Random Forest

Divya Krishnan

Monday, December 7, 2015

Random Forest

```
# Stardard libraries
library(pROC)
library(randomForest)
```

Importing Titanic dataset and splitting the dataset into train and test sets.

```
# Importing data
titanic<-read.csv("titanic.csv",stringsAsFactors = TRUE)

# Setting seed
set.seed(1)

# Sampling the indexes that form the training set
train<-sample(1:nrow(titanic),round(0.8*nrow(titanic),0))
# Exploring Training set
str(titanic[train,])
```

```
## 'data.frame':    1047 obs. of  14 variables:
##  $ pclass   : int  2 2 3 3 1 3 3 3 3 1 ...
##  $ survived : int  0 0 0 0 1 0 0 1 1 0 ...
##  $ name     : Factor w/ 1307 levels "Abbing, Mr. Anthony",..: 147 712 297 1073 1098 1056 1161 531 45
##  $ sex      : Factor w/ 2 levels "female","male": 2 2 2 2 1 2 1 1 1 2 ...
##  $ age      : num  42 24 34 NA 39 14.5 2 26 16 NA ...
##  $ sibsp    : int  0 0 1 2 1 8 0 0 0 0 ...
##  $ parch    : int  0 0 1 0 0 2 1 0 0 0 ...
##  $ ticket   : Factor w/ 929 levels "110152","110413",..: 130 757 458 257 94 779 437 915 576 58 ...
##  $ fare     : num  13 10.5 14.4 21.7 55.9 ...
##  $ cabin    : Factor w/ 187 levels "","A10","A11",..: 1 1 1 1 163 1 186 1 1 1 ...
##  $ embarked : Factor w/ 4 levels "","C","Q","S": 4 4 4 2 4 4 4 4 3 4 ...
##  $ boat     : Factor w/ 28 levels "","1","10","11",..: 1 1 1 1 4 1 1 1 12 1 ...
##  $ body     : int  NA 108 197 NA NA 67 NA NA NA NA ...
##  $ home.dest: Factor w/ 370 levels "","?Havana, Cuba",..: 198 1 311 1 100 1 1 1 72 276 ...
```

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

```
## 'data.frame':    262 obs. of  14 variables:
##  $ pclass   : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ survived : int  1 1 1 1 1 1 0 0 1 0 ...
##  $ name     : Factor w/ 1307 levels "Abbing, Mr. Anthony",..: 73 93 94 112 132 133 135 157 194 203 .
##  $ sex      : Factor w/ 2 levels "female","male": 1 1 2 1 2 1 2 2 1 2 ...
##  $ age      : num  24 26 80 47 40 30 42 NA 53 33 ...
```

```
##  $ sibsp    : int  0 0 0 1 0 0 0 0 0 0 ...
##  $ parch    : int  0 0 0 1 0 0 0 0 0 0 ...
##  $ ticket   : Factor w/ 929 levels "110152","110413",..: 796 119 297 71 22 627 5 25 823 689 ...
##  $ fare     : num  69.3 78.8 30 52.6 31 ...
##  $ cabin    : Factor w/ 187 levels "","A10","A11",..: 35 1 10 133 14 102 127 1 1 47 ...
##  $ embarked : Factor w/ 4 levels "","C","Q","S": 2 4 4 4 2 4 4 2 2 4 ...
##  $ boat     : Factor w/ 28 levels "","1","10","11",..: 23 19 25 16 20 21 1 1 19 1 ...
##  $ body     : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ home.dest: Factor w/ 370 levels "","?Havana, Cuba",..: 259 1 159 238 132 369 209 265 350 238 ...
```

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

Suppose we use the data to construct a new predictor variable based on a passenger's listed title (i.e. Mr., Mrs., Miss., Master).

Using custom function to add the predictor to your dataset.

```
# A function to construct a feature that looks at passenger titles
f <- function(name) {
for (title in c("Master", "Miss", "Mrs.", "Mr.")) {
if (grepl(title, name)) {
return(title)
}
}
return("Nothing")
}

tempTitle = vector()
# Extracting title for each passenger
for(i in 1:(nrow(titanic)))
{
  tempTitle<-c(tempTitle,f(titanic$name[i]))
}
# Adding the title variable to titanic dataset
titanic$title<-tempTitle
```

Random Forest Classification Another very popular classifier used in data science is called a random forest.

