Práctica Machine Learning

Fernández Hernández, Alberto

4/28/2021

[IIIII]  
 )"""(  
 / \  
 / \  
 |`-...-'|  
 |aspirin|  
 \_ |`-...-'j \_  
 (\)`-.\_\_\_.(I) \_(/)  
 (I) (/)(I)(\)  
 (I)

# 1. Introducción y descripción de los datos

El objetivo del presente proyecto consiste en **elaborar un modelo de clasificación binaria que permita predecir si un paciente presentará o será más propenso a padecer una complicación hospitalaria tras una intervención quirúrgica** [[1]](#footnote-1). Originalmente, el fichero (extraído de la plataforma Kaggle) contiene tres variables objetivo, dos continuas:

1. *ccsComplicationRate*: incidencia general de complicaciones hospitalarias por cada tipo de intervención quirúrgica.
2. *complication\_rsi*: índice de estratificación de riesgo en complicaciones hospitalarias.

Y una binaria:

1. *complication*: **si el paciente ha sufrido una complicación (1) o no (0)**.

Por tanto, de cara a la práctica tendremos únicamente en cuenta, como variable objetivo, la columna *complication*, descartando las dos variables continuas anteriores.

En relación con las posibles variables *input*, nos encontramos con las siguientes:

**CONTINUAS**

1. *bmi*: **índice de masa corporal**.
2. *Age*: **edad del paciente**.
3. *baseline\_charlson*: **índice de comorbidad de Charlson, el cual predice la mortalidad a diez años de un paciente que puede tener una variedad de condiciones comórbidas (como una enfermedad cardíaca, SIDA o cáncer)**.
4. *ahrq\_ccs*: **tipo de procedimiento/intervención quirúrgica, etiquetado por la Agencia estadounidense para la Investigación Sanitaria** [[2]](#footnote-2). Dicha variable contiene un total de 22 valores únicos, por lo que se ha decidido mantener la variable como numérica.
5. *ccsMort30Rate*: **incidencia general de mortalidad a los 30 días por cada intervención (dado por el código de la columna ahrq\_ccs)**.
6. *hour*: **hora a la que se realizó la intervención**.
7. *mortality\_rsi*: **índice de estratificación de riesgo en la mortalidad a los 30 días**.

**CATEGÓRICAS**

1. *asa\_status*: **estado físico del paciente establecido por la Sociedad Americana de Anestesiología** [[3]](#footnote-3). Contiene tres categorías:

* 0: **estado I-II** (paciente sano / paciente con enfermedad sistémica leve).
* 1: **estado III** (paciente con enfermedad sistémica grave).
* 2: **estado IV-VI** (paciente con enfermedad muy grave / no espera sobrevivir sin la operación / muerte cerebral).

1. *baseline\_cancer*: **¿El paciente padece algún cáncer?** (1 = Sí; 0 = No)
2. *baseline\_cvd*: **¿El paciente sufre alguna enfermedad cardio o cerebrovascular?** (1 = Sí; 0 = No)
3. *baseline\_dementia*: **¿El paciente sufre algún trastorno por demencia?** (1 = Sí; 0 = No)
4. *baseline\_diabetes*: **¿El paciente sufre diabetes?** (1 = Sí; 0 = No)
5. *baseline\_digestive*: **¿El paciente sufre alguna enfermedad gastro-intestinal?** (1 = Sí; 0 = No)
6. *baseline\_osteoart*: **¿El paciente padece osteoartritis**[[4]](#footnote-4)**?** (1 = Sí; 0 = No)
7. *baseline\_psych*: **¿El paciente padece algún desorden psiquiátrico?** (1 = Sí; 0 = No)
8. *baseline\_pulmonar*: **¿El paciente sufre alguna enfermedad pulmonar?** (1 = Sí; 0 = No)
9. *dow* o *day of week*: **día de la semana en el que se realizó la intervención** (0 = Lunes; 1 = Martes; 2 = Miércoles; 3 = Jueves; 4 = Viernes).
10. *month*: **mes en el que se realizó la intervención**.
11. *moonphase*: **fase lunar que tuvo lugar durante la intervención quirúrgica** (0 = Luna nueva; 1 = Cuarto creciente; 2 = Luna llena; 3 = Cuarto menguante).
12. *mort30*: **¿El paciente presenta algún riesgo de fallecer a los 30 días?** (1 = Sí; 0 = No)
13. *gender*: **Sexo del paciente** (0 = Hombre; 1 = Mujer)
14. *race*: **raza del paciente** (0 = Caucásico; 1 = Afroameriano; 2 = Otro)

# 2. Librerías empleadas

A continuación, se expone un listado de las librerías empleadas en el desarrollo del proyecto:

1. *caret*: tuneo de hipérparámetros de los diferentes algoritmos de clasificación.
2. *data.table*: estructura de datos, similar al *data.frame*, aunque mucho más eficiente en memoria.
3. *ggplot2*: librería gráfica.
4. *scorecard*: cálculo del valor de información (IV), así como el peso de la evidencia (WOE).
5. *dummies*: transformación de variables categóricas a *dummies*.
6. *forcats*: tratamiento de variables categóricas.
7. *inspectdf*: libreria para inspeccionar las caracteristicas principales de un *dataset*, incluyendo variables categóricas, valores *missing* o distribución de las variables continuas.
8. *dplyr*: manipulación de datos.
9. *psych*: información general de data.frames y/o data.tables (media, asimetría, desviación típica, entre otros).
10. *doParallel* y *parallel*: paralelización de funciones.
11. *readxl*: lectura de ficheros *Excel* (.xlsx).
12. *purrr*: herramientas de programación funcional.
13. *visualpred*: visualización de predicciones por diferentes algoritmos de clasificación.
14. *h2o*: *auto Machine Learning (autoML)*.
15. **Librerías y funciones proporcionadas por el profesor**.

#--- Librerias  
suppressPackageStartupMessages({  
 library(caret) # Data partitioning  
 library(data.table) # Lectura de ficheros mucho mas rapido que read.csv  
 library(dplyr) # Manipulacion de datos  
 library(ggplot2) # Libreria grafica  
 library(scorecard) # Woebin + Woebin\_plot + Information Value (IV)  
 library(dummies) # Creacion variables dummy  
 library(psych) # Informacion estadistica de dataframes  
 library(ranger) # Random Forest (+ rapido que caret)  
 library(forcats) # Tratamiento variables categoricas  
 library(inspectdf) # EDA Automatico (II)  
 library(purrr) # Programacion Funcional  
   
 source("./librerias/librerias\_propias.R")  
})

# 3. Depuración de los datos

Inicialmente, comenzamos con la lectura del fichero:

# Lectura del fichero  
surgical\_dataset <- fread("./data/Surgical-deepnet.csv", data.table = FALSE)  
  
# Eliminamos las dos variables objetivo continuas  
surgical\_dataset$ccsComplicationRate <- NULL; surgical\_dataset$complication\_rsi <- NULL  
  
dim(surgical\_dataset) # Filas x columnas

## [1] 14635 23

Nos encontramos con 14.635 observaciones, junto con las 23 variables descritas anteriormente. Si echamos un vistazo a la variable objetivo, podemos observar el desbalanceo entre ambas categorías:

table(surgical\_dataset$complication)

##   
## 0 1   
## 10945 3690

## 3.1 Codificación a factor

En primer lugar, **codificamos como *factor* tanto la variable objetivo como el resto de variables categóricas**:

# Codificamos como factor la variable objetivo...  
surgical\_dataset$complication <- as.factor(surgical\_dataset$complication)  
# ...Asi como el resto de variables categoricas mencionadas anteriormente  
cat\_columns <- c("gender", "race", "asa\_status", "baseline\_cancer", "baseline\_cvd",   
 "baseline\_dementia", "baseline\_diabetes", "baseline\_digestive",   
 "baseline\_osteoart", "baseline\_psych", "baseline\_pulmonary",   
 "dow", "month", "moonphase", "mort30")  
surgical\_dataset[,cat\_columns] <- lapply(surgical\_dataset[, cat\_columns], factor)

A continuación, almacenamos los nombres de cada variable en un vector por separado, **en función de si es continua o categórica**:

# Separamos las variables en numericas, categoricas y target  
# [-16] => Salvo la variable objetivo  
cat\_columns <- names(Filter(is.factor, surgical\_dataset))[-16]  
num\_columns <- names(Filter(is.numeric, surgical\_dataset))  
target <- "complication"

## 3.2 Valores NA

Como se puede comprobar a continuación, el *dataset* **no contiene valores *missing* en ninguna de las variables**:

sum(is.na(surgical\_dataset))

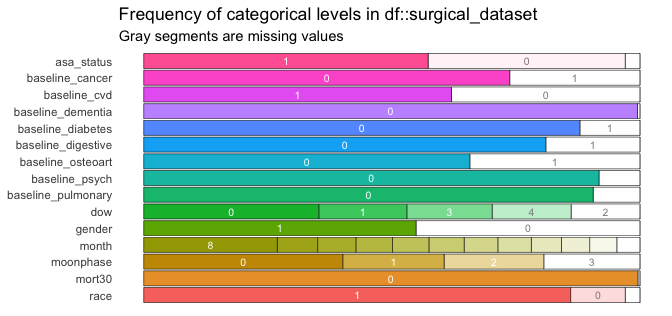
## [1] 0

## 3.3 Variables categóricas

Tras almacenar los nombres de cada variable, mediante la librería *inspectdf* se realizó un primer análisis exploratorio de datos automático con el que **analizar el dataset en primera instancia**. Dado que el contenido del informe es muy extenso, se incluirá en la memoria el contenido esencial (el informe completo se incluye, desglosado, en los anexos *00\_EDA\_report.pdf* y *WOEBIN\_factor\_variables.pdf* ).

