Analysis of the Breast Cancer Dataset

Neha Dixit, Ravindra Thanniru, Saloni Bhatia, Sangrae Cho

12/2/2019

# 1. Introduction

Breast cancer is the most common cancer among women and one of the major causes of death among women worldwide. Every year approximately 124 out of 100,000 women are diagnosed with breast cancer, and the estimation is that 23 out of the 124 women will die of this disease. When detected in its early stages, there is a 30% chance that the cancer can be treated eﬀectively, but the late detection of advanced-stage tumors makes the treatment more diﬃcult. Currently, the most used techniques to detect breast cancer in early stages are: mammography (63% to 97% correctness), FNA (Fine Needle Aspiration) with visual interpretation (65% to 98% correctness) and surgical biopsy (approximately 100% correctness). Therefore, mammography and FNA with visual interpretation correctness varies widely, and the surgical biopsy, although reliable, is invasive and costly. [Ref: <https://www.cancer.org/cancer/breast-cancer/about/how-common-is-breast-cancer.html>]

The breast cancer data set used for analysis in this project has numerous measurements taken from tumor biopsies. The goal of using this data set is to address various objectives of the project and build models to predict using the metrics alone if the biopsy is cancer or not.

# 2. Dataset and Description

The data set used for analysis is available at: <https://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/wdbc.data>

Observations in the data set are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. The data is comprised of 31 numeric features and one categorical feature, which contain 569 observations. The first two columns gives Sample ID and Classes(diagnosis) which has M, malignant and B, benign breast mass. For each cell nucleaus, the following ten charactoristics were measured: Radius, Texture, Perimeter, Area, Smoothness, Compactness, Concavity, Concave points, Symmetry, Fractal dimension. For each characteristic three measures are given: Mean, Standard error, Largest/ “worst”.  
The dataset we chose for this project was a publicly shared, breast cancer Wisconsin(diagnostic) data made available through Kaggle in csv format.

The dataset consists of 569 rows and 32 attributes. We have both categorical and continous attributes in the dataset. The attributes are summarized as below:

1. id ID number
2. diagnosis The diagnosis of breast tissues (M = malignant, B = benign)
3. radius\_mean mean of distances from center to points on the perimeter
4. texture\_mean standard deviation of gray-scale values
5. perimeter\_mean mean size of the core tumor
6. area\_mean
7. smoothness\_mean mean of local variation in radius lengths
8. compactness\_mean mean of perimeter^2 / area - 1.0
9. concavity\_mean mean of severity of concave portions of the contour
10. concave points\_mean mean for number of concave portions of the contour
11. symmetry\_mean
12. fractal\_dimension\_mean mean for “coastline approximation” - 1
13. radius\_se standard error for the mean of distances from center to points on the perimeter
14. texture\_se standard error for standard deviation of gray-scale values
15. perimeter\_se
16. area\_se
17. smoothness\_se standard error for local variation in radius lengths
18. compactness\_se standard error for perimeter^2 / area - 1.0
19. concavity\_se standard error for severity of concave portions of the contour
20. concave points\_se standard error for number of concave portions of the contour
21. symmetry\_se
22. fractal\_dimension\_se standard error for “coastline approximation” - 1
23. radius\_worst “worst” or largest mean value for mean of distances from center to points on the perimeter
24. texture\_worst “worst” or largest mean value for standard deviation of gray-scale values
25. perimeter\_worst
26. area\_worst
27. smoothness\_worst “worst” or largest mean value for local variation in radius lengths
28. compactness\_worst “worst” or largest mean value for perimeter^2 / area - 1.0
29. concavity\_worst “worst” or largest mean value for severity of concave portions of the contour
30. concave points\_worst “worst” or largest mean value for number of concave portions of the contour
31. symmetry\_worst
32. fractal\_dimension\_worst “worst” or largest mean value for “coastline approximation” - 1

# 3. Exploratory Data Analysis

We start Exploratory Data Analysis (EDA) of the dataset by checking the number of observations, structure of each variable and summary statistics:

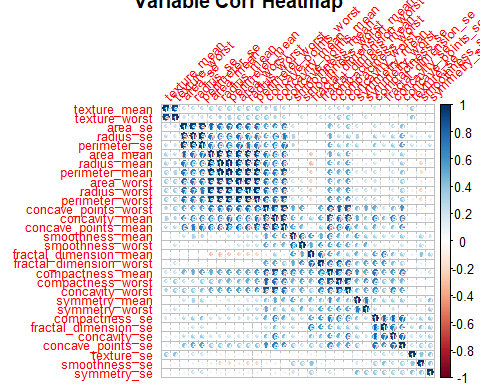
## 3-1. Summary statistics and glimpse of the data set

summary(bc)

## diagnosis radius\_mean texture\_mean perimeter\_mean   
## 0:357 Min. : 6.981 Min. : 9.71 Min. : 43.79   
## 1:212 1st Qu.:11.700 1st Qu.:16.17 1st Qu.: 75.17   
## Median :13.370 Median :18.84 Median : 86.24   
## Mean :14.127 Mean :19.29 Mean : 91.97   
## 3rd Qu.:15.780 3rd Qu.:21.80 3rd Qu.:104.10   
## Max. :28.110 Max. :39.28 Max. :188.50   
## area\_mean smoothness\_mean compactness\_mean concavity\_mean   
## Min. : 143.5 Min. :0.05263 Min. :0.01938 Min. :0.00000   
## 1st Qu.: 420.3 1st Qu.:0.08637 1st Qu.:0.06492 1st Qu.:0.02956   
## Median : 551.1 Median :0.09587 Median :0.09263 Median :0.06154   
## Mean : 654.9 Mean :0.09636 Mean :0.10434 Mean :0.08880   
## 3rd Qu.: 782.7 3rd Qu.:0.10530 3rd Qu.:0.13040 3rd Qu.:0.13070   
## Max. :2501.0 Max. :0.16340 Max. :0.34540 Max. :0.42680   
## concave\_points\_mean symmetry\_mean fractal\_dimension\_mean  
## Min. :0.00000 Min. :0.1060 Min. :0.04996   
## 1st Qu.:0.02031 1st Qu.:0.1619 1st Qu.:0.05770   
## Median :0.03350 Median :0.1792 Median :0.06154   
## Mean :0.04892 Mean :0.1812 Mean :0.06280   
## 3rd Qu.:0.07400 3rd Qu.:0.1957 3rd Qu.:0.06612   
## Max. :0.20120 Max. :0.3040 Max. :0.09744   
## radius\_se texture\_se perimeter\_se area\_se   
## Min. :0.1115 Min. :0.3602 Min. : 0.757 Min. : 6.802   
## 1st Qu.:0.2324 1st Qu.:0.8339 1st Qu.: 1.606 1st Qu.: 17.850   
## Median :0.3242 Median :1.1080 Median : 2.287 Median : 24.530   
## Mean :0.4052 Mean :1.2169 Mean : 2.866 Mean : 40.337   
## 3rd Qu.:0.4789 3rd Qu.:1.4740 3rd Qu.: 3.357 3rd Qu.: 45.190   
## Max. :2.8730 Max. :4.8850 Max. :21.980 Max. :542.200   
## smoothness\_se compactness\_se concavity\_se   
## Min. :0.001713 Min. :0.002252 Min. :0.00000   
## 1st Qu.:0.005169 1st Qu.:0.013080 1st Qu.:0.01509   
## Median :0.006380 Median :0.020450 Median :0.02589   
## Mean :0.007041 Mean :0.025478 Mean :0.03189   
## 3rd Qu.:0.008146 3rd Qu.:0.032450 3rd Qu.:0.04205   
## Max. :0.031130 Max. :0.135400 Max. :0.39600   
## concave\_points\_se symmetry\_se fractal\_dimension\_se  
## Min. :0.000000 Min. :0.007882 Min. :0.0008948   
## 1st Qu.:0.007638 1st Qu.:0.015160 1st Qu.:0.0022480   
## Median :0.010930 Median :0.018730 Median :0.0031870   
## Mean :0.011796 Mean :0.020542 Mean :0.0037949   
## 3rd Qu.:0.014710 3rd Qu.:0.023480 3rd Qu.:0.0045580   
## Max. :0.052790 Max. :0.078950 Max. :0.0298400   
## radius\_worst texture\_worst perimeter\_worst area\_worst   
## Min. : 7.93 Min. :12.02 Min. : 50.41 Min. : 185.2   
## 1st Qu.:13.01 1st Qu.:21.08 1st Qu.: 84.11 1st Qu.: 515.3   
## Median :14.97 Median :25.41 Median : 97.66 Median : 686.5   
## Mean :16.27 Mean :25.68 Mean :107.26 Mean : 880.6   
## 3rd Qu.:18.79 3rd Qu.:29.72 3rd Qu.:125.40 3rd Qu.:1084.0   
## Max. :36.04 Max. :49.54 Max. :251.20 Max. :4254.0   
## smoothness\_worst compactness\_worst concavity\_worst concave\_points\_worst  
## Min. :0.07117 Min. :0.02729 Min. :0.0000 Min. :0.00000   
## 1st Qu.:0.11660 1st Qu.:0.14720 1st Qu.:0.1145 1st Qu.:0.06493   
## Median :0.13130 Median :0.21190 Median :0.2267 Median :0.09993   
## Mean :0.13237 Mean :0.25427 Mean :0.2722 Mean :0.11461   
## 3rd Qu.:0.14600 3rd Qu.:0.33910 3rd Qu.:0.3829 3rd Qu.:0.16140   
## Max. :0.22260 Max. :1.05800 Max. :1.2520 Max. :0.29100   
## symmetry\_worst fractal\_dimension\_worst  
## Min. :0.1565 Min. :0.05504   
## 1st Qu.:0.2504 1st Qu.:0.07146   
## Median :0.2822 Median :0.08004   
## Mean :0.2901 Mean :0.08395   
## 3rd Qu.:0.3179 3rd Qu.:0.09208   
## Max. :0.6638 Max. :0.20750

As seen above, there are no missing values and the Class distribution is: 357 Benign, 212 Malignant. First column represents the sample ID so we deleted the column. Also we changed response variable from M, B to 1 and 0.

