Ingreso de los hogares: trees y random forests

Este es el código para preparar los datos, donde tomamos unas cuantas variables de la encuesta Enigh 2016. **En este caso ignoraremos el hecho de que estos datos resultan de un diseño complejo de muestra**. En este caso, convendría diseñar un esquema de validación apropiado (extrayendo unidades primarias de muestreo completas, por ejemplo), y usar los factores de expansión de la muestra.

```
library(readr)
library(dplyr)
concentrado <- read_csv('https://raw.githubusercontent.com/felipegonzalez/aprendizaje-maquina-2017/maste
hogares <- read_csv('https://raw.githubusercontent.com/felipegonzalez/aprendizaje-maquina-2017/master/da
problems(concentrado)
head(concentrado)
names(concentrado)
concen_2 <- left_join(concentrado, hogares)
names(concen_2)[1] <- "folioviv"
datos <- concen_2 %>% select(folioviv, foliohog, tam_loc, educa_jefe,
                             celular, tv_paga, conex_inte, num_auto, num_tosta, num_lavad,
                             num_compu, ing_cor, factor) %>%
                      mutate(tam_loc = recode(tam_loc, `1`='100 mil+',`2`='15mil-100mil',
                                              `3`='2.5mil-15mil',`4`='Menos de 2.5 mil')) %>%
                      mutate(celular = recode(celular, `1`='Si', `2`='No')) %>%
                      mutate(tv_paga = recode(tv_paga, `1`='Si', `2`='No')) %>%
                      mutate(celular = recode(celular, `1`='Si', `2`='No')) %>%
                      mutate(conex_inte = recode(conex_inte, `1`='Si', `2`='No'))

write_csv(datos, path ='./vars_enigh_2016.csv')
```

Datos

Buscamos predecir el ingreso corriente trimestral de los hogares a partir de algunas de sus características, el tamaño de la localidad, y la educación del jefe(a) del hogar. Para este ejemplo usamos una muestra:

```
set.seed(293)
library(tidyverse)
```

```
## Loading tidyverse: ggplot2
## Loading tidyverse: tibble
## Loading tidyverse: tidyr
## Loading tidyverse: readr
## Loading tidyverse: purrr
## Loading tidyverse: dplyr
```

```
## Warning: package 'dplyr' was built under R version 3.4.2
```

```
## Conflicts with tidy packages ----------------------------------------------
```

```
## filter(): dplyr, stats
## lag():    dplyr, stats
```

```
datos <- read_csv(file = './vars_enigh_2016.csv')
```

```
## Parsed with column specification:
## cols(
##   folioviv = col_character(),
```

```
##   foliohog = col_integer(),
##   tam_loc = col_character(),
##   educa_jefe = col_character(),
##   celular = col_character(),
##   tv_paga = col_character(),
##   conex_inte = col_character(),
##   num_auto = col_integer(),
##   num_tosta = col_integer(),
##   num_lavad = col_integer(),
##   num_compu = col_integer(),
##   ing_cor = col_double(),
##   factor = col_integer()
## )
```

```
datos <- sample_n(datos, 10000)
```

Vamos a predecir el log del ingreso:

```
datos$ingreso_log <- log(1 + datos$ing_cor)
#escala log
quantile(datos$ingreso_log, probs = seq(0,1,0.1))
```

```
##        0%       10%       20%       30%       40%       50%       60%
##  0.000000  9.337049  9.689651  9.925379 10.125587 10.309623 10.496600
##       70%       80%       90%      100%
## 10.704291 10.957557 11.293807 15.263128
```

```
#escala original
exp(quantile(datos$ingreso_log, probs = seq(0,1,0.1)))
```