Sobre dicho informe, **comenzamos remarcando la frecuencia de aparición de los niveles de cada variable categórica**:

x <- inspectdf::inspect\_cat(surgical\_dataset[, cat\_columns], include\_int = TRUE)  
show\_plot(x)

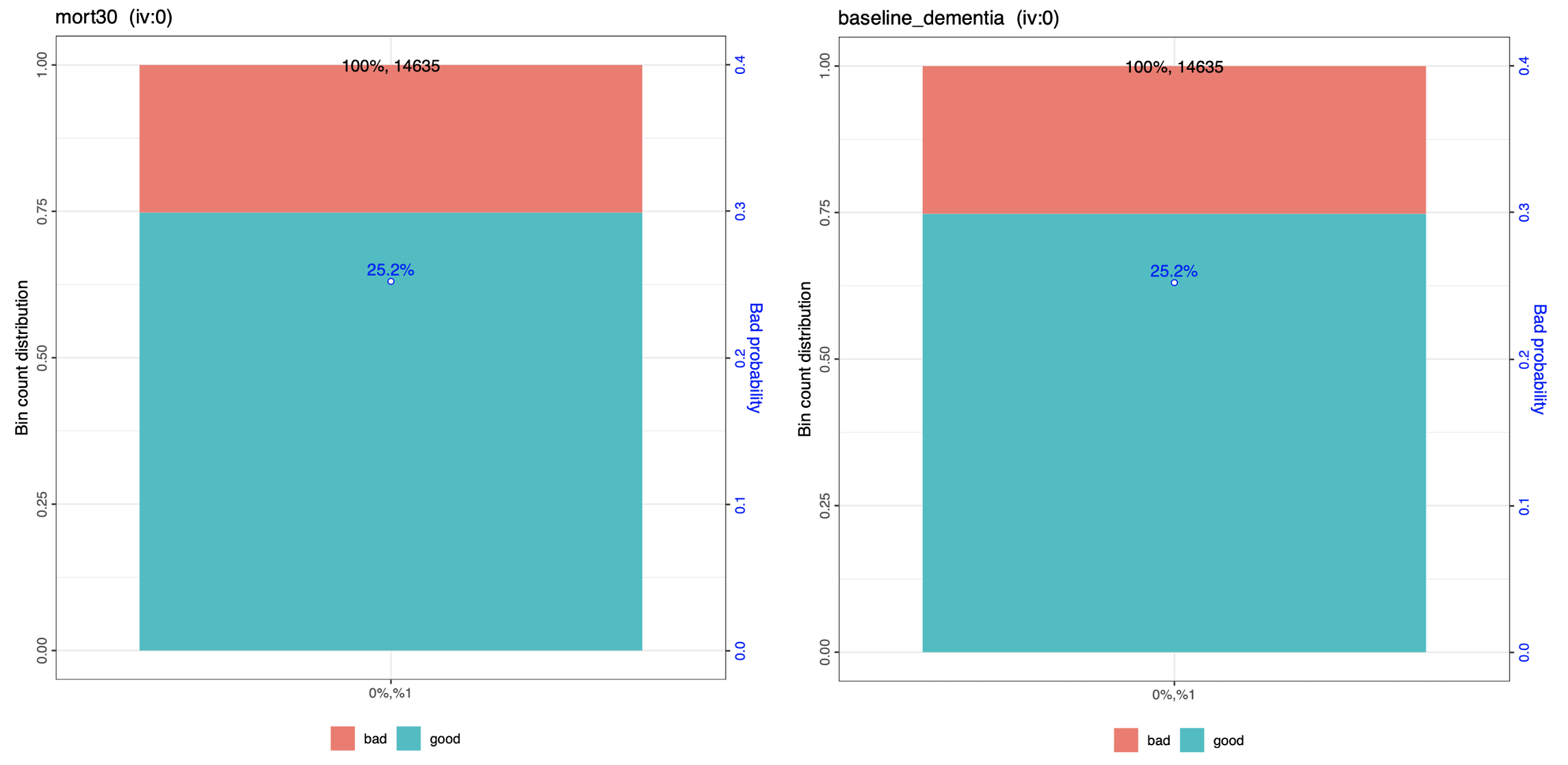


A simple vista, prácticamente todas las categorías presentan una alta frecuencia de aparición, **salvo por *baseline\_dementia* y *mort30***, donde el número de observaciones a 1 es de 71 y 58, respectivamente.

surgical\_dataset[, c("baseline\_dementia", "mort30")] %>% map(table)

## $baseline\_dementia $mort30  
##  
## 0 1 0 1  
## 14564 71 14577 58

Es decir, se tratan de variables con pocas observaciones con valor 1. De hecho, si analizamos el valor de información, haciendo uso del paquete *scorecard*, Observamos que el valor de información es cero, dada la poca representatividad de los valores a 1, de forma que el paquete *scorecard* acaba uniendo ambas categorías, lo que se traduce en un escaso poder predictivo:



Mort 30 y Baseline dementia (IV)

Por otro lado, si analizamos la proporción de aparición de la variable objetivo sobre cada categoría:

#-- baseline\_dementia  
surgical\_dataset %>%  
 count(baseline\_dementia, complication) %>%  
 group\_by(complication)

## # A tibble: 4 x 3  
## # Groups: complication [2]  
## baseline\_dementia complication n  
## <fct> <fct> <int>  
## 1 0 0 10913  
## 2 0 1 3651  
## 3 1 0 32  
## 4 1 1 39

#-- mort30  
surgical\_dataset %>%  
 count(mort30, complication) %>%  
 group\_by(complication)

## # A tibble: 4 x 3  
## # Groups: complication [2]  
## mort30 complication n  
## <fct> <fct> <int>  
## 1 0 0 10924  
## 2 0 1 3653  
## 3 1 0 21  
## 4 1 1 37

A simple vista, en ambas variables **no existe una clara separación sobre la variable objetivo**. Por tanto, se ha tomado la decisión de descartar ambas columnas del conjunto de datos.

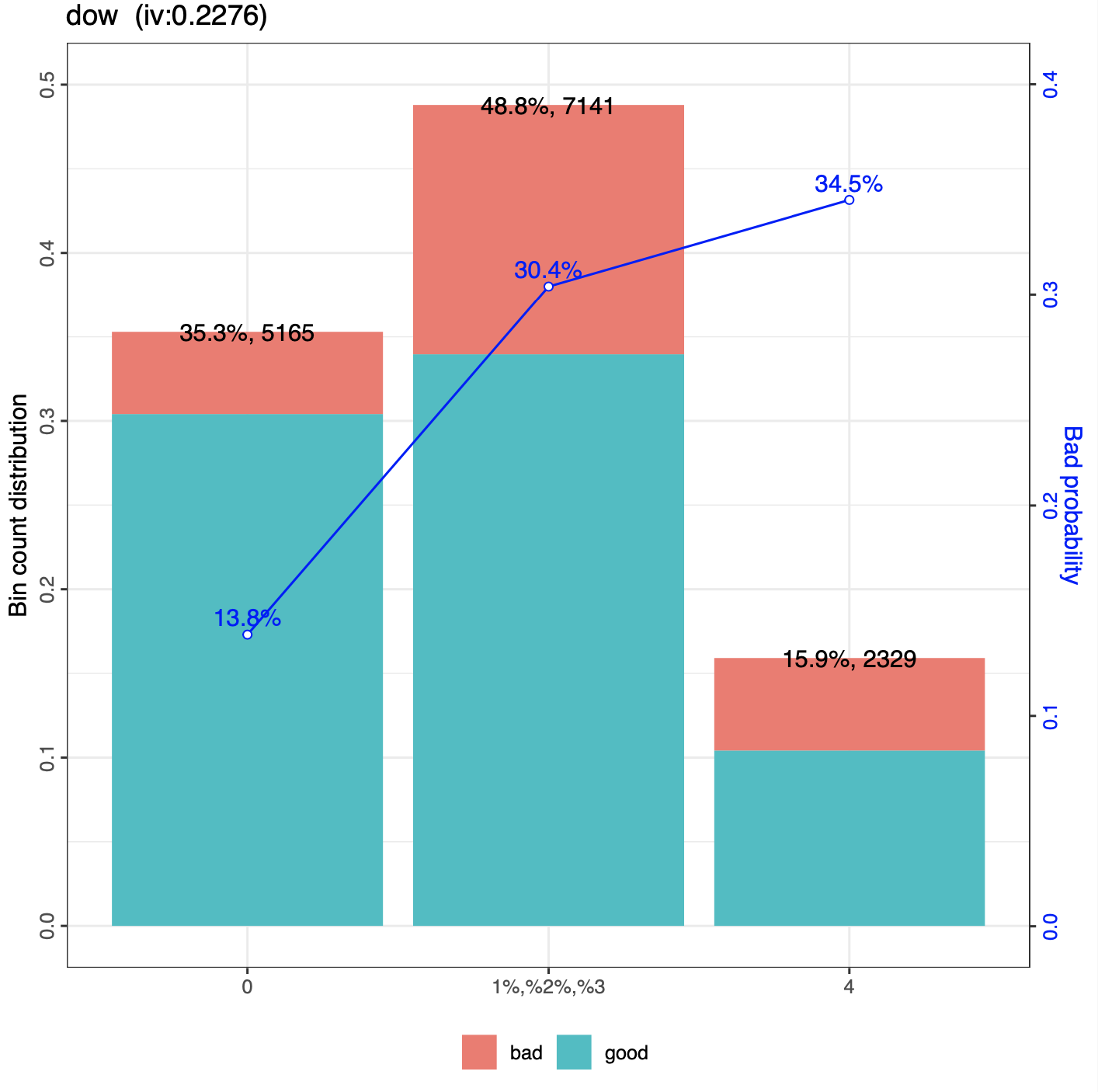
surgical\_dataset$baseline\_dementia <- NULL; surgical\_dataset$mort30 <- NULL   
  
