M3 - Tarea - ML Student

Daniel Aláez Riaño

23 de noviembre de 2017

# TAREA M3 - MACHINE LEARNING DATASET STUDENT

## INTRODUCCIÓN

A continuación se generará un modelo de Machine Learning (ML) sobre el dataset ["student"](https://archive.ics.uci.edu/ml/datasets/Student+Performance) de manera que tras un proceso de análisis descriptivo inicial, se realizará un análisis exploratorio junto con métodos de ML No Supervisado, y para finalizar, se aplicarán 2 modelos de ML Supervisado con el objetivo de ser capaces de predecir el valor de la nota final G3.

getwd()  
library(knitr)  
library(readr)  
library(dplyr)  
library(plotrix)  
library(ggplot2)  
library(grid)  
library(gridExtra)  
library(rpart)  
library(caret)  
library(corrplot)

## CARGA DE LOS DATOS Y ANÁLISIS DESCRIPTIVO

Comenzamos con la carga de los datos de los estudiantes en 2 datasets *df\_mat* y *df\_por* respectivamente. Posteriormente, creamos un dataset conjunto con el fin de disponer de todos los datos juntos (sería útil para la realización de análisis conjuntos e incluso estudiar posibles relaciones entre campos comunes).

# Cargamos los 2 datasets relativos a los students de matemáticas y portugués.  
df\_mat = read.table("student-mat.csv",sep=";",header=TRUE)  
df\_por = read.table("student-por.csv",sep=";",header=TRUE)   
  
# Creamos el dataset conjunto con los datos recogidos de ambos datasets con la función merge  
df\_all = merge(df\_mat,df\_por,  
 by=c("school","sex","age","address","famsize","Pstatus","Medu","Fedu","Mjob","Fjob","reason","nursery","internet"),   
 all = FALSE,   
 suffixes=c(".mat",".por"))  
  
names(df\_all)

Realizamos un análisis descriptivo de dichos datos basado en el estudio de los datos extraidos del uso de las funciones **summary**, **str** y **head**.

# Analizamos los datos descriptivos mediante la función summary.  
summary(df\_all)

# Analizamos los datos descriptivos mediante la función summary.  
str(df\_all)

## LIMPIEZA Y PREPARACIÓN DE LOS DATOS

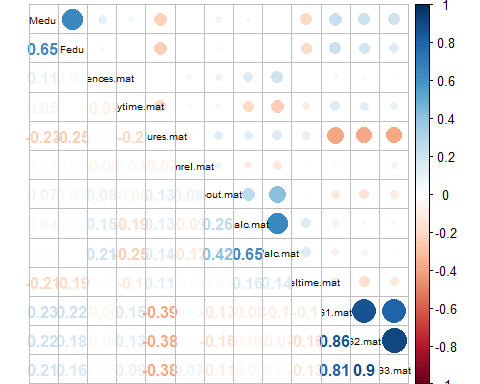
Examinando la información recogida podemos destacar que en total disponemos de un dataset con 382 observaciones y 53 variables o columnas (de las cuales se distinguen algunas de éstas entre las correspondientes a los estudiantes de matemáticas y los estudiantes de portugués. P.e.: Distinguiendo las ausencias de una clase o de la otra *absences.mat* o *absences.por*).

En este caso en concreto, vamos a centrarnos en la variable *G3.mat*, de manera que podamos predecir los resultados finales de matemáticas que van a tener diferentes estudiantes en función de los valores del resto de sus variables y principalmente basándonos en las siguientes variables numéricas *Medu*, *Fedu*,*absences.mat*,*studytime.mat*, *failures.mat*, *goout.mat*, *Dalc.mat*,*Walc.mat*, *traveltime.mat*, *G1.mat*,*G2.mat*, *famrel.mat*

Además, y de cara a futuros análisis clasificatorios, se crean las variables *pass.mat* y *pass.por* que recogen un valor binario en función de si la nota es < o > de 10. Es decir, distinguiendo entre "aprobados" y "suspensos".

