Introducción a la analítica

Profesores César Augusto Gómez, Mauricio Alejandro Mazo y Juan Carlos Salazar

Ejemplo LR: Datos del titanic[1]

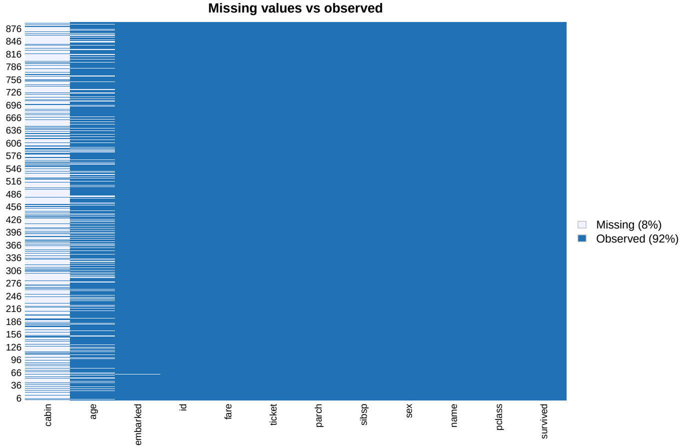
El archivo train.csv contiene datos de 887 de los pasajeros reales del Titanic. Cada fila representa a una persona. Las columnas describen diferentes atributos sobre la persona, incluyendo si sobrevivieron (survived), su edad (age), su clase de pasajero (pclass), su sexo (sex) y la tarifa que pagaron (fare) entre otras. Ajuste un modelo de LR múltiple a estos datos usando como respuesta la variable survived.

[1] On April 15, 1912, the largest passenger liner ever made collided with an iceberg during her maiden voyage. When the Titanic sank it killed 1502 out of 2224 passengers and crew. This sensational tragedy shocked the international community and led to better safety regulations for ships. One of the reasons that the shipwreck resulted in such loss of life was that there were not enough lifeboats for the passengers and crew. Although there was some element of luck involved in surviving the sinking, some groups of people were more likely to survive than others.
https://web.stanford.edu/class/archive/cs/cs109/cs109.1166/problem12.html

Ejemplo: Datos del titanic.

```
## survived   pclass     name      sex      age    sibsp    parch   ticket
##        0        0        0        0      177        0        0        0
##     fare    cabin embarked       id
##        0      687        2        0

## survived   pclass     name      sex      age    sibsp    parch   ticket
##        2        3      891        2       89        7        7      681
##     fare    cabin embarked       id
##      247      148        4      891
```

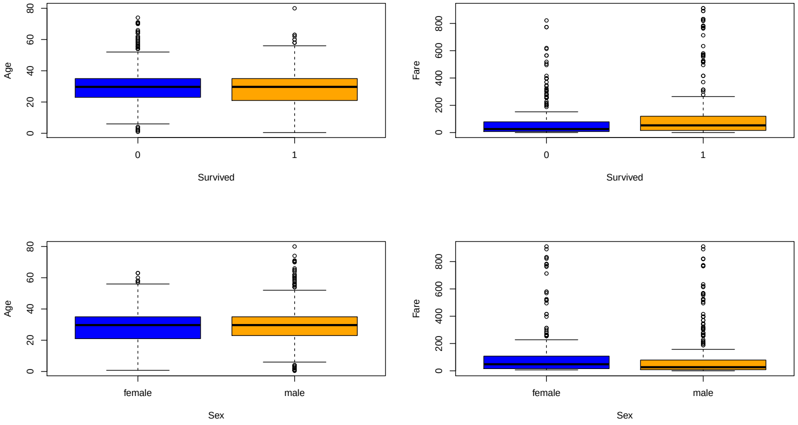
1) The data cleaning process.



1) The data cleaning process.

```
data <- subset(training.data.raw,select=c(1,2,4,5,6,7,9,11))
data$fare<-as.numeric(paste(data$fare))
data$fare[is.na(data$fare)] <- mean(data$fare,na.rm=T)
data$age[is.na(data$age)] <- mean(data$age,na.rm=T)
data <- data[!is.na(data$embarked),]
rownames(data) <- NULL
```

2) Descriptive analysis.