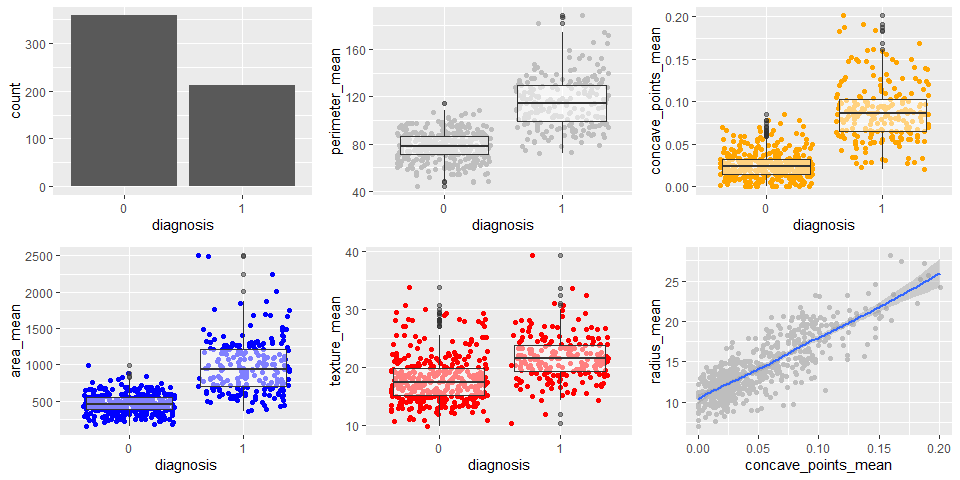
## 3-2. Correlation HeatMap

Plotting Correlation Heatmap for the data to observe the nature and extent of correlation between various features in the dataset:  


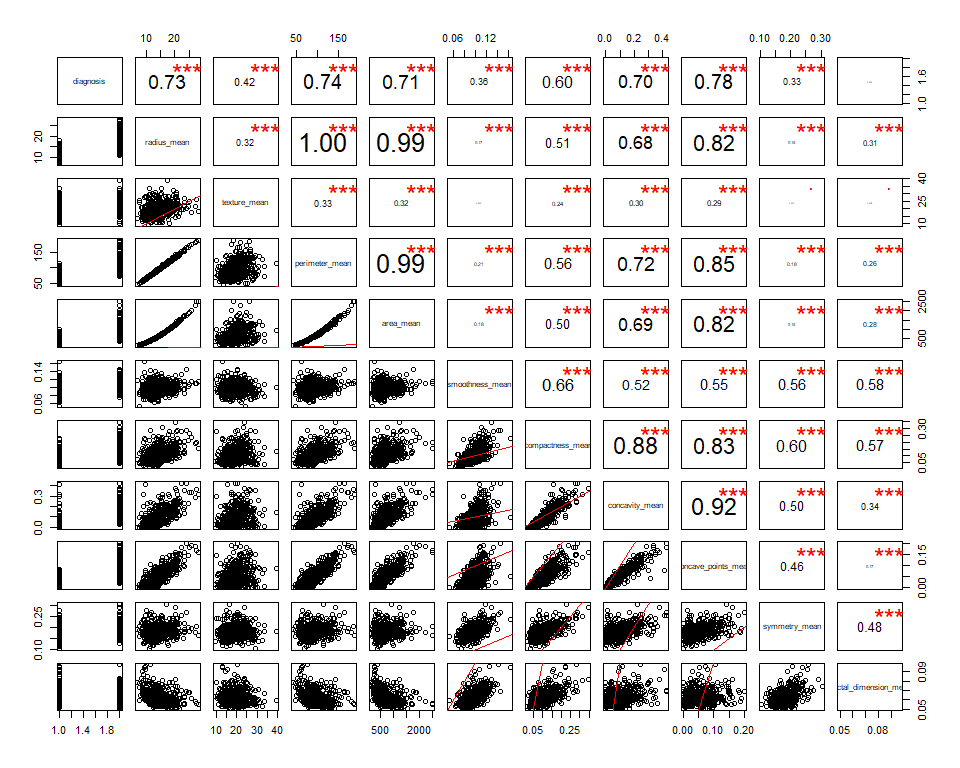
As seen in the heatmap, there is strong correlation between:  
1. Texture\_mean and Texture\_worst  
2. area\_se, radius\_se, perimeter\_se, area\_mean, radius\_mean, perimeter\_mean, area\_worst, radius\_worst, perimeter\_worst, concave.points\_worst, concavity\_mean, concave.points\_mean  
3. smoothness\_mean, smoothness\_worst, fractal\_dimension\_mean, fractal\_dimension\_worst  
4. compactness\_mean, compactness\_worst, concavity\_worst  
5. symmetry\_mean, symmetry\_worst  
6. compactness\_se, fractal\_dimension\_se, concavity\_se, concavity.points\_se

## 3-3. EDA with various graphics

Of the observations around 40% have Malignant tumor. In malignant group, we see high values on perimeter\_mean, convave.point\_mean, area\_mean.



## 3-3. Pairwise Plots



The pairwise plot also confirms that lots of features are correlated, especially the radius, parameter and area are highly correlated as expected from their relationship. So, from these we can use anyone of them.

Compactness\_mean, concavity\_mean and concave point\_mean are highly correlated so we can use compactness\_mean from here.

Based on the EDA, we think the following features would be of great importance for use in the analysis:  
[‘texture\_mean’,‘perimeter\_mean’,‘smoothness\_mean’,‘concave\_point\_mean’,‘symmetry\_mean’]

# 4. Split the available data set into Train, Validate and Test portions:

For the sake of our analysis we split the available data set into Train, Validate and Test subsets.

# Objective 1:

Perform logistic regression analysis and provide interpretation of the regression coefficients including confidence intervals.

# 4-1. Problem Statement:

The purpose of this analysis is to build a model to predict using the metrics alone (most appropriate risk factors) if the biopsy is cancer or not.

Model selection techniques will be used and a final model will be selected based on various performance criteria. Interpretation of the selected model will be provided as well.

## 4-2. Model Selection & Logistic Regression

For the model selection, we use LASSO and manual/intuition models and compare them to see which gives us the best results.

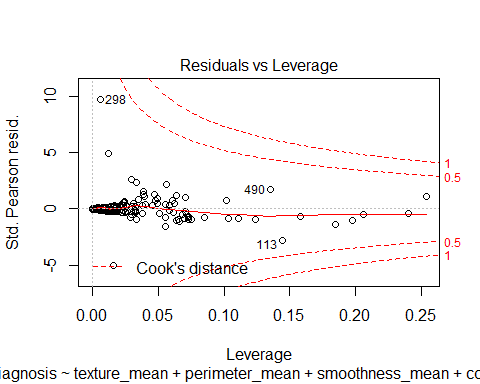
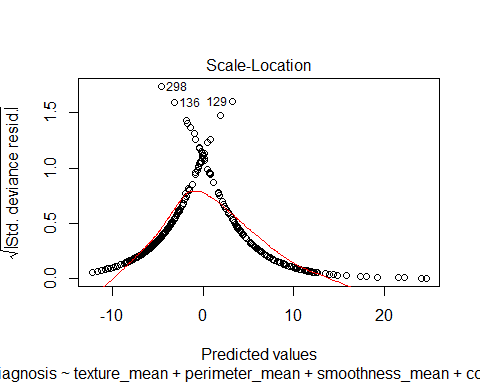
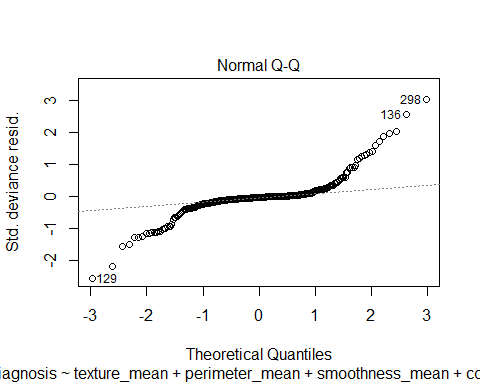
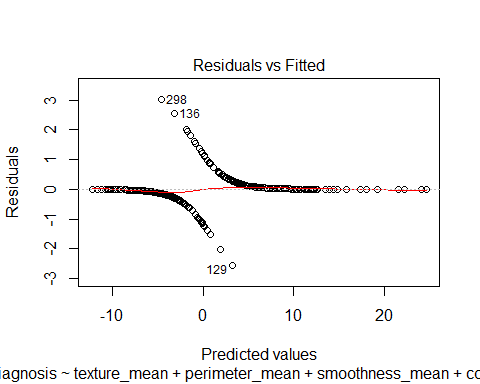
### Key Assumptions:

Before running logistic regression we proceed making sure the following two key assumptions are met.  
1. Logistic regression requires the observations to be independent of each other. For this data set we do not have information if any of the observations recorded belonged to members from the same family. And hence we assume that all the observations are independent of each other.  
2. Logistic regression requires there to be little or no multicollinearity among the independent variables. This means that the independent variables should not be too highly correlated with each other. During the EDA we identified highly correlated variables and we make sure to remove them when building the models.