Sobre el propio código y de cara a la realización de análisis clasificatorios basados en variables categóricas se dejan comentados los análisis y visualización de información realizados sobre otro tipo de variables como son *school*, *Mjob*, *sex* o *activities.mat* de manera que se permita ver mediante diferentes gráficas información de las notas finales obtenidas en función de los valores de dichas variables.

#Dado que vamos a centrarnos principalmente en las   
df\_all$pass.mat <- ifelse(df\_all$G3.mat>9, 1, 0)  
df\_all$pass.por <- ifelse(df\_all$G3.por>9, 1, 0)  
  
#### Análisis Exploratorio - Correlación  
# Visualización de la matriz de correlación entre las variables numéricas seleccionadas con el fin de discernir aquellas variables más correlacionadas con G3.mat  
  
#Visualización - Con valores numéricos  
corrplot.mixed(corr <- cor(df\_all[,c("Medu","Fedu","absences.mat","studytime.mat", "failures.mat","famrel.mat", "goout.mat","Dalc.mat","Walc.mat","traveltime.mat","G1.mat","G2.mat","G3.mat")],  
 method = "pearson"), tl.col="black", tl.cex= .6, tl.pos = "d")



#### Análisis Exploratorio - Variables categóricas "school", "Mjob",   
  
#ggplot(df\_all, aes(Mjob)) + geom\_bar() + facet\_wrap(~pass.mat) +   
# ggtitle("Fig 2.1 Diagrama de barras trabajo materno, por nota final") +  
# theme(plot.title = element\_text(vjust=+1.5, size=12))

## MODELO NO SUPERVISADO - CLUSTERING K-MEANS

Con el fin de poder examinar mejor las diferencias entre los estudiantes que aprueban y los que suspenden Matemáticas, vamos a realizar un análisis mediante un modelo No Supervisado de Clustering de manera que sobre una agrupación con 2 cluster podamos comparar su agrupación con respecto a los alumnos "aprobados" y "suspensos" y las características de sus variables. Para este análisis vamos a utilizar únicamente el dataset relativo a los alumnos de matemáticas para evitar confusión con el Portugués.

# Creamos la variable pass sobre el dataset relativo a los estudiantes de matemáticas  
df\_mat$pass <- ifelse(df\_mat$G3>9, 1, 0)  
df\_mat\_mod <- df\_mat   
# Nos quedamos únicamente con las variables numéricas de interés para aplicar K-MEANS sobre éstas  
df\_mat\_mod <- df\_mat\_mod %>% select(G1, G2, Medu, Fedu, absences, failures, goout, Dalc, Walc, traveltime, studytime, famrel)  
  
# Mediante el método de elbow seleccionamos el número de clusters  
# Aplicamos el algoritmo K-Means sobre 4 cluster en base al método de elbow aplicado anteriormente  
set.seed(1234)  
kmeans.clust <- kmeans(df\_mat\_mod, 4)  
table(df\_mat$pass, kmeans.clust$cluster)

##   
## 1 2 3 4  
## 0 3 0 33 94  
## 1 2 158 42 63

#Visualizamos las variables más representativas  
#{plot(df\_mat\_mod %>% select(G1, G2, studytime, absences, famrel), col = kmeans.clust$cluster)   
#Visualizamos la relación mediante un gráfico radial  
#radial.plot(kmeans.clust$centers[1,], labels=names(kmeans.clust$centers[1,]), rp.type="s", radial.lim=c(0,10), point.symbols=13, point.col="red", mar = c(2,1,5,2))  
df\_mat\_mod\_final <- df\_mat\_mod %>% mutate(cluster\_id = kmeans.clust$cluster)   
#kable(head(df\_mat\_mod\_final))  
  
#Examinamos la distribución de valores de todas las variables utilizadas dentro del cluster 2 el cual representa la totalidad de sus observaciones como "aprobados"  
df\_mat\_clust2 <- subset(df\_mat\_mod\_final, df\_mat\_mod\_final$cluster\_id==2)  
#hist(df\_mat\_clust2)

Se observa de una manera lógica la relación entre las variables utilizadas en un conjunto de muestras donde todos los estudiantes han aprobado la asignatura de Matemáticas. P.e: *G1* y *G2* aprobados, *Medu*, *Fedu* y *famrel* con valores altos, *Absences*, *failures*, *Dalc*, *Walc*, *traveltime*, con valores bajos y curiosamente *goout* y *studytime* se encuentran en valores medios y no destacan por ser valores bajos y altos respectivamente.

