# Prediction of Breast Cancer Diagnosis

Eliza Chai, Ingrid Luo

2022-11-29

## Scientific Background

Breast cancer is the second most common cause of death from cancer in women in the US after lung cancer. Genetic, environmental and behavioral factors contribute to the wide variation in the clinical course of breast cancer, and the length of time that patients can expect to live. Breast cancer occurs as a result of disordered proliferation and constant growth of cells in the breast tissue, known as tumor. A tumor can be benign (not cancerous) or malignant (cancerous). Benign tumors do not invade other tissues and do not spread to other parts of the body, and can be removed surgically. In other words, these tumors are not life-threatening. Malignant tumors grow and multiply far more quickly than benign tumors. They spread systematically and invade vital organs, which deteriorate the health of patients. Whether a tumor is benign or malignant is determined by pathological examination, such as fine-needle aspiration (FNA). In this technique, a thin (23–25 gauge), hollow needle is inserted into the mass for sampling of cells that, after being stained, will be examined under a microscope (biopsy). Fine-needle aspiration biopsies are very safe minor surgical procedures.

## Goal of the Study

The identification and classification of breast tumor tissues is a critical step for clinicians to accurately and effectively diagnose breast cancer in patients at an early stage. The objective of the study is to use classification techniques (logistic regression model and random forest) to predict the diagnosis of breast tumor tissues as either malignant or benign based on ten cytological features of each FNA of a breast mass.

## Problem of Interest

The problem of interest is choosing the most related features in predicting malignant or benign breast cancer and test the performance of the selected algorithm for breast cancer diagnosis.

## Data Description

The Breast Cancer (Wisconsin) Diagnosis dataset contains the diagnosis and a set of 30 features describing the characteristics of the cell nuclei in the breast tissue present in the digitized image of a fine needle aspirate (FNA) of a breast mass. One feature is an identification number, another is the cancer diagnosis that is coded as “M” to indicate malignant or “B” to indicate benign. The other 30 laboratory measurements include the mean, standard error and worst (i.e. largest) value for 10 features for each cell nucleus, which are as follows:

- radius (mean of distances from center to points on the perimeter)
- texture (standard deviation of gray-scale values)
- perimeter
- area

- smoothness (local variation in radius lengths)
- compactness (perimeter^2 / area - 1.0)
- concavity (severity of concave portions of the contour)
- concave points (number of concave portions of the contour)
- symmetry
- fractal dimension ("coastline approximation" - 1)

Cancer diagnosis is the qualitative response variable (M = malignant, B = benign). The above 10 features are the predictors/covariates and they are quantitative.

## Data Preparation

### Missing data

The dataset used is clean and does not have any missing values in the covariates.

