Poverty Mapping in the Age of Machine Learning

Marco Espinoza

02/05/2024

De acuerdo a *Poverty Mapping in the Age of Machine Learning* se prueban tres modelos de XGBoosting machines, los cuales son:

* Modelo XGBoost Machine con covariables de geoubicación únicamente.
* Modelo XGBoost Machine con covariables censales únicamente.
* Modelo XGBoost Machine con todas las covariables.

rm(list = ls())

# Librerias

x <- c('tidyverse','xgboost','gridExtra')  
lapply(x, require, character.only = TRUE)

## Loading required package: tidyverse

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.4 ✔ readr 2.1.5  
## ✔ forcats 1.0.0 ✔ stringr 1.5.1  
## ✔ ggplot2 3.5.0 ✔ tibble 3.2.1  
## ✔ lubridate 1.9.3 ✔ tidyr 1.3.1  
## ✔ purrr 1.0.2   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors  
## Loading required package: xgboost  
##   
##   
## Attaching package: 'xgboost'  
##   
##   
## The following object is masked from 'package:dplyr':  
##   
## slice  
##   
##   
## Loading required package: gridExtra  
##   
##   
## Attaching package: 'gridExtra'  
##   
##   
## The following object is masked from 'package:dplyr':  
##   
## combine

## [[1]]  
## [1] TRUE  
##   
## [[2]]  
## [1] TRUE  
##   
## [[3]]  
## [1] TRUE

set.seed(1998)  
  
options(scipen=999)

# Datos

encuesta\_mrp <- readRDS("C:/Users/mespinoza/Documents/SAExML/R/ingreso/datos/encuesta\_mrp1.rds")

# Encuesta quitamos ingreso, lp, li y fep  
encuesta\_mrp <- encuesta\_mrp %>%   
 select(  
 # -X2016\_crops.coverfraction, -X2016\_urban.coverfraction,  
 # -X2016\_gHM, -accessibility, -accessibility\_walking\_only,  
 -area1, -sexo2,-edad2,-edad3, -edad4,-edad5,-anoest2,  
 -anoest3, -anoest4, -discapacidad1,-etnia1,-lp,-li,-fep,-pobreza) %>%   
 mutate(  
 dam = recode(dam,  
 "01" = 1,  
 "02" = 2,  
 "03" = 3,  
 "04" = 4,  
 "05" = 5,  
 "06" = 6)) %>%  
 mutate\_if(is.character, as.factor) %>%   
 mutate(dam = as.factor(dam))  
  
summary(encuesta\_mrp)