# Actualizamos el vector con las variables categoricas  
cat\_columns <- setdiff(cat\_columns, c("baseline\_dementia", "mort30"))

### 3.3.1 Agrupación de variables categóricas

Por otro lado, nos encontramos con dos variables cuyas categorías pueden ser agrupadas, según la información proporcionada por el paquete *scorecard*:

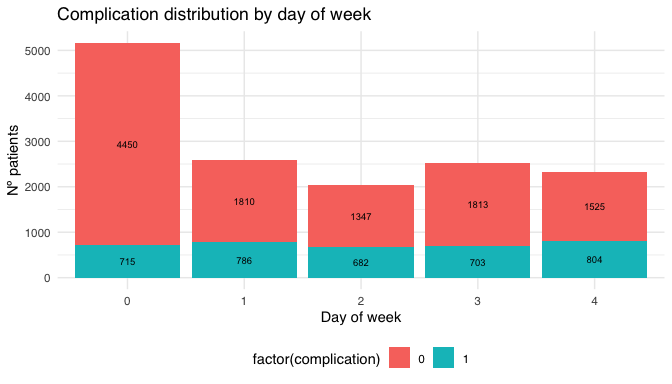
**DÍA DE LA SEMANA** (dow):

Sobre dicha variable, **observamos una relación “lineal” en la distribución de la variable objetivo a lo largo de los diferentes días de la semana**, comenzando por el Lunes (0), con el menor porcentaje de complicaciones hospitalarias (alrededor del 14 %), seguido de los Martes-Miércoles-Jueves, donde el porcentaje aumenta hasta el 30.4 %, y finalizando con los viernes, donde se alcanza el mayor porcentaje de complicaciones hospitalarias: 34.5 %:



Dia de la semana o dow (IV)

Por otro lado, si analizamos detenidamente el gráfico de distribución:



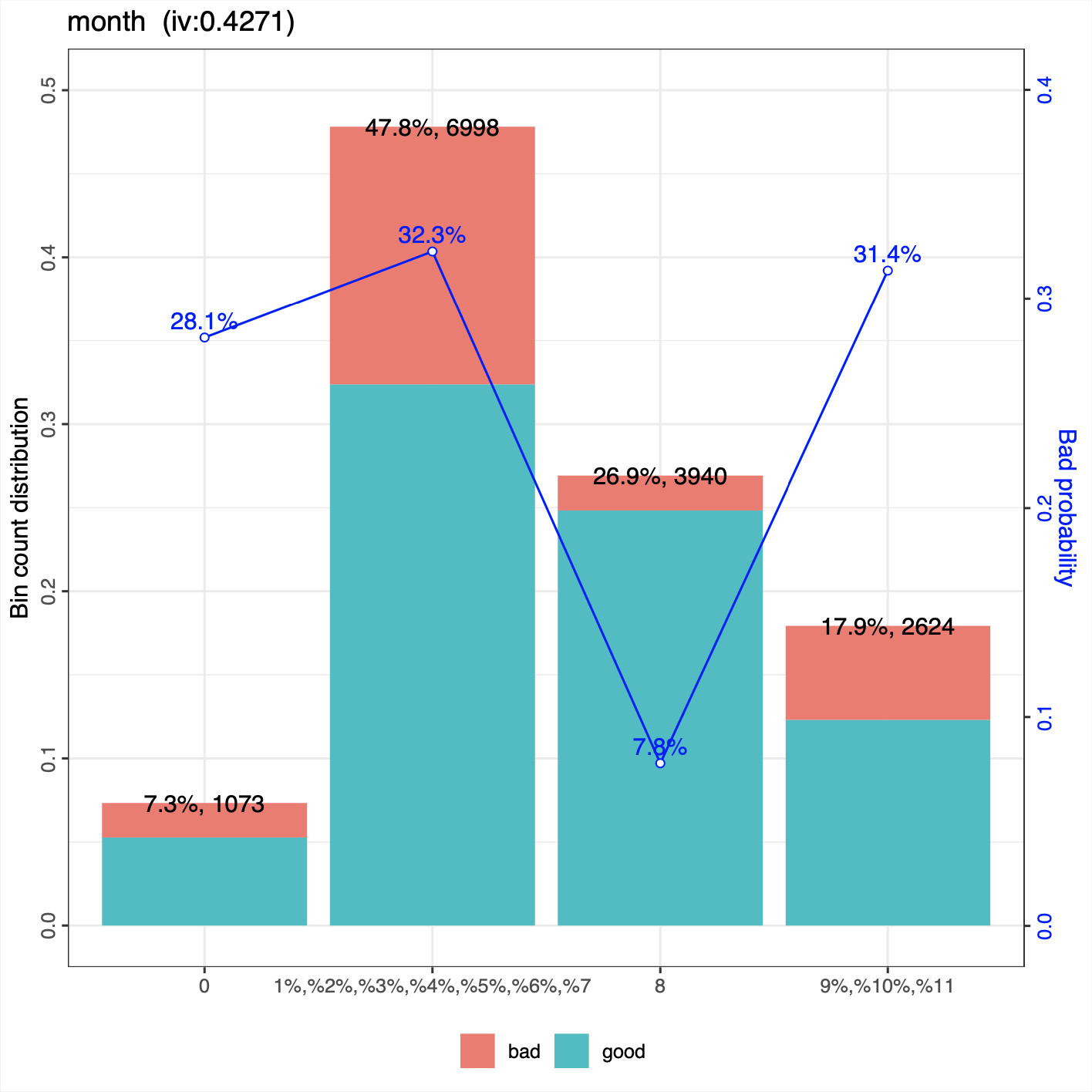
Observamos que la proporción de aparición de pacientes con complicaciones es muy similar entre los martes, miércoles y jueves:

## dow sin.comp con.comp total prop.complicacion  
## 1 1 4450 715 2596 30.3  
## 2 2 1810 786 2029 33.6  
## 3 3 1347 682 2516 27.9  
## 4 En conjunto (1-2-3) 1813 703 7141 30.4  
## 5 4 1525 804 2329 34.5

En conjunto, acumulan alrededor del 30.4 % de pacientes con complicaciones, mientras que con tan solo el viernes aumenta hasta alcanzar el 34 %. Por tanto, dado que los martes, miércoles y jueves presentan una proporción de aparición similar, **las agrupamos en torno a una misma categoría**:

1. **Lunes (0)**
2. **Martes-Miercoles-Jueves (1-3)**
3. **Viernes (4)**

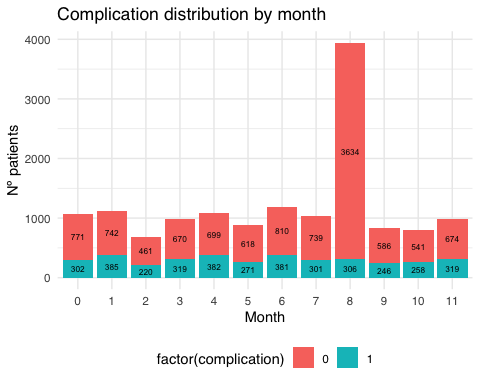
**MES** (month):



Month (IV)

En este caso, llaman la atención tres principales grupos: en primer lugar el mes de enero (0), con un 28.1 % de las complicaciones hospitalarias, **seguido de los meses de febrero (1) hasta agosto (7) con un total acumulado del 32.3 % de los pacientes con complicaciones**, es decir, el mes de enero tiene un porcentaje similar de pacientes con complicaciones que los siguientes 7 meses en conjunto. Por el contrario, **durante el mes de septiembre (8) el porcentaje se desploma hasta el 7.8 %**, porcentaje que vuelve a aumentar en los tres meses siguientes (octubre, noviembre y diciembre), hasta el 31.4 %.