2) Descriptive analysis.

```
## $sex_by_survived
##
##            0   1
##   female  81 231
##   male   468 109
##
## $embarked_by_survived
##
##       0   1
##   C  75  93
##   Q  47  30
##   S 427 217
##
## $Age_summary
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.42   22.00   29.70   29.65   35.00   80.00
##
## $Fare_summary
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00   10.50   29.12   91.31   83.47  910.79
```

3) Model fitting.

```
##
## Call:
## glm(formula = survived ~ ., family = binomial(link = "logit"),
##     data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.5965  -0.5829  -0.4262   0.6305   2.4475
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  4.8801670  0.5517676   8.845  < 2e-16 ***
## pclass      -1.0591096  0.1351872  -7.834 4.71e-15 ***
## sexmale     -2.7840758  0.2137356 -13.026  < 2e-16 ***
## age         -0.0393843  0.0083994  -4.689 2.75e-06 ***
## sibsp       -0.3258694  0.1169999  -2.785  0.00535 **
## parch       -0.1362905  0.1274912  -1.069  0.28506
## fare         0.0019689  0.0007323   2.689  0.00717 **
## embarkedQ    0.2082736  0.4136491   0.504  0.61461
## embarkedS   -0.1061956  0.2665338  -0.398  0.69031
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1065.39  on 799  degrees of freedom
## Residual deviance:  702.22  on 791  degrees of freedom
## AIC: 720.22
##
## Number of Fisher Scoring iterations: 5
```

4) Assessing the predictive ability of the model.

```
## 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                      799    1065.39
## pclass    1   83.607       798     981.79 < 2.2e-16 ***
## sex       1  240.014       797     741.77 < 2.2e-16 ***
## age       1   17.495       796     724.28 2.881e-05 ***
## sibsp     1   10.842       795     713.43  0.000992 ***
## parch     1    0.863       794     712.57  0.352873
## fare      1    9.428       793     703.14  0.002137 **
## embarked  2    0.927       791     702.22  0.628933
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
## fitting null model for pseudo-r2
```

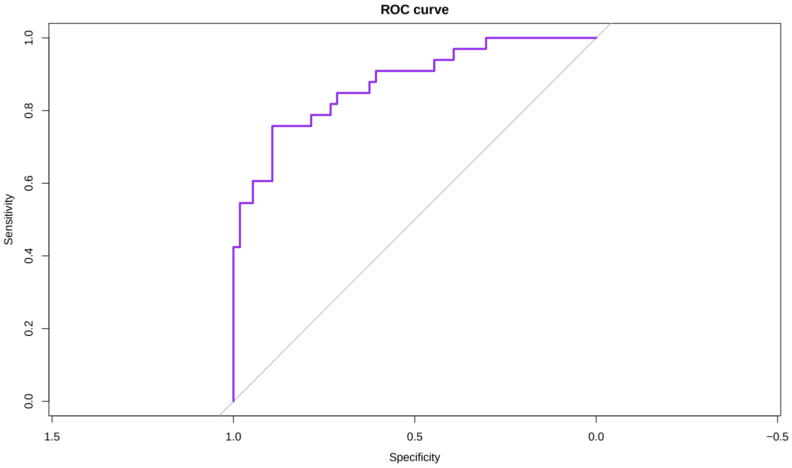
```
##          llh      llhNull           G2     McFadden         r2ML         r2CU
## -351.1079355 -532.6961008  363.1763307    0.3408851    0.3648985    0.4957977
```

```
## [1] "Accuracy 0.831460674157303"
```

5) Confussion Matrix.

