test

Miguel Ángel Castaño Ibáñez

5/18/2021

Librerías

Cargo todas las librerías utlilizadas para para este ejercicio

```
# LIBRERIES


# install
# install.packages('tinytex')
# install.packages("reshape")

# load
suppressPackageStartupMessages({
library(ggplot2)
library(inspectdf)      # EDAs automaticos
library(plotly)
library(dummies)
library(MASS)
library(caret)
library(plyr)
library(reshape)
library(randomForest)
# library(tinytex)
})
```

Ejercicios

A continuacion, los ejercicios propuestos a resolver en este modulo

a) Se deben realizar pruebas suficientes para obtener una buena selección de variables, obteniendo uno o varios conjuntos de variables tentativos

b) Se requiere la comparación entre los mejores algoritmos y regresión logística

c) Se comprobará el efecto de la variación de los parámetros básicos de cada algoritmo (tuneado) (número de nodos en redes, shrink en gradient boosting, etc.).

d) Los algoritmos a utilizar son obligatoriamente y como mínimo:

- Redes Neuronales

- Regresión Logística
- Bagging
- Random Forest
- Gradient Boosting
- Support Vector Machines
- También si se quiere y para comprender los datos se puede probar con un simple árbol pero no es obligatorio.

e) Es necesario utilizar validación cruzada, validación cruzada repetida o como mínimo training/test repetido.

f) Es necesario hacer alguna prueba de ensamblado.

Lectura ficheros

Nuestro dataset contiene una muestra de 5000 pacientes de diferentes edades, donde podemos observar quien de ellos a sufrido un ictus. Este dataset presenta 12 variables, 11 inputs y una dependiente objetivo binaria.

Fuente: *https://www.kaggle.com/fedesoriano/stroke-prediction-dataset*

Cargamos los ficheros donde están el conjunto de entrenamiento y el de test, ademas de tener un dataset solo el id y la variable objetivo

```
data <- read.csv("./healthcare-dataset-stroke-data.csv")
```

Analisís exploratorio (EDA)

Antes de empezar a crear los modelos vamos a hacer un analisis exploratorio de nuestra variable por si fuera necesario, imputar valores o cambiar el tipo de alguna de estas.

```
#compruebo los tipos
str(data)
```

```
## 'data.frame':    5110 obs. of  12 variables:
##  $ id               : int  9046 51676 31112 60182 1665 56669 53882 10434 27419 60491 ...
##  $ gender           : chr  "Male" "Female" "Male" "Female" ...
##  $ age              : num  67 61 80 49 79 81 74 69 59 78 ...
##  $ hypertension     : int  0 0 0 0 1 0 1 0 0 0 ...
##  $ heart_disease    : int  1 0 1 0 0 0 1 0 0 0 ...
##  $ ever_married     : chr  "Yes" "Yes" "Yes" "Yes" ...
##  $ work_type        : chr  "Private" "Self-employed" "Private" "Private" ...
##  $ Residence_type   : chr  "Urban" "Rural" "Rural" "Urban" ...
##  $ avg_glucose_level: num  229 202 106 171 174 ...
##  $ bmi              : chr  "36.6" "N/A" "32.5" "34.4" ...
##  $ smoking_status   : chr  "formerly smoked" "never smoked" "never smoked" "smokes" ...
##  $ stroke           : int  1 1 1 1 1 1 1 1 1 1 ...
```