## dam area ingreso sexo anoest edad   
## 1:17756 0:10271 Min. : 0 1:15397 1 : 1031 1:6139   
## 2: 3110 1:21692 1st Qu.: 96879 2:16566 2 :11846 2:7387   
## 3: 2259 Median : 170417 3 :11655 3:6773   
## 4: 2807 Mean : 275676 4 : 4949 4:7676   
## 5: 3272 3rd Qu.: 320000 98: 2482 5:3988   
## 6: 2759 Max. :16630000   
## discapacidad F182013\_stable\_lights X2016\_crops.coverfraction  
## 0:31354 Min. : 26906 Min. : 69431   
## 1: 609 1st Qu.: 35356 1st Qu.:172050   
## Median :104975 Median :172050   
## Mean : 74817 Mean :196693   
## 3rd Qu.:104975 3rd Qu.:232074   
## Max. :104975 Max. :389867   
## X2016\_urban.coverfraction X2016\_gHM accessibility   
## Min. : 4618 Min. :1434 Min. : 129933   
## 1st Qu.: 7403 1st Qu.:2766 1st Qu.: 462765   
## Median :51434 Median :3335 Median : 670616   
## Mean :31641 Mean :3032 Mean : 681345   
## 3rd Qu.:51434 3rd Qu.:3335 3rd Qu.: 670616   
## Max. :51434 Max. :3335 Max. :1668339   
## accessibility\_walking\_only tiene\_alcantarillado tiene\_electricidad  
## Min. :1056441 Min. :0.7286 Min. :0.1228   
## 1st Qu.:1558951 1st Qu.:0.7286 1st Qu.:0.1228   
## Median :1558951 Median :0.7286 Median :0.1228   
## Mean :1827739 Mean :0.8271 Mean :0.2349   
## 3rd Qu.:1816410 3rd Qu.:0.9467 3rd Qu.:0.2506   
## Max. :3916513 Max. :1.0000 Max. :0.7202   
## tiene\_acueducto tiene\_gas eliminar\_basura tiene\_internet   
## Min. :0.09656 Min. :0.08182 Min. :0.1198 Min. :0.6086   
## 1st Qu.:0.09656 1st Qu.:0.08182 1st Qu.:0.1198 1st Qu.:0.6086   
## Median :0.09656 Median :0.08182 Median :0.1198 Median :0.6086   
## Mean :0.17940 Mean :0.12107 Mean :0.2733 Mean :0.7122   
## 3rd Qu.:0.27274 3rd Qu.:0.16223 3rd Qu.:0.4519 3rd Qu.:0.8464   
## Max. :0.31354 Max. :0.21130 Max. :0.5388 Max. :0.8569   
## piso\_tierra material\_paredes material\_techo rezago\_escolar   
## Min. :0.006272 Min. :0.09714 Min. :0.08633 Min. :0.1065   
## 1st Qu.:0.006272 1st Qu.:0.09714 1st Qu.:0.08633 1st Qu.:0.1250   
## Median :0.006272 Median :0.09714 Median :0.08633 Median :0.2628   
## Mean :0.015931 Mean :0.13167 Mean :0.12174 Mean :0.2044   
## 3rd Qu.:0.025423 3rd Qu.:0.16477 3rd Qu.:0.15997 3rd Qu.:0.2628   
## Max. :0.037085 Max. :0.22252 Max. :0.20493 Max. :0.2628   
## alfabeta hacinamiento tasa\_desocupacion   
## Min. :0.01711 Min. :0.01718 Min. :0.001833   
## 1st Qu.:0.01711 1st Qu.:0.01718 1st Qu.:0.002801   
## Median :0.01711 Median :0.01718 Median :0.022012   
## Mean :0.02863 Mean :0.02584 Mean :0.013330   
## 3rd Qu.:0.04238 3rd Qu.:0.03910 3rd Qu.:0.022012   
## Max. :0.06029 Max. :0.04218 Max. :0.022012

# Modelo con todas las covariables

## Validación cruzada

# Eta: Magnitud de las correciones que son hechas por cada predictor nuevo (learning rate)  
# max depth: profundidad de los árboles  
grid <- expand\_grid(max\_depth = seq(3, 6, 1), eta = seq(.2, .35, .01))  
y = encuesta\_mrp[,3]  
encuesta\_mrp\_m <- xgb.DMatrix(data = as.matrix(sapply(encuesta\_mrp[,-3], as.numeric)), label = as.matrix(sapply(y, as.numeric)))  
  
  
xgb\_train\_rmse <- numeric(nrow(grid))  
xgb\_test\_rmse <- numeric(nrow(grid))  
  
  
for(i in 1:nrow(grid)){  
 xgb\_untuned = xgb.cv(  
 data = encuesta\_mrp\_m,  
 params = list(  
 objective = "reg:squarederror",  
 eta = grid$eta[i],  
 max\_depth = grid$max\_depth[i]  
 ),  
 nrounds = 100,  
 early\_stopping\_rounds = 3, # training with a validation set will stop if the performance does not improve for k rounds (3)  
 nfold = 5  
 )  
   
 xgb\_train\_rmse[i] <-  
 xgb\_untuned$evaluation\_log$train\_rmse\_mean[xgb\_untuned$best\_iteration]  
 xgb\_test\_rmse[i] <-  
 xgb\_untuned$evaluation\_log$test\_rmse\_mean[xgb\_untuned$best\_iteration]  
   
 cat(i, "\n")  
}

xgb\_train\_rmse <- xgb\_train\_rmse %>%   
 tibble() %>%   
 mutate(simulacion = seq(1:64))  
  
xgb\_test\_rmse <- xgb\_test\_rmse %>%   
 tibble() %>%   
 mutate(simulacion = seq(1:64))  
  
xgb\_train\_rmse %>% ggplot(aes(x = simulacion, y = .)) + geom\_line()  
  
#Guardamos las iteraciones para el primer modelo de la validación cruzada  
saveRDS(xgb\_train\_rmse, file = "ingreso/output/xgb\_train\_rmse\_all.rds")  
  
  
grid[which.min(xgb\_train\_rmse$.), ]

vamos a ajustar el modelo con los siguientes hiperparámetros:

* max\_depth = 6
* eta = 0.30

modelo <- xgboost(params = list(  
 objective = "reg:squarederror",  
 eta = 0.30,  
 max\_depth = 6),  
 nrounds = 1000,  
 # early\_stopping\_rounds = 3,  
 data = encuesta\_mrp\_m  
)  
  
ggplot(modelo$evaluation\_log, aes(  
 x = iter,  
 y = train\_rmse  
)) + geom\_line() + geom\_vline(xintercept = 71, col = "red")  
  
  
modelo$evaluation\_log[which.min(modelo$evaluation\_log$train\_rmse), ]  
  
modelo$evaluation\_log[71, ]  
  
saveRDS(modelo, file = "ingreso/output/modelo.rds")

* Podemos observar que en la creación de los árboles, luego de 71 iteraciones el Error Cuadrático Medio se estabiliza y tiene un cambio mínimo con respecto a la última iteración.

## importancia de variables

importancia <- xgb.importance(  
 feature\_names = names(encuesta\_mrp[,-3]),  
 model = modelo  
)  
  
p1 <- xgb.ggplot.importance(importancia)

# Modelo con solo covariables geo espaciales

rm(list = ls())

## Datos

encuesta\_mrp <- readRDS("C:/Users/mespinoza/Documents/SAExML/R/ingreso/datos/encuesta\_mrp1.rds")

# Encuesta quitamos ingreso, lp, li y fep  
encuesta\_mrp <- encuesta\_mrp %>%   
 select(  
 F182013\_stable\_lights, X2016\_crops.coverfraction, X2016\_urban.coverfraction, X2016\_gHM, accessibility,   
 accessibility\_walking\_only, ingreso  
 )  
  
summary(encuesta\_mrp)

## Validación cruzada

# Eta: Magnitud de las correciones que son hechas por cada predictor nuevo (learning rate)  
# max depth: profundidad de los árboles  
grid <- expand\_grid(max\_depth = seq(3, 6, 1), eta = seq(.2, .35, .01))  
y = encuesta\_mrp[,7]  
encuesta\_mrp\_m <- xgb.DMatrix(data = as.matrix(sapply(encuesta\_mrp[,-7], as.numeric)), label = as.matrix(sapply(y, as.numeric)))  
  
  
xgb\_train\_rmse <- numeric(nrow(grid))  
xgb\_test\_rmse <- numeric(nrow(grid))  
  
  
for(i in 1:nrow(grid)){  
 xgb\_untuned = xgb.cv(  
 data = encuesta\_mrp\_m,  
 params = list(  
 objective = "reg:squarederror",  
 eta = grid$eta[i],  
 max\_depth = grid$max\_depth[i]  
 ),  
 nrounds = 100,  
 early\_stopping\_rounds = 3, # training with a validation set will stop if the performance does not improve for k rounds (3)  
 nfold = 5  
 )  
   
 xgb\_train\_rmse[i] <-  
 xgb\_untuned$evaluation\_log$train\_rmse\_mean[xgb\_untuned$best\_iteration]  
 xgb\_test\_rmse[i] <-  
 xgb\_untuned$evaluation\_log$test\_rmse\_mean[xgb\_untuned$best\_iteration]  
   
 cat(i, "\n")  
}

xgb\_train\_rmse <- xgb\_train\_rmse %>%   
 tibble() %>%   
 mutate(simulacion = seq(1:64))  
  
xgb\_test\_rmse <- xgb\_test\_rmse %>%   
 tibble() %>%   
 mutate(simulacion = seq(1:64))  
  
xgb\_train\_rmse %>% ggplot(aes(x = simulacion, y = .)) + geom\_line()  
  
saveRDS(xgb\_train\_rmse, file = "ingreso/output/xgb\_train\_rmse\_geo.rds")  
  
grid[which.min(xgb\_train\_rmse$.), ]

max depth = 5 eta = 0.23

modelo1 <- xgboost(params = list(  
 objective = "reg:squarederror",  
 eta = 0.23,  
 max\_depth = 5),  
 nrounds = 1000,  
 # early\_stopping\_rounds = 3,  
 data = encuesta\_mrp\_m  
)  
  
saveRDS(modelo1, file = "ingreso/output/modelo1.rds")  
  
ggplot(modelo1$evaluation\_log, aes(  
 x = iter,  
 y = train\_rmse  
)) + geom\_line() + geom\_vline(xintercept = 13.5, col = "red")  
  
  
modelo1$evaluation\_log[which.min(modelo1$evaluation\_log$train\_rmse), ]  
  
modelo1$evaluation\_log[13, ]  
modelo1$evaluation\_log[14, ]

