Week 6 Discussion Assignment

Lucas Hennessey

10/3/2021

## To get ready to investigate breast cancer data we need to load our libraries and import the data

# opening our libraries, maybe more than I need  
  
library(data.table)

## Warning: package 'data.table' was built under R version 4.0.5

library(dplyr)

## Warning: package 'dplyr' was built under R version 4.0.5

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:data.table':  
##   
## between, first, last

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

library(car)

## Warning: package 'car' was built under R version 4.0.5

## Loading required package: carData

##   
## Attaching package: 'car'

## The following object is masked from 'package:dplyr':  
##   
## recode

library(caret)

## Warning: package 'caret' was built under R version 4.0.5

## Loading required package: lattice

## Loading required package: ggplot2

library(caTools)

## Warning: package 'caTools' was built under R version 4.0.5

library(pROC)

## Warning: package 'pROC' was built under R version 4.0.5

## Type 'citation("pROC")' for a citation.

##   
## Attaching package: 'pROC'

## The following objects are masked from 'package:stats':  
##   
## cov, smooth, var

library(MASS)

##   
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':  
##   
## select

library(tidyverse)

## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

## v tibble 3.0.6 v purrr 0.3.4  
## v tidyr 1.1.2 v stringr 1.4.0  
## v readr 1.4.0 v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse\_conflicts() --  
## x dplyr::between() masks data.table::between()  
## x dplyr::filter() masks stats::filter()  
## x dplyr::first() masks data.table::first()  
## x dplyr::lag() masks stats::lag()  
## x dplyr::last() masks data.table::last()  
## x purrr::lift() masks caret::lift()  
## x car::recode() masks dplyr::recode()  
## x MASS::select() masks dplyr::select()  
## x purrr::some() masks car::some()  
## x purrr::transpose() masks data.table::transpose()

# Importing our dataset and converting to a data frame. Used notepad to add column names before importing   
  
breast.cancer.wisconsin <- read.csv("~/Python\_class/Week 6 Logistic Regression Stuff/breast-cancer-wisconsin.data")  
  
dt <- as.data.table(breast.cancer.wisconsin)  
  
# setting the seed for later analysis  
  
set.seed(1)

## I’m going to take a quick look at our data, then clean and prepare it for our regresssion

# removing id  
  
dt <- dt[, c("id") := NULL]  
  
#checking structure of df  
  
str(dt)

## Classes 'data.table' and 'data.frame': 699 obs. of 10 variables:  
## $ clump\_thickness : int 5 5 3 6 4 8 1 2 2 4 ...  
## $ cell\_size : int 1 4 1 8 1 10 1 1 1 2 ...  
## $ cell\_shape : int 1 4 1 8 1 10 1 2 1 1 ...  
## $ marginal\_adhesion: int 1 5 1 1 3 8 1 1 1 1 ...  
## $ se\_cell\_size : int 2 7 2 3 2 7 2 2 2 2 ...  
## $ bare\_nucleoli : chr "1" "10" "2" "4" ...  
## $ bland\_chromatin : int 3 3 3 3 3 9 3 3 1 2 ...  
## $ normal\_nucleoli : int 1 2 1 7 1 7 1 1 1 1 ...  
## $ mitoses : int 1 1 1 1 1 1 1 1 5 1 ...  
## $ diagnosis : int 2 2 2 2 2 4 2 2 2 2 ...  
## - attr(\*, ".internal.selfref")=<externalptr>

# checking summary statistics  
  
summary(dt)

