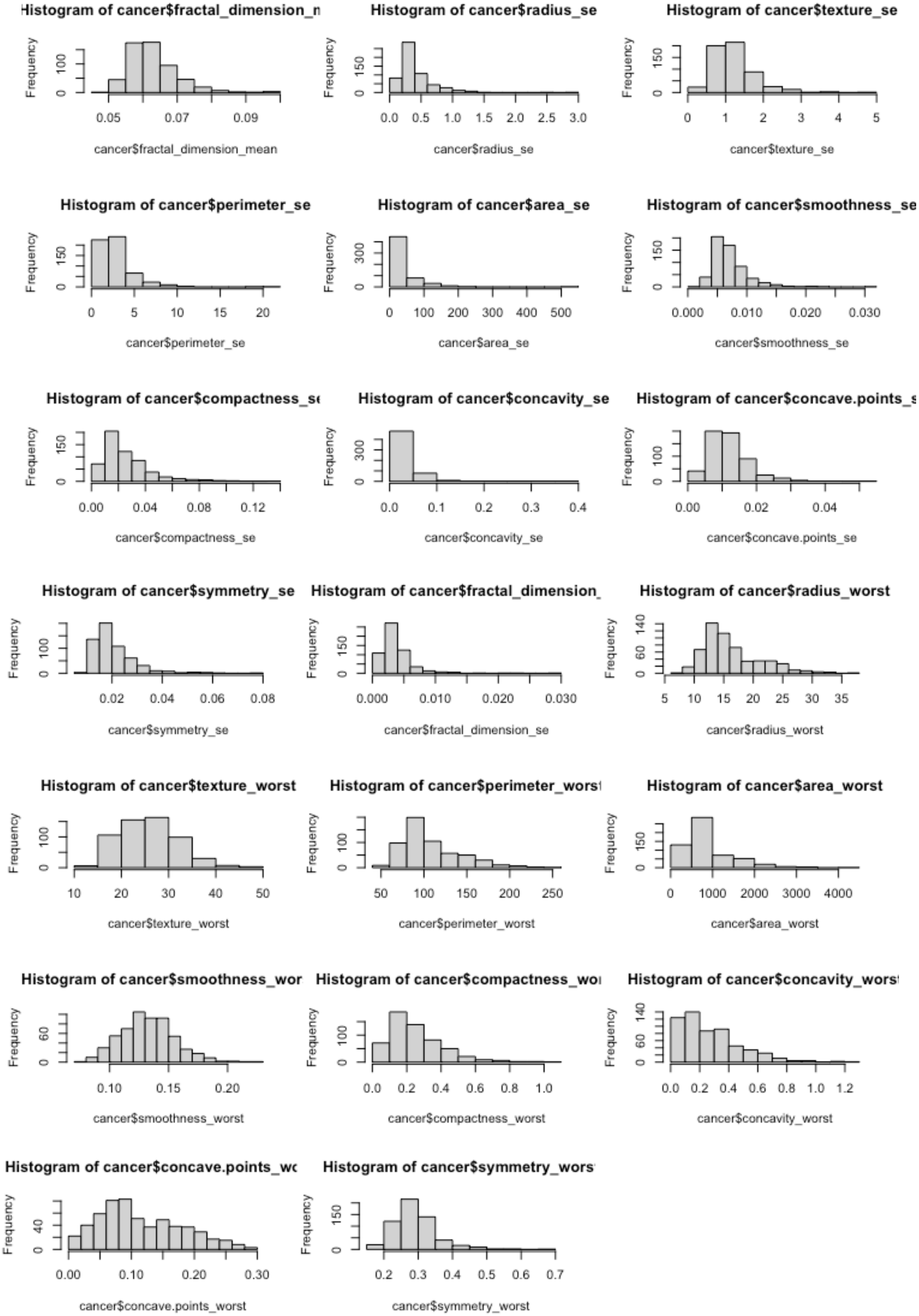
“Common Cancer Sites - Cancer Stat Facts.” *National Cancer Institute*, 2023, https://seer.cancer.gov/statfacts/html/common.html.