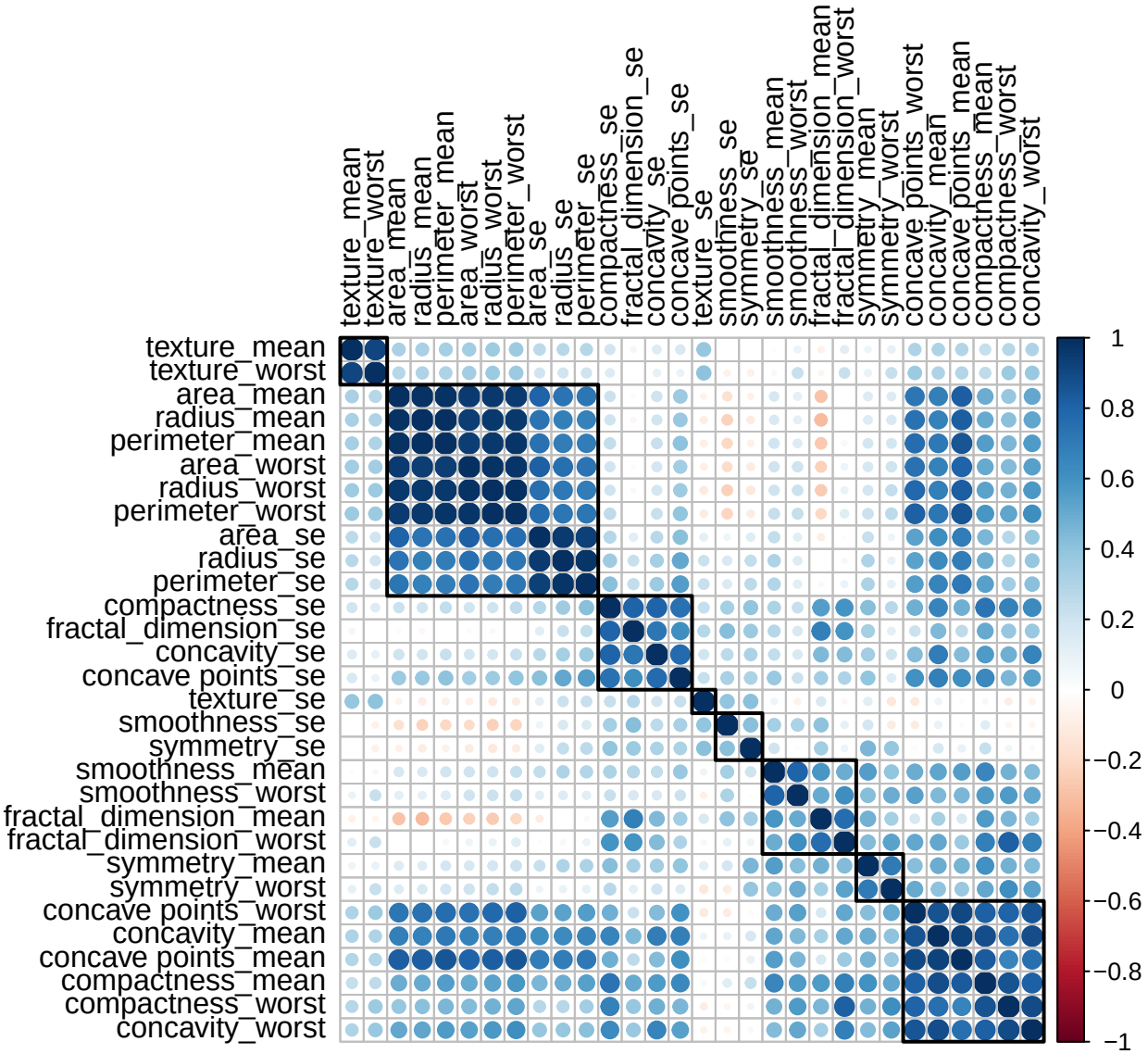
### Balance data

We first check whether our response variable is balanced in the dataset. 63% of observations were benign, and 27% of observations were malignant, which suggests that our data is slightly unbalanced.

### Correlation

We then check for correlations. From the correlation plot, we observe that quite a few variables are correlated. Since features with high correlation are more linearly dependent and would have similar effect on the diagnosis (response variable), we use findcorrelation() from the caret package to remove highly correlated predictors based on whose correlation is above 0.7. This function uses a heuristic algorithm to search through the correlation matrix and determine which variable should be removed that would reduce pairwise correlations.

After removing the correlated variables, we have 10 covariates remaining in the dataset (21 variables were removed).


texture_mean
texture_worst
area_mean
radius_mean
perimeter_mean
area_worst
radius_worst
perimeter_worst
area_se
radius_se
perimeter_se
compactness_se
fractal_dimension_se
concavity_se
concave points_se
texture_se
smoothness_se
symmetry_se
smoothness_mean
smoothness_worst
fractal_dimension_mean
fractal_dimension_worst
symmetry_mean
symmetry_worst
concave points_worst
concavity_mean
concave points_mean
compactness_mean
compactness_worst
concavity_worst
1
0.8
0.6
0.4
0.2
0
−0.2
−0.4
−0.6
−0.8
−1