## clump\_thickness cell\_size cell\_shape marginal\_adhesion  
## Min. : 1.000 Min. : 1.000 Min. : 1.000 Min. : 1.000   
## 1st Qu.: 2.000 1st Qu.: 1.000 1st Qu.: 1.000 1st Qu.: 1.000   
## Median : 4.000 Median : 1.000 Median : 1.000 Median : 1.000   
## Mean : 4.418 Mean : 3.134 Mean : 3.207 Mean : 2.807   
## 3rd Qu.: 6.000 3rd Qu.: 5.000 3rd Qu.: 5.000 3rd Qu.: 4.000   
## Max. :10.000 Max. :10.000 Max. :10.000 Max. :10.000   
## se\_cell\_size bare\_nucleoli bland\_chromatin normal\_nucleoli   
## Min. : 1.000 Length:699 Min. : 1.000 Min. : 1.000   
## 1st Qu.: 2.000 Class :character 1st Qu.: 2.000 1st Qu.: 1.000   
## Median : 2.000 Mode :character Median : 3.000 Median : 1.000   
## Mean : 3.216 Mean : 3.438 Mean : 2.867   
## 3rd Qu.: 4.000 3rd Qu.: 5.000 3rd Qu.: 4.000   
## Max. :10.000 Max. :10.000 Max. :10.000   
## mitoses diagnosis   
## Min. : 1.000 Min. :2.00   
## 1st Qu.: 1.000 1st Qu.:2.00   
## Median : 1.000 Median :2.00   
## Mean : 1.589 Mean :2.69   
## 3rd Qu.: 1.000 3rd Qu.:4.00   
## Max. :10.000 Max. :4.00

# converting data types to allow for logistic regression  
  
dt$bare\_nucleoli <- as.integer(dt$bare\_nucleoli)

## Warning: NAs introduced by coercion

dt$diagnosis <- factor(dt$diagnosis, labels =c('benign', 'malignant'))  
  
str(dt)

## Classes 'data.table' and 'data.frame': 699 obs. of 10 variables:  
## $ clump\_thickness : int 5 5 3 6 4 8 1 2 2 4 ...  
## $ cell\_size : int 1 4 1 8 1 10 1 1 1 2 ...  
## $ cell\_shape : int 1 4 1 8 1 10 1 2 1 1 ...  
## $ marginal\_adhesion: int 1 5 1 1 3 8 1 1 1 1 ...  
## $ se\_cell\_size : int 2 7 2 3 2 7 2 2 2 2 ...  
## $ bare\_nucleoli : int 1 10 2 4 1 10 10 1 1 1 ...  
## $ bland\_chromatin : int 3 3 3 3 3 9 3 3 1 2 ...  
## $ normal\_nucleoli : int 1 2 1 7 1 7 1 1 1 1 ...  
## $ mitoses : int 1 1 1 1 1 1 1 1 5 1 ...  
## $ diagnosis : Factor w/ 2 levels "benign","malignant": 1 1 1 1 1 2 1 1 1 1 ...  
## - attr(\*, ".internal.selfref")=<externalptr>

# removing NAs  
  
dt <- dt[complete.cases(dt)]  
  
# finally split our data into a training and a test set  
  
samp <- sample.split(dt$diagnosis, SplitRatio = 0.8)  
train <- subset(dt, samp == TRUE)  
test <- subset(dt, samp == FALSE)

## Now to model our cleaned, prepped data on our training set and take a look

# creating the model  
  
model <- glm(diagnosis ~ ., data = train, family = "binomial")  
  
# looking at the results  
  
summary(model)

##   
## Call:  
## glm(formula = diagnosis ~ ., family = "binomial", data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -3.4921 -0.1236 -0.0556 0.0150 2.4024   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -10.03688 1.35360 -7.415 1.22e-13 \*\*\*  
## clump\_thickness 0.54984 0.17353 3.169 0.001532 \*\*   
## cell\_size 0.24492 0.28598 0.856 0.391759   
## cell\_shape 0.26000 0.30637 0.849 0.396078   
## marginal\_adhesion 0.23523 0.13987 1.682 0.092598 .   
## se\_cell\_size 0.03456 0.17802 0.194 0.846089   
## bare\_nucleoli 0.46406 0.11937 3.888 0.000101 \*\*\*  
## bland\_chromatin 0.29953 0.19611 1.527 0.126666   
## normal\_nucleoli 0.22741 0.13558 1.677 0.093487 .   
## mitoses 0.61735 0.33894 1.821 0.068549 .   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 706.888 on 545 degrees of freedom  
## Residual deviance: 78.196 on 536 degrees of freedom  
## AIC: 98.196  
##   
## Number of Fisher Scoring iterations: 8