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 50  9
##          1  6 24
##
##                Accuracy : 0.8315
##                  95% CI : (0.7373, 0.9025)
##     No Information Rate : 0.6292
##     P-Value [Acc > NIR] : 2.521e-05
##
##                   Kappa : 0.6319
##
##  Mcnemar's Test P-Value : 0.6056
##
##             Sensitivity : 0.8929
##             Specificity : 0.7273
##          Pos Pred Value : 0.8475
##          Neg Pred Value : 0.8000
##              Prevalence : 0.6292
##          Detection Rate : 0.5618
##    Detection Prevalence : 0.6629
##       Balanced Accuracy : 0.8101
##
##        'Positive' Class : 0
##
```

6) Assesing the performance of the classifier: Curva ROC.



6) Assesing the performance of the classifier: Curva ROC. Area under ROC curve (AUC ROC statistic).

```
## $AUC
## [1] 0.8755411
```

Ejemplo: Bayes db, LDA db y LR db:

```
#CODE TO SIMULATE A FIGURE SIMILAR TO 4.6 ISLR BUT IT IS BASED ON
#BOTH BAYES CLASSIFIER, LDA AND LR for p=2

library(MASS)
library(class)
library( mixtools )
library(caret)

# require(foreign)
# require(nnet)
# require(ggplot2)
# require(reshape2)

library(foreign)
library(nnet)
library(ggplot2)
library(reshape2)

set.seed(1234)
n_k=20
rho=0.75
Sigma=matrix(c(1,rho,rho,1),2,2)
mu_1=c(-2 , 2)
mu_2=c(2,3)
mu_3=c(1,5.5)

m1<-mvrnorm(n = n_k, mu=mu_1, Sigma=Sigma, tol = 1e-6, empirical = FALSE, EISPACK = FALSE)
x1=m1[,1]
x2=m1[,2]
y<-rep(1,n_k)
m1=cbind(x1,x2,y)
```

Ejemplo: Bayes db, LDA db y LR db:

```
m2<-mvrnorm(n = n_k, mu=mu_2, Sigma=Sigma, tol = 1e-6, empirical = FALSE, EISPACK = FALSE)
x1=m2[,1]
x2=m2[,2]
y<-rep(2,n_k)
m2=cbind(x1,x2,y)

m3<-mvrnorm(n = n_k, mu=mu_3, Sigma=Sigma, tol = 1e-6, empirical = FALSE, EISPACK = FALSE)
x1=m3[,1]
x2=m3[,2]
y<-rep(3,n_k)
m3=cbind(x1,x2,y)
par(mfrow=c(1,1))

df<-data.frame(rbind(m1,m2,m3))

d_1<-function(x1,x2){matrix(c(x1,x2),1,2)%*%
    ginv(Sigma)%*%matrix(mu_1,2,1)-
    0.5*matrix(mu_1,1,2)%*%ginv(Sigma)%*%matrix(mu_1,2,1)+log(1/3)
}

d_2<-function(x1,x2){matrix(c(x1,x2),1,2)%*%
    ginv(Sigma)%*%matrix(mu_2,2,1)-
    0.5*matrix(mu_2,1,2)%*%ginv(Sigma)%*%matrix(mu_2,2,1)+log(1/3)
}

d_3<-function(x1,x2){matrix(c(x1,x2),1,2)%*%
    ginv(Sigma)%*%matrix(mu_3,2,1)-
    0.5*matrix(mu_3,1,2)%*%ginv(Sigma)%*%matrix(mu_3,2,1)+log(1/3)
}
```

Ejemplo: Bayes db, LDA db y LR db:

```
###################BAYES DECISION BOUNDARIES##################
df1<-df[(df[,3]==1 ),]
df2<-df[(df[,3]==2 ),]
df3<-df[(df[,3]==3 ),]
df123<-rbind(df1,df2,df3)
resolution=100
xnew1 <- seq(-5, 5, len=resolution)
ynew1 <- seq(-5, 8, len=resolution)
xnew <- expand.grid(x1 = xnew1, x2 = ynew1)
d1<-rep(NA,nrow(xnew))
for(i in 1:nrow(xnew)){
  d1[i]<-d_1(xnew[i,1],xnew[i,2])
}

d2<-rep(NA,nrow(xnew))
for(i in 1:nrow(xnew)){
  d2[i]<-d_2(xnew[i,1],xnew[i,2])
}