* Para este caso podemos observar que luego de 34 iteraciones el modelo se detuvo debido a que el modelo no mejoró luego de 34 por lo que ahí se detuvo.
* Por otro lado, se puede observar que luego de 14 iteraciones el modelo estabiliza el Error Cuadrático Medio.

## importancia de variables

importancia <- xgb.importance(  
 feature\_names = names(encuesta\_mrp[,-7]),  
 model = modelo1  
)  
  
p2 <- xgb.ggplot.importance(importancia)

# Modelo con solo las variables censales

## Datos

rm(list = ls())

encuesta\_mrp <- readRDS("C:/Users/mespinoza/Documents/SAExML/R/ingreso/datos/encuesta\_mrp1.rds")

# Encuesta quitamos ingreso, lp, li y fep  
encuesta\_mrp <- encuesta\_mrp %>%   
 select(  
 dam, area, ingreso,sexo, anoest, edad, discapacidad,   
 tiene\_alcantarillado, tiene\_electricidad, tiene\_acueducto,  
 tiene\_gas, eliminar\_basura, tiene\_internet, piso\_tierra,  
 material\_paredes, material\_techo, rezago\_escolar, alfabeta,  
 hacinamiento, tasa\_desocupacion  
 ) %>%   
 mutate(  
 dam = recode(dam,  
 "01" = 1,  
 "02" = 2,  
 "03" = 3,  
 "04" = 4,  
 "05" = 5,  
 "06" = 6)) %>%  
 mutate\_if(is.character, as.factor) %>%   
 mutate(dam = as.factor(dam))  
  
summary(encuesta\_mrp)

## Validación cruzada

# Eta: Magnitud de las correciones que son hechas por cada predictor nuevo (learning rate)  
# max depth: profundidad de los árboles  
grid <- expand\_grid(max\_depth = seq(3, 6, 1), eta = seq(.2, .35, .01))  
y = encuesta\_mrp[,3]  
encuesta\_mrp\_m <- xgb.DMatrix(data = as.matrix(sapply(encuesta\_mrp[,-3], as.numeric)), label = as.matrix(sapply(y, as.numeric)))  
  
  
xgb\_train\_rmse <- numeric(nrow(grid))  
xgb\_test\_rmse <- numeric(nrow(grid))  
  
  
for(i in 1:nrow(grid)){  
 xgb\_untuned = xgb.cv(  
 data = encuesta\_mrp\_m,  
 params = list(  
 objective = "reg:squarederror",  
 eta = grid$eta[i],  
 max\_depth = grid$max\_depth[i]  
 ),  
 nrounds = 100,  
 early\_stopping\_rounds = 3, # training with a validation set will stop if the performance does not improve for k rounds (3)  
 nfold = 5  
 )  
   
 xgb\_train\_rmse[i] <-  
 xgb\_untuned$evaluation\_log$train\_rmse\_mean[xgb\_untuned$best\_iteration]  
 xgb\_test\_rmse[i] <-  
 xgb\_untuned$evaluation\_log$test\_rmse\_mean[xgb\_untuned$best\_iteration]  
   
 cat(i, "\n")  
}

xgb\_train\_rmse <- xgb\_train\_rmse %>%   
 tibble() %>%   
 mutate(simulacion = seq(1:64))  
  
xgb\_test\_rmse <- xgb\_test\_rmse %>%   
 tibble() %>%   
 mutate(simulacion = seq(1:64))  
  
xgb\_train\_rmse %>% ggplot(aes(x = simulacion, y = .)) + geom\_line()  
  
grid[which.min(xgb\_train\_rmse$.), ]  
saveRDS(xgb\_train\_rmse, file = "ingreso/output/xgb\_train\_rmse\_census.rds")