# checking for collinearity (assumption checking)  
  
vif(model)

## clump\_thickness cell\_size cell\_shape marginal\_adhesion   
## 1.102854 2.701849 2.772975 1.264947   
## se\_cell\_size bare\_nucleoli bland\_chromatin normal\_nucleoli   
## 1.402168 1.162031 1.405300 1.201308   
## mitoses   
## 1.065620

# performing step AIC to remove high p values  
  
stepAIC(model, dirrection = 'both')

## Start: AIC=98.2  
## diagnosis ~ clump\_thickness + cell\_size + cell\_shape + marginal\_adhesion +   
## se\_cell\_size + bare\_nucleoli + bland\_chromatin + normal\_nucleoli +   
## mitoses  
##   
## Df Deviance AIC  
## - se\_cell\_size 1 78.234 96.234  
## - cell\_shape 1 78.902 96.902  
## - cell\_size 1 78.998 96.998  
## <none> 78.196 98.196  
## - bland\_chromatin 1 80.592 98.592  
## - marginal\_adhesion 1 80.912 98.912  
## - mitoses 1 81.008 99.008  
## - normal\_nucleoli 1 81.080 99.080  
## - clump\_thickness 1 91.291 109.291  
## - bare\_nucleoli 1 97.131 115.131  
##   
## Step: AIC=96.23  
## diagnosis ~ clump\_thickness + cell\_size + cell\_shape + marginal\_adhesion +   
## bare\_nucleoli + bland\_chromatin + normal\_nucleoli + mitoses  
##   
## Df Deviance AIC  
## - cell\_shape 1 79.006 95.006  
## - cell\_size 1 79.114 95.114  
## <none> 78.234 96.234  
## - bland\_chromatin 1 80.673 96.673  
## - marginal\_adhesion 1 81.015 97.015  
## - mitoses 1 81.022 97.022  
## - normal\_nucleoli 1 81.114 97.114  
## - clump\_thickness 1 91.348 107.348  
## - bare\_nucleoli 1 97.329 113.329  
##   
## Step: AIC=95.01  
## diagnosis ~ clump\_thickness + cell\_size + marginal\_adhesion +   
## bare\_nucleoli + bland\_chromatin + normal\_nucleoli + mitoses  
##   
## Df Deviance AIC  
## <none> 79.006 95.006  
## - mitoses 1 81.732 95.732  
## - marginal\_adhesion 1 81.767 95.767  
## - bland\_chromatin 1 81.999 95.999  
## - normal\_nucleoli 1 83.040 97.040  
## - cell\_size 1 83.851 97.851  
## - clump\_thickness 1 95.113 109.113  
## - bare\_nucleoli 1 102.809 116.809

##   
## Call: glm(formula = diagnosis ~ clump\_thickness + cell\_size + marginal\_adhesion +   
## bare\_nucleoli + bland\_chromatin + normal\_nucleoli + mitoses,   
## family = "binomial", data = train)  
##   
## Coefficients:  
## (Intercept) clump\_thickness cell\_size marginal\_adhesion   
## -10.1113 0.5930 0.4230 0.2408   
## bare\_nucleoli bland\_chromatin normal\_nucleoli mitoses   
## 0.4915 0.3265 0.2561 0.6154   
##   
## Degrees of Freedom: 545 Total (i.e. Null); 538 Residual  
## Null Deviance: 706.9   
## Residual Deviance: 79.01 AIC: 95.01