d3<-rep(NA,nrow(xnew))
for(i in 1:nrow(xnew)){
  d3[i]<-d_3(xnew[i,1],xnew[i,2])
}
dks123<-data.frame(d1,d2,d3)
maxdks123=apply(dks123,1,max)
ndks123<-data.frame(dks123,maxdks123)
ndks123$cat<-0
yhat123<-ifelse(ndks123$d1==ndks123$maxdks,ndks123$cat<-1,
                ifelse(ndks123$d2==ndks123$maxdks,ndks123$cat<-2,
                       ifelse(ndks123$d3==ndks123$maxdks,ndks123$cat<-3,
                              ndks123$cat<-NA)))
yhat_123<-matrix(yhat123,resolution,resolution)
```

Ejemplo: Bayes db, LDA db y LR db:

```
decisionplot <- function(model, data, class = NULL, predict_type = "class",
                         resolution = 100, showgrid = TRUE, line_color,line_type, ...)
  {
  if(!is.null(class)) cl <- data[,class] else cl <- 1
  data <- data[,1:2]
  k <- length(unique(cl))
  #plot(data, col = as.integer(cl)+1L, pch = as.integer(cl)+1L, ...)
  #plot(data, col = c("orange","blue","limegreen"), pch = 20, ...)
  r <- sapply(data, range, na.rm = TRUE)
  # xs <- seq(r[1,1], r[2,1], length.out = resolution)
  # ys <- seq(r[1,2], r[2,2], length.out = resolution)
  xs <- seq(-5, 5, length.out = resolution)
  ys <- seq(-5, 8, length.out = resolution)
  g <- cbind(rep(xs, each=resolution), rep(ys, time = resolution))
  colnames(g) <- colnames(r)
  g <- as.data.frame(g)
  ### guess how to get class labels from predict
  ### (unfortunately not very consistent between models)
  p <- predict(model, g, type = predict_type)
  if(is.list(p)) p <- p$class
  p <- as.factor(p)
  if(showgrid) points(g, col = as.integer(p)+1L, pch = ".")
  z <- matrix(as.integer(p), nrow = resolution, byrow = TRUE)
  contour(xs, ys, z, add = TRUE, drawlabels = FALSE,
          lwd = 3.5, levels = (1:(k-1))+.5,col=line_color,lty=line_type)
  invisible(z)
}
```

Ejemplo: Bayes db, LDA db y LR db:

```
par(mar=rep(3, 4))
contour(unique(xnew[, 1]), unique(xnew[, 2]), yhat_123, levels = (1:(4-1))+.5,
        labels=" ", xlab='', ylab='',  lwd=2, lty = 2,
        col='black',ylim=c(-5,8),xlim=c(-5,5),
        main="BAYES, LDA, AND LR Decision Boundaries. p=2")
title(xlab=expression(italic('X')[1]), ylab=expression(italic('X')[2]),
      line=2, family='serif', cex.lab=1.0)
points(xnew, pch=20, cex=0.3,
       col=ifelse(yhat_123==1 , "red",ifelse(yhat_123==2,"limegreen",
                              ifelse(yhat_123==3,"blue","red"))))
points(df123[,1:2], bg=ifelse(df123[,3]==1, "red", ifelse(df123[,3]==2,"limegreen",ifelse(df123[,3]==3,"bl
ellipse(mu=c(1,5.5), sigma=matrix(c(1,rho,rho,1),2,2), alpha = .05,
        npoints = 250, newplot = FALSE, draw = TRUE,col="blue",lwd=2)
ellipse(mu=c(2,3), sigma=matrix(c(1,rho,rho,1),2,2), alpha = .05,
        npoints = 250, newplot = FALSE, draw = TRUE,col="limegreen",lwd=2)
ellipse(mu=c(-2,2), sigma=matrix(c(1,rho,rho,1),2,2), alpha = .05,
        npoints = 250, newplot = FALSE, draw = TRUE,col="red",lwd=2)
legend("bottomright",legend=c("BAYES DB","LDA DB","LR DB"),lty=c(2,1,1),
       col=c("black","purple2","forestgreen"),pch=c(19,19,19),cex=0.8)
box()

