Random Forest and ROC curves

Divya Krishnan

December 14, 2015

Random Forest & ROC Curves

```
# Standard libraries
library(RCurl)
library(leaps)
library(car)
library(randomForest)
library(pROC)
library(boot)
library(tree)
library(AER)
```

```
## Warning: package 'sandwich' was built under R version 3.2.3
```

```
library(bestglm)
# Setting seed
set.seed(1)
```

The Wisconsin Breast Cancer dataset is available as a comma-delimited text file on the UCI Machine Learning Repository http://archive.ics.uci.edu/ml. Our goal in this problem will be to predict whether observations (i.e. tumors) are malignant or benign.

(a) Obtain the data, and load it into R by pulling it directly from the web. (Do not download it and import it from a CSV file.) Give a brief description of the data.

The data is about Breast cancer patients who were either diagnosed with benign or malignant cancer. The dataset was created by Dr. William H. Wolberg from the University of Wisconsin Hospitals. The dataset has the following variables - 1. Sample code number
2. Clump Thickness
3. Uniformity of Cell Size
4. Uniformity of Cell Shape
5. Marginal Adhesion
6. Single Epithelial Cell Size
7. Bare Nuclei
8. Bland Chromatin
9. Normal Nucleoli
10. Mitoses
11. Class - Cancer classified as benign(2) or malignant(4)

```
url<-"http://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/breast-cancer-wi
# Reading data from URL
cancer<-read.csv(url,header=FALSE,stringsAsFactors=FALSE)
# Exploring the dataset
str(cancer)
```

```
## 'data.frame':    699 obs. of  11 variables:
##  $ V1 : int  1000025 1002945 1015425 1016277 1017023 1017122 1018099 1018561 1033078 1033078 ...
##  $ V2 : int  5 5 3 6 4 8 1 2 2 4 ...
##  $ V3 : int  1 4 1 8 1 10 1 1 1 2 ...
##  $ V4 : int  1 4 1 8 1 10 1 2 1 1 ...
##  $ V5 : int  1 5 1 1 3 8 1 1 1 1 ...
##  $ V6 : int  2 7 2 3 2 7 2 2 2 2 ...
##  $ V7 : chr  "1" "10" "2" "4" ...
##  $ V8 : int  3 3 3 3 3 9 3 3 1 2 ...
##  $ V9 : int  1 2 1 7 1 7 1 1 1 1 ...
##  $ V10: int  1 1 1 1 1 1 1 1 5 1 ...
##  $ V11: int  2 2 2 2 2 4 2 2 2 2 ...
```

References - http://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/breast-cancer-wisconsin.names

(b) Tidy the data, ensuring that each variable is properly named and cast as the correct data type. Discuss any missing data.