### 4-2-1. Manual/Intuition: Based on EDA

Based on the EDA, we think the following features would be of great importance for use in the analysis:  
[‘texture\_mean’,‘perimeter\_mean’,‘smoothness\_mean’,‘concave\_point\_mean’,‘symmetry\_mean’]

#### 4-2-1-1. Residual diagnostics

Plots here help us examine Cook’s D graph: 

When checking Cook’s D plot, if observations are outside the Cook’s distance (meaning they have a high Cook’s distance score) the observations are influential to the regression results. In this case, from the Cook’s D plot above, we do not see any influential points.

#### 4-2-1-2. Lack of fit test

We then run the Hosmer Lemeshow test to check the lack of fit. And the Hosmer Lemeshow test looks fine since p-value > 0.05. Hence, we conclude that the model is a good fit.

##   
## Hosmer and Lemeshow goodness of fit (GOF) test  
##   
## data: model.check$y, fitted(model.check)  
## X-squared = 2.1064, df = 8, p-value = 0.9776

## Odds ratio 2.5 % 97.5 %  
## (Intercept) 3.075774e-13 5.244866e-19 1.803742e-07  
## texture\_mean 1.390704e+00 1.196232e+00 1.616791e+00  
## perimeter\_mean 1.147257e+00 1.064732e+00 1.236179e+00  
## smoothness\_mean 1.771197e+14 9.605177e-15 3.266090e+42  
## concave\_points\_mean 3.546396e+34 1.366051e+14 9.206772e+54  
## symmetry\_mean 1.396018e+05 1.312158e-06 1.485237e+16

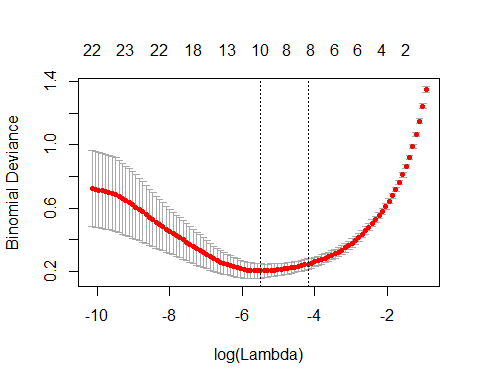
### 4-2-2. LASSO

We now try to build the model using LASSO. Since LASSO is a regularized model, binomial deviance can be used for model selection. Figure below shows the binomial deviance. The binomial deviance is minimized near the vertical lines (10: best prediction, 8: most simple to interpret).

coef(bc\_cvfit, s = c("lambda.1se"))

## 31 x 1 sparse Matrix of class "dgCMatrix"  
## 1  
## (Intercept) -17.2202783  
## radius\_mean .   
## texture\_mean .   
## perimeter\_mean .   
## area\_mean .   
## smoothness\_mean .   
## compactness\_mean .   
## concavity\_mean .   
## concave\_points\_mean 24.2431949  
## symmetry\_mean .   
## fractal\_dimension\_mean .   
## radius\_se 1.0709922  
## texture\_se .   
## perimeter\_se .   
## area\_se .   
## smoothness\_se .   
## compactness\_se .   
## concavity\_se .   
## concave\_points\_se .   
## symmetry\_se .   
## fractal\_dimension\_se .   
## radius\_worst 0.5067433  
## texture\_worst 0.1098493  
## perimeter\_worst .   
## area\_worst .   
## smoothness\_worst 7.3417090  
## compactness\_worst .   
## concavity\_worst 0.6644741  
## concave\_points\_worst 12.3281541  
## symmetry\_worst 4.7495948  
## fractal\_dimension\_worst .

plot(bc\_cvfit)



The features selected at the “most simple to interpret” line are: concave\_points\_mean, radius\_se, radius\_worst, texture\_worst, smoothness\_worst, concavity\_worst, concave\_points\_worst, symmetry\_worst.

model.LASSO <-glm(diagnosis ~ concave\_points\_mean + radius\_se + radius\_worst + texture\_worst + smoothness\_worst +   
 concavity\_worst + concave\_points\_worst + symmetry\_worst,  
 data = training, family = binomial(link = "logit"))

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

summary(model.LASSO)

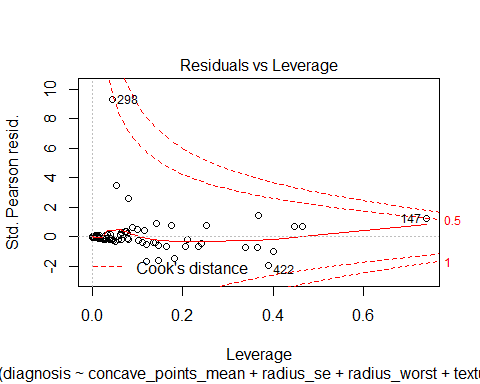
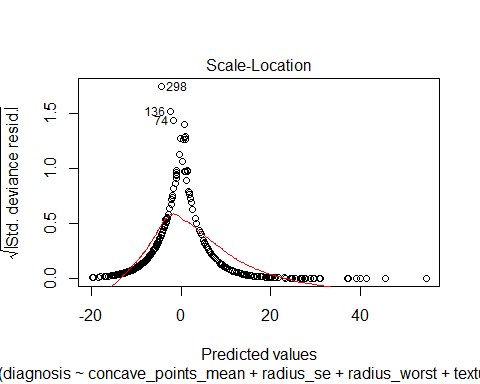
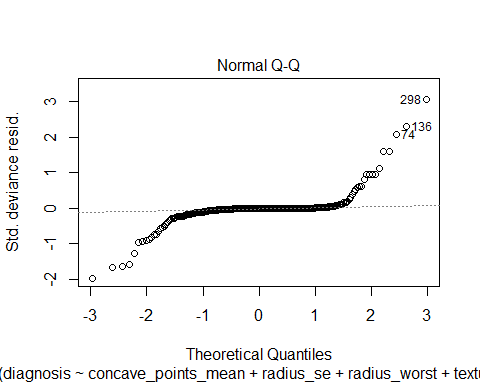
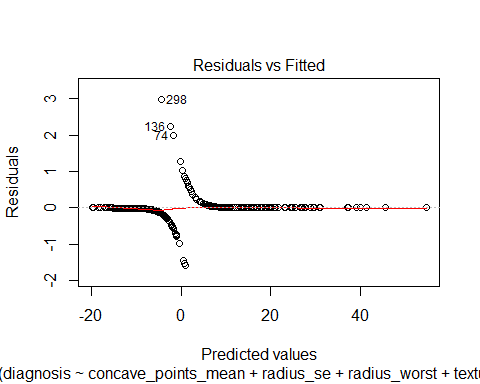
##   
## Call:  
## glm(formula = diagnosis ~ concave\_points\_mean + radius\_se + radius\_worst +   
## texture\_worst + smoothness\_worst + concavity\_worst + concave\_points\_worst +   
## symmetry\_worst, family = binomial(link = "logit"), data = training)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.56977 -0.03962 -0.00234 0.00040 2.97372   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -53.66376 12.75367 -4.208 2.58e-05 \*\*\*  
## concave\_points\_mean 4.68992 39.66150 0.118 0.905871   
## radius\_se 16.24499 6.85122 2.371 0.017735 \*   
## radius\_worst 1.46875 0.39659 3.703 0.000213 \*\*\*  
## texture\_worst 0.31784 0.09467 3.357 0.000787 \*\*\*  
## smoothness\_worst 71.68613 33.04072 2.170 0.030035 \*   
## concavity\_worst 3.76145 4.53398 0.830 0.406758   
## concave\_points\_worst 17.84345 22.27645 0.801 0.423131   
## symmetry\_worst 9.58595 6.50216 1.474 0.140409   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 460.260 on 340 degrees of freedom  
## Residual deviance: 43.067 on 332 degrees of freedom  
## AIC: 61.067  
##   
## Number of Fisher Scoring iterations: 10

confint.default(model.LASSO, level = 0.95)

## 2.5 % 97.5 %  
## (Intercept) -78.6604908 -28.6670319  
## concave\_points\_mean -73.0451904 82.4250232  
## radius\_se 2.8168479 29.6731317  
## radius\_worst 0.6914469 2.2460514  
## texture\_worst 0.1322926 0.5033966  
## smoothness\_worst 6.9275128 136.4447521  
## concavity\_worst -5.1249954 12.6478989  
## concave\_points\_worst -25.8175793 61.5044888  
## symmetry\_worst -3.1580534 22.3299490

From the above output we see that only the following parameters are **statistically significant**: radius\_se, radius\_worst, texture\_worst and smoothness\_worst.

#### 4-2-2-1. Residual diagnostics

Plots here help us examine Cook’s D graph: 

When checking Cook’s D plot, if observations are outside the Cook’s distance (meaning they have a high Cook’s distance score) the observations are influential to the regression results. In this case, from the Cook’s D plot above, we do not see any influential points.

#### 4-2-2-2. Lack of fit test

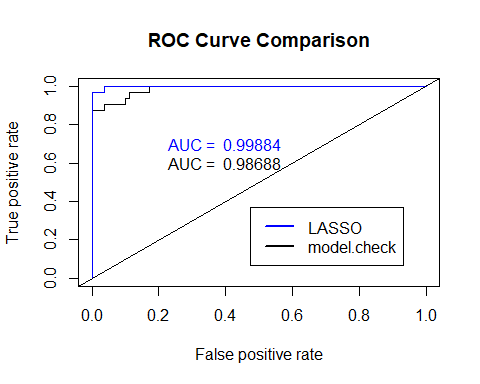
We then run the Hosmer Lemeshow test to check the lack of fit. And the Hosmer Lemeshow test looks fine since p-value > 0.05. Hence, we conclude that the model is a good fit.