####################LDA DECISION BOUNDARIES##################
#LDA MODEL
model <- lda(y ~ ., data=df)
decisionplot(model, df, class = "y", main = "LDA",line_color="purple2",line_type=1)

####################LR DECISION BOUNDARIES##################
#MULTINOMIAL LOGISTIC REGRESSION MODEL LR

model <- multinom(y ~., data = df,trace=FALSE)
decisionplot(model, df, class = "y", main = "Logistic Regression",line_color="forestgreen",line_type=1)
```

Ejemplo: Bayes db, LDA db y LR db:

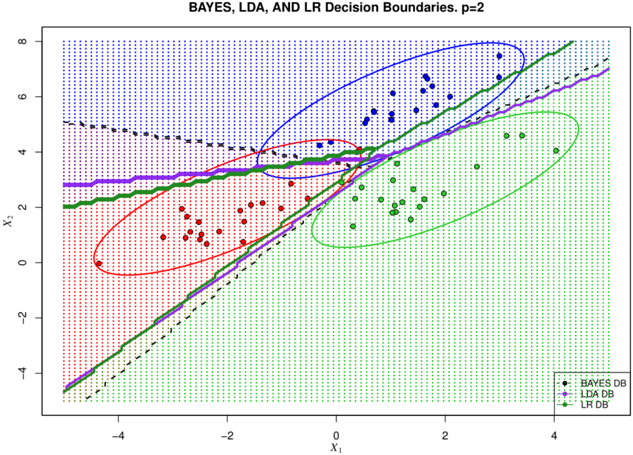


Figura 1: Bayes vs LDA and LR

GRÁFICOS DISCRIMINANTE DE FISHER Y DE PARTICIÓN

Cuando hay clases K, el análisis discriminante lineal puede ser visto exactamente en un diagrama dimensional de orden $K-1$. ¿Por qué? Debido a que esencialmente, LDA, clasifica al centroide más cercano y ellos generan un plano de dimensión $K-1$. Incluso cuando $K>3$, se puede encontrar el mejor plano bidimensional para visualizar la regla discriminante.

GRÁFICOS DISCRIMINANTE DE FISHER Y DE PARTICIÓN

Gráficos de partición de LDA El uso de la función *partimat* del paquete de R *klaR* proporciona una forma alternativa de trazar las funciones discriminantes lineales. *partimat* ofrece una variedad de gráficos para cada combinación de dos variables. Piense en cada gráfico como una representación distinta de los mismos datos. Las regiones coloreadas delinean cada área de clasificación. Se prevé que cualquier observación que se encuentre dentro de una región sea de una clase específica. Cada gráfico también incluye la tasa de error aparente para esa vista particular de los datos.

GRÁFICOS DISCRIMINANTE DE FISHER Y DE PARTICIÓN

```
library(foreign)
library(nnet)
library(ggplot2)
library(reshape2)
library(MASS)
library(tidyverse)
library(klaR)

set.seed(1234)
n_k=20
rho1=0.75
rho2=-0.95

Sigma1=matrix(c(1.2,rho2,rho2,1),2,2)
Sigma=matrix(c(1.5,rho1,rho1,1),2,2)

mu_1=c(-7 , 0)
mu_2=c(1.5,-1)
mu_3=c(6,0.5)