Tras los resultados vistos en el apartado anterior podemos concluir a llevar a factor aquellas variables que consideramos categoricas, y transformar a "Yes/No", nuestra variable objetivo binaria.

```
# transfromamos en yes or no la variable obj
data$stroke<-ifelse(data$stroke==1,"Yes","No")

# convert to factor
data[,c(2,4,5,6,7,8,11,12)] <- lapply(data[,c(2,4,5,6,7,8,11,12)], factor)

# convert to nummeric
data$bmi <- as.numeric(data$bmi)

# comprobar tipos
str(data)
```

```
## 'data.frame':    5110 obs. of  12 variables:
##  $ id               : int  9046 51676 31112 60182 1665 56669 53882 10434 27419 60491 ...
##  $ gender           : Factor w/ 3 levels "Female","Male",..: 2 1 2 1 1 2 2 1 1 1 ...
##  $ age              : num  67 61 80 49 79 81 74 69 59 78 ...
##  $ hypertension     : Factor w/ 2 levels "0","1": 1 1 1 1 2 1 2 1 1 1 ...
##  $ heart_disease    : Factor w/ 2 levels "0","1": 2 1 2 1 1 1 2 1 1 1 ...
##  $ ever_married     : Factor w/ 2 levels "No","Yes": 2 2 2 2 2 2 2 2 1 2 2 ...
##  $ work_type        : Factor w/ 5 levels "children","Govt_job",..: 4 5 4 4 5 4 4 4 4 4 ...
##  $ Residence_type   : Factor w/ 2 levels "Rural","Urban": 2 1 1 2 1 2 1 2 1 2 ...
##  $ avg_glucose_level: num  229 202 106 171 174 ...
##  $ bmi              : num  36.6 NA 32.5 34.4 24 29 27.4 22.8 NA 24.2 ...
##  $ smoking_status   : Factor w/ 4 levels "formerly smoked",..: 1 2 2 3 2 1 2 2 4 4 ...
##  $ stroke           : Factor w/ 2 levels "No","Yes": 2 2 2 2 2 2 2 2 2 2 ...
```

```
# Comprobar observaciones de la var objetivo
length(filter(data, stroke == "Yes")[,1])
```

```
## [1] 249
```

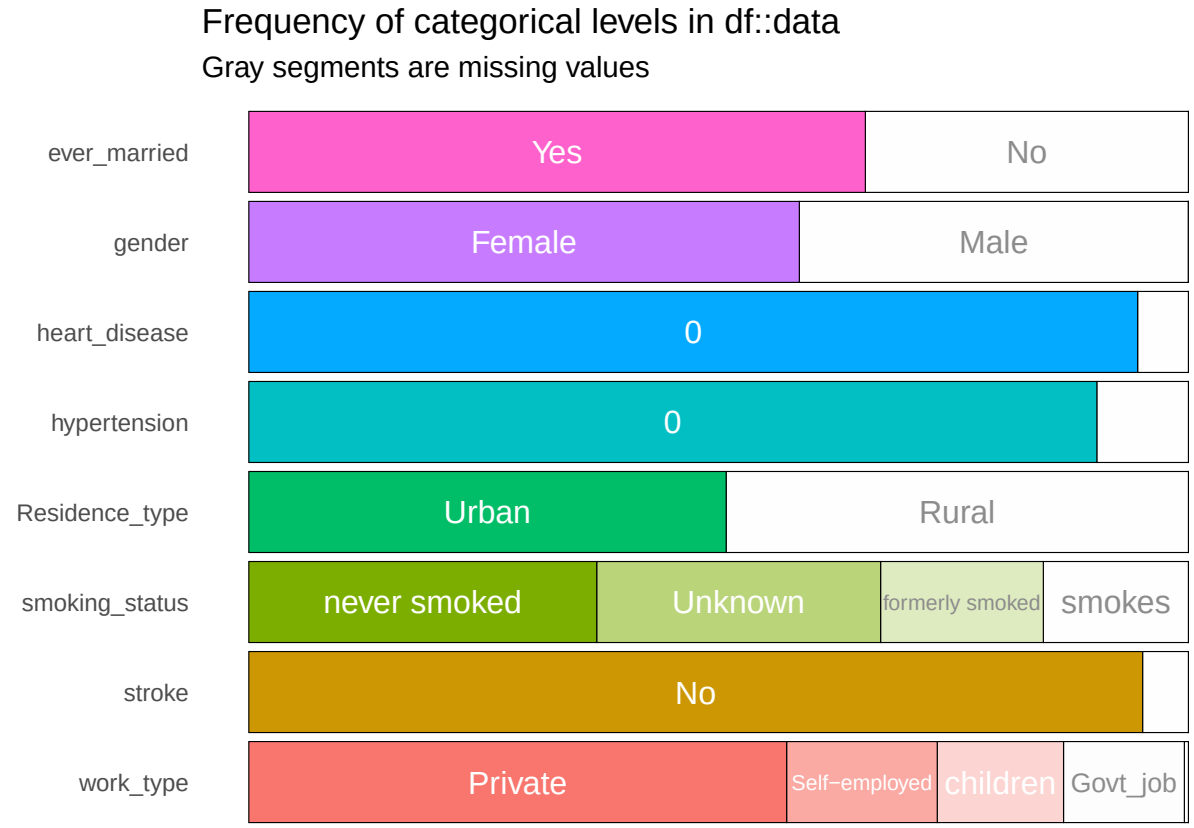
```
length(filter(data, stroke == "No")[,1])
```

```
## [1] 4861
```