##   
## Hosmer and Lemeshow goodness of fit (GOF) test  
##   
## data: model.LASSO$y, fitted(model.LASSO)  
## X-squared = 2.5289, df = 8, p-value = 0.9604

## Odds ratio 2.5 % 97.5 %  
## (Intercept) 4.944525e-24 6.889424e-35 3.548675e-13  
## concave\_points\_mean 1.088441e+02 1.891807e-32 6.262284e+35  
## radius\_se 1.135297e+07 1.672405e+01 7.706858e+12  
## radius\_worst 4.343798e+00 1.996602e+00 9.450347e+00  
## texture\_worst 1.374163e+00 1.141442e+00 1.654331e+00  
## smoothness\_worst 1.357975e+31 1.019954e+03 1.808019e+59  
## concavity\_worst 4.301082e+01 5.946245e-03 3.111091e+05  
## concave\_points\_worst 5.614537e+07 6.131515e-12 5.141148e+26  
## symmetry\_worst 1.455875e+04 4.250841e-02 4.986245e+09

### 4-2-3. Model validation

Below is ROC curve comparison of the two models above. And we see that **LASSO** seems to work better.



### 4-2-4. Interpretation:

Based on the Odds ratio output above, interpretations can be made as below for the parameters that are statiscally significant:

**For radius\_se**: For every 0.01 unit increase in radius\_se, the odds of diagnosis being Malignant will increase by 17.6% [exp(16.24499 \* 0.01)], holding all other explanatories fixed. A 95% confidence interval for the percentage increase is between [ 2.9% [exp(2.8168479 \* 0.01)] and 34.5% [exp(29.6731317 \* 0.01)] ].

**For radius\_worst**: For every 0.1 unit increase in radius\_worst, the odds of diagnosis being Malignant will increase by 15.8% [exp(1.46875 \* 0.1)], holding all other explanatories fixed. A 95% confidence interval for the percentage increase is between [ 7.2% [exp(0.6914469 \* 0.1)] and 25.2% [exp(2.2460514 \* 0.1)] ].

**For texture\_worst**: For every 0.1 unit increase in texture\_worst, the odds of diagnosis being Malignant will increase by 3.2% [exp(0.31784 \* 0.1)], holding all other explanatories fixed. A 95% confidence interval for the percentage increase is between [ 1.3% [exp(0.1322926 \* 0.1)] and 5.2% [exp(0.5033966 \* 0.1)] ].

**For smoothness\_worst**: For every 0.001 unit increase in smoothness\_worst, the odds of diagnosis being Malignant will increase by 7.4% [exp(71.68613 \* 0.001)], holding all other explanatories fixed. A 95% confidence interval for the percentage increase is between [ 0.6% [exp(6.9275128 \* 0.001)] and 14.6% [exp(136.4447521 \* 0.001)] ].

### 4-2-3. Building a complex model using interactions

In order to build a complex model using interactions we use the statistically significant variables derived using LASSO and based on manual intuition add the following interactions: texture\_mean:perimeter\_mean and smoothness\_mean:concave\_points\_mean.

model.null<-glm(diagnosis ~ 1, data=training,family = binomial(link="logit"))  
  
model.complex<-glm(diagnosis ~ radius\_se+radius\_worst+texture\_worst+smoothness\_worst+ texture\_mean:perimeter\_mean+smoothness\_mean:concave\_points\_mean  
 , data=training,family = binomial(link="logit"))

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

step(model.null,  
 scope = list(upper=model.complex),  
 direction="forward",  
 test="Chisq",  
 data=training)

## Start: AIC=462.26  
## diagnosis ~ 1  
##   
## Df Deviance AIC LRT Pr(>Chi)  
## + radius\_worst 1 137.87 141.87 322.39 < 2.2e-16  
## + smoothness\_mean:concave\_points\_mean 1 182.06 186.06 278.20 < 2.2e-16  
## + texture\_mean:perimeter\_mean 1 220.90 224.90 239.36 < 2.2e-16  
## + radius\_se 1 275.88 279.88 184.38 < 2.2e-16  
## + texture\_worst 1 386.87 390.87 73.39 < 2.2e-16  
## + smoothness\_worst 1 390.29 394.29 69.98 < 2.2e-16  
## <none> 460.26 462.26   
##   
## + radius\_worst \*\*\*  
## + smoothness\_mean:concave\_points\_mean \*\*\*  
## + texture\_mean:perimeter\_mean \*\*\*  
## + radius\_se \*\*\*  
## + texture\_worst \*\*\*  
## + smoothness\_worst \*\*\*  
## <none>   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Step: AIC=141.87  
## diagnosis ~ radius\_worst

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Df Deviance AIC LRT Pr(>Chi)  
## + smoothness\_worst 1 85.11 91.11 52.759 3.771e-13  
## + smoothness\_mean:concave\_points\_mean 1 87.19 93.19 50.678 1.088e-12  
## + texture\_worst 1 110.11 116.11 27.762 1.372e-07  
## + radius\_se 1 124.64 130.63 13.233 0.0002750  
## + texture\_mean:perimeter\_mean 1 126.02 132.02 11.847 0.0005776  
## <none> 137.87 141.87   
##   
## + smoothness\_worst \*\*\*  
## + smoothness\_mean:concave\_points\_mean \*\*\*  
## + texture\_worst \*\*\*  
## + radius\_se \*\*\*  
## + texture\_mean:perimeter\_mean \*\*\*  
## <none>   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Step: AIC=91.11  
## diagnosis ~ radius\_worst + smoothness\_worst

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Df Deviance AIC LRT Pr(>Chi)  
## + texture\_worst 1 63.535 71.535 21.5750 3.403e-06  
## + texture\_mean:perimeter\_mean 1 69.258 77.258 15.8513 6.852e-05  
## + radius\_se 1 77.780 85.780 7.3297 0.006782  
## + smoothness\_mean:concave\_points\_mean 1 79.758 87.758 5.3518 0.020700  
## <none> 85.110 91.110   
##   
## + texture\_worst \*\*\*  
## + texture\_mean:perimeter\_mean \*\*\*  
## + radius\_se \*\*   
## + smoothness\_mean:concave\_points\_mean \*   
## <none>   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##   
## Step: AIC=71.53  
## diagnosis ~ radius\_worst + smoothness\_worst + texture\_worst

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Df Deviance AIC LRT Pr(>Chi)  
## + radius\_se 1 54.381 64.381 9.1535 0.002482  
## + smoothness\_mean:concave\_points\_mean 1 55.367 65.367 8.1675 0.004265  
## <none> 63.535 71.535   
## + texture\_mean:perimeter\_mean 1 63.509 73.509 0.0260 0.871895  
##   
## + radius\_se \*\*  
## + smoothness\_mean:concave\_points\_mean \*\*  
## <none>   
## + texture\_mean:perimeter\_mean   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##   
## Step: AIC=64.38  
## diagnosis ~ radius\_worst + smoothness\_worst + texture\_worst +   
## radius\_se

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Df Deviance AIC LRT Pr(>Chi)   
## + smoothness\_mean:concave\_points\_mean 1 51.053 63.053 3.3284 0.06809 .  
## <none> 54.381 64.381   
## + texture\_mean:perimeter\_mean 1 54.065 66.065 0.3157 0.57423   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##   
## Step: AIC=63.05  
## diagnosis ~ radius\_worst + smoothness\_worst + texture\_worst +   
## radius\_se + smoothness\_mean:concave\_points\_mean

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Df Deviance AIC LRT Pr(>Chi)  
## <none> 51.053 63.053   
## + texture\_mean:perimeter\_mean 1 49.612 63.612 1.4408 0.23

##   
## Call: glm(formula = diagnosis ~ radius\_worst + smoothness\_worst + texture\_worst +   
## radius\_se + smoothness\_mean:concave\_points\_mean, family = binomial(link = "logit"),   
## data = training)  
##   
## Coefficients:  
## (Intercept) radius\_worst   
## -51.782 1.672   
## smoothness\_worst texture\_worst   
## 82.703 0.321   
## radius\_se smoothness\_mean:concave\_points\_mean   
## 9.712 401.641   
##   
## Degrees of Freedom: 340 Total (i.e. Null); 335 Residual  
## Null Deviance: 460.3   
## Residual Deviance: 51.05 AIC: 63.05

hoslem.test(model.complex$y, fitted(model.complex), g=10)

##   
## Hosmer and Lemeshow goodness of fit (GOF) test  
##   
## data: model.complex$y, fitted(model.complex)  
## X-squared = 2.1914, df = 8, p-value = 0.9746

# validate model.complex  
y\_obs<-as.numeric(as.character(validation$diagnosis))  
y\_hat\_complex <-predict(model.complex, newdata = validation, type = 'response')  
binomial\_deviance(y\_obs, y\_hat\_complex)

## [1] 23.24822

pred\_complex <-prediction(y\_hat\_complex, y\_obs)  
AUC.complex <-performance(pred\_complex, "auc")@y.values[[1]] #AUC: 0.989  
perf\_complex<-performance(pred\_complex,measure="tpr", x.measure = "fpr")

Make note of the changes to the binomial\_deviance value by adding interaction terms. At the end of this report we will compare various models. We will also add a graphical comparison at the end of this report. Below is complex model that we built and binomial\_deviance. Hosmer Lemeshow test was run to check the lack of fit. And the Hosmer Lemeshow test looks fine since p-value > 0.05.

model.complex<-glm(diagnosis ~ radius\_se+radius\_worst+texture\_worst+smoothness\_worst+ texture\_mean:perimeter\_mean+smoothness\_mean:concave\_points\_mean  
 , data=training,family = binomial(link="logit"))

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

binomial\_deviance(y\_obs, y\_hat\_complex)

## [1] 23.24822

### 4-2-3. LDA

Another competing model was built using all the avilable continuous predictors and LDA. Misclassification error was then calculated as 0.002, in other words, overal accuracy is 0.998. We then check the binomial\_deviance of LDA which is 13.51.

