Logistic Regression and Prediction

Divya Krishnan

Monday, December 7, 2015

Logistic Regression and Prediction

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

Train-Test split As part of this assignment we will evaluate the performance of a few different statistical learning methods. We will fit a particular statistical learning method on a set of training observations and measure its performance on a set of test observations.

(a) Discuss the advantages of using a training/test split when evaluating statistical models.

Splitting the dataset into training and test is important to evaluate models. When we use the training/test split, we can train the model on training set and see it's performance on the test set. This would give us the confusion matrix, which looks at predicted values versus actual values. The confusion matrix gives us idea about Type-I and Type-II error.

(b) Split your data into a training and test set based on an 80-20 split, in other words, 80% of the observations will be in the training set.

```r
# Importing data
titanic<-read.csv("titanic.csv",stringsAsFactors = TRUE)
# Exploring the dataset
str(titanic)
```

```
## 'data.frame':    1309 obs. of  14 variables:
##  $ pclass   : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ survived : int  1 1 0 0 0 1 1 0 1 0 ...
##  $ name     : Factor w/ 1307 levels "Abbing, Mr. Anthony",..: 22 24 25 26 27 31 46 47 51 55 ...
##  $ sex      : Factor w/ 2 levels "female","male": 1 2 1 2 1 2 1 2 1 2 ...
##  $ age      : num  29 0.917 2 30 25 ...
##  $ sibsp    : int  0 1 1 1 1 0 1 0 2 0 ...
##  $ parch    : int  0 2 2 2 2 0 0 0 0 0 ...
##  $ ticket   : Factor w/ 929 levels "110152","110413",..: 188 50 50 50 50 125 93 16 77 826 ...
##  $ fare     : num  211 152 152 152 152 ...
##  $ cabin    : Factor w/ 187 levels "","A10","A11",..: 45 81 81 81 81 151 147 17 63 1 ...
##  $ embarked : Factor w/ 4 levels "","C","Q","S": 4 4 4 4 4 4 4 4 4 2 ...
##  $ boat     : Factor w/ 28 levels "","1","10","11",..: 13 4 1 1 1 14 3 1 28 1 ...
##  $ body     : int  NA NA NA 135 NA NA NA NA NA 22 ...
##  $ home.dest: Factor w/ 370 levels "","?Havana, Cuba",..: 310 232 232 232 232 238 163 25 23 230 ...
```

```r
# 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 531 452
##  $ 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 ...
```

```r
# 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 ...
```

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

Logistic Regression In this problem set our goal is to predict the survival of passengers. First consider training a logistic regression model for survival that controls for the socioeconomic status of the passenger.

(a) Fit the model described above using the glm function in R.

Since pclass variable gives the passenger class, it can be used as the socioeconomic status of the passenger. Hence the logistic regression was done using pclass as the predictor.

```
# Converting survived variable into factor
titanic$survived<-as.factor(titanic$survived)
# Logistic regression
modPclass<-glm(survived ~ pclass,data=titanic,
             family=binomial,subset=titanicTrain)
# Summary of the model
summary(modPclass)
```

```
##
## Call:
## glm(formula = survived ~ pclass, family = binomial, data = titanic,
##     subset = titanicTrain)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.3710  -0.7745  -0.7745   0.9955   1.6434
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   1.1917     0.1882   6.332 2.43e-10 ***
## pclass       -0.7474     0.0795  -9.402  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1390.7  on 1046  degrees of freedom
## Residual deviance: 1297.6  on 1045  degrees of freedom
## AIC: 1301.6
##
## Number of Fisher Scoring iterations: 4
```

(b) What might you conclude based on this model about the probability of survival for lower class passengers? Note: If your model looks unstable you might consider using the bayesglm function from the arm package.

The 1st, 2nd and 3rd class of passengers is assigned as 1,2 and 3 respectively. The logistic regression suggests that as we move from one class to next(in the order of 1,2,3), the log odds of survival changes by -0.75. This conclusion is statistically significant as p-value is very close to zero. So the probability of survival is highest for 1st class passengers, lower for 2nd class passengers and the lowest for 3rd class passengers.

```
# Coefficient estimates
modPclass$coefficients
```

```
## (Intercept)      pclass
##   1.1916919  -0.7473992
```

```
# p-value of beta1 coefficient
summary(modPclass)$coefficients[2,4]
```

```
## [1] 5.376761e-21
```

Model Performance Next, let's consider the performance of this model.