Por otro lado, si analizamos el gráfico de distribución:



Sucede un comportamiento similar al de la variable *dow*: salvo el mes de septiembre, **la distribución de la variable objetivo sobre cada mes es muy similar, de forma que podemos agrupar varios de los meses en una misma categoría, tal y como hemos comprobado anteriormente**. A modo de ejemplo, analicemos la proporción en el número de complicaciones hospitalarias de cada mes por separado y en conjunto:

## dow sin.comp con.comp total prop.complicacion  
## 1 0 771 302 1073 28.1  
## 2 1 742 385 1127 34.2  
## 3 2 461 220 681 32.3  
## 4 3 670 319 989 32.3  
## 5 4 699 382 1081 35.3  
## 6 5 618 271 889 30.5  
## 7 6 810 381 1191 32.0  
## 8 7 739 301 1040 28.9  
## 9 Uniendo 1-7 4739 2259 6998 32.3  
## 10 8 3634 306 3940 7.8  
## 11 9 586 246 832 29.6  
## 12 10 541 258 799 32.3  
## 13 11 674 319 993 32.1  
## 14 Uniendo 9-11 1801 823 2624 31.4

Como podemos comprobar, **la proporción de pacientes con complicaciones hospitalarias, entre los meses de febrero (1) y agosto (7), se sitúa en torno al 32 % si agrupamos dichas categorías**. Del mismo modo, **los meses de octubre, noviembre y diciembre (9, 10 y 11) se sitúan en torno al 31 %**. Como consecuencia, y dado que dichas categorías presentan una proporción similar en cuanto a pacientes con complicaciones se refiere, las agrupamos:

1. **Enero (0)**
2. **Febrero a Agosto (1-7)**
3. **Septiembre (8)**
4. **Octubre, Noviembre y Diciembre (9-10-11)**

En relación con el resto de variables categóricas, si analizamos su valor de información:

## variable iv  
## 1 baseline\_osteoart 0.5246  
## 2 month 0.4271  
## 3 dow 0.2276  
## 4 moonphase 0.2119  
## 5 baseline\_cancer 0.1358  
## 6 baseline\_cvd 0.0429  
## 7 gender 0.0221  
## 8 baseline\_digestive 0.0134  
## 9 asa\_status 0.0062  
## 10 baseline\_pulmonary 0.0053  
## 11 baseline\_diabetes 0.0013  
## 12 race 0.0002  
## 13 baseline\_psych 0.0001

Por lo general, la mayoría de las variables categóricas presentan, como primera impresión, un buen poder predicitivo, con ciertas excepciones como *asa\_status*, *baseline\_pulmonary*, *baseline\_diabetes*, *race* o *baseline\_pysch*, cuyo IV no alcanza el 0.01 (como normal general, un valor de información inferior a 0.1-0.02 se traduce en un poder predictivo muy bajo)[[5]](#footnote-5). De este modo, de cara a la selección de variables, **en caso de ser necesario descartar alguna variable categórica, utilizaremos esta tabla como referencia**.

## 3.4 Variables continuas

En relación con las variables continuas, nos encontramos con algunas variables con un número de valores únicos muy reducidos:

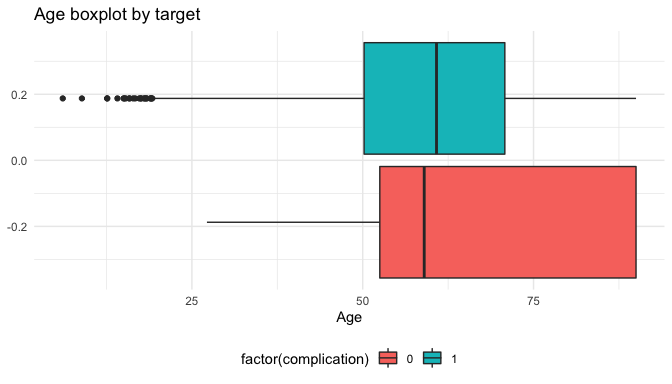
#-- Nº Valores unicos (continuas)  
apply(surgical\_dataset[, num\_columns], 2, function(x) {length(unique(x))})

## bmi Age baseline\_charlson ahrq\_ccs   
## 3095 672 14 22   
## ccsMort30Rate hour mortality\_rsi   
## 20 725 633

A modo de ejemplo, variables como *baseline\_charlson*, *ahrq\_ccs* o *ccsMort30Rate* presentan 14, 22 o 20 valores únicos, respectivamente. No obstante, se tomó la decisión de mantener dichas variables como continuas, principalmente para facilitar la selección de variables, evitando con ello crear nuevas variables *dummy* (una por cada categoría), utilizando con ello menos parámetros para describir el conjunto de datos.

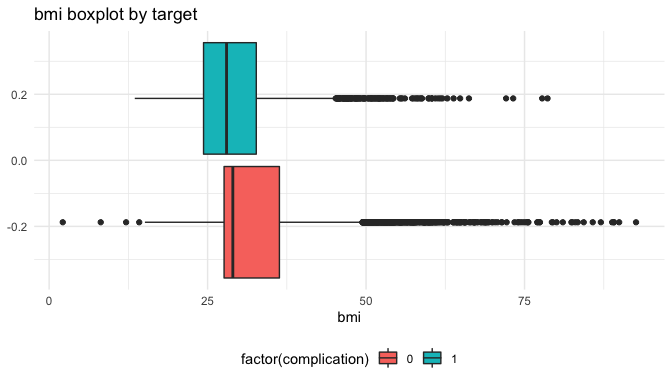
Por otro lado, en un primer análisis exploratorio se observó que en las variables continuas, por lo general, **no existe una clara separación entre ambas variables objetivo**. A modo de ejemplo, observemos las dos siguientes variables: *age* y *bmi*:

**AGE**:



En este caso, **bien es cierto que los pacientes que no sufren complicaciones oscilan entre los 25 y los 90 años, aproximadamente, mientras que existe un número de pacientes (menores a 25 años), donde padecen alguna complicación hospitalaria**. Pese a ello, la separación no es tan clara.

**BMI**:



Incluso con el índice de masa de corporal, **a simple vista no existe una clara separación que diferencie a ambas variables objetivo en función de dicho índice**.

## 3.5 Estandarización de variables continuas

Una vez realizado el primer análisis, **procedemos a la estandarización de las variables continuas**:

# --- Estandarizacion de variables  
surgical\_dataset\_stnd <- surgical\_dataset  
  
# media  
media <- sapply(surgical\_dataset\_stnd[, num\_columns], mean)  
  
# sd  
desv.tipica <- sapply(surgical\_dataset\_stnd[, num\_columns], sd)  
  
surgical\_dataset\_stnd[, num\_columns] <- scale(surgical\_dataset\_stnd[, num\_columns],   
 center = media,   
 scale = desv.tipica)

## 3.6 Creacion de variables dummy

A continuación, convertimos las variables categóricas (con más de una categoría) en variables *dummy*, mediante la función *dummy.data.frame*, conviritiendo de este modo todas las variables a formato numérico:

columnas\_dummy <- c("asa\_status", "dow", "month", "moonphase", "race")  
surgical\_dataset\_stnd\_dummy <- dummy.data.frame(surgical\_dataset\_stnd[, columnas\_dummy],   
 sep = ".")

Una vez codificadas las variables, **comprobamos si existen variables *dummy* con una frecuencia de aparición menor a 100**:

names(surgical\_dataset\_stnd\_dummy[, colSums(surgical\_dataset\_stnd\_dummy == 0) < 100,   
 drop = FALSE])

## character(0)

Como podemos comprobar, todas las variables *dummy* codificadas presentan una frecuencia superior. A continuación, **unimos en un único *data.table* tanto las variables numéricas, las variables categóricas (binarias), las variables *dummy*, así como la variable objetivo**:

surgical\_dataset\_final <- cbind(  
 surgical\_dataset\_stnd[, num\_columns],  
 surgical\_dataset\_stnd[, cat\_columns[!cat\_columns %in% columnas\_dummy]],  
 surgical\_dataset\_stnd\_dummy,  
 surgical\_dataset\_stnd[, target]  
 )

Finalmente, de cara a la elaboración de los modelos **codificamos la variable objetivo como “Yes” / “No”**, renombrando la variable como *target*:

#-- Renombramos la variable objetivo como "target"  
names(surgical\_dataset\_final)[33] <- "target"  
  
# Renombramos las columnas para adecuarlas a formulas  
names(surgical\_dataset\_final) <- make.names(names(surgical\_dataset\_final))  
  
# 1 - "Yes" ; 0 - "No"  
surgical\_dataset\_final$target <- ifelse(  
 surgical\_dataset\_final$target == 1,  
 "Yes",  
 "No"  
)  
  
surgical\_dataset\_final$target <- as.factor(surgical\_dataset\_final$target)

cat("Numero de variables finales (dummies incluidas): ", length(names(surgical\_dataset\_final)[-33]))

## Numero de variables finales (dummies incluidas): 32

# 2. Selección de variables bajo logística

## 2.1 Selección de una submuestra

Durante la selección de variables, e incluso con determinados modelos como redes neuronales o SVM, el tiempo de cómputo que requiere era demasiado elevado. Como consecuencia, se planteó utilizar, en lugar del *dataset* completo con 14.000 observaciones, **un subconjunto de menor tamaño**. Dado que la variable objetivo se encuentra desbalanceada, como pudimos comprobar en el apartado de Depuración, recurrimos al **muestreo estratificado**, el cual nos asegura la proporción de la variable objetivo. Para ello, *caret* dispone de la función *createDataPartition*, con el que obtener una submuestra estratificada. Concretamente, para el desarrollo del proyecto empleamos un subconjunto con el **40 % de las observaciones**:

#-- Muestreo estratificado  
set.seed(1234)  
partitions <- createDataPartition(surgical\_dataset\_final$target, p = 0.40, list = FALSE)  
  
surgical\_dataset\_final\_est <- surgical\_dataset\_final[partitions, ]  
  
# Mantenemos la misma proporcion en la variable objetivo  
table(surgical\_dataset\_final\_est$target)

##   
## No Yes   
## 4378 1476

No obstante, con un 40 % de los datos ¿Es suficiente? Más importante aún ¿Se obtienen los mismos resultados o similares? Es decir, si nos fijamos en la depuración final, nos encontramos con 14.635 / 32 ~ 457 observaciones por variable, aproximadamente. Sin embargo, con tan solo el 40 %, obtendríamos (14.635 \* 0.4) / 32 ~ 183 observaciones por variable. Es decir, con una menor proporción de observaciones por variable, es posible que los modelos no sean exactamente iguales con ambos conjuntos.