(a) Use the randomForest function to fit a random forest model with passenger class and title as predictors. Make predictions for the test set using the random forest model. Save these predictions as yhat3.

```
# Converting title and pclass as factors
titanic$title<-as.factor(titanic$title)
titanic$pclass<-as.factor(titanic$pclass)
# Random forest model on trianing set
rfTitanic<-randomForest(survived ~ pclass + title,data=titanic,subset=train)
```

```
## Warning in randomForest.default(m, y, ...): The response has five or fewer
## unique values. Are you sure you want to do regression?
```

```
# Model
rfTitanic
```

```
##
## Call:
##  randomForest(formula = survived ~ pclass + title, data = titanic,      subset = train)
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 1
##
##           Mean of squared residuals: 0.1460325
##                     % Var explained: 38.03
```

```
# Importance of each predictor
rfTitanic$importance
```

```
##        IncNodePurity
## pclass      23.40264
## title       60.77914
```

```
# Predicted values based on random forest model
yhat3<-predict(rfTitanic,newdata=titanic[-train,])
```

(b) Develop your own random forest model, attempting to improve the model performance. Make predictions for the test set using your new random forest model. Save these predictions as yhat4.

```
# New random forest model incldуing Sex
rfTitanicSex<-randomForest(survived ~ pclass + title + sex,data=titanic,subset=train)
```

```
## Warning in randomForest.default(m, y, ...): The response has five or fewer
## unique values. Are you sure you want to do regression?
```

```
# Model
rfTitanicSex
```

```
##
## Call:
##  randomForest(formula = survived ~ pclass + title + sex, data = titanic,      subset = train)
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 1
##
##           Mean of squared residuals: 0.1473489
##                     % Var explained: 37.47
```

```
# Importance of each predictor
rfTitanicSex$importance
```

```
##        IncNodePurity
## pclass      16.89456
## title       37.03776
## sex         30.46830
```

```r
# Predicted values based on new random forest model
yhat4<-predict(rfTitanicSex,newdata=titanic[-train,])
```

(c) Compare the accuracy of each of the models from this problem set using ROC curves. Comment on which statistical learning method works best for predicting survival of the titanic passengers.

Looking at the ROC curves for both models, the model with pclass, title and sex as predictors (Model 4) performs better than the model with only pclass and title as predictor (Model 3). The AUC value for Model 4 is 0.77 whereas for Model 3 is 0.73. Hence Model 4 should be considered for predicting survival of titanic passengers.

```r
# Confusion matrix for test set for Model 3
table(yhat3,titanic[-train,]$survived)
```

```
##
## yhat3                0  1
##   0.163432661542604  2  1
##   0.16497660730111  23  2
##   0.167659865292938 82 11
##   0.340808144614493 19 11
##   0.360664704126479  8  3
##   0.440637257626849 13 13
##   0.457532815982571  6  9
##   0.546746072486315  4  4
##   0.801270939489689  2  7
##   0.841866151998016  1  8
##   0.844150242202728  0  1
##   0.862750539556711  0  2
##   0.865540445582099  0  9
##   0.875165708216851  0 21
```

```r
# Confusion matrix for test set for Model 4
table(yhat4,titanic[-train,]$survived)
```

```
##
## yhat4                0  1
##   0.159399846176182 82 11
##   0.162226015865973 23  2
##   0.175407538877119  2  0
##   0.279715763115094 19 11
##   0.341649466293579  8  3
##   0.414775209428898  4  1
##   0.526846820305996 13 13
##   0.555812913103855  6  9
##   0.562749568792698  0  1
##   0.610777259663362  0  1
##   0.66831745883369   0  2
##   0.765936208769774  0  3
##   0.811867496248972  2  7
##   0.829069108294939  1  8
##   0.854743110514836  0  9
##   0.86314072793212   0 21
```