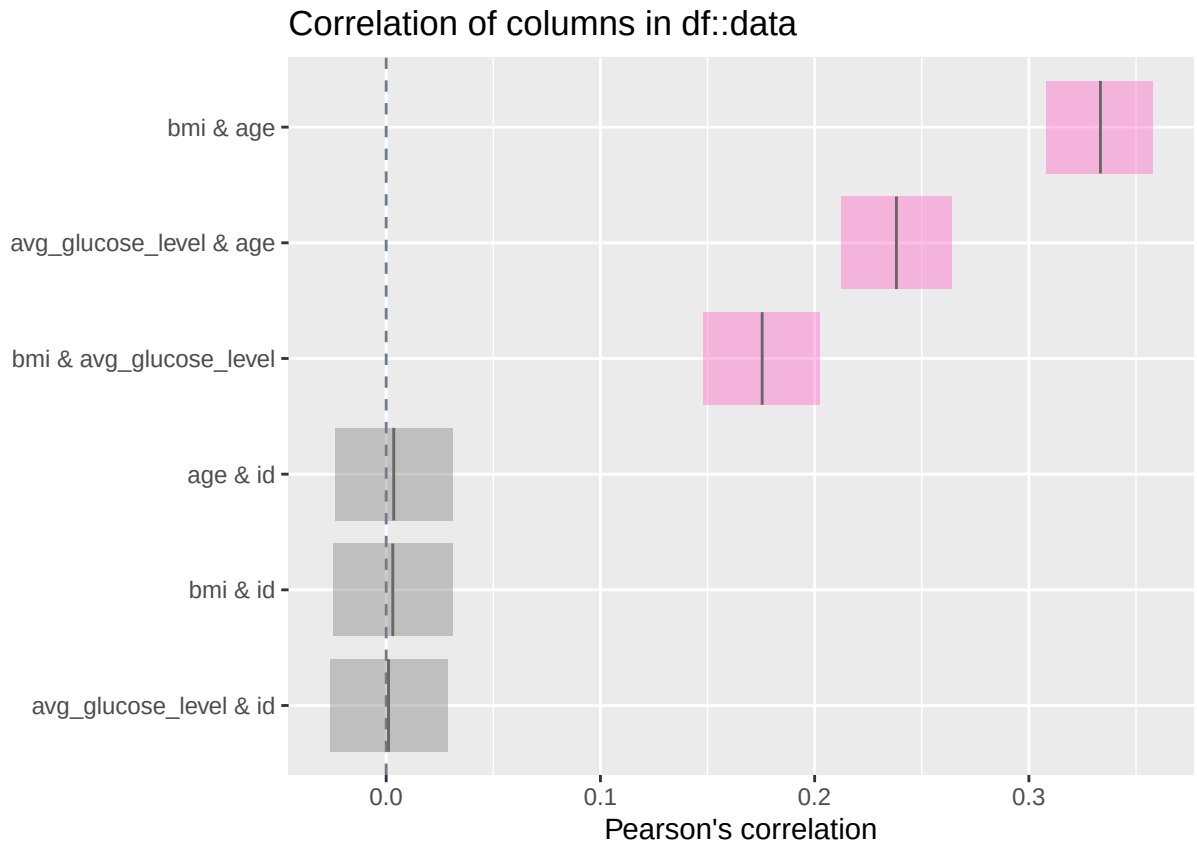
El numero de casos que si presentan ataque cerebral es muy pequeño respecto al numero de los que no, y esto prodria llevarnos a conclusiones sesgadas, pese a tener un accuracy bastante, por ello debemos ir con cuidado. Tras la eleccion del modelo seria conveniente comprobar este modelos con un dataset mas grande de datos.

En segundo lugar vamos a comprobar las variables categoricas que hay, la correlacion de variable por si fuera necesario quitar alguna de estas y el numero de NAs.