# se cell shape, cell shape and cell size don't appear to be helpful in our regression, so I'll remove them and model again  
  
model <- glm(formula = diagnosis ~ clump\_thickness + marginal\_adhesion + bare\_nucleoli + bland\_chromatin + normal\_nucleoli + mitoses, family = "binomial", data = train)  
  
summary(model)

##   
## Call:  
## glm(formula = diagnosis ~ clump\_thickness + marginal\_adhesion +   
## bare\_nucleoli + bland\_chromatin + normal\_nucleoli + mitoses,   
## family = "binomial", data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -3.9688 -0.1250 -0.0523 0.0140 2.1549   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -10.7163 1.3467 -7.958 1.76e-15 \*\*\*  
## clump\_thickness 0.6874 0.1591 4.321 1.55e-05 \*\*\*  
## marginal\_adhesion 0.3608 0.1291 2.795 0.00518 \*\*   
## bare\_nucleoli 0.5237 0.1097 4.773 1.81e-06 \*\*\*  
## bland\_chromatin 0.5011 0.1585 3.161 0.00157 \*\*   
## normal\_nucleoli 0.3374 0.1176 2.870 0.00410 \*\*   
## mitoses 0.7080 0.3143 2.253 0.02429 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 706.888 on 545 degrees of freedom  
## Residual deviance: 83.851 on 539 degrees of freedom  
## AIC: 97.851  
##   
## Number of Fisher Scoring iterations: 8

# predicting on our training data using our simplified model  
  
trainpreds <- predict(model, type = 'response', train)  
  
# setting prediction values. Assuming that '2' means negative and '4' means positive  
  
trainp <- factor(trainpreds >= 0.5, labels = c('benign', 'malignant'))  
  
# Building a confusion matrix to evaluate our model  
  
trainCM <- confusionMatrix(train$diagnosis, trainp)  
trainCM

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction benign malignant  
## benign 348 7  
## malignant 9 182  
##   
## Accuracy : 0.9707   
## 95% CI : (0.9528, 0.9832)  
## No Information Rate : 0.6538   
## P-Value [Acc > NIR] : <2e-16   
##   
## Kappa : 0.9354   
##   
## Mcnemar's Test P-Value : 0.8026   
##   
## Sensitivity : 0.9748   
## Specificity : 0.9630   
## Pos Pred Value : 0.9803   
## Neg Pred Value : 0.9529   
## Prevalence : 0.6538   
## Detection Rate : 0.6374   
## Detection Prevalence : 0.6502   
## Balanced Accuracy : 0.9689   
##   
## 'Positive' Class : benign   
##

## So far our simplified model looks fantastic. Let’s test it against our test data set

# predict on the test data   
testpreds <- predict(model, type = 'response', test)  
  
# set cutoff labels for predictions  
testp <- factor(testpreds >= 0.5, labels = c('benign', 'malignant'))  
  
# Build a confusion matrix to see results  
testCM <- confusionMatrix(test$diagnosis, testp)  
testCM

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction benign malignant  
## benign 86 3  
## malignant 2 46  
##   
## Accuracy : 0.9635   
## 95% CI : (0.9169, 0.988)  
## No Information Rate : 0.6423   
## P-Value [Acc > NIR] : <2e-16   
##   
## Kappa : 0.9202   
##   
## Mcnemar's Test P-Value : 1   
##   
## Sensitivity : 0.9773   
## Specificity : 0.9388   
## Pos Pred Value : 0.9663   
## Neg Pred Value : 0.9583   
## Prevalence : 0.6423   
## Detection Rate : 0.6277   
## Detection Prevalence : 0.6496   
## Balanced Accuracy : 0.9580   
##   
## 'Positive' Class : benign   
##

## Our model looks really good against the test data too, so we’ll create ROC curves for our model on the training set and our model on the test set and we’ll look at confusion matrices for all of our models to evaluate the performance of each