(a) Predict the survival of passengers for each observation in your test set using the model fit in Problem 2. Save these predictions as yhat.

```
# Predicting the survival for test set
yhat<-predict(modPclass,titanic[!titanicTrain,],type="response")
# Exploring predicted values
str(yhat)
```

```
##  Named num [1:262] 0.609 0.609 0.609 0.609 0.609 ...
##  - attr(*, "names")= chr [1:262] "13" "14" "15" "22" ...
```

(b) Use a threshold of 0.5 to classify predictions. What is the number of false positives on the test data? Interpret this in your own words.

The number of false positives on the test data is 23. This means that the logistic regression model predicted 23 cases of survival when they had actually not survived. 23 passengers who did not survive were wrongly classified by the model as survived.

```
# Actual survival values in the test set
survivedTest<-titanic$survived[!titanicTrain]

pred<-rep(0,nrow(titanic[!titanicTrain,]))
# Predicting survival based on threshold probability of 0.5
pred[yhat>0.5]<-1

# Looking at error in prediction
table(pred,survivedTest)
```

```
##     survivedTest
## pred   0   1
##    0 137  55
##    1  23  47
```

```
# False positives
table(pred,survivedTest)[2,1]
```

```
## [1] 23
```

(c) Using the roc function, plot the ROC curve for this model. Discuss what you find.

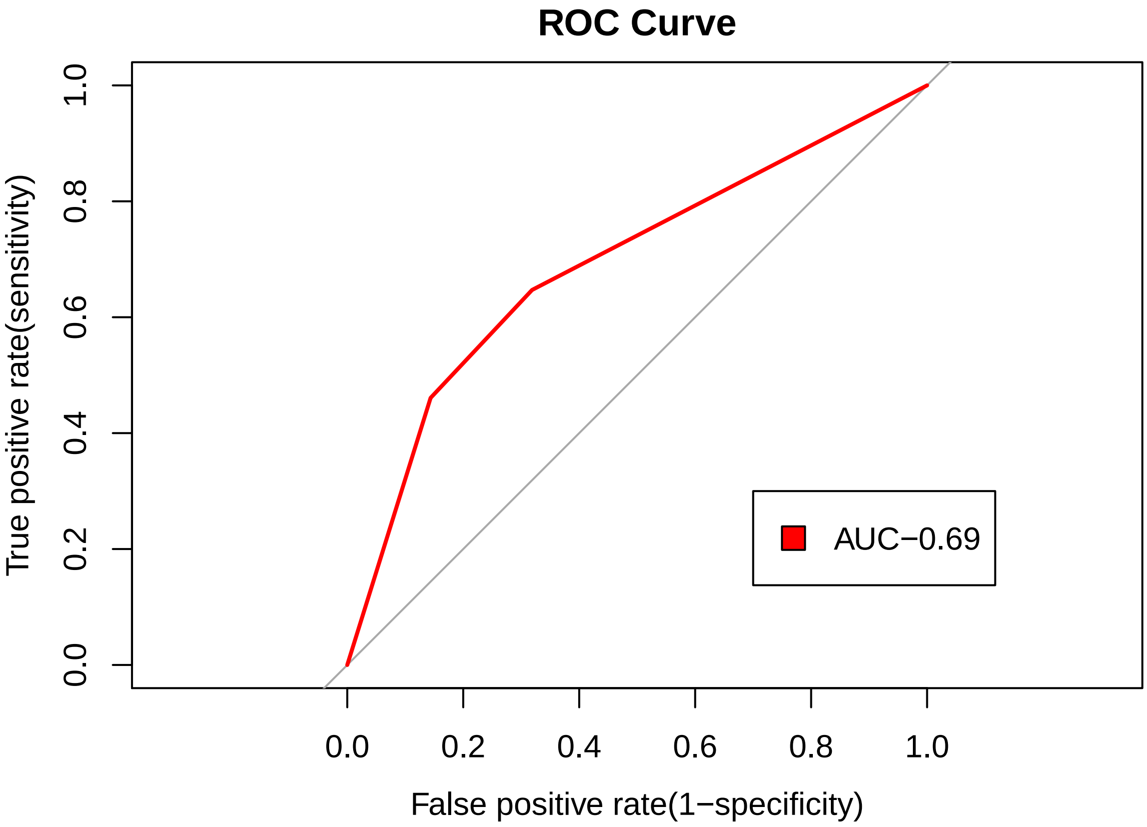
An ideal ROC curve will hug the top-left corner of the plot. The ROC curve of the model shows that it is not an ideal and the AUC is 0.69. Hence there is scope to improve the model.

```
# ROC curve
roc1<-roc(survivedTest,yhat)

# AUC for the model
auc1<-round(roc1$auc,2)
plot.roc(roc1,main="ROC Curve",col=2,legacy.axes=TRUE,
         xlab="False positive rate(1-specificity)",ylab="True positive rate(sensitivity)")
```

```
##
## Call:
## roc.default(response = survivedTest, predictor = yhat)
##
## Data: yhat in 160 controls (survivedTest 0) < 102 cases (survivedTest 1).
## Area under the curve: 0.6911
```

```
legend(0.3,0.3,c(paste0("AUC-",auc1)),2)
```