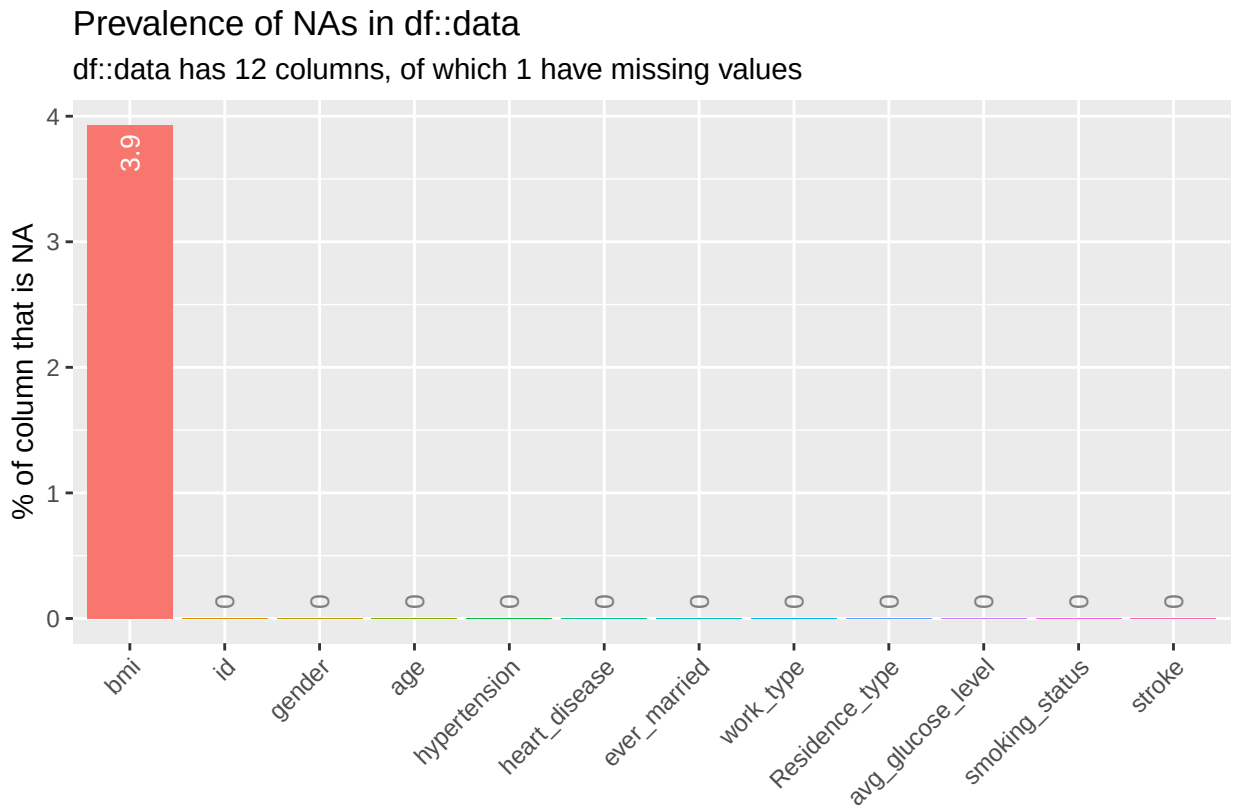
```
# Horizontal bar plot for categorical column composition
x <- inspect_cat(data)
show_plot(x)
```




```
# Correlation betwee numeric columns + confidence intervals
x <- inspect_cor(data)
show_plot(x)
```




```
# Occurence of NAs in each column ranked in descending order
x <- inspect_na(data)
show_plot(x)
```



Feature engineering

Nuestro primer paso para a aplicar feature engineering a nuestras variables es separarlas en continuas, categoricas y objetivo. La variable id no tiene relevancia en este dataset por ello he decidido dejarla fuera del analisis.

```
# lista de variables
# dput(names(data))
list_int <- c("id")
list_cont <- c("avg_glucose_level", "bmi", "age")
list_cat <- c("gender", "hypertension", "heart_disease", "ever_married",
              "work_type", "Residence_type", "smoking_status")
var_dep <- c("stroke")
col_var_dep<-data[,var_dep]
```