# ROC curve for the Train data  
train\_roc\_curve <- roc(train$diagnosis, trainpreds)

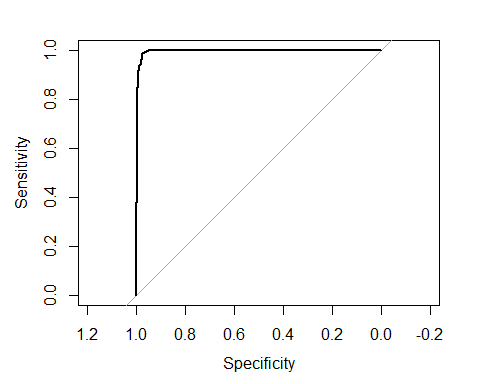
## Setting levels: control = benign, case = malignant

## Setting direction: controls < cases

train\_roc\_curve

##   
## Call:  
## roc.default(response = train$diagnosis, predictor = trainpreds)  
##   
## Data: trainpreds in 355 controls (train$diagnosis benign) < 191 cases (train$diagnosis malignant).  
## Area under the curve: 0.9962

plot(train\_roc\_curve)



train\_rocc <- coords(roc=train\_roc\_curve, x = 'best', best.method = 'closest.topleft')  
train\_rocc

## threshold specificity sensitivity  
## 1 0.2476884 0.9746479 0.9842932

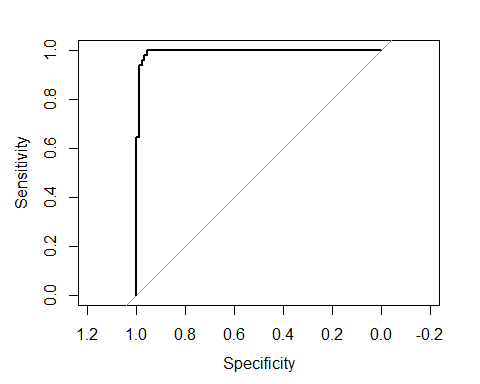
# ROC curve for the Test data  
test\_roc\_curve <- roc(test$diagnosis, testpreds)

## Setting levels: control = benign, case = malignant  
## Setting direction: controls < cases

test\_roc\_curve

##   
## Call:  
## roc.default(response = test$diagnosis, predictor = testpreds)  
##   
## Data: testpreds in 89 controls (test$diagnosis benign) < 48 cases (test$diagnosis malignant).  
## Area under the curve: 0.9946

plot(test\_roc\_curve)



test\_rocc <- coords(roc=test\_roc\_curve, x = 'best', best.method = 'closest.topleft')  
test\_rocc

## threshold specificity sensitivity  
## 1 0.3655996 0.9662921 0.9791667

# predict on training data using ROC cut-off  
trainrocp <- factor(trainpreds >= as.numeric(train\_rocc[1]), labels = c('benign', 'malignant'))  
  
# confusion matrix  
trainROCCM <- confusionMatrix(train$diagnosis, trainrocp)  
trainROCCM

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction benign malignant  
## benign 346 9  
## malignant 3 188  
##   
## Accuracy : 0.978   
## 95% CI : (0.9619, 0.9886)  
## No Information Rate : 0.6392   
## P-Value [Acc > NIR] : <2e-16   
##   
## Kappa : 0.952   
##   
## Mcnemar's Test P-Value : 0.1489   
##   
## Sensitivity : 0.9914   
## Specificity : 0.9543   
## Pos Pred Value : 0.9746   
## Neg Pred Value : 0.9843   
## Prevalence : 0.6392   
## Detection Rate : 0.6337   
## Detection Prevalence : 0.6502   
## Balanced Accuracy : 0.9729   
##   
## 'Positive' Class : benign   
##