Column nuclei has 16 records of missing data, which has been coded as '?'. The missing values have been omitted from the dataset for analysis.

```
# Renaming the columns appropriately
colnames(cancer)<-c("id","cThickness","cellSize","cellShape","adhesion",
                    "eCellSize","nuclei","chromatin","nucleoli","mitoses","class")
# Reformatting the response variable as factor
cancer$class<-as.factor(cancer$class)
# Summary of the dataset
summary(cancer)
```

```
##        id              cThickness        cellSize        cellShape
##  Min.   :   61634   Min.   : 1.000   Min.   : 1.000   Min.   : 1.000
##  1st Qu.:  870688   1st Qu.: 2.000   1st Qu.: 1.000   1st Qu.: 1.000
##  Median : 1171710   Median : 4.000   Median : 1.000   Median : 1.000
##  Mean   : 1071704   Mean   : 4.418   Mean   : 3.134   Mean   : 3.207
##  3rd Qu.: 1238298   3rd Qu.: 6.000   3rd Qu.: 5.000   3rd Qu.: 5.000
##  Max.   :13454352   Max.   :10.000   Max.   :10.000   Max.   :10.000
##     adhesion        eCellSize         nuclei            chromatin
##  Min.   : 1.000   Min.   : 1.000   Length:699         Min.   : 1.000
##  1st Qu.: 1.000   1st Qu.: 2.000   Class :character   1st Qu.: 2.000
##  Median : 1.000   Median : 2.000   Mode  :character   Median : 3.000
##  Mean   : 2.807   Mean   : 3.216                      Mean   : 3.438
##  3rd Qu.: 4.000   3rd Qu.: 4.000                      3rd Qu.: 5.000
##  Max.   :10.000   Max.   :10.000                      Max.   :10.000
##     nucleoli         mitoses       class
##  Min.   : 1.000   Min.   : 1.000   2:458
##  1st Qu.: 1.000   1st Qu.: 1.000   4:241
##  Median : 1.000   Median : 1.000
##  Mean   : 2.867   Mean   : 1.589
##  3rd Qu.: 4.000   3rd Qu.: 1.000
##  Max.   :10.000   Max.   :10.000
```

```r
# Exploring missing data in nuclei column
table(cancer$nuclei)
```

```
##
##   ?   1  10   2   3   4   5   6   7   8   9
##  16 402 132  30  28  19  30   4   8  21   9
```

```r
# Substituting the missing value with NA
cancer$nuclei<-sub("\\?",NA,as.character(cancer$nuclei))
# Omitting missing data
cancer<-na.omit(cancer)
# Reformatting the nuclei variable
cancer$nuclei<-as.numeric(cancer$nuclei)
```

(c) Split the data into a training and validation set such that a random 70% of the observations are in the training set.

```r
# Sampling the indexes that form the training set
train<-sample(1:nrow(cancer),round(0.7*nrow(cancer),0))
# Exploring Training set
str(cancer[train,])
```

```
## 'data.frame':    478 obs. of  11 variables:
##  $ id         : int  1206841 263538 1223543 1253955 1183240 1183596 1326892 1268804 566509 1102573 ..
##  $ cThickness: int  10 5 1 8 4 3 3 3 5 5 ...
##  $ cellSize   : int  5 10 2 7 1 1 1 1 1 6 ...
##  $ cellShape : int  6 10 1 4 2 3 1 1 1 5 ...
##  $ adhesion   : int  10 6 3 4 1 1 1 1 1 6 ...
##  $ eCellSize : int  6 10 2 5 2 3 2 2 2 10 ...
##  $ nuclei     : num  10 10 1 3 1 4 1 5 1 1 ...
##  $ chromatin : int  7 10 1 5 2 1 2 1 1 3 ...
##  $ nucleoli   : int  7 6 2 10 1 1 1 1 1 1 ...
##  $ mitoses    : int  10 5 1 1 1 1 1 1 1 1 ...
##  $ class      : Factor w/ 2 levels "2","4": 2 2 1 2 1 1 1 1 1 2 ...
##  - attr(*, "na.action")=Class 'omit'  Named int [1:16] 24 41 140 146 159 165 236 250 276 293 ...
##    .. ..- attr(*, "names")= chr [1:16] "24" "41" "140" "146" ...
```

```r
# Exploring Test set
str(cancer[-train,])
```

```
## 'data.frame':    205 obs. of  11 variables:
##  $ id         : int  1002945 1015425 1017023 1017122 1018099 1033078 1043999 1047630 1049815 1054590
##  $ cThickness: int  5 3 4 8 1 4 1 7 4 7 ...
##  $ cellSize   : int  4 1 1 10 1 2 1 4 1 3 ...
##  $ cellShape : int  4 1 1 10 1 1 1 6 1 2 ...
##  $ adhesion   : int  5 1 3 8 1 1 1 4 1 10 ...
##  $ eCellSize : int  7 2 2 7 2 2 2 6 2 5 ...
##  $ nuclei     : num  10 2 1 10 10 1 3 1 1 10 ...
##  $ chromatin : int  3 3 3 9 3 2 3 4 3 5 ...
##  $ nucleoli   : int  2 1 1 7 1 1 1 3 1 4 ...
##  $ mitoses    : int  1 1 1 1 1 1 1 1 1 4 ...
```

```
##  $ class     : Factor w/ 2 levels "2","4": 1 1 1 2 1 1 1 2 1 2 ...
##  - attr(*, "na.action")=Class 'omit'  Named int [1:16] 24 41 140 146 159 165 236 250 276 293 ...
##   .. ..- attr(*, "names")= chr [1:16] "24" "41" "140" "146" ...
```

```
# Creating logical vectors for training set
cancerTrain<-rep(FALSE,nrow(cancer))
cancerTrain[train]<-TRUE
```