“NCI Dictionary of Cancer Terms.” *National Cancer Institute*, 2023, https://www.cancer.gov/publications/dictionaries/cancer-terms/def/benign.

“NCI Dictionary of Cancer Terms.” *National Cancer Institute*, https://www.cancer.gov/publications/dictionaries/cancer-terms/def/malignant**.**

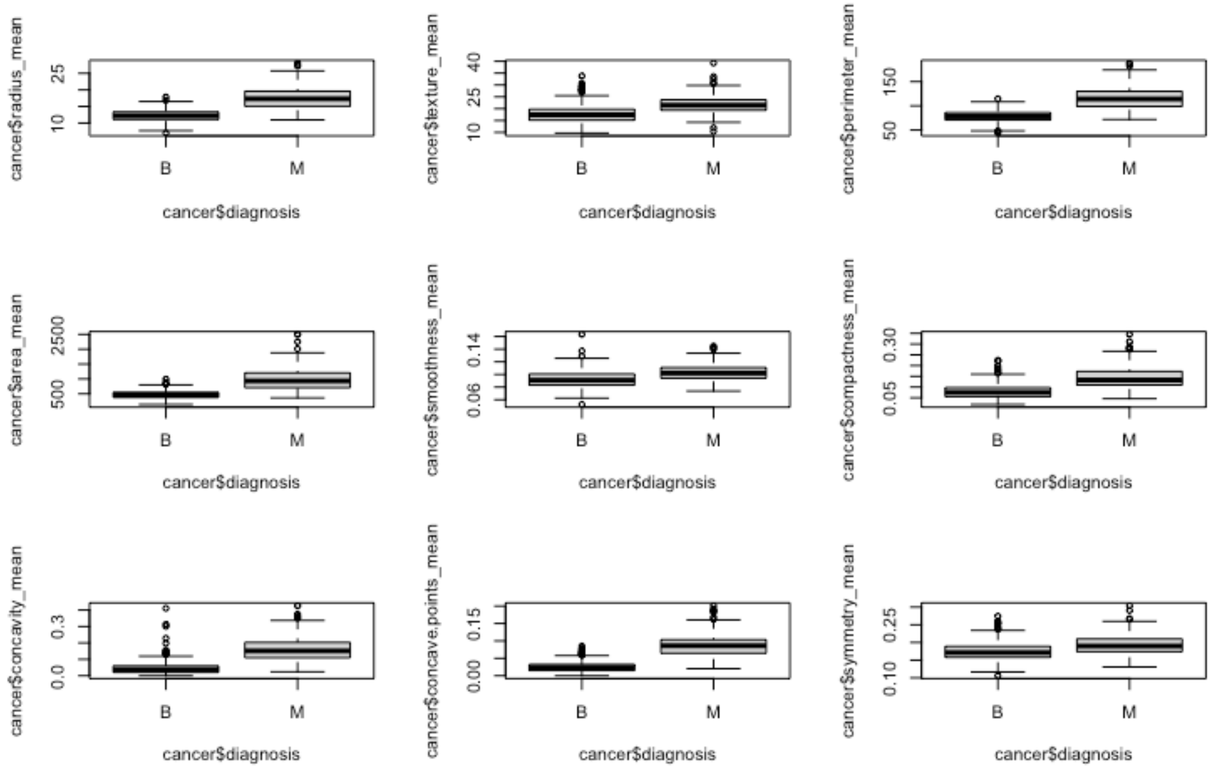
Appendix:



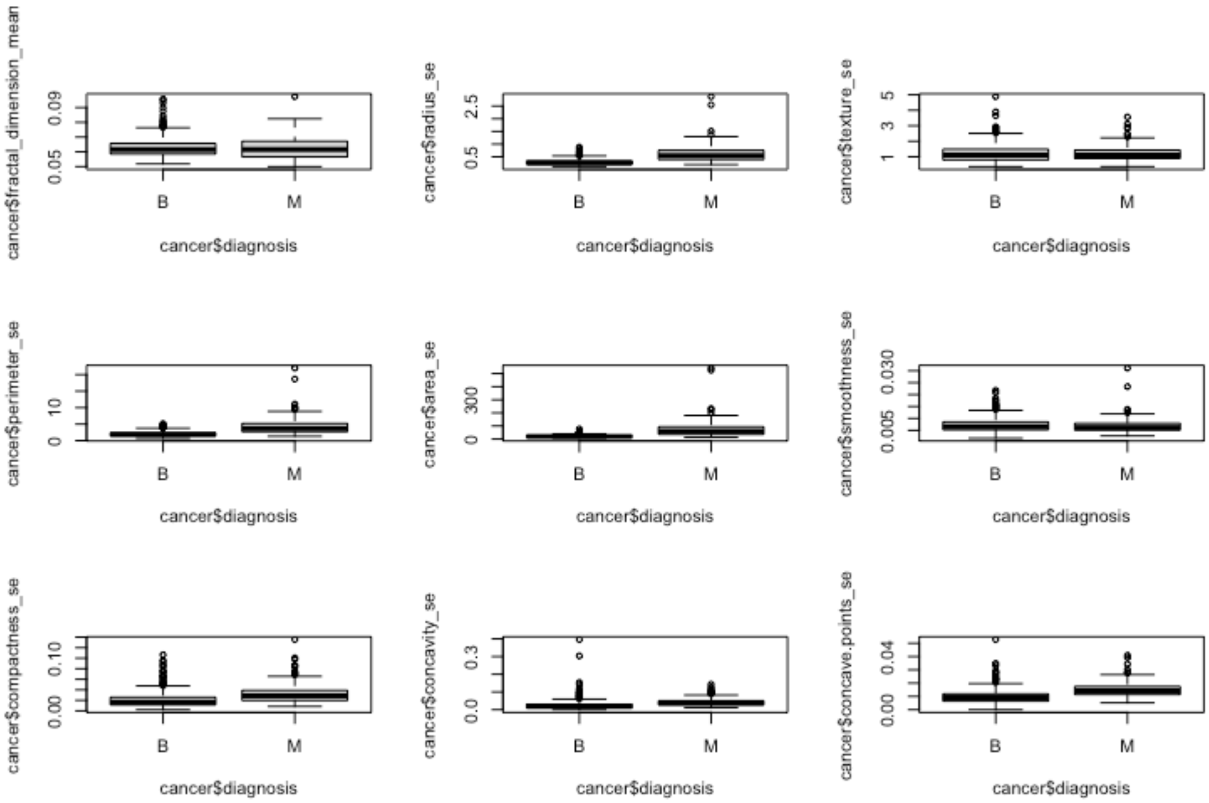


Histogram of cancer$fractal_dimension_m
Frequency
0
100
0.05
0.07
0.09
cancer$fractal_dimension_mean
Histogram of cancer$radius_se
Frequency
0
150
0.0
0.5
1.0
1.5
2.0
2.5
3.0
cancer$radius_se
Histogram of cancer$texture_se
Frequency
0
150
0
1
2
3
4
5
cancer$texture_se
Histogram of cancer$perimeter_se
Frequency
0
150
0
5
10
15
20
cancer$perimeter_se
Histogram of cancer$area_se
Frequency
0
300
0
100
200
300
400
500
cancer$area_se
Histogram of cancer$smoothness_se
Frequency
0
150
0.000
0.010
0.020
0.030
cancer$smoothness_se
Histogram of cancer$compactness_se
Frequency
0
150
0.00
0.04
0.08
0.12
cancer$compactness_se
Histogram of cancer$concavity_se
Frequency
0
300
0.0
0.1
0.2
0.3
0.4
cancer$concavity_se
Histogram of cancer$concave.points_s
Frequency
0
100
0.00
0.02
0.04
cancer$concave.points_se
Histogram of cancer$symmetry_se
Frequency
0
100
0.02
0.04
0.06
0.08
cancer$symmetry_se
Histogram of cancer$fractal_dimension_
Frequency
0
150
0.000
0.010
0.020
0.030
cancer$fractal_dimension_se
Histogram of cancer$radius_worst
Frequency
0
60
140
5
10
15
20
25
30
35
cancer$radius_worst
Histogram of cancer$texture_worst
Frequency
0
100
10
20
30
40
50
cancer$texture_worst
Histogram of cancer$perimeter_worst
Frequency
0
100
50
100
150
200
250
cancer$perimeter_worst
Histogram of cancer$area_worst
Frequency
0
150
0
1000
2000
3000
4000
cancer$area_worst
Histogram of cancer$smoothness_wor
Frequency
0
60
0.10
0.15
0.20
cancer$smoothness_worst
Histogram of cancer$compactness_wo
Frequency
0
100
0.0
0.2
0.4
0.6
0.8
1.0
cancer$compactness_worst
Histogram of cancer$concavity_worst
Frequency
0
60
140
0.0
0.2
0.4
0.6
0.8
1.0
1.2
cancer$concavity_worst
Histogram of cancer$concave.points_wo
Frequency
0
40
0.00
0.10
0.20
0.30
cancer$concave.points_worst
Histogram of cancer$symmetry_worst
Frequency
0
150
0.2
0.3
0.4
0.5
0.6
0.7
cancer$symmetry_worst