**Multicollinearity**

Multicollinearity occurs when there is a correlation between 2 or more independent variables in the regression model which would affect the classification accuracy. Since we have correlated variables in the dataset, we examine multicollinearity within the dataset to remove variables that are multicollinear. We fit a logistic regression model with all 10 variables and examine the variance inflation factor (VIF), which is a measure of the amount of multicollinearity in regression analysis. From the result, we observe that the variable 'concavity_worst' has the highest VIF (vif = 13.23).

```
##            texture_mean          symmetry_worst         perimeter_worst
##                1.788144                1.808895                1.975938
##            perimeter_se  fractal_dimension_mean  `concave points_worst`
##                2.422001                5.456711                7.368343
## fractal_dimension_worst     `concave points_se`            concavity_se
##                8.447648                8.805221               12.148037
##         concavity_worst
##               13.233101
```

Therefore, we drop the variable 'concavity_worst', which has the high VIF (vif >10 indicates multicollinearity). We then refit the logistic regression model with 9 variables and confirm all variables have VIF less than 10 from the regression model output.

```
##
## Call:
## glm(formula = diagnosis ~ ., family = binomial, data = data2)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.0623  -0.0582  -0.0075   0.0030   3.3441
##
## Coefficients:
##                           Estimate Std. Error z value Pr(>|z|)
## (Intercept)             -32.06101    7.41713  -4.323 1.54e-05 ***
## texture_mean              0.40717    0.08461   4.812 1.49e-06 ***
## fractal_dimension_mean   25.94451   88.46825   0.293 0.769321
## perimeter_se              1.26766    0.37099   3.417 0.000633 ***
## concavity_se             -7.46263   19.25771  -0.388 0.698376
## `concave points_se`    -131.51136  114.55268  -1.148 0.250951
## perimeter_worst           0.11923    0.03452   3.454 0.000552 ***
## `concave points_worst`   66.69859   16.51216   4.039 5.36e-05 ***
## symmetry_worst            9.75953    5.76224   1.694 0.090322 .
## fractal_dimension_worst  -38.52084   32.77397  -1.175 0.239855
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 750.507  on 567  degrees of freedom
## Residual deviance:  97.356  on 558  degrees of freedom
## AIC: 117.36
##
## Number of Fisher Scoring iterations: 9
```

```
##         perimeter_worst          symmetry_worst            texture_mean
```

```
##                1.730923                1.791675                1.800094
##           perimeter_se           concavity_se  `concave points_worst`
##                2.275838                3.580453                4.525713
##  fractal_dimension_mean     `concave points_se` fractal_dimension_worst
##                4.873821                5.013565                6.061464
```

### Outliers

All features have outliers which would affect the accuracy of the models. However, as our dataset is not large, we decide to keep the outliers in order to keep as much data as possible.