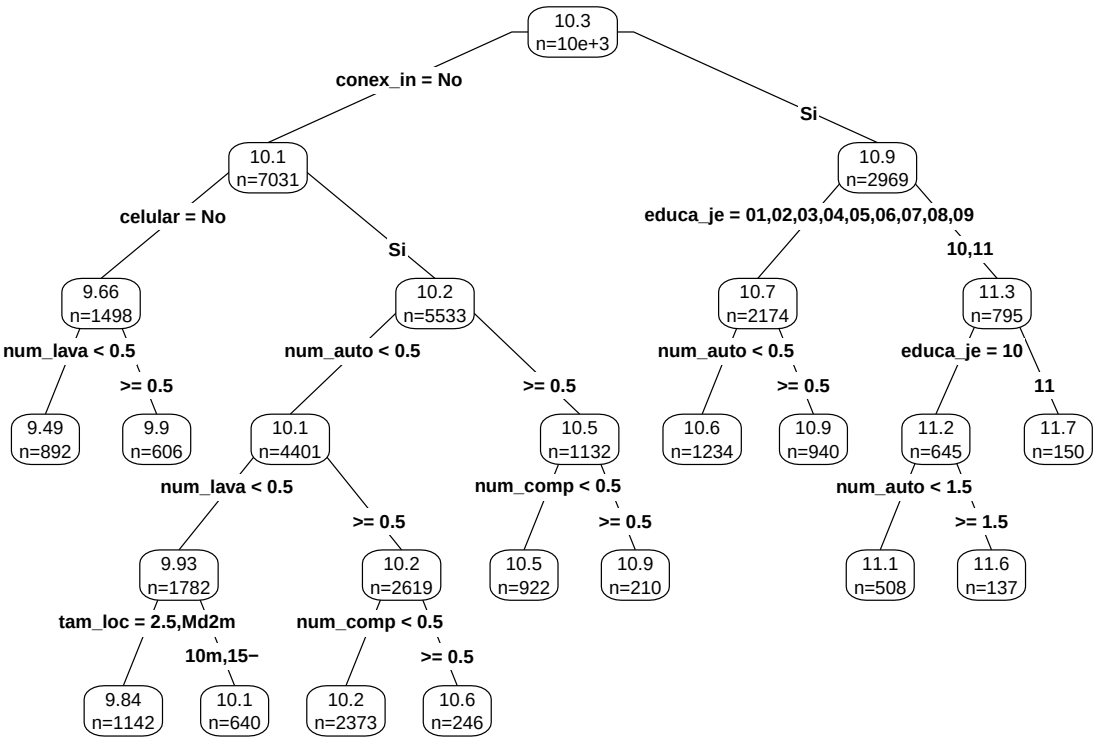
```
##         0%        10%        20%        30%        40%        50%
##       1.00   11350.86   16149.60   20442.66   24973.90   30020.10
##        60%        70%        80%        90%       100%
##   36192.25   44546.57   57386.08   80322.63 4252971.16
```

Árboles

Corre el siguiente código

```
library(rpart)
library(rpart.plot)

arbol_grande <- rpart(ingreso_log ~ tam_loc + educa_jefe +
      celular+ conex_inte + num_auto+ num_tosta+ num_lavad+ num_compu + factor,
      data= datos, cp=0)
prp(prune(arbol_grande, cp=0.004), type=4, extra=1, digits=3)
```