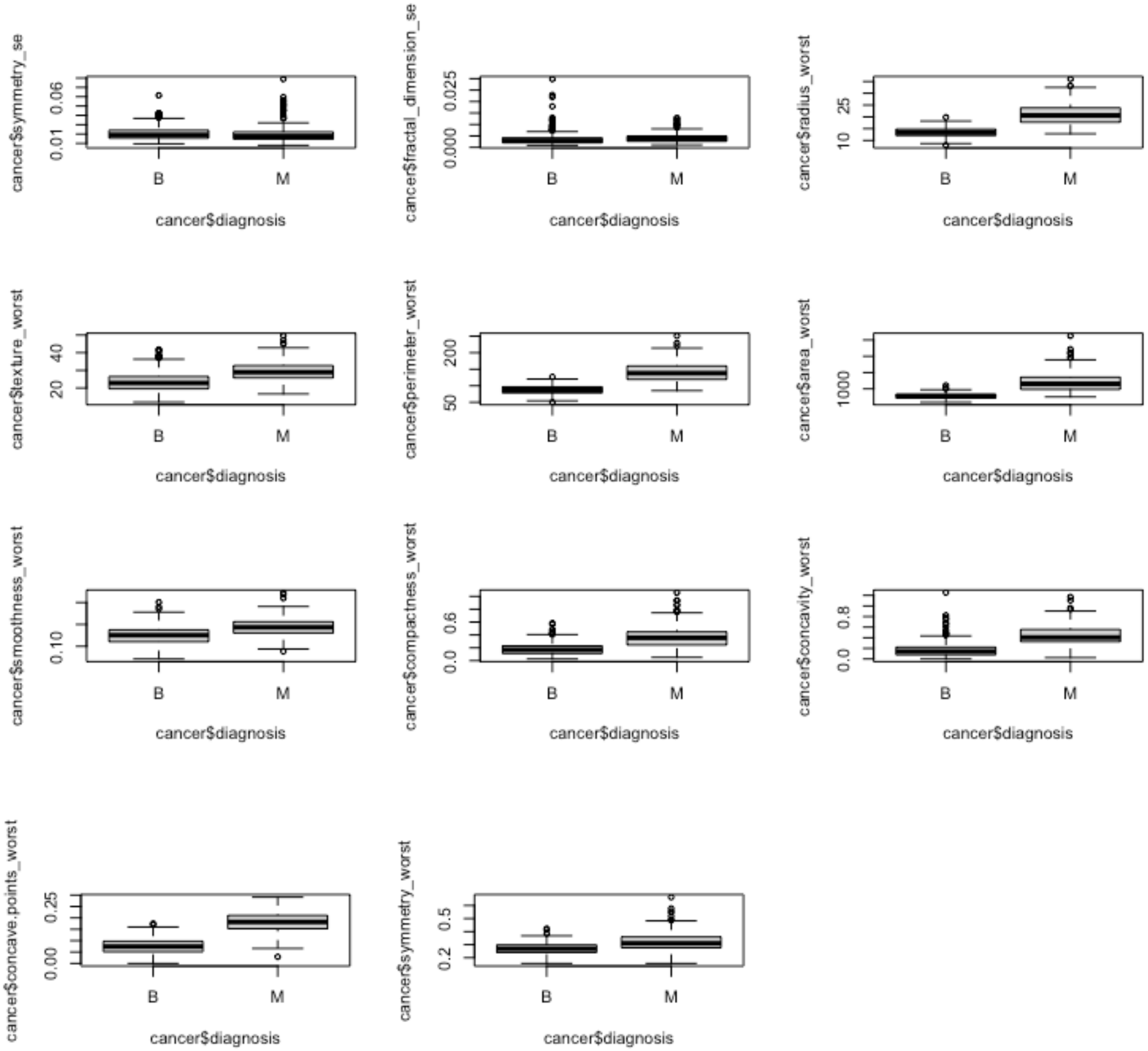
Box Plots



cancer$radius_mean
10
25
B
M
cancer$diagnosis
cancer$texture_mean
10
25
40
B
M
cancer$diagnosis
cancer$perimeter_mean
50
150
B
M
cancer$diagnosis
cancer$area_mean
500
2500
B
M
cancer$diagnosis
cancer$smoothness_mean
0.06
0.14
B
M
cancer$diagnosis
cancer$compactness_mean
0.05
0.30
B
M
cancer$diagnosis
cancer$concavity_mean
0.0
0.3
B
M
cancer$diagnosis
cancer$concave.points_mean
0.00
0.15
B
M
cancer$diagnosis
cancer$symmetry_mean
0.10
0.25
B
M
cancer$diagnosis



cancer$fractal_dimension_mean
0.05
0.09
B
M
cancer$diagnosis
cancer$radius_se
0.5
2.5
B
M
cancer$diagnosis
cancer$texture_se
1
3
5
B
M
cancer$diagnosis
cancer$perimeter_se
0
10
B
M
cancer$diagnosis
cancer$area_se
0
300
B
M
cancer$diagnosis
cancer$smoothness_se
0.005
0.030
B
M
cancer$diagnosis
cancer$compactness_se
0.00
0.10
B
M
cancer$diagnosis
cancer$concavity_se
0.0
0.3
B
M
cancer$diagnosis
cancer$concave.points_se
0.00
0.04
B
M
cancer$diagnosis


cancer$symmetry_se
0.01
0.06
B
M
cancer$diagnosis
cancer$fractal_dimension_se
0.000
0.025
B
M
cancer$diagnosis
cancer$radius_worst
10
25
B
M
cancer$diagnosis
cancer$texture_worst
20
40
B
M
cancer$diagnosis
cancer$perimeter_worst
50
200
B
M
cancer$diagnosis
cancer$area_worst
1000
B
M
cancer$diagnosis
cancer$smoothness_worst
0.10
B
M
cancer$diagnosis
cancer$compactness_worst
0.0
0.6
B
M
cancer$diagnosis
cancer$concavity_worst
0.0
0.8
B
M
cancer$diagnosis
cancer$concave.points_worst
0.00
0.25
B
M
cancer$diagnosis
cancer$symmetry_worst
0.2
0.5
B
M
cancer$diagnosis