m1<-mvrnorm(n = n_k, mu=mu_1, Sigma=Sigma1, tol = 1e-6, empirical = FALSE, EISPACK = FALSE)
x1=m1[,1]
x2=m1[,2]
y<-rep(1,n_k)
m1=cbind(x1,x2,y)
```

GRÁFICOS DISCRIMINANTE DE FISHER Y DE PARTICIÓN

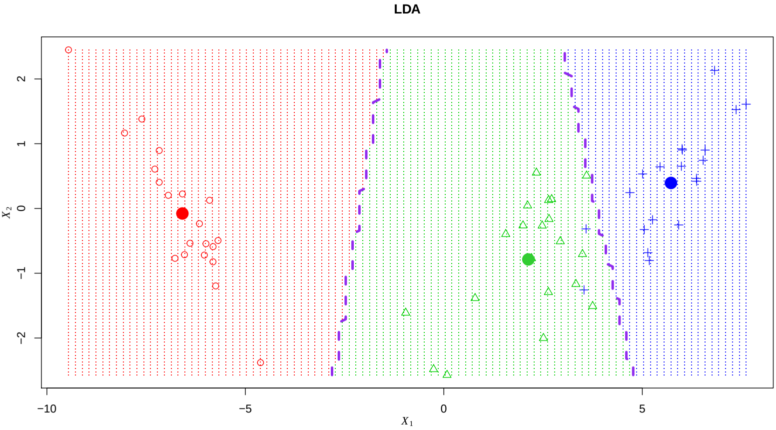
```
m2<-mvrnorm(n = n_k, mu=mu_2, Sigma=Sigma, tol = 1e-6, empirical = FALSE, EISPACK = FALSE)
x1=m2[,1]
x2=m2[,2]
y<-rep(2,n_k)
m2=cbind(x1,x2,y)

m3<-mvrnorm(n = n_k, mu=mu_3, Sigma=Sigma, tol = 1e-6, empirical = FALSE, EISPACK = FALSE)
x1=m3[,1]
x2=m3[,2]
y<-rep(3,n_k)
m3=cbind(x1,x2,y)
par(mfrow=c(1,1))

df<-data.frame(rbind(m1,m2,m3))
means_1<-tapply(df$x1, as.factor(df$y), mean)
means_2<-tapply(df$x2, as.factor(df$y), mean)
means<-cbind(means_1,means_2)
```

GRÁFICOS DISCRIMINANTE DE FISHER Y DE PARTICIÓN

```r
decisionplot <- function(model, data, class = NULL, predict_type = "class",
                         resolution = 100, showgrid = TRUE, line_color,line_type, ...)
{
  if(!is.null(class)) cl <- data[,class] else cl <- 1
  data <- data[,1:2]
  k <- length(unique(cl))
  plot(data, col = as.integer(cl)+1L, pch = as.integer(cl-1)+1L,xlab="",ylab="", cex=1.2,...)
  r <- sapply(data, range, na.rm = TRUE)
   xs <- seq(r[1,1], r[2,1], length.out = resolution)
   ys <- seq(r[1,2], r[2,2], length.out = resolution)
  g <- cbind(rep(xs, each=resolution), rep(ys, time = resolution))
  colnames(g) <- colnames(r)
  g <- as.data.frame(g)
  p <- predict(model, g, type = predict_type)
  if(is.list(p)) p <- p$class
  p <- as.factor(p)
  if(showgrid) points(g, col = as.integer(p)+1L, pch = ".")
  z <- matrix(as.integer(p), nrow = resolution, byrow = TRUE)
  contour(xs, ys, z, drawlabels = FALSE, add = TRUE,
          lwd = 3.5, levels = (1:(k-1))+.5,col=line_color,lty=line_type)
  invisible(z)
  points(means[1,1],means[1,2],col="red",pch=19,cex=2.3)
  points(means[2,1],means[2,2],col="limegreen",pch=19,cex=2.3)
  points(means[3,1],means[3,2],col="blue",pch=19,cex=2.3)
  title(xlab=expression(italic('X')[1]), ylab=expression(italic('X')[2]),
              line=2, family='serif', cex.lab=1.0)
  }
```

GRÁFICOS DISCRIMINANTE DE FISHER Y DE PARTICIÓN

```
#LDA MODEL
model <- lda(y ~ ., data=df)
###################Fisher´s Discriminant Plot###################
decisionplot(model, df, class = "y", main = "LDA",line_color="purple2",line_type=2)
```



```r
# LDA Partition Plots

partimat(as.factor(df$y)~x2+x1, data=df, method="lda")
```