# Ejercicio5

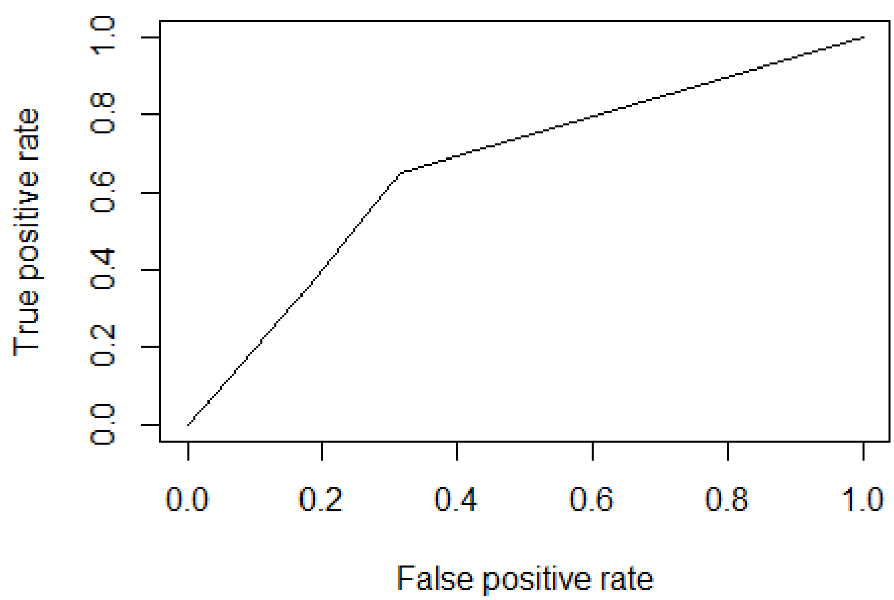
Javier

10/11/2020

```r
library(caret)

## Warning: package 'caret' was built under R version 4.0.3

## Loading required package: lattice

## Loading required package: ggplot2

## Warning: package 'ggplot2' was built under R version 4.0.3

library(nnet)
library(lattice)
library(ggplot2)
library(ROCR)

## Warning: package 'ROCR' was built under R version 4.0.3

# Leemos el csv
data <- read.csv2("Titanic.csv",sep = ";",stringsAsFactors =
FALSE,header=TRUE)
# Lo limpiamos
data <- data[!apply(is.na(data) | data == "", 1, all), ]

#Separamos el contenido del csv en 75% train y 25% test
size_ <- floor(0.75*nrow(data))
size_

## [1] 1200

#Generamos valores aleatorios
set.seed(5)
train_ind <- sample(seq_len(nrow(data)),size=size_)
train <- data[train_ind,]
test <- data[-train_ind,]
model_1 <- glm(Survived ~ Pclass,
family=binomial(link='logit'),data=train)
summary(model_1)

##
## Call:
## glm(formula = Survived ~ Pclass, family = binomial(link = "logit"),
##     data = train)
##
## Deviance Residuals:
```

```
##      Min        1Q    Median        3Q       Max
## -1.4138  -0.7485  -0.7485   0.9581   1.6789
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  1.37533    0.17842   7.708 1.28e-14 ***
## Pclass      -0.83486    0.07522 -11.098  < 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: 1588.8  on 1199  degrees of freedom
## Residual deviance: 1457.1  on 1198  degrees of freedom
## AIC: 1461.1
##
## Number of Fisher Scoring iterations: 4
```