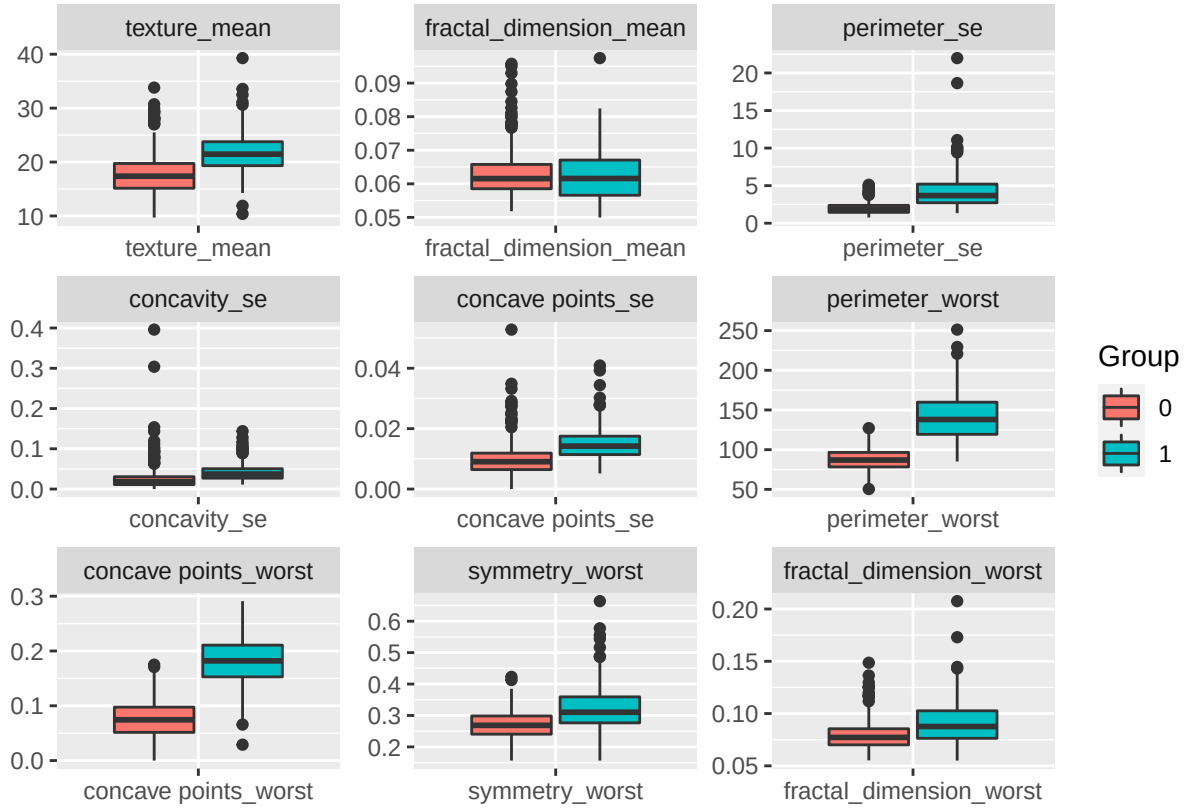



Figure 1. Features

### Descriptive Statistics

In general, malignant diagnoses have higher scores in all features.

Table 1: Distribution of the selected features, stratified by two groups of diagnosis. Most of the features are right skewed, and are summarized as median (IQR).

| Feature | Benign, N = 356 | Malignant, N = 212 |
|---|---|---|
| texture_mean | 17.38 (15.14, 19.74) | 21.46 (19.33, 23.76) |
| fractal_dimension_mean | 0.06 (0.06, 0.07) | 0.06 (0.06, 0.07) |
| perimeter_se | 1.85 (1.44, 2.37) | 3.68 (2.72, 5.21) |
| concavity_se | 0.02 (0.01, 0.03) | 0.04 (0.03, 0.05) |
| concave points_se | 0.01 (0.01, 0.01) | 0.01 (0.01, 0.02) |
| perimeter_worst | 86.94 (78.28, 96.61) | 138.00 (119.33, 159.80) |
| concave points_worst | 0.07 (0.05, 0.10) | 0.18 (0.15, 0.21) |
| symmetry_worst | 0.27 (0.24, 0.30) | 0.31 (0.28, 0.36) |
| fractal_dimension_worst | 0.08 (0.07, 0.09) | 0.09 (0.08, 0.10) |

**Test-Train split**

We randomly assign 80% of the observations in the dataset to a training set, and the remaining 20% of the observations to a test set to prevent overfitting and to accurately evaluate the model.

**Turn dataset into numeric**

We create a separate dataset by converting the original dataset into numeric to prepare for random forest classification.

## Modeling

**Logistic regression**

We fit a logistic regression model with diagnosis as the outcome and the main effect of 8 features for each cell nucleus. We used model-based standard error estimates to construct confidence intervals and for hypothesis testing to assess significant predictors (Wald test).

**Logistic regression plus stepwise forward selection with AIC**

We implement the stepwise forward selection algorithm to select the best subset model based on AIC. The stepwise regression model begins with a model only with intercept and adds in variables one by one. This algorithm is beneficial in reducing training times and the chances of overfitting. It also helps us simplify the model which makes it more interpretable. We used model-based standard error estimates to construct confidence intervals and for hypothesis testing to assess significant predictors (Wald test).