## Truth  
## pred.lda 0 1  
## 0 81 2  
## 1 0 30

## [1] 0.002

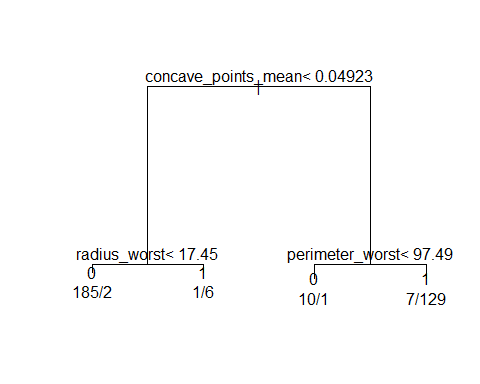
## [1] 0.998

## [1] 0.9984568

## [1] 13.50906

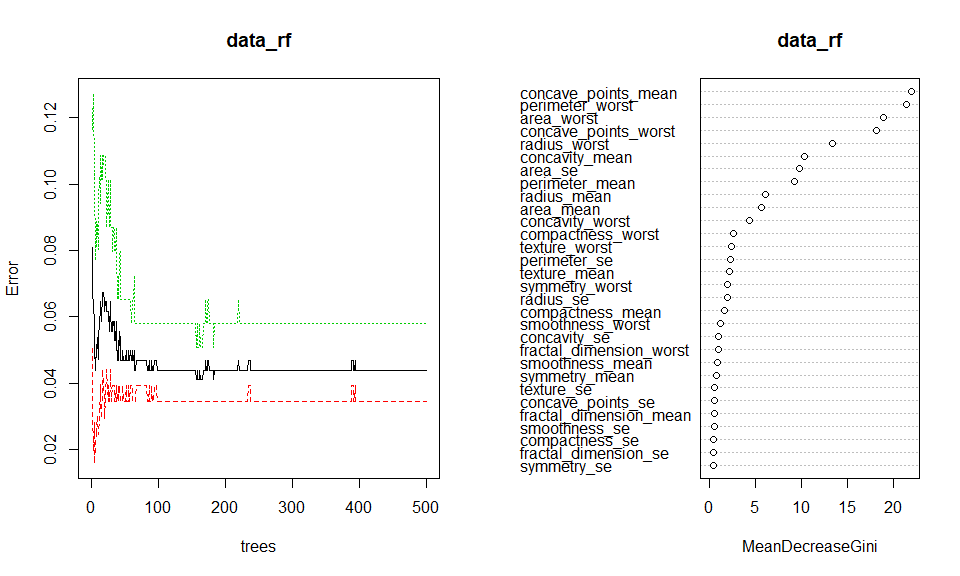
### 4-2-4. Tree model

Tree model gives us very simple and intuitive model as shown below. However, it gives us lower accuracy and higher binomial\_deviance. We will have random forest model as well later. We will add a graphical comparison at the end of this report.



### 4-2-5. Random Forest

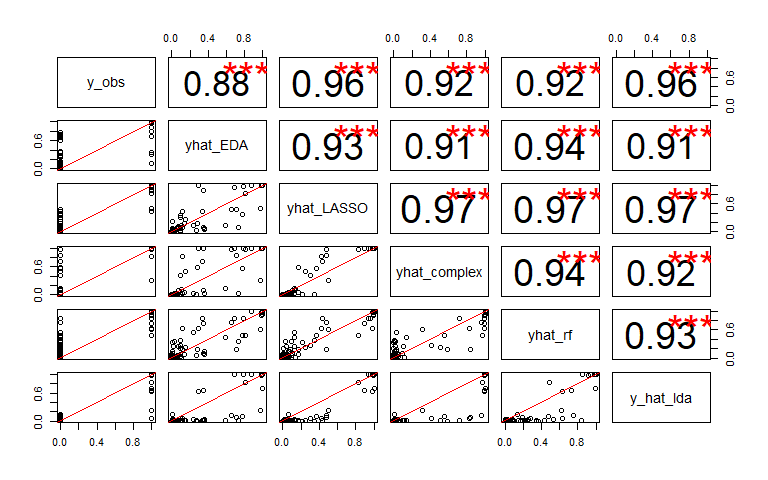
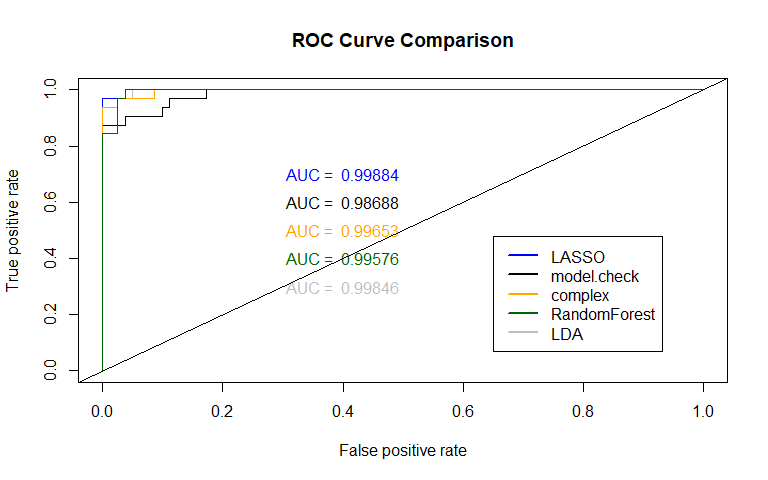
Random forest model has better accuarcy than tree model and lower binomial\_deviance than tree model. We can confirm the reduction of MSE by adding more trees.



# 5. Model comparison

The table below compares various models using AUC and Binomial Deviation values. From the table we can confirm that LASSO has best prediction performance and LDA has least error. From pair plot, we can confirms that LAASO or LDA has best coefficient with observation. Also we see LASSO, complex model (*this was expected*), random forest and LDA are highly correlated (coef: 0.97)

## method auc bin\_dev  
## 1 LASSO 0.9988426 15.53480  
## 2 LDA 0.9984568 13.50906  
## 3 RandomForest 0.9957562 27.05367  
## 4 Complex 0.9965278 23.24822  
## 5 Model.check 0.9868827 31.60577  
## 6 Tree 0.9403935 63.07844



# 6. Conclusion

Based on the analysis, we choose the model from LASSO as the final model, to predict using the metrics alone (most appropriate risk factors) if the biopsy is cancer or not. When we apply this model to new data (test), the performance of LAASO model is seen as: binomial deviance: 14.98, AUC: 0.998

y\_obs\_test <- as.numeric(as.character(test$diagnosis))  
yhat\_glmnet\_test <-predict(bc\_cvfit, s="lambda.min", newx = xx[test\_idx,], type = "response")  
yhat\_glmnet\_test <- yhat\_glmnet\_test[,1]  
pred\_glmnet\_test <- prediction(yhat\_glmnet\_test, y\_obs\_test)  
performance(pred\_glmnet\_test, "auc")@y.values[[1]]

## [1] 0.9983692

binomial\_deviance(y\_obs\_test, yhat\_glmnet\_test)

## [1] 14.98807

# APPENDIX

All the code used to generate this report and complete the analysis is included below:

---

title: "Analysis of the Breast Cancer Dataset"

author: "Neha Dixit, Ravindra Thanniru, Saloni Bhatia, Sangrae Cho"

date: "12/2/2019"

output:

word\_document: default

pdf\_document: default

html\_document:

df\_print: paged

---

```{r setup, include=FALSE}

knitr::opts\_chunk$set(echo = TRUE)

```

```{r requirments, message=FALSE, include=FALSE, echo=FALSE}

# Importing required libraries

library(dplyr)

library(ggplot2)

library(MASS)

library(glmnet)

library(randomForest)

library(gbm)

library(rpart)

library(boot)

library(data.table)

library(ROCR)

library(gridExtra)

library(ResourceSelection)

# library imports

library(tidyverse)

# Date manipulation

library(lubridate)

# Plotting

library(olsrr)

# RMLSE

library(MLmetrics)

# Correlation Matrix

library(ggcorrplot)

# Lasso

library(glmnet)

# AIC commands

library(MASS)

#library(devtools)

library(tidyverse)

library("tree")

library(gridExtra)

library(plyr)

library(dplyr)

library(tidyr)

library(ggplot2)

library(corrplot)

library(e1071)

library(ggthemes)

library(caret)

library(tidyverse)

library(scales)

library("openxlsx")

# Importing the data set

bc<-read.table("https://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/wdbc.data",header=F,sep=",")

names(bc)<- c('id\_number', 'diagnosis', 'radius\_mean',

'texture\_mean', 'perimeter\_mean', 'area\_mean',

'smoothness\_mean', 'compactness\_mean',

'concavity\_mean','concave\_points\_mean',

'symmetry\_mean', 'fractal\_dimension\_mean',

'radius\_se', 'texture\_se', 'perimeter\_se',

'area\_se', 'smoothness\_se', 'compactness\_se',

'concavity\_se', 'concave\_points\_se',

'symmetry\_se', 'fractal\_dimension\_se',

'radius\_worst', 'texture\_worst',

'perimeter\_worst', 'area\_worst',

'smoothness\_worst', 'compactness\_worst',

'concavity\_worst', 'concave\_points\_worst',

'symmetry\_worst', 'fractal\_dimension\_worst')