(d) Fit a regression model to predict whether tissue samples are malignant or benign. Classify cases in the validation set. Compute and discuss the resulting confusion matrix.

The logistic regression correctly predicted the survival 96.59%. The confusion matrix shows that number of false positives were 4 and false negatives were 3. The false positive rate(Type I error) is about 0.02 and the true positive rate is about 0.96.

```
# Logistic regression
glm.cancer<-glm(class ~ .,data=cancer,
             family=binomial,subset=cancerTrain)
# Summary of the model
summary(glm.cancer)
```

```
##
## Call:
## glm(formula = class ~ ., family = binomial, data = cancer, subset = cancerTrain)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.7183  -0.0937  -0.0367   0.0107   1.7657
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.198e+01  2.069e+00  -5.789 7.06e-09 ***
## id           3.557e-07  3.609e-07   0.985 0.324415
## cThickness   6.337e-01  2.211e-01   2.866 0.004159 **
## cellSize     2.801e-01  3.331e-01   0.841 0.400397
## cellShape    6.540e-02  3.922e-01   0.167 0.867561
## adhesion     2.939e-01  1.587e-01   1.852 0.064046 .
## eCellSize    2.017e-01  2.031e-01   0.993 0.320530
## nuclei       5.522e-01  1.537e-01   3.592 0.000328 ***
## chromatin    4.090e-01  2.237e-01   1.829 0.067411 .
## nucleoli     1.898e-01  1.341e-01   1.415 0.156964
## mitoses      7.405e-01  3.834e-01   1.931 0.053463 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 623.414  on 477  degrees of freedom
## Residual deviance:  58.039  on 467  degrees of freedom
## AIC: 80.039
##
## Number of Fisher Scoring iterations: 9
```

```
# Coefficient estimates
glm.cancer$coefficients
```

```
##   (Intercept)            id    cThickness      cellSize     cellShape
## -1.197620e+01  3.556576e-07  6.337255e-01  2.800668e-01  6.539793e-02
##      adhesion     eCellSize        nuclei     chromatin      nucleoli
##  2.939442e-01  2.017179e-01  5.522176e-01  4.090496e-01  1.898227e-01
##       mitoses
##  7.405047e-01
```

```
# Predicting the survival for test set
yhat<-predict(glm.cancer,cancer[!cancerTrain,],type="response")
# Exploring predicted values
str(yhat)
```

```
##  Named num [1:205] 0.97545 0.00446 0.00867 0.99999 0.09467 ...
##  - attr(*, "names")= chr [1:205] "2" "3" "5" "6" ...
```

```
# Actual survival values in the test set
classTest<-cancer$class[!cancerTrain]
glm.pred<-rep(2,nrow(cancer[!cancerTrain,]))
# Predicting survival based on threshold probability of 0.5
glm.pred[yhat>0.5]<-4

# Looking at error in prediction
table(glm.pred,classTest)
```

```
##         classTest
## glm.pred   2   4
##        2 133   4
##        4   4  64
```

```
# Prediction accuracy
round(mean(glm.pred==classTest)*100,2)
```

```
## [1] 96.1
```

```
# False positives (Type I error)
falsePos<-table(glm.pred,classTest)[2,1]
falsePos
```

```
## [1] 4
```

```
# False negatives (Type II error)
falseNeg<-table(glm.pred,classTest)[1,2]
falseNeg
```

```
## [1] 4
```

```r
# False positive rate (Type I error)
falsePos/sum(table(glm.pred,classTest))
```

```
## [1] 0.0195122
```

```r
# True positive
truePos<-table(glm.pred,classTest)[2,2]
truePos
```

```
## [1] 64
```

```r
# True positive rate (Power)
truePos/(truePos+falseNeg)
```

```
## [1] 0.9411765
```

(e) Fit a random forest model to predict whether tissue samples are malignant or benign. Classify cases in the validation set. Compute and discuss the resulting confusion matrix.