* max depth = 6
* eta = 0.2

modelo2 <- xgboost(params = list(  
 objective = "reg:squarederror",  
 eta = 0.2,  
 max\_depth = 6),  
 nrounds = 1000,  
 # early\_stopping\_rounds = 3,  
 data = encuesta\_mrp\_m  
)  
  
saveRDS(modelo2, file = "ingreso/output/modelo2.rds")  
  
ggplot(modelo2$evaluation\_log, aes(  
 x = iter,  
 y = train\_rmse  
)) + geom\_line() + geom\_vline(xintercept = 104, col = "red")  
  
modelo2$evaluation\_log[which.min(modelo2$evaluation\_log$train\_rmse), ]

* Para este caso podemos observar que luego de 927 iteraciones el modelo se detuvo debido a que el modelo no mejoró luego de 927 por lo que ahí se detuvo.
* Por otro lado, se puede observar que luego de 104 iteraciones el modelo estabiliza el Error Cuadrático Medio.

## importancia de variables

importancia <- xgb.importance(  
 feature\_names = names(encuesta\_mrp[,-7]),  
 model = modelo2  
)  
  
p3 <- xgb.ggplot.importance(importancia)

# Evaluación del RMSE para los tres modelos

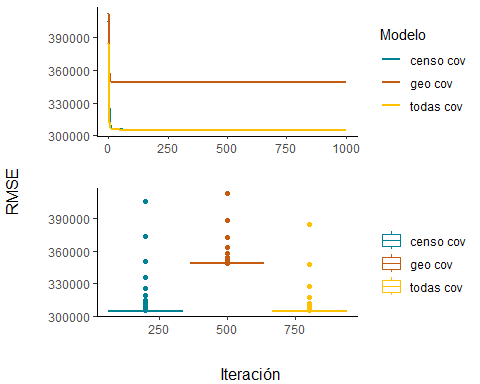
rm(list = ls())

modelo <- readRDS("C:/Users/mespinoza/Documents/SAExML/R/ingreso/output/modelo.rds")  
modelo1 <- readRDS("C:/Users/mespinoza/Documents/SAExML/R/ingreso/output/modelo1.rds")   
modelo2 <- readRDS("C:/Users/mespinoza/Documents/SAExML/R/ingreso/output/modelo2.rds")

modelos <- rbind(modelo$evaluation\_log, modelo1$evaluation\_log,  
 modelo2$evaluation\_log)  
  
modelos$mods <- rep(c("todas cov","geo cov","censo cov"), each = 1000)

p1 <- ggplot(modelos, aes(x = iter, y = train\_rmse, col = mods)) + geom\_line(linewidth = 1) + labs(color = "Modelo") + xlab(" ") + ylab(" ") + theme\_classic() + scale\_color\_manual(values=c("#008194", "#C55A11","#FFC000"))  
  
p2 <- ggplot(modelos, aes(x = iter, y = train\_rmse, col = mods)) + geom\_boxplot() + labs(color = " ") + xlab(" ") + ylab(" ") + theme\_classic() + scale\_color\_manual(values=c("#008194", "#C55A11","#FFC000"))

grid.arrange(p1, p2, nrow = 2, left = "RMSE", bottom = "Iteración")



## Predicción con la encuesta

pred <- predict(modelo, newdata = encuesta\_mrp\_m)  
  
saveRDS(pred, file = "ingreso/output/pred\_encuesta.rds")

pred <- readRDS("C:/Users/mespinoza/Documents/SAExML/R/ingreso/output/pred\_encuesta.rds")  
  
encuesta\_mrp$pred <- pred

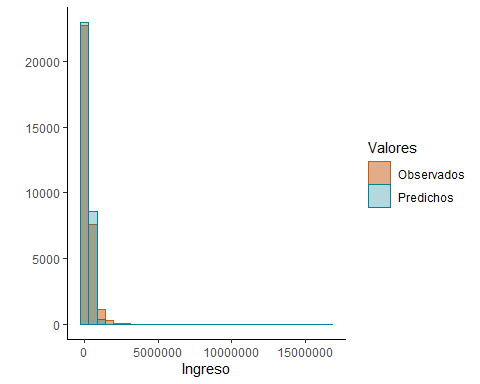
matrix(c(mean(encuesta\_mrp$ingreso, mean(encuesta\_mrp$pred))),  
 ncol = 2, dimnames = list(" " = "media", "Estimación" = c("Encuesta", "Predicción") ))