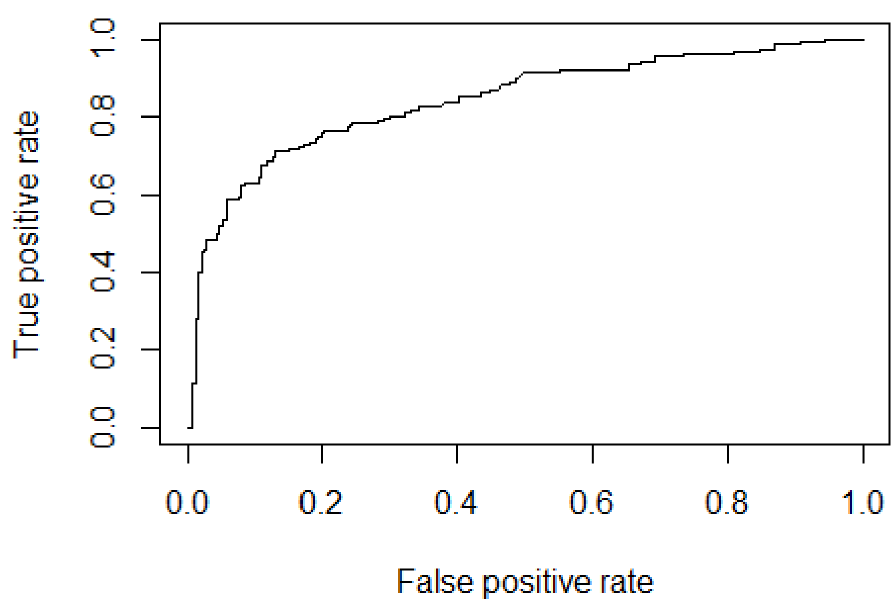
```
#Hacemos la prediccion del modelo
predict_1 <- predict(model_1,
newdata=subset(test,select=c(2,3,4,5,6,7,8)), type="response")
pred_1 <- ROCR::prediction(predict_1, test$Survived)
perf_1 <- performance(pred_1, measure = "tpr", x.measure = "fpr")
plot(perf_1)
```



```
auc_1 <- performance(pred_1, measure = "auc")
auc_1 <- auc_1@y.values[[1]]
auc_1

## [1] 0.6656053

confusionMatrix(table(ifelse(predict_1 < 0.5, 0, 1), test$Survived), dnn
= c("predicted", "actual"))

## Confusion Matrix and Statistics
##
##
##       0   1
##   0 195 105
##   1  42  58
##
##                Accuracy : 0.6325
##                  95% CI : (0.5832, 0.6799)
##     No Information Rate : 0.5925
##     P-Value [Acc > NIR] : 0.05677
##
##                   Kappa : 0.1901
##
##  Mcnemar's Test P-Value : 3.16e-07
##
##             Sensitivity : 0.8228
##             Specificity : 0.3558
##          Pos Pred Value : 0.6500
##          Neg Pred Value : 0.5800
##              Prevalence : 0.5925
##          Detection Rate : 0.4875
##    Detection Prevalence : 0.7500
##       Balanced Accuracy : 0.5893
##
##        'Positive' Class : 0
##

#Creamos el segundo modelo
model_2 <- glm(Survived ~., family=binomial(link='logit'),data=train)
summary(model_2)

##
## Call:
## glm(formula = Survived ~ ., family = binomial(link = "logit"),
##     data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.3836  -0.5721  -0.4266   0.6074   2.4635
##
## Coefficients:
```

```
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  5.310708   0.489957  10.839  < 2e-16 ***
## Pclass      -1.131442   0.125030  -9.049  < 2e-16 ***
## Gendermale  -2.821954   0.176153 -16.020  < 2e-16 ***
## Age         -0.043105   0.006855  -6.288 3.21e-10 ***
## SibSp       -0.311784   0.095534  -3.264   0.0011 **
## Parch       -0.103614   0.103622  -1.000   0.3173
## Fare         0.002355   0.001821   1.293   0.1960
## EmbarkedQ    0.098657   0.342093   0.288   0.7730
## EmbarkedS   -0.158895   0.210262  -0.756   0.4498
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1588.8  on 1199  degrees of freedom
## Residual deviance: 1039.7  on 1191  degrees of freedom
## AIC: 1057.7
##
## Number of Fisher Scoring iterations: 5
```

```r
#Hacemos la prediccion del modelo
predict_2 <- predict(model_2,
newdata=subset(test,select=c(2,3,4,5,6,7,8)), type="response")
pred_2 <- ROCR::prediction(predict_2, test$Survived)
perf_2 <- performance(pred_2, measure = "tpr", x.measure = "fpr")
plot(perf_2)
```



```r
auc_2 <- performance(pred_2, measure = "auc")
auc_2 <- auc_2@y.values[[1]]
auc_2
```

```
## [1] 0.8435712
```

```r
confusionMatrix(table(ifelse(predict_2 < 0.5, 0, 1), test$Survived), dnn
= c("predicted", "actual"))
```

```
## Confusion Matrix and Statistics
##
##
##       0   1
##   0 198  46
##   1  39 117
##
##                Accuracy : 0.7875
##                  95% CI : (0.7441, 0.8266)
##     No Information Rate : 0.5925
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.557
##
##  Mcnemar's Test P-Value : 0.5152
##
##             Sensitivity : 0.8354
##             Specificity : 0.7178
##          Pos Pred Value : 0.8115
##          Neg Pred Value : 0.7500
##              Prevalence : 0.5925
##          Detection Rate : 0.4950
##    Detection Prevalence : 0.6100
##       Balanced Accuracy : 0.7766
##
##        'Positive' Class : 0
##
```