The confusion matrix shows that number of false positives were 6 and false negatives were 8. The false positive rate(Type I error) is about 0.013 and the true positive rate is about 0.952.

```r
# Random forest function
rfcancer<-randomForest(class ~ .,data=cancer,subset=train)
# Model
rfcancer
```

```
##
## Call:
##  randomForest(formula = class ~ ., data = cancer, subset = train)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 3
##
##         OOB estimate of  error rate: 2.09%
## Confusion matrix:
##     2   4 class.error
## 2 300   7  0.02280130
## 4   3 168  0.01754386
```

```r
# Importance of each predictor
rfcancer$importance
```

```
##             MeanDecreaseGini
## id                  3.471653
## cThickness          8.885145
## cellSize           57.807003
## cellShape          49.254616
## adhesion            5.189877
## eCellSize          19.627093
```

```
## nuclei            35.941832
## chromatin         16.650322
## nucleoli          21.147082
## mitoses            1.401263
```

```
# Predicted values based on random forest model
rfyhat<-predict(rfcancer,newdata=cancer[-train,])
# Confusion Matrix
rfcancer$confusion
```

```
##     2   4 class.error
## 2 300   7  0.02280130
## 4   3 168  0.01754386
```

```
# False positives (Type I error)
falsePos<-rfcancer$confusion[2,1]
falsePos
```

```
## [1] 3
```

```
# False negatives (Type II error)
falseNeg<-rfcancer$confusion[1,2]
falseNeg
```

```
## [1] 7
```

```
# False positive rate (Type I error)
falsePos/sum(rfcancer$confusion[1:2,1:2])
```

```
## [1] 0.006276151
```

```
# True positive
truePos<-rfcancer$confusion[2,2]
truePos
```

```
## [1] 168
```

```
# True positive rate (Power)
truePos/(truePos+falseNeg)
```

```
## [1] 0.96
```

(f) Compare the models from part (d) and (e) using ROC curves. Which do you prefer? Be sure to justify your preference.

The ROC curve of the regression model performs slightly better than the random forest model. The AUC for the regression model is about 0.9945 whereas AUC for random forest is 0.9741. Hence, we prefer the regression model.

```r
# Confusion matrix for test set for Logistic regression model
table(glm.pred,classTest)
```

```
##         classTest
## glm.pred   2   4
##        2 133   4
##        4   4  64
```

```r
# Confusion matrix for test set for Random forest model
rfcancer$confusion[1:2,1:2]
```

```
##     2   4
## 2 300   7
## 4   3 168
```