Por ello, durante la selección de variables **aplicaremos las mismas técnicas sobre ambos conjuntos**:

1. *stepwise AIC*
2. *stepwise BIC*
3. *RFE o Recursive Feature Elimination (sobre logística)*
4. *RFE (sobre Random Forest)*

A continuación, y en caso de obtener una selección de variables similar, aplicamos un modelo logístico en ambos casos, comprobando de este modo si los resultados son similares, incluso con tan solo el 40 % de las observaciones. De ser así, de cara al resto de modelos trabajaremos con el subconjunto.

### 2.2 Stepwise AIC

Comenzamos con la selección de variables bajo *stepwise AIC*:

#-- Stepwise AIC (100 repeticiones)  
lista.variables.aic <- steprepetidobinaria(data=surgical\_dataset,  
 vardep=target,listconti=vars,sinicio=1234,  
 sfinal=1334,porcen=0.8,criterio="AIC")  
tabla.aic <- lista.variables.aic[[1]]

## n varible\_dataset\_original variable\_subset  
## 1 1 mortality\_rsi mortality\_rsi  
## 2 2 ccsMort30Rate ccsMort30Rate  
## 3 3 bmi bmi  
## 4 4 month.8 month.8  
## 5 5 baseline\_cvd baseline\_cvd  
## 6 6 dow.0 dow.0  
## 7 7 Age Age  
## 8 8 moonphase.0 moonphase.0  
## 9 9 month.0 month.0  
## 10 10 asa\_status.0 asa\_status.0  
## 11 11 baseline\_osteoart baseline\_osteoart  
## 12 12 baseline\_charlson baseline\_charlson  
## 13 13 ahrq\_ccs ahrq\_ccs  
## 14 14 baseline\_diabetes baseline\_diabetes  
## 15 15 baseline\_cancer -  
## 16 16 baseline\_psych -  
## 17 17 asa\_status.1 -

Como podemos observar, **la diferencia entre el *dataset* original y la submuestra es de tan solo 3 variables: *baseline\_cancer*, *baseline\_psych* y *asa\_status.1***.

### 2.3 Stepwise BIC

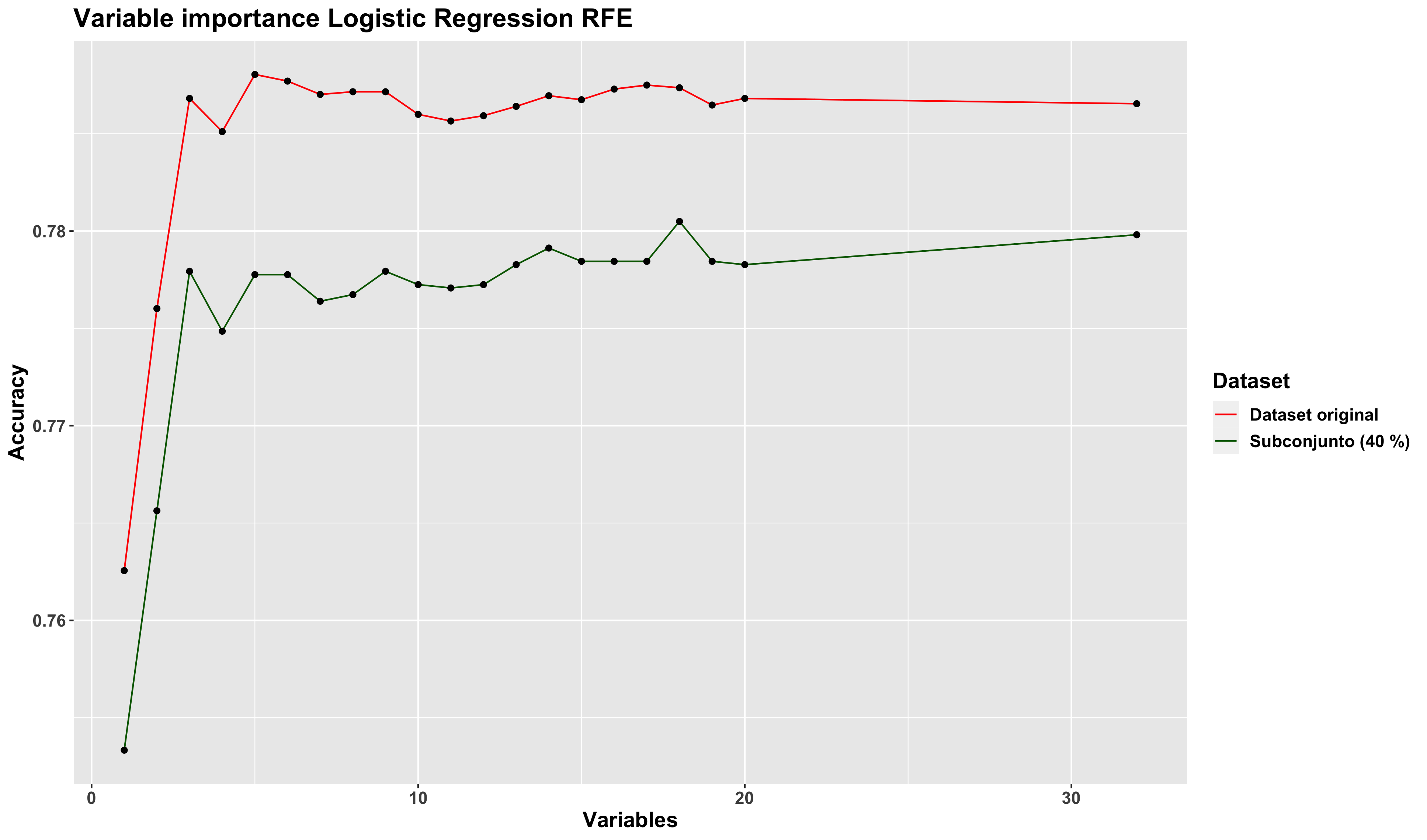
Por otro lado, analicemos la selección de variables bajo *stepwise BIC*:

## n varible\_dataset\_original variable\_subset  
## 1 1 mortality\_rsi mortality\_rsi  
## 2 2 ccsMort30Rate ccsMort30Rate  
## 3 3 bmi bmi  
## 4 4 month.8 month.8  
## 5 5 dow.0 dow.0  
## 6 6 Age Age  
## 7 7 moonphase.0 moonphase.0  
## 8 8 baseline\_osteoart baseline\_osteoart  
## 9 9 asa\_status.0 asa\_status.0  
## 10 10 baseline\_cancer -

En este caso, la diferencia en ambas seleciones es de tan solo una única variable: *baseline\_cancer*.

### 2.4 Recursive Feature Elimination (bajo logística)

No obstante, en ambas selecciones de variables el número de parámetros es bastante elevado, en especial con *stepwise AIC*. Como consecuencia, a los métodos anteriores añadimos una selección de variables mediante *Recursive Feature Elimination*, tanto bajo logística como con *Random Forest*:

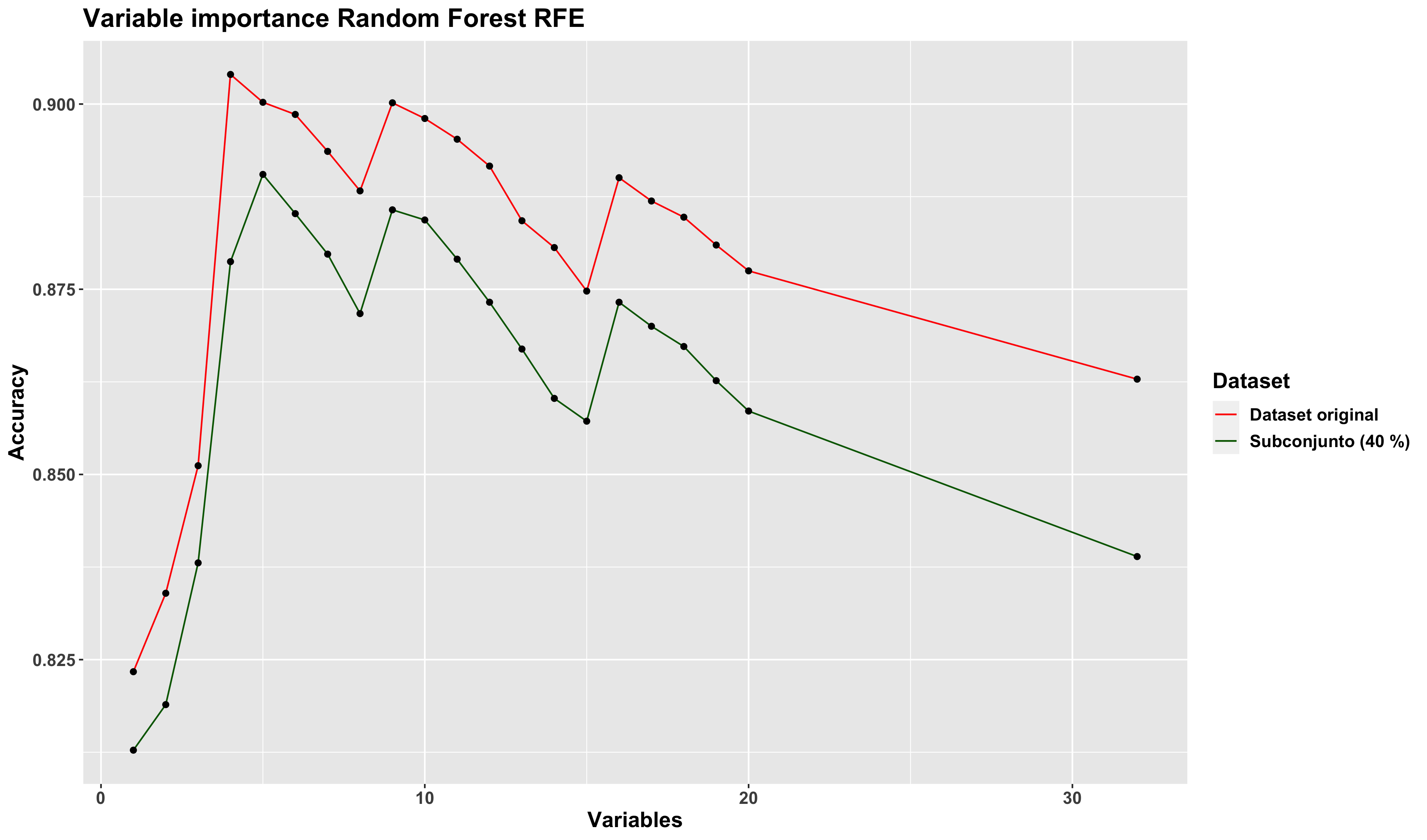


RFE Logistic Regression

Analizando el gráfico resultante, observamos que la diferencia de *Accuracy* en ambos *datasets* no es muy significativa: **mientras que con el *dataset* completo se alcanza un valor en torno a 0.78, con el subconjunto la precisión se reduce a tan solo 0.77**. De hecho, en ambos casos se obtienen muy buenos resultados con tan solo tres variables, y en ambos casos coinciden:

ccsMort30Rate, mortality\_rsi, bmi

### 2.5 Recursive Feature Elimination (bajo Random Forest)



RFE Random Forest

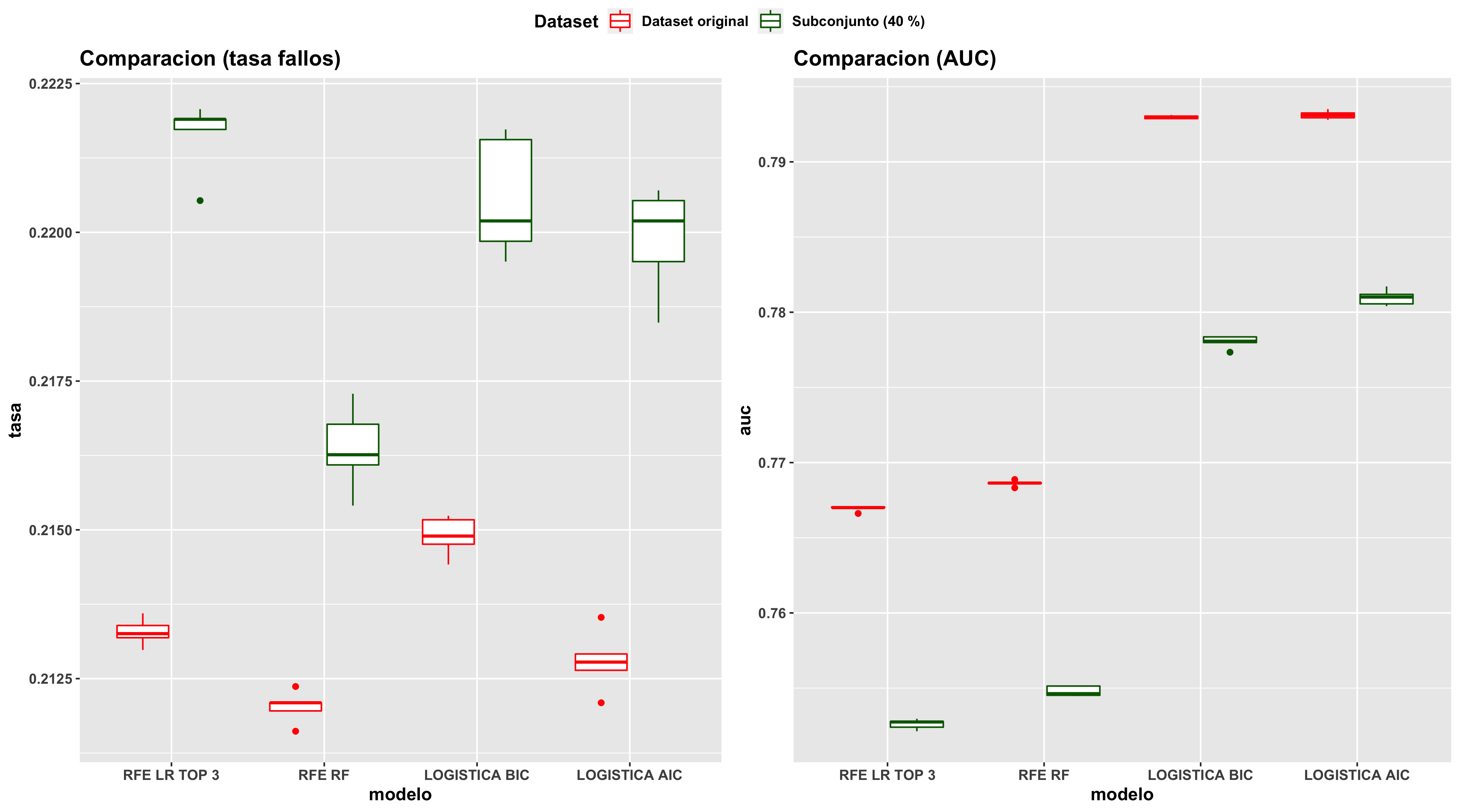
En el caso de *Random Forest*, **con 4-5 parámetros se obtienen resultados muy similares**: en el caso el *dataset* original en torno a 0.90 y 0.88 el caso del subconjunto. De hecho, en el punto de máximo *accuracy*, ambos modelos comparten las mismas variables, a excepción de *ahrq\_ccs*:

## n varible\_dataset\_original variable\_subset  
## 1 1 Age Age  
## 2 2 ccsMort30Rate ccsMort30Rate  
## 3 3 mortality\_rsi mortality\_rsi  
## 4 4 bmi bmi  
## 5 5 - ahrq\_ccs

### 2.6 Selección bajo logística

#### 2.6.1 Comparación datasets

A continuación, y una vez realizada la selección de variables tanto por AIC/BIC como por RFE, **realizamos una primera comparación, bajo logística, de la selección de variables obtenidas en los métodos anteriores, tanto con el *dataset* original como con el subconjunto**:



Comparacion bajo logística (I)

Como primera impresión, y dada la escala del eje Y, la diferencia en cuanto a tasa de fallos y AUC se refiere es muy pequeña, concretamente de 0.01, aproximadamente (0.21 en tasa de fallos y 0.79 en AUC frente a 0.22 y 0.78).

Por tanto, dado que la diferencia entre ambos *datasets* es pequeña, **de cara al resto de la práctica se trabajó con el subconjunto del 40 %**. No obstante, en los últimos apartados (y una vez tuneados los modelos), se realiza una última comparación con el fichero original, comprobando de este modo si el orden de los algoritmos se conserva.

Además, de cara a la evaluación de los modelos se ha tomado la decisión de utilizar el resto de observaciones del *dataset* original como conjunto *test*, **evaluando con ello una matriz de confusión por cada modelo**.

#### 2.6.2 Selección de los mejores sets de variables

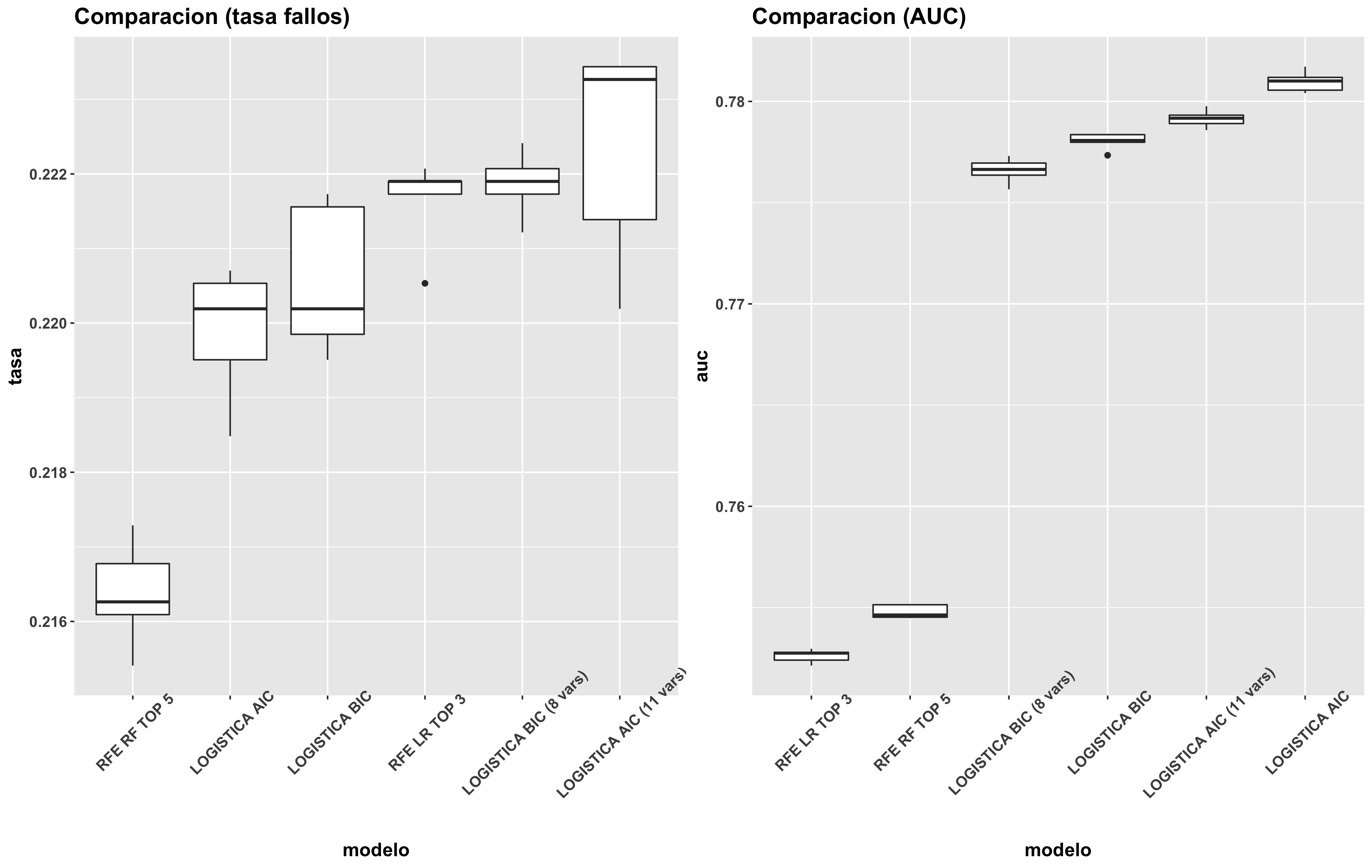
Una vez realizada la comparación con el *dataset* original, nos centramos en la selección de variables obtenida por el subconjunto, tanto por *stepwise AIC/BIC* como por RFE. En primer lugar, y remontándonos al daigrama de cajas anterior, debemos recordar la diferencia en el número de variables entre *stepwise AIC* y *BIC*: con 14 y 9 variables, respectivamente, **la tasa de fallos es prácticamente idéntica, y en relación al valor AUC, la diferencia es de tan solo unas milésimas, pues ambos modelos se sitúan en torno a 0.78**. Por tanto, el hecho de incluir demasiadas variables **no afecta en gran medida al modelo, lo cual puede traducirse en un sobreajuste con el resto de algoritmo**.