**Random forest classification**

We use random forests or random decision forest for classification. The random forest is a classification algorithm that consists of many decision trees to create an uncorrelated forest of trees and returns the class selected by most trees as the output.
## Results (Interpret findings from fitted model):

**Logistic Regression**

Based on the fitted logistic regression model, the variables 'texture_mean', 'perimeter_worst', and 'concave_point_worst' are the significant predictors (Wald test p=0.00006, p=0.0020, p=0.0007 respectively). With the logistic regression model, the misclassification error rate on the test set is 1.77%. The prediction accuracy of the logistic regression model on the test set is 98.23%.

```
##
## Call:
## glm(formula = diagnosis ~ ., family = binomial, data = training_logreg)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9710  -0.0736  -0.0107   0.0061   3.6402
##
## Coefficients:
##                           Estimate Std. Error z value Pr(>|z|)
## (Intercept)             -31.33152    8.50955  -3.682 0.000231 ***
## texture_mean              0.35173    0.08800   3.997 6.42e-05 ***
## fractal_dimension_mean   40.35428  105.86750   0.381 0.703072
## perimeter_se              0.58532    0.51564   1.135 0.256319
## concavity_se             -3.59209   21.09242  -0.170 0.864772
## `concave points_se`     -96.70777  121.10276  -0.799 0.424546
## perimeter_worst           0.12772    0.04126   3.096 0.001964 **
## `concave points_worst`   61.60155   18.23979   3.377 0.000732 ***
## symmetry_worst           11.28046    6.96850   1.619 0.105495
## fractal_dimension_worst -39.22750   35.86823  -1.094 0.274106
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 601.380  on 454  degrees of freedom
## Residual deviance:  82.057  on 445  degrees of freedom
## AIC: 102.06
##
## Number of Fisher Scoring iterations: 9
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 70  1
##          1  1 41
##
##                Accuracy : 0.9823
##                  95% CI : (0.9375, 0.9978)
##     No Information Rate : 0.6283
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.9621
##
##  Mcnemar's Test P-Value : 1
##
```

```
##             Sensitivity : 0.9859
##             Specificity : 0.9762
##          Pos Pred Value : 0.9859
##          Neg Pred Value : 0.9762
##              Prevalence : 0.6283
##          Detection Rate : 0.6195
##    Detection Prevalence : 0.6283
##       Balanced Accuracy : 0.9811
##
##        'Positive' Class : 0
##
```

**Logistic regression plus stepwise forward selection with AIC**

Based on the fitted stepwise forward logistic regression model, the variables 'texture_mean', 'perimeter_worst', 'concave_point_worst', and 'symmetry_worst' are the significant predictors (Wald test p=0.00007, p=0.0000001, p=0.00017, p=0.042 respectively).

```
##            (Intercept)        perimeter_worst `concave points_worst`
##            -32.0784959              0.1502389             41.7761465
##           texture_mean         symmetry_worst
##              0.3258787             13.1430671
```

With the stepwise forward regression model, the misclassification error rate on the test set is 3.54%. The prediction accuracy of the stepwise forward regression on the test set is 96.46%.

```
##
## Call:
## glm(formula = diagnosis ~ . - fractal_dimension_mean - concavity_se -
##     perimeter_se - concavity_se - `concave points_se` - fractal_dimension_worst,
##     family = binomial, data = training_logreg)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2055  -0.0822  -0.0139   0.0093   3.9475
##
## Coefficients:
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)            -32.07850    4.70767  -6.814 9.49e-12 ***
## texture_mean             0.32588    0.08201   3.973 7.08e-05 ***
## perimeter_worst          0.15024    0.02835   5.299 1.17e-07 ***
## `concave points_worst`  41.77615   11.10760   3.761 0.000169 ***
## symmetry_worst          13.14307    6.45396   2.036 0.041707 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 601.380  on 454  degrees of freedom
## Residual deviance:  85.595  on 450  degrees of freedom
## AIC: 95.595
##
## Number of Fisher Scoring iterations: 8
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 69  2
##          1  2 40
##
##                Accuracy : 0.9646
##                  95% CI : (0.9118, 0.9903)
##     No Information Rate : 0.6283
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.9242
##
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 0.9718
##             Specificity : 0.9524
##          Pos Pred Value : 0.9718
##          Neg Pred Value : 0.9524
##              Prevalence : 0.6283
##          Detection Rate : 0.6106
##    Detection Prevalence : 0.6283
##       Balanced Accuracy : 0.9621
##
##        'Positive' Class : 0
##
```