## MODELOS SUPERVISADOS - REGRESIÓN LINEAL Y CLASIFICACIÓN CON ÁRBOLES DE DECISIÓN - ENTRENAMIENTO Y EVALUACIÓN

### Regresión Lineal multivariable con Stepwize Linear Regression

Comenzamos aplicando como Modelo Supervisado la Regresión Lineal con el fin de predecir el valor de la nota final en base a los datos obtenidos de las notas previas y alguna variable adicional.

Para ello, vamos a utilizar el modelo "Stepwize Linear Regression" sobre las variables seleccionadas durante las fases anteriores con el fin de que el propio modelo tras un proceso recurrente de evaluación de todas las variables, nos proporcione aquellas más significativas (según lo visto anteriormente anteriormente deberían ser G1, G2, studytime, absences, famrel, Medu y Fedu).

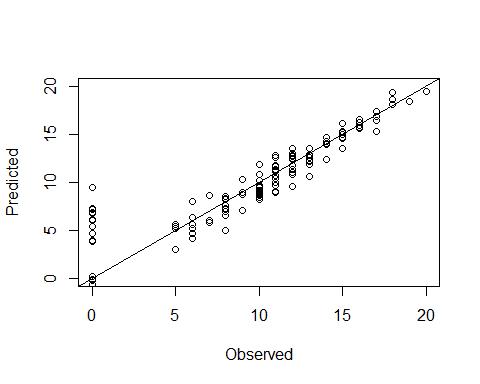
# REGRESIÓN LINEAL  
# Comenzamos seleccionando las variables bajo estudio que serán todas menos G3 y pass.  
# Limpiamos el dataset excluyendo las variables fuera de interés  
df\_all\_lm <- df\_all[,!colnames(df\_all)=="pass.mat"]  
  
# Creamos los dataset de entrenamiento y de test  
set.seed(1234)  
train.sample <- sample(1:nrow(df\_all\_lm),size=(nrow(df\_all\_lm)\*2/3),replace=F)  
train.df\_all\_lm <- df\_all\_lm[train.sample,]  
test.df\_all\_lm <- df\_all\_lm[-train.sample,]  
  
# Entrenamos el modelo con todas las variables posibles, excepto la variable "pass" para detectar variables con c  
lineal.reg.model.df\_all\_lm <- lm(G3.mat ~ + Medu + Fedu + failures.mat + absences.mat + famrel.mat + studytime.mat + goout.mat + Dalc.mat + Walc.mat + traveltime.mat + G1.mat + G2.mat, data=train.df\_all\_lm)  
SLR.model.total <- step(lineal.reg.model.df\_all\_lm)  
summary(SLR.model.total)

Efectivamente, y tal como se preveía, se puede observar que el modelo lineal óptimo es el compuesto por las variables G1, G2, studytime, absences y famrel. Obteniendo un valor bastante elevado de R-squared = 0.8455 (adjusted R-squared = 0.8424), así como un valor de F-Statistic de 271.4 y un p-value muy reducido (<2.2e-16).

Por lo tanto, y una vez construido el modelo vamos a proceder a evaluarlo contra el dataset de test que hemos recogido anteriormente para ver su eficacia.

df\_all.pred.lm <- predict(lineal.reg.model.df\_all\_lm, test.df\_all\_lm)  
head(test.df\_all\_lm$G3.mat)  
head(df\_all.pred.lm)  
#hist(df\_all.pred.lm-test.df\_all\_lm$G3.mat)

# Predecimos los nuevos valores con predict()  
xlim <- range(df\_all$G3.mat)  
{plot(df\_all.pred.lm ~ G3.mat, data=test.df\_all\_lm, xlab="Observed",ylab="Predicted", ylim=xlim, xlim=xlim)  
abline(a=0, b=1)}



analisis\_lm <- c(mean(df\_all.pred.lm), median(df\_all.pred.lm), mean(test.df\_all\_lm$G3.mat), median(test.df\_all\_lm$G3.mat))  
names(analisis\_lm) <- c("Media Pred.", "Mediana Pred.", "Media Test", "Mediana Test")  
analisis\_lm