Por tanto, en ambas selecciones **probamos a eliminar las variables categóricas menos relevantes, concretamente aquellas con un menor poder predictivo**, tal y como pudimos observar en el apartado de depuración, gracias al **Valor de Información**:

1. En el caso de *stepwise AIC*, eliminamos los campos *baseline\_cvd*, *asa\_status.0* y *baseline\_diabetes* (iv: 0.04, 0.0062 y 0.0013, respectivamente).
2. En el caso de *stepwise BIC*, eliminamos el campo *asa\_status.0* (iv: 0.0062).

## n setpwise\_aic stepwise\_bic  
## 1 1 mortality\_rsi mortality\_rsi  
## 2 2 ccsMort30Rate ccsMort30Rate  
## 3 3 bmi bmi  
## 4 4 month.8 month.8  
## 5 5 dow.0 dow.0  
## 6 6 Age Age  
## 7 7 moonphase.0 moonphase.0  
## 8 8 month.0 baseline\_osteoart  
## 9 9 baseline\_osteoart -  
## 10 10 baseline\_charlson -  
## 11 11 ahrq\_ccs -

Analicemos tanto la tasa de fallos como el valor AUC:

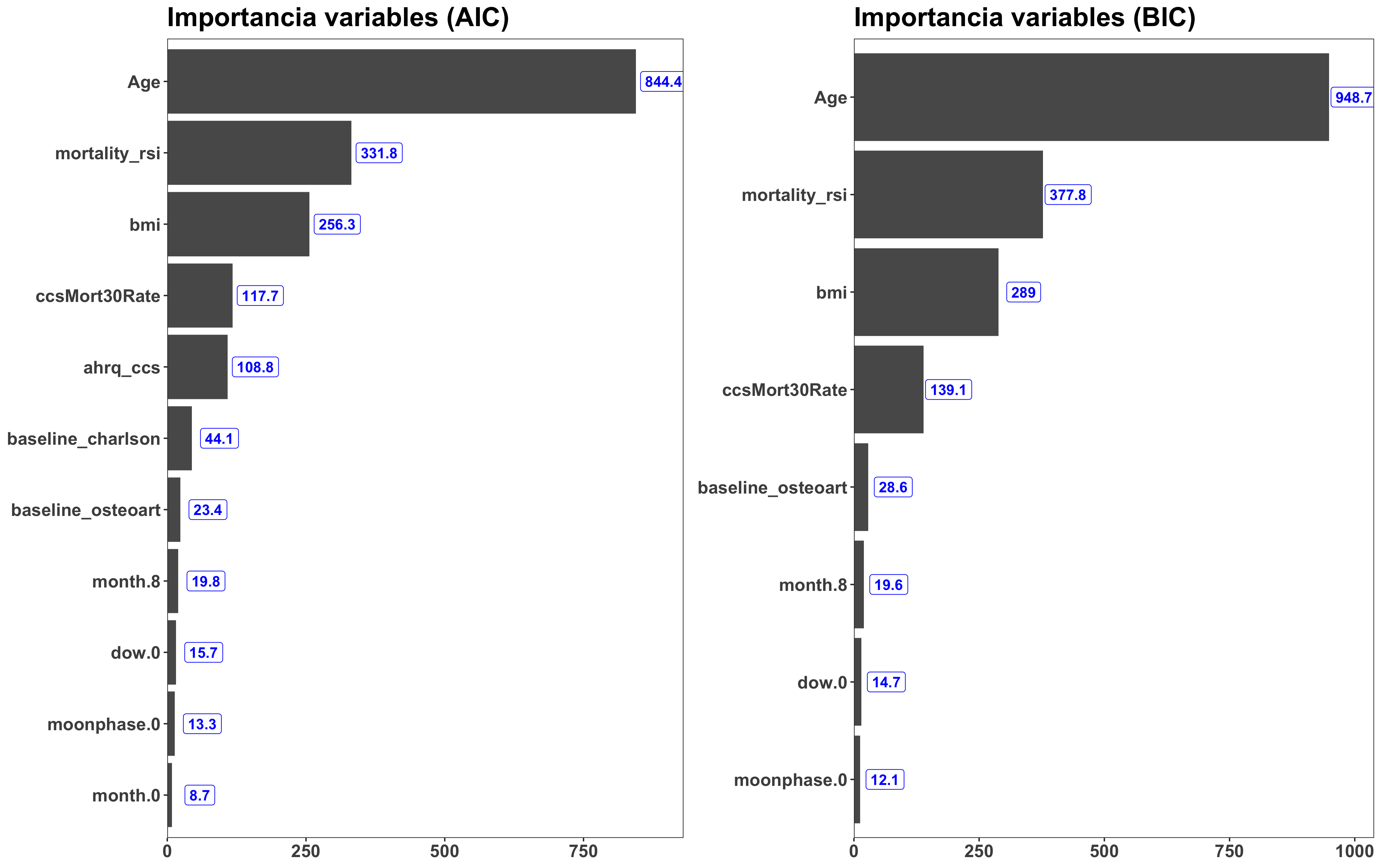


Comparacion bajo logística (II)

Incluso reduciendo el número de variables en ambos casos, aunque la tasa de fallos aumente o el valor auc disiminuya ligeramente, la diferencia no es muy significativa.

No obstante, uno de los aspectos que más ha llamada la atención ha sido la selección de variables RFE con Random Forest, donde con tan solo 5 variables la accuracy aumentaba hasta cerca del 90 %, mucho más alto que un modelo logístico. Por tanto, ¿Y si entrenamos un pequeño modelo random forest para observar la importancia de las variables tanto en *stepwise AIC* como en *BIC*?

#-- mtry (sqrt(numero de variables), por defecto)  
rf\_modelo\_bic <- train\_rf\_model(surgical\_dataset,   
 as.formula(paste0("target~", paste0(candidato.bic.2, collapse = "+"))),  
 ntree = 1000, grupos = 5, repe = 5, nodesize = 10, seed = 1234)