```

```{r basic funcion,message=FALSE, include=FALSE, echo=FALSE}

# Defining Correlation function

panel.cor <- function(x, y, digits=2, prefix="", cex.cor)

{

usr <- par("usr"); on.exit(par(usr))

par(usr = c(0, 1, 0, 1))

r <- abs(cor(x, y))

txt <- format(c(r, 0.123456789), digits=digits)[1]

txt <- paste(prefix, txt, sep="")

if(missing(cex.cor)) cex <- 0.8/strwidth(txt)

test <- cor.test(x,y)

# borrowed from printCoefmat

Signif <- symnum(test$p.value, corr = FALSE, na = FALSE,

cutpoints = c(0, 0.001, 0.01, 0.05, 0.1, 1),

symbols = c("\*\*\*", "\*\*", "\*", ".", " "))

text(0.5, 0.5, txt, cex = cex \* r)

text(.8, .8, Signif, cex=cex, col=2)

}

binomial\_deviance <- function(y\_obs, yhat){

epsilon = 0.0001

yhat = ifelse(yhat < epsilon, epsilon, yhat)

yhat = ifelse(yhat > 1-epsilon, 1-epsilon, yhat)

a = ifelse(y\_obs==0, 0, y\_obs \* log(y\_obs/yhat))

b = ifelse(y\_obs==1, 0, (1-y\_obs) \* log((1-y\_obs)/(1-yhat)))

return(2\*sum(a + b))

} #prediction accuracy

basic.fit.plots <- function(model) {

# get predicted values

Predicted <- model$fitted.values

# get residuals

Resid <- model$residuals

# get studentized residuals

RStudent <- rstudent(model = model)

# build out a data container from the model

fit.data = data.frame(

'Resid' = Resid,

'RStudent' = RStudent,

'Predicted' = Predicted

)

# create qqplot of residuals with reference line

qqplot.resid <- fit.data %>%

ggplot(aes(sample = Resid)) +

geom\_qq() + geom\_qq\_line() +

labs(subtitle = 'QQ Plot of Residuals',

x = 'Theoretical Quantile',

y = 'Acutal Quantile')

# create histogram of residuals

hist.resid <- fit.data %>%

ggplot(aes(x = Resid)) +

geom\_histogram(bins = 15) +

labs(subtitle = 'Histogram of Residuals',

x = 'Residuals',

y = 'Count')

# create scatter plot of residuals vs

# predicted values

resid.vs.pred <- fit.data %>%

ggplot(aes(x = Predicted, y = Resid)) +

geom\_point() +

geom\_abline(slope = 0) +

labs(subtitle = 'Residuals vs Predictions',

x = 'Predicted Value',

y = 'Residual')

# create scatter plot of studentized

# residuals vs predicted values

rStud.vs.pred <- fit.data %>%

ggplot(aes(x = Predicted, y = RStudent)) +

geom\_point() +

geom\_abline(slope = 0) +

geom\_abline(slope = 0, intercept = -2) +

geom\_abline(slope = 0, intercept = 2) +

labs(subtitle = 'RStudent vs Predictions',

x = 'Predicted Value',

y = 'RStudent')

# add all four plots to grid as

# qqplot histogram

# resid vs pred RStud vs pred

grid.arrange(qqplot.resid,

hist.resid,

resid.vs.pred,

rStud.vs.pred,

nrow = 2,

top = 'Fit Assessment Plots')

}

#### function ends

```

# 1. Introduction

Breast cancer is the most common cancer among women and one of the major causes of death among women worldwide. Every year approximately 124 out of 100,000 women are diagnosed with breast cancer, and the estimation is that 23 out of the 124 women will die of this disease. When detected in its early stages, there is a 30% chance that the cancer can be treated eﬀectively, but the late detection of advanced-stage tumors makes the treatment more diﬃcult. Currently, the most used techniques to detect breast cancer in early stages are: mammography (63% to 97% correctness), FNA (Fine Needle Aspiration) with visual interpretation (65% to 98% correctness) and surgical biopsy (approximately 100% correctness). Therefore, mammography and FNA with visual interpretation correctness varies widely, and the surgical biopsy, although reliable, is invasive and costly.

[Ref: https://www.cancer.org/cancer/breast-cancer/about/how-common-is-breast-cancer.html]

The breast cancer data set used for analysis in this project has numerous measurements taken from tumor biopsies. The goal of using this data set is to address various objectives of the project and build models to predict using the metrics alone if the biopsy is cancer or not.

# 2. Dataset and Description

The data set used for analysis is available at:

https://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/wdbc.data

Observations in the data set are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. The data is comprised of 31 numeric features and one categorical feature, which contain 569 observations. The first two columns gives Sample ID and Classes(diagnosis) which has M, malignant and B, benign breast mass. For each cell nucleaus, the following ten charactoristics were measured:

Radius, Texture,

Perimeter,

Area,

Smoothness,

Compactness,

Concavity,

Concave points,

Symmetry,

Fractal dimension.

For each characteristic three measures are given:

Mean,

Standard error,

Largest/ “worst”.

The dataset we chose for this project was a publicly shared, breast cancer Wisconsin(diagnostic) data made available through Kaggle in csv format.

The dataset consists of 569 rows and 32 attributes. We have both categorical and continous attributes in the dataset. The attributes are summarized as below:

1. id ID number

2. diagnosis The diagnosis of breast tissues (M = malignant, B = benign)

3. radius\_mean mean of distances from center to points on the perimeter

4. texture\_mean standard deviation of gray-scale values

5. perimeter\_mean mean size of the core tumor

6. area\_mean

7. smoothness\_mean mean of local variation in radius lengths

8. compactness\_mean mean of perimeter^2 / area - 1.0

9. concavity\_mean mean of severity of concave portions of the contour

10. concave points\_mean mean for number of concave portions of the contour

11. symmetry\_mean

12. fractal\_dimension\_mean mean for "coastline approximation" - 1

13. radius\_se standard error for the mean of distances from center to points on the perimeter

14. texture\_se standard error for standard deviation of gray-scale values

15. perimeter\_se

16. area\_se

17. smoothness\_se standard error for local variation in radius lengths

18. compactness\_se standard error for perimeter^2 / area - 1.0

19. concavity\_se standard error for severity of concave portions of the contour

20. concave points\_se standard error for number of concave portions of the contour

21. symmetry\_se

22. fractal\_dimension\_se standard error for "coastline approximation" - 1

23. radius\_worst "worst" or largest mean value for mean of distances from center to points on the perimeter

24. texture\_worst "worst" or largest mean value for standard deviation of gray-scale values

25. perimeter\_worst

26. area\_worst

27. smoothness\_worst "worst" or largest mean value for local variation in radius lengths

28. compactness\_worst "worst" or largest mean value for perimeter^2 / area - 1.0

29. concavity\_worst "worst" or largest mean value for severity of concave portions of the contour

30. concave points\_worst "worst" or largest mean value for number of concave portions of the contour

31. symmetry\_worst

32. fractal\_dimension\_worst "worst" or largest mean value for "coastline approximation" - 1

# 3. Exploratory Data Analysis

We start Exploratory Data Analysis (EDA) of the dataset by checking the number of observations, structure of each variable and summary statistics:

## 3-1. Summary statistics and glimpse of the data set

```{r EDA, message = FALSE, include = FALSE, echo = TRUE}

# glimpse(bc)

bc<-bc %>% dplyr::select(-id\_number) #deleting ID

bc$diagnosis <- factor(ifelse(bc$diagnosis == 'B', 0, 1)) # class variable change to 1(M) and 0(B)

glimpse(bc)

```

```{r summary, echo=TRUE}

summary(bc)

```

As seen above, there are no missing values and the Class distribution is: 357 Benign, 212 Malignant. First column represents the sample ID so we deleted the column. Also we changed response variable from M, B to 1 and 0.

## 3-2. Correlation HeatMap

Plotting Correlation Heatmap for the data to observe the nature and extent of correlation between various features in the dataset:

```{r HeatMap, echo = FALSE}

correlator <- function(df){

df %>%

keep(is.numeric) %>%

cor %>%

corrplot( addCoef.col = "white", number.digits = 2,

number.cex = 0.5, method="circle",

order="hclust", title="Variable Corr Heatmap",

tl.srt=45, tl.cex = 0.8)

}

correlator(bc)

```

As seen in the heatmap, there is strong correlation between:

1. Texture\_mean and Texture\_worst

2. area\_se, radius\_se, perimeter\_se, area\_mean, radius\_mean, perimeter\_mean, area\_worst, radius\_worst, perimeter\_worst, concave.points\_worst, concavity\_mean, concave.points\_mean

3. smoothness\_mean, smoothness\_worst, fractal\_dimension\_mean, fractal\_dimension\_worst

4. compactness\_mean, compactness\_worst, concavity\_worst

5. symmetry\_mean, symmetry\_worst

6. compactness\_se, fractal\_dimension\_se, concavity\_se, concavity.points\_se

## 3-3. EDA with various graphics

Of the observations around 40% have Malignant tumor. In malignant group, we see high values on perimeter\_mean, convave.point\_mean, area\_mean.

```{r EDA graphics, message = FALSE, include = FALSE, echo = FALSE}

## visualize with some feature and response

p1 <- bc %>% ggplot(aes(diagnosis)) + geom\_bar()

p2 <- bc %>% ggplot(aes(diagnosis,perimeter\_mean)) +

geom\_jitter(col='gray') +

geom\_boxplot(alpha=.5) # alpha: transparency

p3 <- bc %>% ggplot(aes(diagnosis,concave\_points\_mean)) +

geom\_jitter(col='orange') +

geom\_boxplot(alpha=.5)

p4 <- bc %>% ggplot(aes(diagnosis,area\_mean)) +

geom\_jitter(col='blue') +

geom\_boxplot(alpha=0.5)

p5 <- bc %>% ggplot(aes(diagnosis,texture\_mean)) +

geom\_jitter(col='red') +

geom\_boxplot(alpha=0.5)

p6 <- bc %>% ggplot(aes(concave\_points\_mean, radius\_mean)) +

geom\_jitter(col='grey') +

geom\_smooth()

```

```{r EDA explain, echo=FALSE, fig.width=10, fig.height=5, message = FALSE}

grid.arrange(p1,p2,p3,p4,p5,p6,ncol=3)

```

## 3-3. Pairwise Plots

```{r EDA explain pairplot, fig.width=10, fig.height=8, message = FALSE, echo=FALSE}

pairs(bc %>% dplyr::select(diagnosis, ends\_with('\_mean')) %>%

sample\_n(min(1000, nrow(bc))),

lower.panel = function(x,y){points(x,y); abline(0,1,col='red')},

upper.panel = panel.cor)

```

The pairwise plot also confirms that lots of features are correlated, especially the radius, parameter and area are highly correlated as expected from their relationship. So, from these we can use anyone of them.