## Estimación  
## Encuesta Predicción  
## media 170416.7 170416.7

ggplot(encuesta\_mrp) +   
 geom\_histogram(aes(x = ingreso),stat = "density", adjust = 80, fill = "#C55A11",col = "#C55A11") +  
 geom\_histogram(aes(x = pred), stat = "density", adjust = 80, fill = "#008194",col = "#008194", alpha = 0.3) +  
 theme\_classic()

ggplot(encuesta\_mrp) +   
 geom\_histogram(aes(x = ingreso,fill = "Observados"),   
 bins = 30,  
 color = "#C55A11",  
 alpha = 0.5,  
 size = 0.2) +  
 geom\_histogram(aes(x = pred, fill = "Predichos"),   
 bins = 30,  
 color = "#008194",  
 alpha = 0.3,  
 size = 0.2) +  
scale\_fill\_manual(values = c("#C55A11", "#008194")) + labs(x = "Ingreso", y = " ", fill = "Valores") +  
 theme\_classic()

## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.  
## ℹ Please use `linewidth` instead.  
## This warning is displayed once every 8 hours.  
## Call `lifecycle::last\_lifecycle\_warnings()` to see where this warning was  
## generated.



* De acá se puede observar que el modelo con solo las covariables geoespaciales no están brindando suficiente información al modelo, generando así un Error Cuadrático Medio mucho más alto que sus otros competidores.
* Por otro lado, la ganancia que hay con el modelo con todas las covariables es muy poca con respecto al modelo con solo las covariables del censo.

censo\_mrp <- readRDS("C:/Users/mespinoza/Documents/SAExML/R/ingreso/datos/censo\_mrp1.rds") %>% ungroup() %>%   
 select(dam, area, sexo, anoest, edad, discapacidad,  
 F182013\_stable\_lights, X2016\_crops.coverfraction,  
 X2016\_urban.coverfraction, X2016\_gHM,   
 accessibility, accessibility\_walking\_only,  
 tiene\_alcantarillado, tiene\_electricidad,  
 tiene\_acueducto, tiene\_gas, eliminar\_basura,  
 tiene\_internet, piso\_tierra,material\_paredes,  
 material\_techo, rezago\_escolar, alfabeta,  
 hacinamiento,tasa\_desocupacion  
 )

censo\_mrp1 <- xgb.DMatrix(data = as.matrix(sapply(censo\_mrp, as.numeric)))

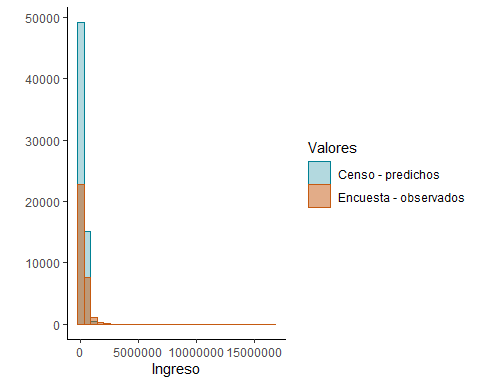
pred <- predict(modelo, newdata = censo\_mrp1)  
  
saveRDS(pred, file = "ingreso/output/pred.rds")

pred <- readRDS("C:/Users/mespinoza/Documents/SAExML/R/ingreso/output/pred.rds")  
censo\_mrp$pred <- pred

matrix(c(mean(encuesta\_mrp$ingreso), mean(censo\_mrp$pred)),  
 ncol = 2, dimnames = list(" " = "media", "Estimación" = c("Encuesta", "Censo") ))

## Estimación  
## Encuesta Censo  
## media 275675.8 239002.8

ggplot() +  
 geom\_histogram(data = censo\_mrp, aes(x = pred, fill = "Censo - predichos"),  
 bins = 30,  
 color = "#008194",  
 alpha = 0.3,  
 size = 0.2) +  
 geom\_histogram(data = encuesta\_mrp, aes(x = ingreso, fill = "Encuesta - observados"),  
 bins = 30,  
 color = "#C55A11",  
 alpha = 0.5,  
 size = 0.2) +  
 scale\_fill\_manual(values = c("#008194","#C55A11")) +   
 labs(x = "Ingreso", y = " ", fill = "Valores") +  
 theme\_classic()



importancia <- xgb.importance(  
 feature\_names = names(encuesta\_mrp[,-c(3,27)]),  
 model = modelo  
)  
  
xgb.ggplot.importance(importancia)