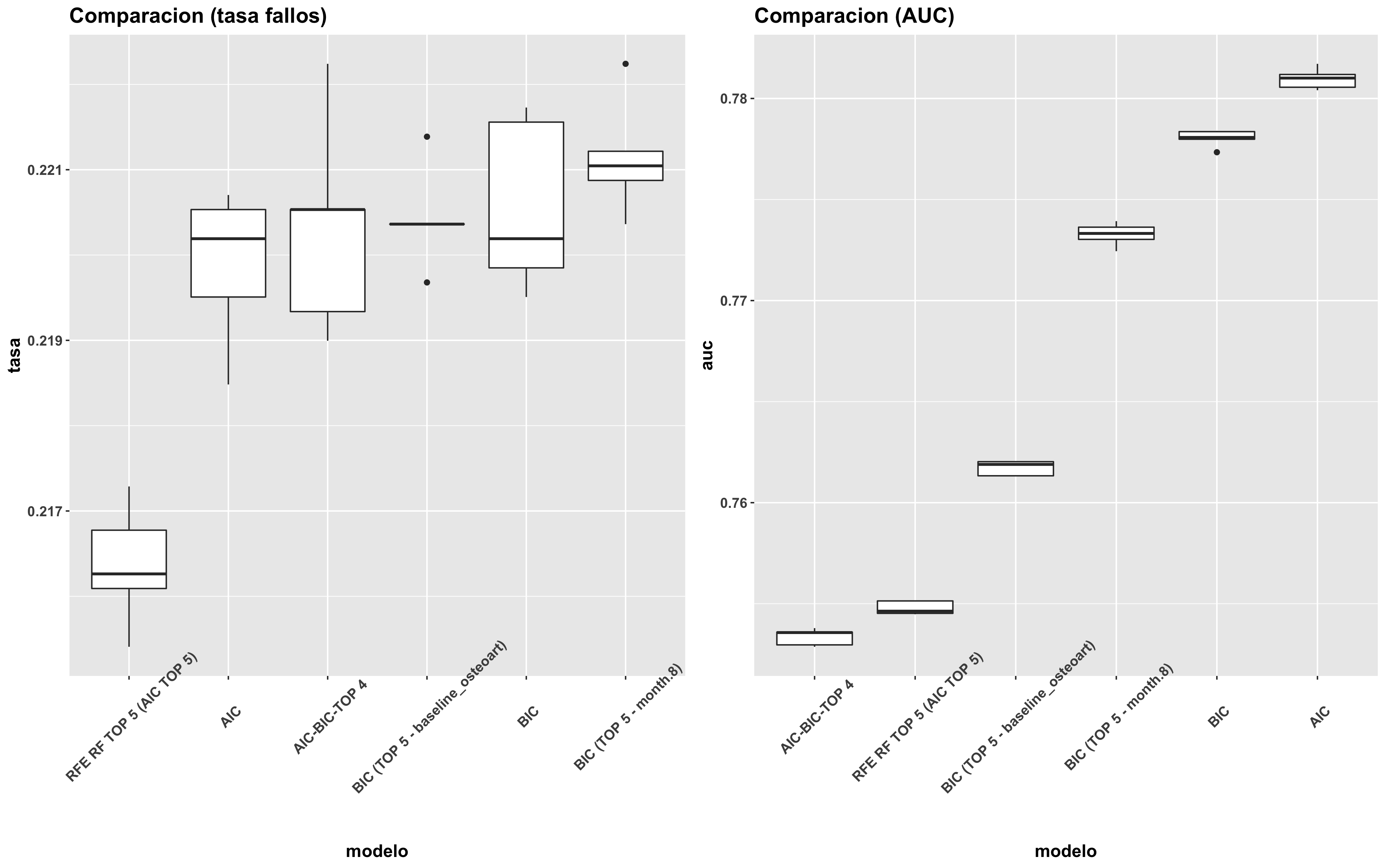


Importancia variables Random Forest (AIC y BIC)

Analizando el gráfico de importancia, detectamos:

1. Tanto en *AIC* como en *BIC*, **coinciden las primeras cuatro variables**: *Age*, *mortality\_rsi*, *bmi* y *ccsMort30Rate*.
2. Con el criterio *AIC*, **las cinco variables más importantes coinciden con las obtenidas en RFE mediante *Random Forest***: *Age*, *mortality\_rsi*, *bmi*, *ccsMort30Rate* y *ahrq\_ccs*.
3. Por el contrario, *BIC* considera de mayor importancia a variables como *baseline\_osteart* o *month.8*.

Por tanto, realizamos una última comparación bajo logística, incluyendo las cuatro mejores variables en ambos métodos *stepwise*, así como las cinco variables más importantes tanto en *AIC* y *BIC* (en este último caso, probaremos tanto con *baseline\_osteoart* como con *month.8*, dado que la importancia de ambos parámetros es muy similar):



Comparacion bajo logística (III)

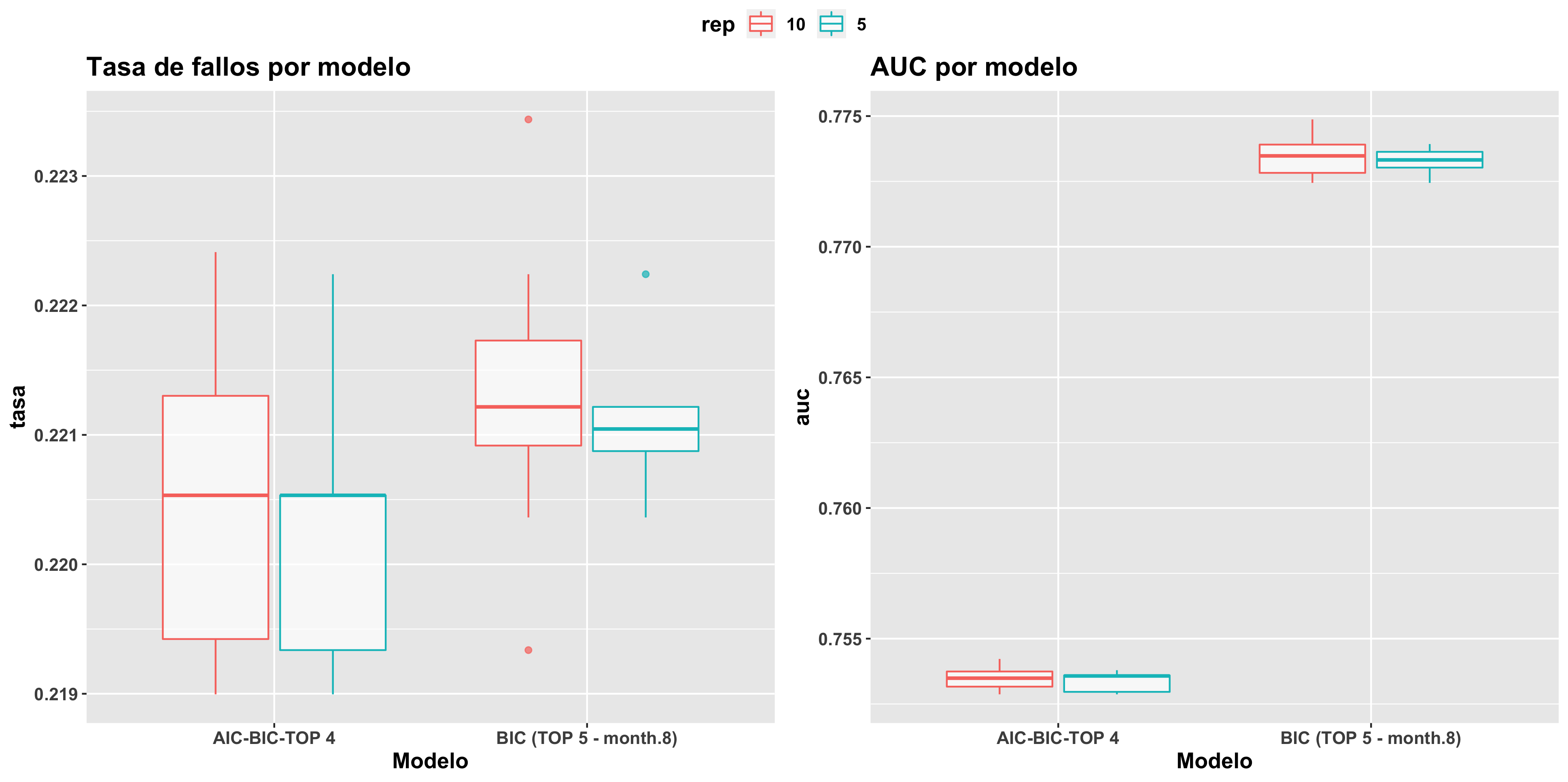
Como podemos comprobar, aunque la tasa de fallos sea similar tanto con *basline\_osteoart* como con *month.8*, el valor AUC es ligeramente superior, incluso cercano a los valores obtenidos por la selección *AIC*, *BIC* original. Por otro lado, con tan solo cuatro variables, aunque el valor AUC sea ligeramente inferior con respecto al resto de *sets*, la diferencia no es tan significativa (0.755 a 0.775, aproximadamente).

Por tanto, de cara al resto de modelos **emplamos dos modelos candidatos**:

## n modelo1 modelo2  
## 1 1 Age Age  
## 2 2 mortality\_rsi mortality\_rsi  
## 3 3 bmi bmi  
## 4 4 ccsMort30Rate ccsMort30Rate  
## 5 5 month.8 -

### 2.7 Tuneo y comparación final

Finalmente, realizamos una última comparación de ambos modelos candidatos, **aumentando la validación cruzada de 5 a 10 repeticiones**:



Comparacion final logística (5-10 rep.)

A primera vista, al aumentar el número de repeticiones, la varianza en ambos modelos es muy similar, con una ventaja en el segundo modelo en cuanto a AUC se refiere (0.775 frente a 0.755, aproximadamente). Por otro lado, si nos fijamos en las estadísticas obtenidas a partir de los datos *test*:

## modelo sensibilidad especificidad valor\_pred\_pos valor\_pred\_neg  
## 1 Modelo 1 (BIC TOP 5) 0.67730 0.81011 0.33013 0.94783  
## 2 Modelo 2 (top 4) 0.66707 0.79434 0.25194 0.95830

Observamos que la sensibilidad de ambos modelos ronda el 66-67 %, mientras que la especificidad aumenta hasta el 80 %, aproximadamente.

1. <https://www.kaggle.com/omnamahshivai/surgical-dataset-binary-classification> [↑](#footnote-ref-1)
2. <https://www.hcup-us.ahrq.gov/toolssoftware/ccs10/CCSCategoryNames(FullLabels).pdf> [↑](#footnote-ref-2)
3. <https://www.asahq.org/standards-and-guidelines/asa-physical-status-classification-system> [↑](#footnote-ref-3)
4. <https://dicciomed.usal.es/palabra/osteoartritis> [↑](#footnote-ref-4)
5. <https://docs.tibco.com/pub/sfire-dsc/6.5.0/doc/html/TIB_sfire-dsc_user-guide/GUID-07A78308-525A-406F-8221-9281F4E9D7CF.html> [↑](#footnote-ref-5)