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

(a) Why might this be an interesting variable to help predict passenger survival?

When a disaster occurs, usually the children and the ladies are rescued first, before rescuing the men. Hence it would be intersting to see whether more children and ladies survived compared to men. The title variable would help us in answering this question.

(b) Use the following custom function to add this 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
```

(c) Fit a second logistic regression model including this new feature. Use the summary function to look at the model. Did this new feature improve the model?

The summary of the new model suggest that the strongest association is between survival of passengers and the title “Mr.” (-2.25 approximately). The model also shows that there is a statistically significant association only between survived and pclass, ‘Mr.’ title and ‘Nothing’ title(or missing titles).

Comparing the AIC of the 2 models, shows that the new model (AIC=991 approximately) using pclass and title is better than the old model (AIC=1301 approximately) using only pclass as the predictor variable.

```
# Logistic regression, including title
glmTitanic<-glm(survived ~ pclass + title,data=titanic,
             family=binomial,subset=titanicTrain)
# Summary of the model
summary(glmTitanic)
```

```
##
## Call:
## glm(formula = survived ~ pclass + title, family = binomial, data = titanic,
##     subset = titanicTrain)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1900  -0.6405  -0.4156   0.6709   2.2325
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    2.6175     0.4105   6.376 1.82e-10 ***
## pclass        -0.9259     0.1020  -9.075  < 2e-16 ***
## titleMiss      0.3989     0.3381   1.180  0.23809
## titleMr.      -2.2455     0.3320  -6.763 1.35e-11 ***
## titleMrs.      0.6111     0.3666   1.667  0.09549 .
## titleNothing  -1.8974     0.5851  -3.243  0.00118 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1390.69  on 1046  degrees of freedom
## Residual deviance:  978.77  on 1041  degrees of freedom
## AIC: 990.77
##
## Number of Fisher Scoring iterations: 4
```

```
# AIC of Model 1
modPclass$aic
```

```
## [1] 1301.551
```

```
# AIC of Model 2
glmTitanic$aic
```

```
## [1] 990.7688
```

(d) Comment on the overall fit of this model. For example, you might consider exploring when misclassification occurs.

The logistic regression correctly predicted the survival 79.39%. The confusion matrix shows that number of false positives were 22 and false negatives were 32. The false positive rate(Type I error) is about 0.084 and the true positive rate is about 0.686.

```
# Predicting the probability of survival in test set
glm.prob<-predict(glmTitanic,titanic[!titanicTrain,],type="response")
glm.pred<-rep(0,nrow(titanic[!titanicTrain,]))
# Assigning survival as 1 for threshold probability of 0.5
glm.pred[glm.prob>0.5]<-1
# Misclassification displayed as confusion matrix
table(glm.pred,survivedTest)
```

```
##         survivedTest
## glm.pred   0   1
##        0 138  32
##        1  22  70
```

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

```
## [1] 79.39
```

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

```
## [1] 22
```

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

```
## [1] 32
```

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

```
## [1] 0.08396947
```

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

```
## [1] 70
```

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

```
## [1] 0.6862745
```

(e) Predict the survival of passengers for each observation in your test data using the new model. Save these predictions as yhat2.

```r
# Predicted values based on the second model
yhat2<-predict(glmTitanic,newdata=data.frame(titanic[!titanicTrain,]),type="response")
```