Estandarización de variables continuas

Estandarizamos las variables continuas con valores desde 0 hasta 1

```
# calc estandarizacion
means <-apply(data[,list_cont],2,mean,na.rm=TRUE)
sds<-sapply(data[,list_cont],sd,na.rm=TRUE)
```

```
# var continuas estandarizadas
stroke_data <- scale(data[,list_cont], center = means, scale = sds)
# stroke_data <- data.frame(cbind(stroke_data,col_var_dep))

# union continuas y categoricas
index_cont<-which(colnames(data)%in%list_cont) # index cont
stroke_data<-cbind(stroke_data,data[,-index_cont]) # join
```

Eliminacion de missings

En lugar de imputar las observaciones missing he considerado eliminarlas del dataset a estudiar

```
stroke_data<-na.omit(stroke_data,(!is.na(stroke_data)))

# comprobamos suficientes observaciones YES y NO
length(filter(stroke_data, stroke == "Yes")[,1])
```

```
## [1] 209
```

```
length(filter(stroke_data, stroke == "No")[,1])
```

```
## [1] 4700
```

Dummies

Vamos a aplicar dummies a las variables categoricas, tratandolas de convertir a binarias mediante metodologia one-hot, es decir valor 1 si se cumple la observacion para esa variable categoriaca y 0 cuando no.

```
stroke_data<- dummy.data.frame(stroke_data, list_cat, sep = ".")

#eliminamos los dummies pocos representados
stroke_data$work_type.Never_worked <- NULL
```

He decidico eliminar las dummi work_type.Never_worked, ya que apenas hay observaciones para esta variable.

Por ultimo para evitar futuros conflictos aplico la siguiente funcion, para cambiar el nombre de las columnas que puedan tener palabras reservadas.

```
# Make Valid Column Names
colnames(stroke_data) <- make.names(colnames(stroke_data))
```

Seleccion de variables

En este paso coincidiendo con el apartado a) vamos a tratar de obtener las variables mas relevantes para generar modelos en nuestro dataset.

En primer lugar vamos a aplicar el algoritmo stepwise backward and fordwar y con criterio tonto AIC como BIC para ver cuales son las variables que mas importancia tienen en la regresion logistica.

```r
# Selección de variables por el metodo stepAIC
full<-glm(factor(stroke)~.,data=stroke_data,family = binomial(link="logit"))
null<-glm(factor(stroke)~1,data=stroke_data,family = binomial(link="logit"))

# aplicamos stepAIC
seleccionAIC<-stepAIC(null,scope=list(upper=full),direction="both")
seleccionBIC<-stepAIC(null,scope=list(upper=full),direction="both")
```

Obtenemos la importancia de las variables calculadas para nuestro modelo mediante estos metodos

```r
# AIC
seleccionAIC
```

```
##
## Call:  glm(formula = factor(stroke) ~ age + avg_glucose_level + hypertension.0 +
##     work_type.Self.employed + smoking_status.smokes + heart_disease.0,
##     family = binomial(link = "logit"), data = stroke_data)
##
## Coefficients:
##             (Intercept)                      age        avg_glucose_level
##                 -3.3850                   1.6360                   0.2174
##          hypertension.0  work_type.Self.employed    smoking_status.smokes
##                 -0.5507                  -0.3907                   0.3932
##         heart_disease.0
##                 -0.3577
##
## Degrees of Freedom: 4908 Total (i.e. Null);  4902 Residual
## Null Deviance:       1728
## Residual Deviance: 1366  AIC: 1380
```

```r
# BIC
seleccionBIC
```

```
##
## Call:  glm(formula = factor(stroke) ~ age + avg_glucose_level + hypertension.0 +
##     work_type.Self.employed + smoking_status.smokes + heart_disease.0,
##     family = binomial(link = "logit"), data = stroke_data)
##
## Coefficients:
##             (Intercept)                      age        avg_glucose_level
##                 -3.3850                   1.6360                   0.2174
##          hypertension.0  work_type.Self.employed    smoking_status.smokes
##                 -0.5507                  -0.3907                   0.3932
##         heart_disease.0
##                 -0.3577
##
## Degrees of Freedom: 4908 Total (i.e. Null);  4902 Residual
## Null Deviance:       1728
## Residual Deviance: 1366  AIC: 1380
```

Otra forma de poder seleccionar estas variables es repitiendo el proceso un numero determinado de veces, obteniendo varios modelos y observar las variables mas frecuentes en estos.