Logistic Regression on Titanic Dataset

Abhay Padda

29 October 2017

## R Markdown

## Load Titanic library to get the dataset  
library(titanic)  
  
## Load the datasets  
data("titanic\_train")  
data("titanic\_test")  
  
## Setting Survived column for test data to NA  
titanic\_test$Survived <- NA  
  
## Combining Training and Testing dataset  
complete\_data <- rbind(titanic\_train, titanic\_test)  
  
## Check data structure  
str(complete\_data)

## 'data.frame': 1309 obs. of 12 variables:  
## $ PassengerId: int 1 2 3 4 5 6 7 8 9 10 ...  
## $ Survived : int 0 1 1 1 0 0 0 0 1 1 ...  
## $ Pclass : int 3 1 3 1 3 3 1 3 3 2 ...  
## $ Name : chr "Braund, Mr. Owen Harris" "Cumings, Mrs. John Bradley (Florence Briggs Thayer)" "Heikkinen, Miss. Laina" "Futrelle, Mrs. Jacques Heath (Lily May Peel)" ...  
## $ Sex : chr "male" "female" "female" "female" ...  
## $ Age : num 22 38 26 35 35 NA 54 2 27 14 ...  
## $ SibSp : int 1 1 0 1 0 0 0 3 0 1 ...  
## $ Parch : int 0 0 0 0 0 0 0 1 2 0 ...  
## $ Ticket : chr "A/5 21171" "PC 17599" "STON/O2. 3101282" "113803" ...  
## $ Fare : num 7.25 71.28 7.92 53.1 8.05 ...  
## $ Cabin : chr "" "C85" "" "C123" ...  
## $ Embarked : chr "S" "C" "S" "S" ...

## Let's check for any missing values in the data  
colSums(is.na(complete\_data))

## PassengerId Survived Pclass Name Sex Age   
## 0 418 0 0 0 263   
## SibSp Parch Ticket Fare Cabin Embarked   
## 0 0 0 1 0 0

## Checking for empty values  
colSums(complete\_data=='')

## PassengerId Survived Pclass Name Sex Age   
## 0 NA 0 0 0 NA   
## SibSp Parch Ticket Fare Cabin Embarked   
## 0 0 0 NA 1014 2

## Check number of uniques values for each of the column to find out columns which we can convert to factors  
sapply(complete\_data, function(x) length(unique(x)))

## PassengerId Survived Pclass Name Sex Age   
## 1309 3 3 1307 2 99   
## SibSp Parch Ticket Fare Cabin Embarked   
## 7 8 929 282 187 4

## Missing values imputation  
complete\_data$Embarked[complete\_data$Embarked==""] <- "S"  
complete\_data$Age[is.na(complete\_data$Age)] <- median(complete\_data$Age,na.rm=T)  
  
## Removing Cabin as it has very high missing values, passengerId, Ticket and Name are not required  
library(dplyr)  
titanic\_data <- complete\_data %>% select(-c(Cabin, PassengerId, Ticket, Name))  
  
## Converting "Survived","Pclass","Sex","Embarked" to factors  
for (i in c("Survived","Pclass","Sex","Embarked")){  
 titanic\_data[,i]=as.factor(titanic\_data[,i])  
}  
  
## Create dummy variables for categorical variables  
library(dummies)  
titanic\_data <- dummy.data.frame(titanic\_data, names=c("Pclass","Sex","Embarked"), sep="\_")