**Random forest classification**

Based on the random forest classification, the variables 'texture_mean', 'perimeter_se', 'perimeter_worst', 'concave_point_worst', and 'symmetry_worst' are the significant predictors (Wald test p=0.0099, p=0.0099, p=0.0099, p=0.0099, p=0.049 respectively).

```
##                          importance     pvalue
## texture_mean            0.018991106 0.00990099
## perimeter_se            0.031271884 0.00990099
## concave_points_se       0.007458072 0.10891089
## perimeter_worst         0.171086598 0.00990099
## concave_points_worst    0.132254738 0.00990099
## symmetry_worst          0.008754654 0.04950495
## fractal_dimension_worst 0.004268827 0.21782178
```

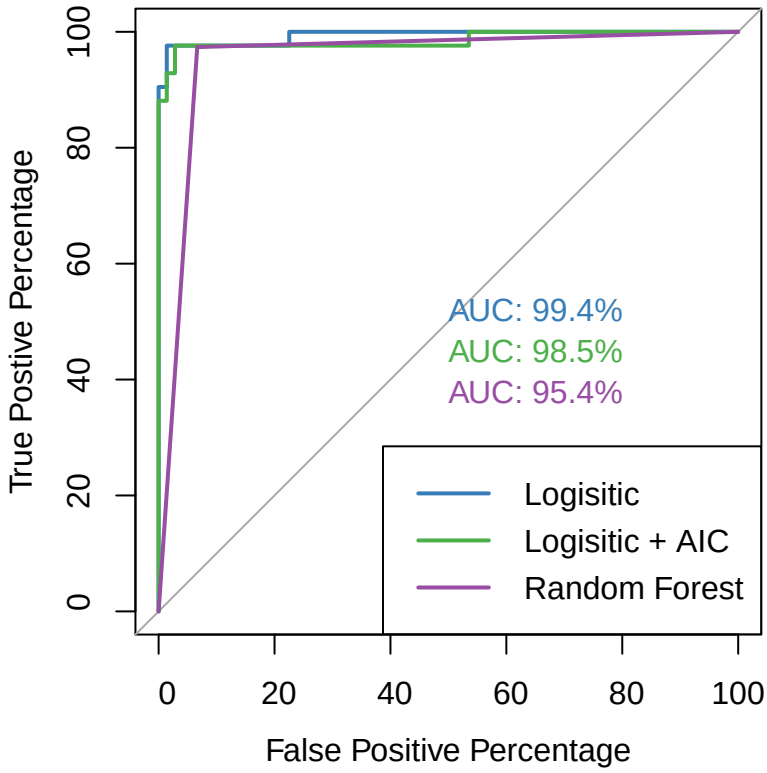
With the random forest classification, the misclassification error rate on the test set is 5.31%. The prediction accuracy for the random forest on the test set is 94.69%.

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 70  1
##          1  5 37
##
```