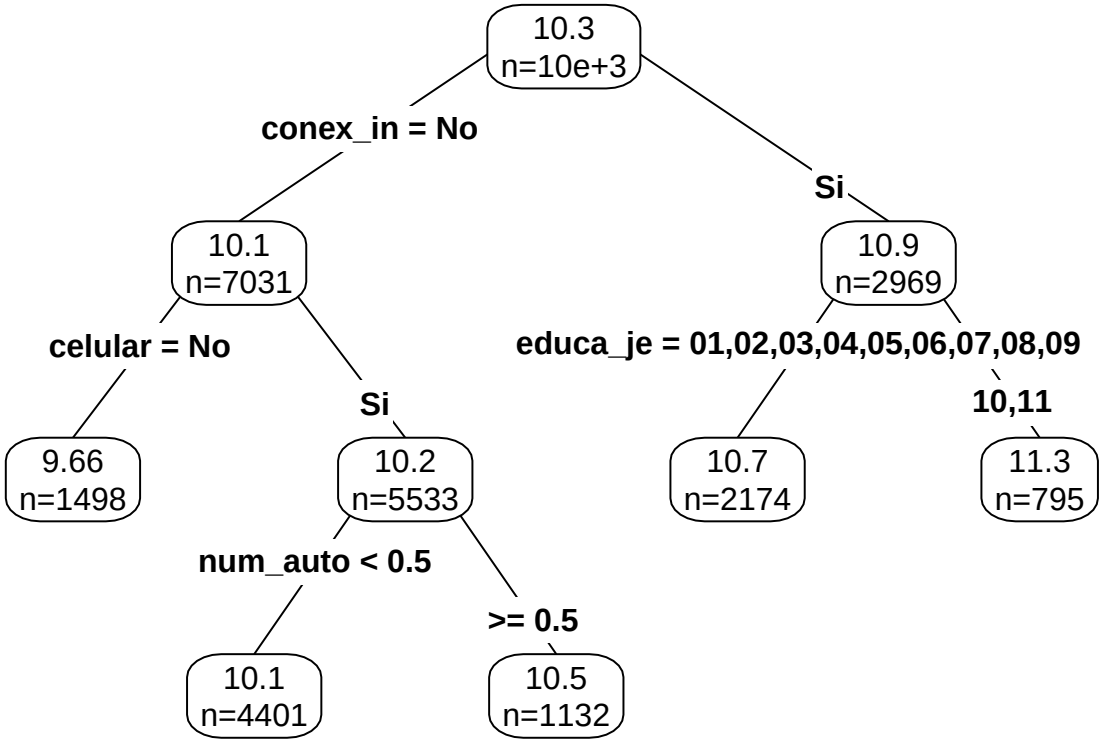


1. ¿Qué significa la información que hay en cada nodo? Nota: puedes interpretar diferencias de log ingreso rápidamente si tomas en cuenta que una diferencia en la escala logarítmica (para diferencias más chicas) es aproximadamente cambio porcentual en ingreso. Por ejemplo la diferencia de ingreso en escala log de 4.7 a 4.9 es aproximadamente un incremento de 20%.

Respuesta: Cada nodo representa un nivel de ingreso, en escala lograrítmica. Por ejemplo, el primer nodo tiene un nivel de ingreso de 29,732.62 pesos, si cuenta con conexion a internet, el sueldo se incrementa en un 82% para llegar a 54,176.36 pesos, de lo contrario el sueldo disminuye en un 18% para quedar en 24,343.01 pesos.

2. Poda el árbol para mostrar solamente un árbol con 5 nodos terminales. Evalúa el error de entrenamiento para este árbol.

```
arbol_5<-prune(arbol_grande, cp=0.02)
prp(arbol_5, type=4, extra=1, digits=3)
```



El error de entrenamiento es el siguiente:

```
mean((predict(arbol_5, data=datos)-datos$ingreso_log)^2)
```

```
## [1] 0.419654
```

Bosques aleatorios

1. Usa un bosque aleatorio para predecir el log ingreso. Prueba algunos valores de m (mtry) y escoge un modelo final usando el error out-of-bag. Grafica cómo evoluciona la estimación OOB del error conforme aumenta el número de árboles.

Con $m = 6$:

```
library(randomForest)
#utiliza estos datos, que tienen las variables categóricas convertidas a factores.
datos_df <- data.frame(unclass(datos))
bosque_enigh <-randomForest(ingreso_log ~ tam_loc + educa_jefe +
      celular+ conex_inte + num_auto+ num_tosta+ num_lavad+ num_compu + factor, data = datos_df,
                          ntree = 1500, mtry = 6, importance=TRUE)
```

Call: randomForest(formula = ingreso_log ~ tam_loc + educa_jefe + celular + conex_inte + num_auto + num_tosta + num_lavad + num_compu + factor, data = datos_df, ntree = 1500, mtry = 6, importance = TRUE) Type of random forest: regression Number of trees: 1500 No. of variables tried at each split: 6

Mean of squared residuals: 0.3895266 % Var explained: 37.41

Con $m = 4$:

```
bosque_enigh_4 <-randomForest(ingreso_log ~ tam_loc + educa_jefe +
      celular+ conex_inte + num_auto+ num_tosta+ num_lavad+ num_compu + factor, data = datos_df,
                          ntree = 1500, mtry = 4, importance=TRUE)
```

Call: randomForest(formula = ingreso_log ~ tam_loc + educa_jefe + celular + conex_inte + num_auto + num_tosta + num_lavad + num_compu + factor, data = datos_df, ntree = 1500, mtry = 4, importance = TRUE) Type of random forest: regression Number of trees: 1500 No. of variables tried at each split: 4