Compactness\_mean, concavity\_mean and concave point\_mean are highly correlated so we can use compactness\_mean from here.

Based on the EDA, we think the following features would be of great importance for use in the analysis:

['texture\_mean','perimeter\_mean','smoothness\_mean','concave\_point\_mean','symmetry\_mean']

# 4. Split the available data set into Train, Validate and Test portions:

For the sake of our analysis we split the available data set into Train, Validate and Test subsets.

```{r data split, message=FALSE, include=FALSE, echo=FALSE}

# data spilit 60:20:20 (training: validate: test)

set.seed(1606)

n <- nrow(bc)

idx <- 1:n

training\_idx <- sample(idx, n \* .60)

idx <- setdiff(idx, training\_idx)

validate\_idx <- sample(idx, n \* .20)

test\_idx <- setdiff(idx, validate\_idx)

training <- bc[training\_idx,]

validation <- bc[validate\_idx,]

test <- bc[test\_idx,]

pairs(training[,2:10],col=training$diagnosis)

```

# Objective 1:

Perform logistic regression analysis and provide interpretation of the regression coefficients including confidence intervals.

# 4-1. Problem Statement:

The purpose of this analysis is to build a model to predict using the metrics alone (most appropriate risk factors) if the biopsy is cancer or not.

Model selection techniques will be used and a final model will be selected based on various performance criteria. Interpretation of the selected model will be provided as well.

## 4-2. Model Selection & Logistic Regression

For the model selection, we use LASSO and manual/intuition models and compare them to see which gives us the best results.

### Key Assumptions:

Before running logistic regression we proceed making sure the following two key assumptions are met.

1. Logistic regression requires the observations to be independent of each other. For this data set we do not have information if any of the observations recorded belonged to members from the same family. And hence we assume that all the observations are independent of each other.

2. Logistic regression requires there to be little or no multicollinearity among the independent variables. This means that the independent variables should not be too highly correlated with each other. During the EDA we identified highly correlated variables and we make sure to remove them when building the models.

### 4-2-1. Manual/Intuition: Based on EDA

Based on the EDA, we think the following features would be of great importance for use in the analysis:

['texture\_mean','perimeter\_mean','smoothness\_mean','concave\_point\_mean','symmetry\_mean']

```{r, message = FALSE, include = FALSE, echo = FALSE}

model.check <-glm(diagnosis ~ texture\_mean+perimeter\_mean+smoothness\_mean+concave\_points\_mean+symmetry\_mean, data = training, family = binomial(link = "logit"))

summary(model.check)

```

#### 4-2-1-1. Residual diagnostics

Plots here help us examine Cook's D graph:

```{r, echo = FALSE}

plot(model.check)

```

When checking Cook's D plot, if observations are outside the Cook's distance (meaning they have a high Cook's distance score) the observations are influential to the regression results. In this case, from the Cook's D plot above, we do not see any influential points.

#### 4-2-1-2. Lack of fit test

We then run the Hosmer Lemeshow test to check the lack of fit. And the Hosmer Lemeshow test looks fine since p-value > 0.05.

```{r, echo = FALSE}

hoslem.test(model.check$y, fitted(model.check), g=10) # it is ok

exp(cbind("Odds ratio" = coef(model.check), confint.default(model.check, level = 0.95))) # ODDS ratio

```

```{r validation, message = FALSE, include = FALSE, echo = FALSE}

y\_obs<-as.numeric(as.character(validation$diagnosis))

y\_hat\_lm\_check <-predict(model.check, newdata = validation, type = 'response')

binomial\_deviance(y\_obs, y\_hat\_lm\_check)

pred\_lm\_check <-prediction(y\_hat\_lm\_check, y\_obs)

AUC.lm.check <-performance(pred\_lm\_check, "auc")@y.values[[1]] #AUC: 0.989

perf\_lm\_check<-performance(pred\_lm\_check,measure="tpr", x.measure = "fpr")

```

### 4-2-2. LASSO

We now try to build the model using LASSO. Since LASSO is a regularized model, binomial deviance can be used for model selection. Figure below shows the binomial deviance. The binomial deviance is minimized near the vertical lines (10: best prediction, 8: most simple to interpret).

```{r LASSO, message = FALSE, include = FALSE, echo = FALSE}

xx <- model.matrix(diagnosis ~ .-1, bc) #-1: no need of intercept

x <-xx[training\_idx,]

y <-as.numeric(as.character(training$diagnosis))

glimpse(x)

## using cv.glmnet() to fit

bc\_cvfit<-cv.glmnet(x,y, family = 'binomial')

plot(bc\_cvfit)

###fidning: at 10: best prediction(lambda.min), at 7: most simple to interpret(lambda.1se)

coef(bc\_cvfit, s = c("lambda.1se"))

coef(bc\_cvfit, s = c("lambda.min"))

## validate model

predict.cv.glmnet(bc\_cvfit, s="lambda.min", newx = x[1:5,], type = 'response')

yhat\_glmnet <-predict(bc\_cvfit, s = "lambda.min", newx=xx[validate\_idx,], type = 'response')

yhat\_glmnet <-yhat\_glmnet[,1] # change to a vector from [n\*1] matrix

pred\_glmnet <-prediction(yhat\_glmnet, y\_obs)

AUC.glm <-performance(pred\_glmnet,"auc")@y.values[[1]]

binomial\_deviance(y\_obs, yhat\_glmnet)

perf\_glmnet<-performance(pred\_glmnet, measure = "tpr", x.measure = "fpr")

```

```{r LASSO graph}

coef(bc\_cvfit, s = c("lambda.1se"))

plot(bc\_cvfit)

```

The features selected at the "most simple to interpret" line are: concave\_points\_mean, radius\_se, radius\_worst, texture\_worst, smoothness\_worst, concavity\_worst, concave\_points\_worst, symmetry\_worst.

```{r interpretation, echo = TRUE}

model.LASSO <-glm(diagnosis ~ concave\_points\_mean + radius\_se + radius\_worst + texture\_worst + smoothness\_worst +

concavity\_worst + concave\_points\_worst + symmetry\_worst,

data = training, family = binomial(link = "logit"))

summary(model.LASSO)

confint.default(model.LASSO, level = 0.95)

```

From the above output we see that only the following parameters are \*\*statistically significant\*\*: radius\_se, radius\_worst, texture\_worst and smoothness\_worst.

#### 4-2-2-1. Residual diagnostics

Plots here help us examine Cook's D graph:

```{r, echo = FALSE}

plot(model.LASSO)

```

When checking Cook's D plot, if observations are outside the Cook's distance (meaning they have a high Cook's distance score) the observations are influential to the regression results. In this case, from the Cook's D plot above, we do not see any influential points.

#### 4-2-2-2. Lack of fit test

We then run the Hosmer Lemeshow test to check the lack of fit. And the Hosmer Lemeshow test looks fine since p-value > 0.05.

```{r, echo = FALSE}

hoslem.test(model.LASSO$y, fitted(model.LASSO), g=10) # it is ok

exp(cbind("Odds ratio" = coef(model.LASSO), confint.default(model.LASSO, level = 0.95))) # ODDS ratio

```

### 4-2-3. Model validation

Below is ROC curve comparison of the two models above. And we see that \*\*LASSO\*\* seems to work better.

```{r model validation, echo=FALSE}

plot(perf\_lm\_check, col='black', main="ROC Curve Comparison")

text(x = .40, y = .6,paste("AUC = ", round(AUC.lm.check,5)), col='black')

plot(perf\_glmnet, col = 'blue', add = TRUE)

text(x = .40, y = .7,paste("AUC = ", round(AUC.glm,5)), col='blue')

abline(0,1)

legend('bottomright', inset = .1,

legend = c("LASSO", "model.check"),

col=c('blue','black'), lty=1, lwd=2)

```

### 4-2-4. Interpretation:

Based on the Odds ratio output above, interpretations can be made as below for the parameters that are statiscally significant:

\*\*For radius\_se\*\*: For every 0.01 unit increase in radius\_se, the odds of diagnosis being Malignant will increase by 17.6% [exp(16.24499 \* 0.01)], holding all other explanatories fixed. A 95% confidence interval for the percentage increase is between [ 2.9% [exp(2.8168479 \* 0.01)] and 34.5% [exp(29.6731317 \* 0.01)] ].

\*\*For radius\_worst\*\*: For every 0.1 unit increase in radius\_worst, the odds of diagnosis being Malignant will increase by 15.8% [exp(1.46875 \* 0.1)], holding all other explanatories fixed. A 95% confidence interval for the percentage increase is between [ 7.2% [exp(0.6914469 \* 0.1)] and 25.2% [exp(2.2460514 \* 0.1)] ].

\*\*For texture\_worst\*\*: For every 0.1 unit increase in texture\_worst, the odds of diagnosis being Malignant will increase by 3.2% [exp(0.31784 \* 0.1)], holding all other explanatories fixed. A 95% confidence interval for the percentage increase is between [ 1.3% [exp(0.1322926 \* 0.1)] and 5.2% [exp(0.5033966 \* 0.1)] ].