# Predict on the test data   
testp <- factor(testpreds >= as.numeric(test\_rocc[1]), labels = c('benign', 'malignant'))  
  
# confusion matrix  
testROCCM <- confusionMatrix(test$diagnosis, testp)  
testROCCM

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction benign malignant  
## benign 86 3  
## malignant 1 47  
##   
## Accuracy : 0.9708   
## 95% CI : (0.9269, 0.992)  
## No Information Rate : 0.635   
## P-Value [Acc > NIR] : <2e-16   
##   
## Kappa : 0.9365   
##   
## Mcnemar's Test P-Value : 0.6171   
##   
## Sensitivity : 0.9885   
## Specificity : 0.9400   
## Pos Pred Value : 0.9663   
## Neg Pred Value : 0.9792   
## Prevalence : 0.6350   
## Detection Rate : 0.6277   
## Detection Prevalence : 0.6496   
## Balanced Accuracy : 0.9643   
##   
## 'Positive' Class : benign   
##

#View all the Confusion matrices  
trainCM

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction benign malignant  
## benign 348 7  
## malignant 9 182  
##   
## Accuracy : 0.9707   
## 95% CI : (0.9528, 0.9832)  
## No Information Rate : 0.6538   
## P-Value [Acc > NIR] : <2e-16   
##   
## Kappa : 0.9354   
##   
## Mcnemar's Test P-Value : 0.8026   
##   
## Sensitivity : 0.9748   
## Specificity : 0.9630   
## Pos Pred Value : 0.9803   
## Neg Pred Value : 0.9529   
## Prevalence : 0.6538   
## Detection Rate : 0.6374   
## Detection Prevalence : 0.6502   
## Balanced Accuracy : 0.9689   
##   
## 'Positive' Class : benign   
##

trainROCCM

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction benign malignant  
## benign 346 9  
## malignant 3 188  
##   
## Accuracy : 0.978   
## 95% CI : (0.9619, 0.9886)  
## No Information Rate : 0.6392   
## P-Value [Acc > NIR] : <2e-16   
##   
## Kappa : 0.952   
##   
## Mcnemar's Test P-Value : 0.1489   
##   
## Sensitivity : 0.9914   
## Specificity : 0.9543   
## Pos Pred Value : 0.9746   
## Neg Pred Value : 0.9843   
## Prevalence : 0.6392   
## Detection Rate : 0.6337   
## Detection Prevalence : 0.6502   
## Balanced Accuracy : 0.9729   
##   
## 'Positive' Class : benign   
##

testCM

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction benign malignant  
## benign 86 3  
## malignant 2 46  
##   
## Accuracy : 0.9635   
## 95% CI : (0.9169, 0.988)  
## No Information Rate : 0.6423   
## P-Value [Acc > NIR] : <2e-16   
##   
## Kappa : 0.9202   
##   
## Mcnemar's Test P-Value : 1   
##   
## Sensitivity : 0.9773   
## Specificity : 0.9388   
## Pos Pred Value : 0.9663   
## Neg Pred Value : 0.9583   
## Prevalence : 0.6423   
## Detection Rate : 0.6277   
## Detection Prevalence : 0.6496   
## Balanced Accuracy : 0.9580   
##   
## 'Positive' Class : benign   
##

testROCCM

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction benign malignant  
## benign 86 3  
## malignant 1 47  
##   
## Accuracy : 0.9708   
## 95% CI : (0.9269, 0.992)  
## No Information Rate : 0.635   
## P-Value [Acc > NIR] : <2e-16   
##   
## Kappa : 0.9365   
##   
## Mcnemar's Test P-Value : 0.6171   
##   
## Sensitivity : 0.9885   
## Specificity : 0.9400   
## Pos Pred Value : 0.9663   
## Neg Pred Value : 0.9792   
## Prevalence : 0.6350   
## Detection Rate : 0.6277   
## Detection Prevalence : 0.6496   
## Balanced Accuracy : 0.9643   
##   
## 'Positive' Class : benign   
##

Note that the echo = FALSE parameter was added to the code chunk to prevent printing of the R code that generated the plot.