Mean of squared residuals: 0.3682859 % Var explained: 40.82

Con $m = 2$

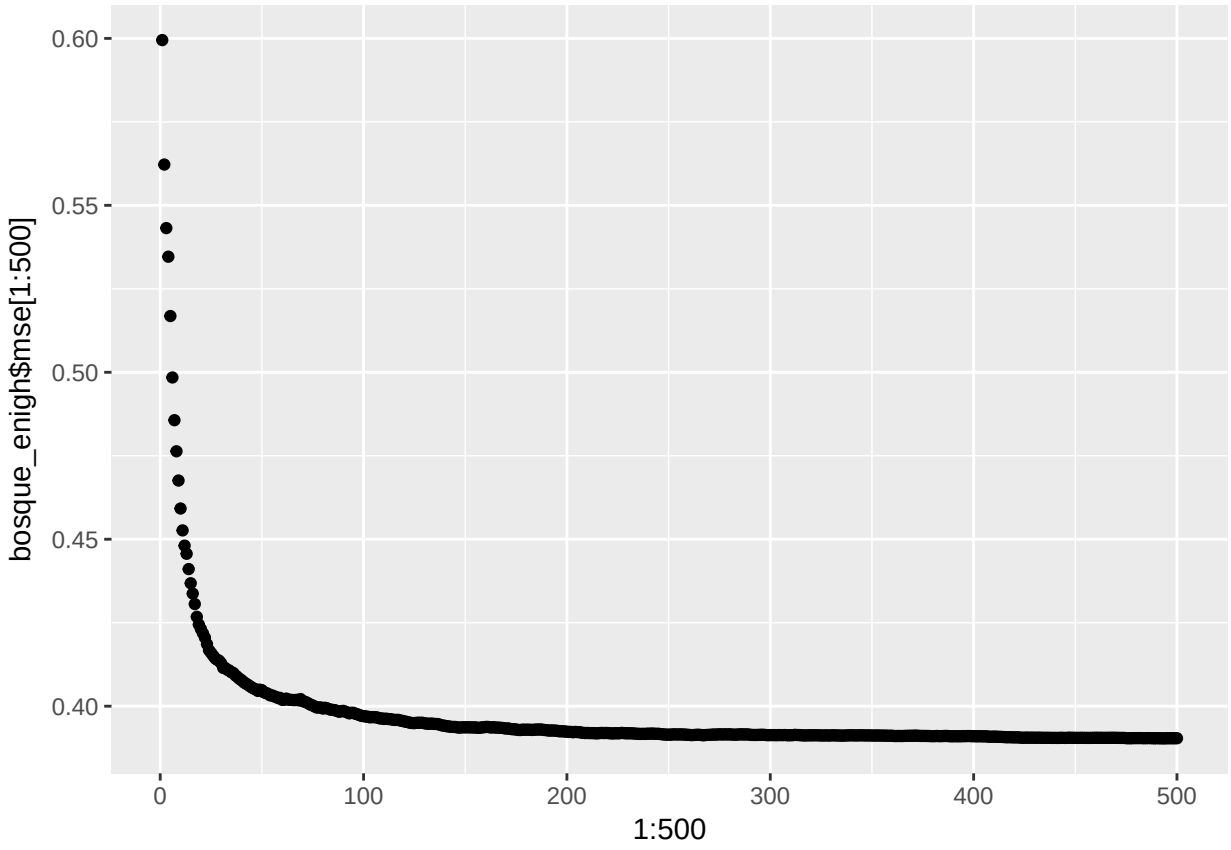
```
bosque_enigh_2 <-randomForest(ingreso_log ~ tam_loc + educa_jefe +
      celular+ conex_inte + num_auto+ num_tosta+ num_lavad+ num_compu + factor, data = datos_df,
                            ntree = 1500, mtry = 4, importance=TRUE)
```

Call: randomForest(formula = ingreso_log ~ tam_loc + educa_jefe + celular + conex_inte + num_auto + num_tosta + num_lavad + num_compu + factor, data = datos_df, ntree = 1500, mtry = 4, importance = TRUE) Type of random forest: regression Number of trees: 1500 No. of variables tried at each split: 4

Mean of squared residuals: 0.3681733 % Var explained: 40.84

Graficamos los errores cuadráticos medios con respecto conforme se incrementa el número de arboles:

```
library(ggplot2)
ggplot()+geom_point(aes(x=1:500,y=bosque_enigh$mse[1:500]))
```



Para el modelo final escogemos uno con 500 arboles pues vemos que con ese valor ya se obtienen un suma de errores al cuadrado muy parecida a la que se obtiene con 1500 arboles.