## Splitting training and test data  
train <- titanic\_data[1:667,]  
test <- titanic\_data[668:889,]  
  
## Model Creation  
model <- glm(Survived ~.,family=binomial(link='logit'),data=train)  
  
## Model Summary  
summary(model)

##   
## Call:  
## glm(formula = Survived ~ ., family = binomial(link = "logit"),   
## data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.3804 -0.6562 -0.4300 0.6392 2.3950   
##   
## Coefficients: (3 not defined because of singularities)  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -1.373105 0.319779 -4.294 1.76e-05 \*\*\*  
## Pclass\_1 2.175104 0.359365 6.053 1.42e-09 \*\*\*  
## Pclass\_2 1.302268 0.271680 4.793 1.64e-06 \*\*\*  
## Pclass\_3 NA NA NA NA   
## Sex\_female 2.677814 0.226863 11.804 < 2e-16 \*\*\*  
## Sex\_male NA NA NA NA   
## Age -0.031671 0.008945 -3.540 0.000399 \*\*\*  
## SibSp -0.248975 0.123365 -2.018 0.043570 \*   
## Parch -0.091603 0.141950 -0.645 0.518718   
## Fare -0.001397 0.003179 -0.440 0.660254   
## Embarked\_C 0.431447 0.271693 1.588 0.112288   
## Embarked\_Q 0.533193 0.369337 1.444 0.148837   
## Embarked\_S NA NA NA NA   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 891.99 on 666 degrees of freedom  
## Residual deviance: 605.78 on 657 degrees of freedom  
## AIC: 625.78  
##   
## Number of Fisher Scoring iterations: 5

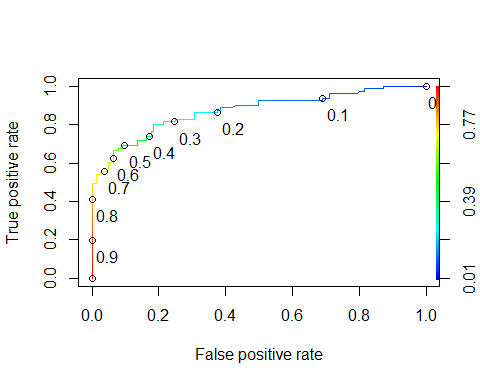
## Using anova() to analyze the table of devaiance  
anova(model, test="Chisq")

## Analysis of Deviance Table  
##   
## Model: binomial, link: logit  
##   
## Response: Survived  
##   
## Terms added sequentially (first to last)  
##   
##   
## Df Deviance Resid. Df Resid. Dev Pr(>Chi)   
## NULL 666 891.99   
## Pclass\_1 1 39.603 665 852.39 3.112e-10 \*\*\*  
## Pclass\_2 1 26.485 664 825.91 2.655e-07 \*\*\*  
## Pclass\_3 0 0.000 664 825.91   
## Sex\_female 1 197.978 663 627.93 < 2.2e-16 \*\*\*  
## Sex\_male 0 0.000 663 627.93   
## Age 1 8.986 662 618.94 0.002721 \*\*   
## SibSp 1 8.114 661 610.83 0.004393 \*\*   
## Parch 1 0.998 660 609.83 0.317889   
## Fare 1 0.044 659 609.79 0.834588   
## Embarked\_C 1 1.936 658 607.85 0.164139   
## Embarked\_Q 1 2.067 657 605.78 0.150485   
## Embarked\_S 0 0.000 657 605.78   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

## Predicting Test Data  
result <- predict(model,newdata=test,type='response')  
result <- ifelse(result > 0.5,1,0)  
  
## Confusion matrix and statistics  
library(caret)  
confusionMatrix(data=result, reference=test$Survived)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 128 25  
## 1 13 56  
##   
## Accuracy : 0.8288   
## 95% CI : (0.7727, 0.8759)  
## No Information Rate : 0.6351   
## P-Value [Acc > NIR] : 1.817e-10   
##   
## Kappa : 0.6187   
## Mcnemar's Test P-Value : 0.07435   
##   
## Sensitivity : 0.9078   
## Specificity : 0.6914   
## Pos Pred Value : 0.8366   
## Neg Pred Value : 0.8116   
## Prevalence : 0.6351   
## Detection Rate : 0.5766   
## Detection Prevalence : 0.6892   
## Balanced Accuracy : 0.7996   
##   
## 'Positive' Class : 0   
##

## ROC Curve and calculating the area under the curve(AUC)  
library(ROCR)  
predictions <- predict(model, newdata=test, type="response")  
ROCRpred <- prediction(predictions, test$Survived)  
ROCRperf <- performance(ROCRpred, measure = "tpr", x.measure = "fpr")  
  
plot(ROCRperf, colorize = TRUE, text.adj = c(-0.2,1.7), print.cutoffs.at = seq(0,1,0.1))



auc <- performance(ROCRpred, measure = "auc")  
auc <- auc@y.values[[1]]  
auc

## [1] 0.8714211