```
##                Accuracy : 0.9469
##                  95% CI : (0.888, 0.9803)
##     No Information Rate : 0.6637
##     P-Value [Acc > NIR] : 3.667e-13
##
##                   Kappa : 0.8841
##
##  Mcnemar's Test P-Value : 0.2207
##
##             Sensitivity : 0.9333
##             Specificity : 0.9737
##          Pos Pred Value : 0.9859
##          Neg Pred Value : 0.8810
##              Prevalence : 0.6637
##          Detection Rate : 0.6195
##    Detection Prevalence : 0.6283
##       Balanced Accuracy : 0.9535
##
##        'Positive' Class : 0
##
```

**ROC analysis**

We plot the ROC curve and compute the area under the curve (AUC) for the three models to compare the prediction accuracy of the three classification methods. From the ROC curve, we see that the logistic regression model gives a curve closest to the top-left corner, indicating a better performance. The logistic regression model also has the highest AUC among the three models, suggesting it has the highest prediction accuracy.




## Conclusion

Overall, the logistic regression model with diagnosis as the outcome and the main effect of 8 features for each cell nucleus has the highest prediction accuracy and the lowest misclassification error rate on the test set among three models. The random forest classification has the lowest accuracy and highest classification error rate on the test set.

## Code appendix

```
knitr::opts_chunk$set(echo=FALSE, message=FALSE, warnings=FALSE, fig.align='center')

# load packages
library(tidyverse)
library(readr)
library(glmnet)
library(caret)
library(ranger)
library(car)
library(janitor)
library(pROC)
library(gtsummary)
library(kableExtra)
library(ggplot2)
library(reshape2)
############
## Read in datasets
############
setwd("~/Downloads")
data <- read_csv("data.csv")

############
## Tidy the data
############
# remove NULL data
data <- data[, -33]
# check for missing values
tmp <- which(complete.cases(data)==FALSE) # no missing data


############
## Transform the data
############
data$diagnosis <- factor(data$diagnosis, levels=c("B","M"), labels=c(0, 1))

############
## Remove correlated variables
############
# check how balanced is our response variable
tab_diagnosis <- round(prop.table(table(data$diagnosis)), 2) # slightly unbalanced
# check for correlations and remove multicollinearity
data_corr <- cor(data %>% select(-c(id, diagnosis)))
corrplot::corrplot(data_corr, tl.col="black", order = "hclust", tl.cex = 1, addrect = 8, insig = "label_
# findcorrelation() from caret package remove highly correlated predictors based on whose correlation i
# This function uses a heuristic algorithm to determine which variable should be removed
data2 <- data %>% select(-findCorrelation(data_corr, cutoff = 0.7))
# number of columns for our new data frame
tmp2 <- head(data2) # 21 variables shorter
# drop variables with high VIF (vif > 10 indicates multi-collinearity)
fit_logistic <- glm(diagnosis ~ ., family = binomial,  data = data2)
fit_logistic_summary <- summary(fit_logistic)
```

```r
sort(vif(fit_logistic))
# remove 'concavity_worst', which has the highest VIF
data2 <- data2 %>% select(-concavity_worst)
fit_logistic_2 <- glm(diagnosis ~ ., family = binomial,  data = data2)
summary(fit_logistic_2)
sort(vif(fit_logistic_2)) # now all variables have VIF less than 10
df.m <- melt(data2, id.var = "diagnosis")
ggplot(df.m, aes(x = variable, y = value)) +
  geom_boxplot(aes(fill = diagnosis)) +
  facet_wrap(~variable, scales = "free") +
  xlab("Figure 1. Features") + ylab("") + guides(fill = guide_legend(title = "Group"))

descrip <-
  data2 %>%
  mutate(diagnosis = case_when(diagnosis == 0 ~ "Benign",
                               diagnosis == 1 ~ "Malignant")) %>%
  tbl_summary(by = diagnosis,
              statistic = all_continuous() ~ "{median} ({p25}, {p75})",
              digits = all_continuous()  ~ c(2, 2)) %>%
  modify_header(list(
    stat_1 ~ "Benign, N = {n}",
    stat_2 ~ "Malignant, N = {n}"
  )) %>%
  as_tibble()
colnames(descrip)[1] <- "Feature"