\*\*For smoothness\_worst\*\*: For every 0.001 unit increase in smoothness\_worst, the odds of diagnosis being Malignant will increase by 7.4% [exp(71.68613 \* 0.001)], holding all other explanatories fixed. A 95% confidence interval for the percentage increase is between [ 0.6% [exp(6.9275128 \* 0.001)] and 14.6% [exp(136.4447521 \* 0.001)] ].

### 4-2-3. Building a complex model using interactions

In order to build a complex model using interactions we use the statistically significant variables derived using LASSO and based on manual intuition add the following interactions: texture\_mean:perimeter\_mean and smoothness\_mean:concave\_points\_mean.

```{r complex model, message = FALSE, include = TRUE, echo = TRUE}

model.null<-glm(diagnosis ~ 1, data=training,family = binomial(link="logit"))

model.complex<-glm(diagnosis ~ radius\_se+radius\_worst+texture\_worst+smoothness\_worst+ texture\_mean:perimeter\_mean+smoothness\_mean:concave\_points\_mean

, data=training,family = binomial(link="logit"))

step(model.null,

scope = list(upper=model.complex),

direction="forward",

test="Chisq",

data=training)

hoslem.test(model.complex$y, fitted(model.complex), g=10)

# validate model.complex

y\_obs<-as.numeric(as.character(validation$diagnosis))

y\_hat\_complex <-predict(model.complex, newdata = validation, type = 'response')

binomial\_deviance(y\_obs, y\_hat\_complex)

pred\_complex <-prediction(y\_hat\_complex, y\_obs)

AUC.complex <-performance(pred\_complex, "auc")@y.values[[1]] #AUC: 0.989

perf\_complex<-performance(pred\_complex,measure="tpr", x.measure = "fpr")

```

Make note of the changes to the binomial\_deviance value by adding interaction terms. At the end of this report we will compare various models. We will also add a graphical comparison at the end of this report. Below is complex model that we built and binomial\_deviance. Hosmer Lemeshow test was run to check the lack of fit. And the Hosmer Lemeshow test looks fine since p-value > 0.05.

```{r complex model\_, echo=TRUE}

model.complex<-glm(diagnosis ~ radius\_se+radius\_worst+texture\_worst+smoothness\_worst+ texture\_mean:perimeter\_mean+smoothness\_mean:concave\_points\_mean

, data=training,family = binomial(link="logit"))

binomial\_deviance(y\_obs, y\_hat\_complex)

```

### 4-2-3. LDA

Another competing model was built using all the avilable continuous predictors and LDA. Misclassification error was then calculated as 0.002, in other words, overal accuracy is 0.998. We then check the binomial\_deviance of LDA which is 13.51.

```{r LDA, echo=FALSE}

# LDA

model.lda<-lda(diagnosis~.,data=training)

pred.lda<-predict(model.lda,newdata=validation)$class

Truth<-validation$diagnosis

x<-table(pred.lda,Truth)

x

#Missclassification Error

ME<-(x[2,1]+x[1,2])/1000

ME

#Calculating overall accuracy

1-ME

y\_hat\_lda<-predict(model.lda,newdata=validation)$posterior

y\_hat\_lda<-y\_hat\_lda[,"1"] # getting y\_hat when "1"

pred\_lda <- prediction(y\_hat\_lda, y\_obs)

performance(pred\_lda, "auc")@y.values[[1]]

binomial\_deviance(y\_obs, y\_hat\_lda)

AUC.lda <-performance(pred\_lda, "auc")@y.values[[1]]

perf\_lda<-performance(pred\_lda,measure="tpr", x.measure = "fpr")

```

### 4-2-4. Tree model

Tree model gives us very simple and intuitive model as shown below. However, it gives us lower accuracy and higher binomial\_deviance. We will have random forest model as well later. We will add a graphical comparison at the end of this report.

```{r tree model, message = FALSE, include = FALSE, echo = FALSE}

# Tree model

bc\_tr <-rpart(diagnosis ~., data = training)

bc\_tr

printcp(bc\_tr)

summary(bc\_tr)

opar <- par(mfrow=c(1,1), xpd = NA)

plot(bc\_tr)

text(bc\_tr, use.n = TRUE)

par(opar)

# dev.off()

#validate bc\_tr

yhat\_tr <- predict(bc\_tr, validation)

yhat\_tr <- yhat\_tr[,"1"]

pred\_tr <- prediction(yhat\_tr, y\_obs)

performance(pred\_tr, "auc")@y.values[[1]]

binomial\_deviance(y\_obs, yhat\_tr)

```

```{r tree graph, echo = FALSE}

opar <- par(mfrow=c(1,1), xpd = NA)

plot(bc\_tr)

text(bc\_tr, use.n = TRUE)

par(opar)

```

### 4-2-5. Random Forest

Random forest model has better accuarcy than tree model and lower binomial\_deviance than tree model. We can confirm the reduction of MSE by adding more trees.

```{r rf,model, message = FALSE, include = FALSE, echo = FALSE}

# random forest

set.seed(1607)

data\_rf <- randomForest(diagnosis ~ ., training)

data\_rf

opar <- par(mfrow=c(1,2))

plot(data\_rf)

varImpPlot(data\_rf)

par(opar)

dev.off()

yhat\_rf <- predict(data\_rf, newdata=validation, type='prob')[,'1']

pred\_rf <- prediction(yhat\_rf, y\_obs)

AUC\_rf <- performance(pred\_rf, "auc")@y.values[[1]]

perf\_rf <- performance(pred\_rf, measure = "tpr", x.measure = "fpr")

binomial\_deviance(y\_obs, yhat\_rf)

```

```{r random forest, fig.height=6, fig.width=10, echo=FALSE}

opar <-par(mfrow = c(1,2))

plot(data\_rf)

varImpPlot(data\_rf)

par(opar)

```

# 5. Model comparison

The table below compares various models using AUC and Binomial Deviation values. From the table we can confirm that LASSO has best prediction performance and LDA has least error. From pair plot, we can confirms that LAASO or LDA has best coefficient with observation. Also we see LASSO, complex model (\*this was expected\*), random forest and LDA are highly correlated (coef: 0.97)

```{r accuracy comp, echo=FALSE, fig.width=8, fig.height=5}

data.frame(method=c('LASSO','LDA', 'RandomForest', 'Complex', 'Model.check','Tree'),

auc = c(performance(pred\_glmnet, "auc")@y.values[[1]],

performance(pred\_lda, "auc")@y.values[[1]],

performance(pred\_rf, "auc")@y.values[[1]],

performance(pred\_complex, "auc")@y.values[[1]],

performance(pred\_lm\_check, "auc")@y.values[[1]],

performance(pred\_tr, "auc")@y.values[[1]]

),

bin\_dev = c(binomial\_deviance(y\_obs, yhat\_glmnet),

binomial\_deviance(y\_obs, y\_hat\_lda),

binomial\_deviance(y\_obs, yhat\_rf),

binomial\_deviance(y\_obs, y\_hat\_complex),

binomial\_deviance(y\_obs, y\_hat\_lm\_check),

binomial\_deviance(y\_obs, yhat\_tr)

)

)

### ROC curve comparison

# png("./plots/1-4.png", 5.5\*1.2, 4\*1.2, units='in', pointsize=9, res=600)

plot(perf\_lm\_check, col='black', main="ROC Curve Comparison")

text(x = .40, y = .6,paste("AUC = ", round(AUC.lm.check,5)), col='black')

plot(perf\_lda, col='grey',add = TRUE)

text(x = .40, y = .3,paste("AUC = ", round(AUC.lda,5)), col='grey')

plot(perf\_glmnet, col = 'blue', add = TRUE)

text(x = .40, y = .7,paste("AUC = ", round(AUC.glm,5)), col='blue')

plot(perf\_complex, col ='orange', add = TRUE)

text(x = .40, y = .5,paste("AUC = ", round(AUC.complex,5)), col='orange')

plot(perf\_rf, col = "dark green", add=TRUE)

text(x = .40, y = .4,paste("AUC = ", round(AUC\_rf,5)), col='dark green')

abline(0,1)

legend('bottomright', inset = .1,

legend = c("LASSO", "model.check", "complex", "RandomForest", "LDA"),

col=c('blue','black','orange','dark green','grey'), lty=1, lwd=2)

#pair plot

pairs(data.frame(y\_obs = y\_obs,

yhat\_EDA = y\_hat\_lm\_check,

yhat\_LASSO = c(yhat\_glmnet),

yhat\_complex = y\_hat\_complex,

yhat\_rf = yhat\_rf,

y\_hat\_lda = y\_hat\_lda),

lower.panel = function(x,y){ points(x,y); abline(0, 1, col ='red')},

upper.panel = panel.cor)

```

# 6. Conclusion

Based on the analysis, we choose the model from LASSO as the final model, to predict using the metrics alone (most appropriate risk factors) if the biopsy is cancer or not. When we apply this model to new data (test), the performance of LAASO model is seen as: binomial deviance: 14.98, AUC: 0.998

```{r}

y\_obs\_test <- as.numeric(as.character(test$diagnosis))

yhat\_glmnet\_test <-predict(bc\_cvfit, s="lambda.min", newx = xx[test\_idx,], type = "response")

yhat\_glmnet\_test <- yhat\_glmnet\_test[,1]

pred\_glmnet\_test <- prediction(yhat\_glmnet\_test, y\_obs\_test)

performance(pred\_glmnet\_test, "auc")@y.values[[1]]

binomial\_deviance(y\_obs\_test, yhat\_glmnet\_test)

```