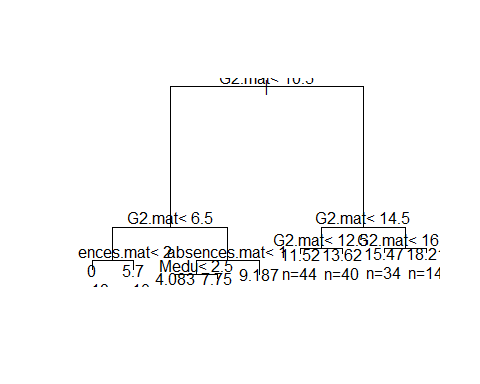
## Media Pred. Mediana Pred. Media Test Mediana Test   
## 10.091028 9.665610 9.742188 10.000000

# Se observa como la media de las predicciones tiende a ser mayor que la media de las notas de nuestro dataset

### Árboles de decisión

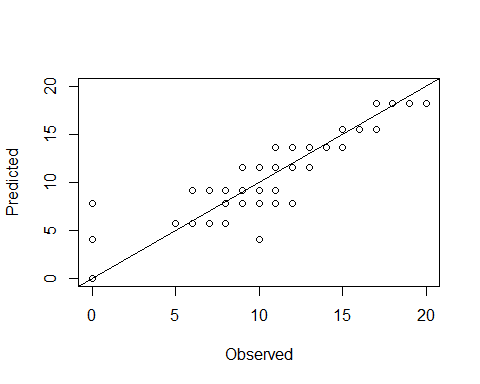
A continuación vamos a construir un modelo de predicción basado en un árbol de decisión para lo cual utilizaremos las mismas variables que en el caso de la regresión lineal empleada anteriormente para comparar los resultados obtenidos.

# Creamos los dataset de entrenamiento y de test  
# Se entrena el modelo con rpart()  
formulaG3 <- G3.mat ~ Medu + Fedu + failures.mat + absences.mat + famrel.mat + studytime.mat + goout.mat + Dalc.mat + Walc.mat + traveltime.mat + G1.mat + G2.mat  
df\_all\_rpart <- rpart(formulaG3, data = train.df\_all\_lm, control = rpart.control(minsplit = 10))  
attributes(df\_all\_rpart)  
print(df\_all\_rpart$cptable)  
print(df\_all\_rpart)  
  
# Podamos el árbol y obtenemos aquel con el mínimo error de predicción  
opt <- which.min(df\_all\_rpart$cptable[,"xerror"])  
cp <- df\_all\_rpart$cptable[opt, "CP"]  
df\_all\_rpart\_prune <- prune(df\_all\_rpart, cp = cp)  
  
# Vemos como quedan las reglas del árbol podado   
print(df\_all\_rpart\_prune)  
  
# Representamos gráficamente el árbol podado   
{plot(df\_all\_rpart\_prune)  
text(df\_all\_rpart\_prune, fheight=0.2, use.n=T)}



A continuación vamos a predecir los resultados de nuestro dataset de test

# Predecimos los nuevos valores con predict()  
df.pred.rpart <- predict(df\_all\_rpart\_prune, newdata=test.df\_all\_lm)   
xlim <- range(df\_all$G3.mat)  
{plot(df.pred.rpart ~ G3.mat, data=test.df\_all\_lm, xlab="Observed",ylab="Predicted", ylim=xlim, xlim=xlim)  
abline(a=0, b=1)}



#hist(d3.pred.rpart-test.d3\_rpart$G3.mat)  
  
analisis\_rpart <- c(mean(df.pred.rpart), median(df.pred.rpart), mean(test.df\_all\_lm$G3.mat), median(test.df\_all\_lm$G3.mat))  
names(analisis\_rpart) <- c("Media Pred.", "Mediana Pred.", "Media Test", "Mediana Test")  
analisis\_rpart

## Media Pred. Mediana Pred. Media Test Mediana Test   
## 10.016482 9.186667 9.742188 10.000000