kable(descrip,
      caption="Distribution of the selected features, stratified by two groups of diagnosis. Most of the
  kable_styling(latex_options = "HOLD_position")
############
## Test-train split (80% vs 20%)
############
set.seed(101)
sampling_index <- createDataPartition(data2$diagnosis, times = 1, p = 0.8, list = FALSE)
training_logreg <- data2[sampling_index, ]
testing_logreg <-  data2[-sampling_index, ]
############
## Fit logistic regression
############
model_logistic_train <- glm(diagnosis ~.,
                            family = binomial, data=training_logreg)
summary(model_logistic_train)
# obtaining test error
obs_test_logreg=testing_logreg[,1]
pred.prob_test_logreg=predict(model_logistic_train, newdata=testing_logreg[,-1], type="response")
pred.class_test_logreg=ifelse(pred.prob_test_logreg > 0.5, 1, 0)
confusionMatrix(data=as.factor(pred.class_test_logreg),
                reference=unlist(as.list(obs_test_logreg))) # accuracy: 98.23%, sensitivity: 98.59%, sp
# misclassification error rate = 1-accuracy = 0.0177
############
## Stepwise forward selection
############
intercept_only <- glm(diagnosis ~ 1, family = binomial,  data = training_logreg) # the model that has o
```

```r
intercept_summary <- summary(intercept_only)

forward <- step(intercept_only, direction="forward", scope=formula(model_logistic_train), trace=0)
forward_anova <-
  forward$anova # displays the forward selection procedure and the variables selected at each step
forward$coefficients # displays the coefficients for the best subset model fitted
############
## Fit logistic regression from stepwise forward
############
model_step_train <- glm(diagnosis ~
                                    .-fractal_dimension_mean-concavity_se-perimeter_se-concavity_se-`co
                            family = binomial,data=training_logreg)
summary(model_step_train)

# obtaining test error
obs_test_step=testing_logreg[,1]
pred.prob_test_step=predict(model_step_train, newdata=testing_logreg[,-1], type="response")
pred.class_test_step=ifelse(pred.prob_test_step > 0.5, 1, 0)
confusionMatrix(data=as.factor(pred.class_test_step),
                reference=unlist(as.list(obs_test_step))) # accuracy: 96.46%, sensitivity: 97.18%, spec
# misclassification error rate = 1-accuracy = 0.0354
############
## Converting the original dataset to numeric
############
turn_to_numeric=function(a){
        if(is.numeric(a)==FALSE) a=as.numeric(a)
}
data3=apply(data2,2,turn_to_numeric)

############
## Test-train split (80% vs 20%)
############
set.seed(101)
a=sample(1:568,455,replace=FALSE) # 0.70*568 ~ 455
training_rf=data3[a,]
test_rf=data3[-a,]

# Fitting the random forest on the training data
model_rf=ranger(diagnosis ~ .-fractal_dimension_mean-concavity_se, data = clean_names(as.data.frame(tra

# Finding which predictors are significant
importance_pvalues(model_rf, method = "altmann", formula = diagnosis ~ .-fractal_dimension_mean-concavi
# Predicting the responses in the test set and obtaining the misclassification
# error rate
pred_rf0=predict(model_rf, data=clean_names(as.data.frame(test_rf))[,-1], type="response")
pred_rf=pred_rf0$predictions
obs_test=test_rf[,1]
err_rf=mean((obs_test - pred_rf)^2)
confusionMatrix(data=as.factor(pred_rf),reference=as.factor(obs_test)) # accuracy: 94.69%, sensitivity:
# misclassification error rate = 1-accuracy = 0.0531
### Plotting the ROC curves
par(pty = "s")
roc(obs_test_logreg$diagnosis, pred.prob_test_logreg, plot=TRUE, legacy.axes=TRUE, percent=TRUE, xlab="
```

```
plot.roc(obs_test_step$diagnosis, pred.prob_test_step, percent=TRUE, col="#4daf4a", lwd=2, print.auc=TRU
plot.roc(obs_test, pred_rf, percent=TRUE, col="#984EA3", lwd=, print.auc=TRUE, add=TRUE, print.auc.y=40)
legend("bottomright", legend=c("Logisitic",
                               "Logisitic + AIC",
                               "Random Forest"), col=c("#377eb8", "#4daf4a", "#984EA3"),lwd=2)
```