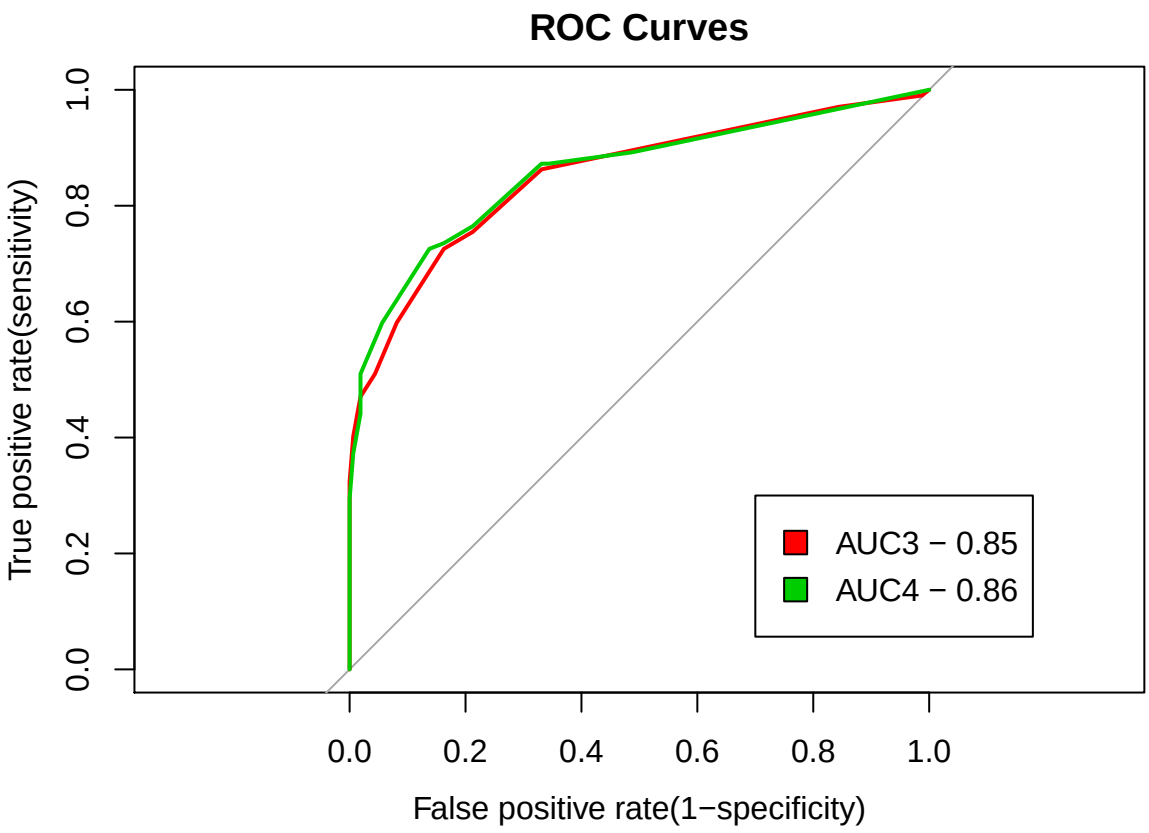
```r
# ROC for Model 3
roc3<-roc(as.numeric(titanic[-train,]$survived),as.numeric(yhat3))
# ROC for Model 4
roc4<-roc(as.numeric(titanic[-train,]$survived),as.numeric(yhat4))

# AUC for Model 3
auc3<-round(roc3$auc,2)
# AUC for Model 4
auc4<-round(roc4$auc,2)
# ROC curve displaying Model 3 & 4
plot.roc(roc3,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(titanic[-train, ]$survived),     predictor = as.numeric(yhat3))
##
## Data: as.numeric(yhat3) in 160 controls (as.numeric(titanic[-train, ]$survived) 0) < 102 cases (as.nu
## Area under the curve: 0.8507
```

```r
plot.roc(roc4,add=TRUE,col=3)
```

```
##
## Call:
## roc.default(response = as.numeric(titanic[-train, ]$survived),     predictor = as.numeric(yhat4))
##
## Data: as.numeric(yhat4) in 160 controls (as.numeric(titanic[-train, ]$survived) 0) < 102 cases (as.nu
## Area under the curve: 0.8573
```

```r
legend(0.3,0.3,c(paste0("AUC3 - ",auc3),paste0("AUC4 - ",auc4)),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)
AUC3 − 0.85
AUC4 − 0.86