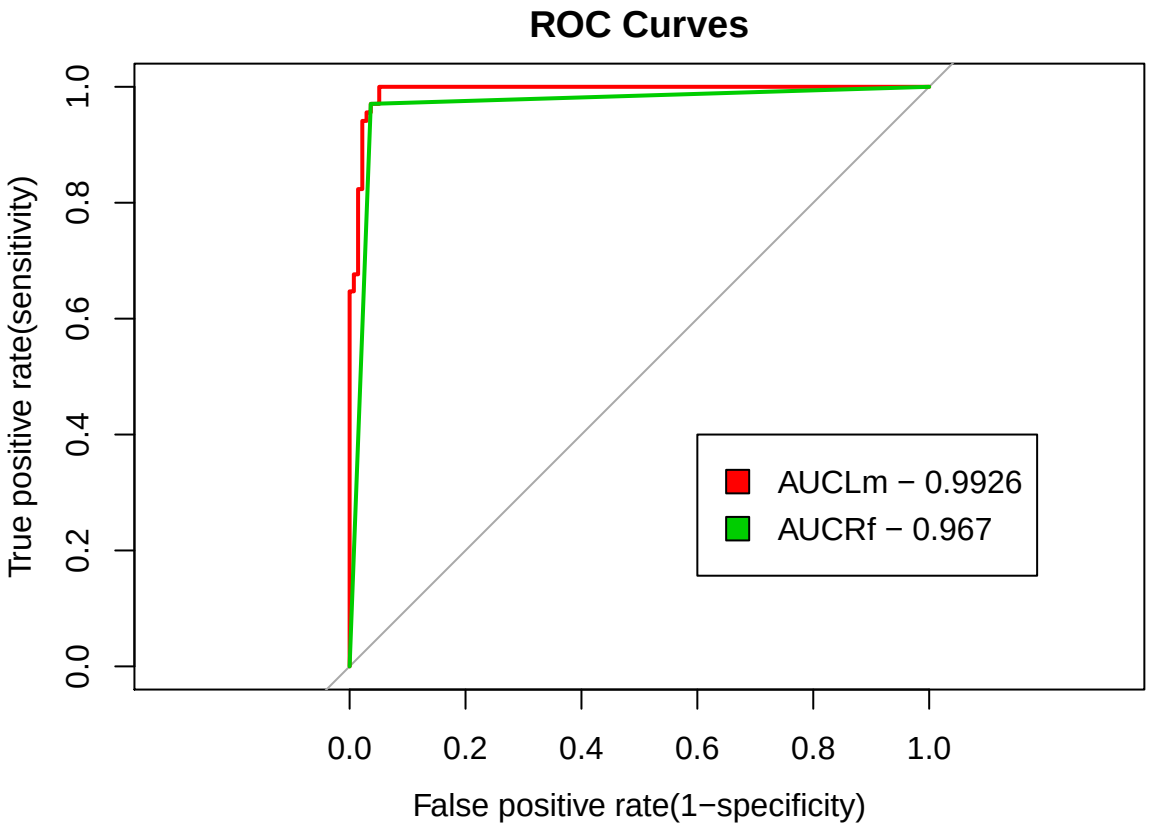
```r
# ROC for Model 3
rocLm<-roc(as.numeric(classTest),as.numeric(yhat))
# ROC for Model 4
roc4Rf<-roc(as.numeric(classTest),as.numeric(rfyhat))

# AUC for Model 3
aucLm<-round(rocLm$auc,4)
# AUC for Model 4
aucRf<-round(roc4Rf$auc,4)
# ROC curve displaying Model 3 & 4
plot.roc(rocLm,main="ROC Curves",col=2,legacy.axes=TRUE,
         xlab="False positive rate(1-specificity)",ylab="True positive rate(sensitivity)")
```

```
##
## Call:
## roc.default(response = as.numeric(classTest), predictor = as.numeric(yhat))
##
## Data: as.numeric(yhat) in 137 controls (as.numeric(classTest) 1) < 68 cases (as.numeric(classTest) 2)
## Area under the curve: 0.9926
```

```r
plot.roc(roc4Rf,add=TRUE,col=3)
```

```
##
## Call:
## roc.default(response = as.numeric(classTest), predictor = as.numeric(rfyhat))
##
## Data: as.numeric(rfyhat) in 137 controls (as.numeric(classTest) 1) < 68 cases (as.numeric(classTest)
## Area under the curve: 0.967
```

```r
legend(0.4,0.4,c(paste0("AUCLm - ",aucLm),paste0("AUCRf - ",aucRf)),2:3)
```


ROC Curves
True positive rate(sensitivity)
0.0
0.2
0.4
0.6
0.8
1.0
False positive rate(1−specificity)
AUCLm − 0.9926
AUCRf − 0.967