2. Examina las importancias de las variables. ¿Cuáles son las 3 variables más importantes?

```
bosque_enigh$importance
```

```
##                %IncMSE IncNodePurity
## tam_loc    0.047893460      267.1733
```

```
## educa_jefe 0.045272332      615.3331
## celular    0.029053120      303.2647
## conex_inte 0.049120251      969.7195
## num_auto   0.065688581      421.9776
## num_tosta  0.006095989      104.4811
## num_lavad  0.031236512      263.6140
## num_compu  0.029168493      484.0319
## factor     0.039680084     1615.5102
```

Las 3 variables más importantes son: tam_loc, educa_jefe y celular.

3. Incluye una o dos variables adicionales que crees que puedan tener importancia alta. ¿En qué lugar aparecen?

Incorporamos las variables `tot_integ` y `sexo_jefe`:

```
# library(tidyverse)
# datos2 <- concen_2 %>% select(folioviv, foliohog, tam_loc, educa_jefe,
#                               celular, tv_paga, conex_inte, num_auto, num_tosta, num_lavad,
#                               num_compu, ing_cor, factor,tot_integ,sexo_jefe) %>%
#                       mutate(tam_loc = recode(tam_loc, `1`='100 mil+',`2`='15mil-100mil',
#                                               `3`='2.5mil-15mil',`4`='Menos de 2.5 mil')) %>%
#                       mutate(celular = recode(celular, `1`='Si', `2`='No')) %>%
#                       mutate(tv_paga = recode(tv_paga, `1`='Si', `2`='No')) %>%
#                       mutate(celular = recode(celular, `1`='Si', `2`='No')) %>%
#                       mutate(conex_inte = recode(conex_inte, `1`='Si', `2`='No'))
# write_csv(datos2, path ='./vars_enigh_2016_2.csv')



set.seed(293)
datos2 <- read_csv(file = './vars_enigh_2016_2.csv')
```

```
## Parsed with column specification:
## cols(
##   folioviv = col_character(),
##   foliohog = col_integer(),
##   tam_loc = col_character(),
##   educa_jefe = col_character(),
##   celular = col_character(),
##   tv_paga = col_character(),
##   conex_inte = col_character(),
##   num_auto = col_integer(),
##   num_tosta = col_integer(),
##   num_lavad = col_integer(),
##   num_compu = col_integer(),
##   ing_cor = col_double(),
##   factor = col_integer(),
##   tot_integ = col_integer(),
##   sexo_jefe = col_integer()
## )
```

```
datos2 <- sample_n(datos2, 10000)
datos2$ingreso_log <- log(1 + datos2$ing_cor)
datos_df_2 <- data.frame(unclass(datos2))

bosque_enigh_2var <-randomForest(ingreso_log ~ tam_loc + educa_jefe +
```

```
      celular+ conex_inte + num_auto+ num_tosta+ num_lavad+ num_compu + factor + tot_integ + sexo_jefe,
                           ntree = 500, mtry = 6, importance=TRUE)

bosque_enigh_2var$importance
```

```
##                %IncMSE IncNodePurity
## tam_loc    0.045200070      300.2831
## educa_jefe 0.054926993      689.3293
## celular    0.014612024      267.4470
## conex_inte 0.046254071      829.7872
## num_auto   0.054412190      459.3356
## num_tosta  0.006229876      106.5317
## num_lavad  0.026123859      266.0655
## num_compu  0.025875835      548.7096
## factor     0.032680961     1270.5845
## tot_integ  0.071954196      578.9117
## sexo_jefe  0.003959411      118.4187
```

Las 2 variables que se incluyeron aparecieron como las 2 últimas.