ISSN: 2602-8085
Vol. 2, N°4.1., p. 42-54, octubre - diciembre, 2018

Comparativa entre classification trees, random forest y gradient boosting; en la predicción de la satisfacción laboral en Ecuador.

Comparison between classification trees, random forest and gradient boosting; in the prediction of job satisfaction in Ecuador.

Vinicio Alexander Andrade Saltos.[1], Pablo Flores M. [2]

Recibido: 09-07-2017 / Revisado: 10-09-2018 Aceptado: 13-10-2018/ Publicado: 01-11-2018

Abstract.

DOI: https://doi.org/10.33262/cienciadigital.v2i4.1..189

In order to find an adequate model to predict the Level of Job Satisfaction in Ecuador, three prediction models based on trees were compared. The "Random Forest" and "Gradient Boosting" models are considered more complex than the "Classification Tree" model and suppose better results; however, when applied to a database obtained from the National Survey of Employment, Unemployment and Underemployment; it was found that the criteria and efficiency of prediction are similar for the three models, reaching approximately 30% error in the classification.
It was concluded that not necessarily a more complex model obtains more precise results.

Keywords: Classification trees, random forest, gradient boosting, job satisfaction, data mining, ENEMDU.

Resumen.

Con el objetivo de encontrar un modelo adecuado para predecir el Nivel de Satisfacción Laboral en Ecuador, se compararon tres modelos de predicción basados en árboles. Los modelos “Random Forest” y “Gradient Boosting” se consideran más complejos que el

[1] Escuela Superior Politécnica de Chimborazo, Facultad de Ciencias, Grupo de Investigación en Ciencia de Datos CIDED, Riobamba - Ecuador, vaas_92@hotmail.com
[2]Escuela Superior Politécnica de Chimborazo, Facultad de Ciencias, Grupo de Investigación en Ciencia de Datos CIDED, Riobamba - Ecuador, p_flores@espoch.edu.ec

ISSN: 2602-8085
Vol. 2, N°4.1., p. 42-54, octubre - diciembre, 2018

modelo “Classification Trees” y suponen mejores resultados; sin embargo, al aplicarlos sobre una base de datos obtenida a partir de la Encuesta Nacional de Empleo, Desempleo y Subempleo; se encontró que los criterios y eficiencia de predicción son similares para los tres modelos, alcanzando aproximadamente un 30% de error en la clasificación.
Se concluyó que no necesariamente un modelo más complejo obtiene resultados más precisos.

Palabras Claves: árboles de clasificación, bosques aleatorios, gradient boosting, satisfacción laboral, minería de datos, ENEMDU.

Introducción .

El objetivo principal del análisis predictivo es obtener modelos, los cuales, mediante una muestra, permitan predecir una variable objetivo para nuevas observaciones en función de un conjunto de variables predictivas. Dependiendo de esta variable de interés, distinguimos entre tareas de clasificación, cuando dicha variable es categórica, clasificando un nuevo individuo en la categoría de la variable que predice el modelo o tareas de regresión cuando la variable es numérica, obteniendo un valor estimado por el modelo para un nuevo individuo (Torgo, 2003). De acuerdo al tipo de tarea que se realice, se pueden proponer diferentes indicadores que permitan determinar la calidad de predicción del modelo.

Uno de los análisis predictivos que más trascendencia y evolución ha tenido en el campo de la minería de datos, es aquel que se basa en modelos de árboles (Classification Trees) (Rokach & Maimon, 2008). Empezando con un desarrollo teórico de la idea (Leo, Friedman, Olshen, & Stone, 1984), el estudio ha ido evolucionando hasta poder ser explicado mediante un enfoque menos formal (Quinlan, 2014; Torgo, 1999), lo cual ha permitido implementar estos modelos a través de algoritmos computacionales (Therneau & Beth, 2018). El proceso algorítmico para desarrollar este modelo consiste básicamente en agrupar a los individuos de acuerdo a las diferentes interacciones que se puedan formar con las distintas variables explicativas. En todas las posibles categorías cruzadas se mide la frecuencia observada respecto a la variable objetivo (variable de salida); luego, las ramas se empiezan a abrir de acuerdo a las frecuencias más relevantes de las variables predictivas (variables de entrada) con las separaciones más grandes que se pueda formar. A menos que exista un indicador que permita podar el árbol, el proceso se sigue repitiendo hasta encontrar categorías para las cuales las frecuencias respecto de la variable objetivo dejan de ser relevantes.

A partir de la idea de árboles de decisión, se han creado algoritmos más complejos con el fin de mejorar el nivel de precisión de la predicción. El problema que presentan los modelos de árboles, es que, al seguir un algoritmo secuencial, siempre eligen a las mismas variables de acuerdo a su nivel de relevancia en términos de frecuencia; lo cual, da una exploración del

ISSN: 2602-8085
Vol. 2, N°4.1., p. 42-54, octubre - diciembre, 2018

espacio de variables demasiado acotada, esto podría derivar en la omisión de un grupo de variables predictivas que quizás son importantes en el contexto del estudio. Se proponen entonces nuevos métodos más complejos que consisten en construir múltiples árboles en sub-espacios aleatorios del espacio de variables, lo cual permite generalizar su clasificación de manera complementaria, y su clasificación combinada puede ser mejorada monótonamente (Kam, 1995).

Al respecto de los modelos más complejos, uno muy utilizado es el denominado "Random Forest" (Breiman, 2001); el cual, es una versión mejorada de modelos que llevan esta misma idea de múltiples árboles en sub-espacios aleatorios como el "bagging" (Breiman, 1996). El algoritmo para construir un random forest, consiste básicamente en seleccionar aleatoriamente m grupos disjuntos de variables aleatorias independientes, sobre cada uno de los cuales se creará un árbol; luego, se promedia la capacidad de predicción de todos los m árboles formados, teniendo así un modelo que podría incluir una variable que posiblemente sea relevante para la predicción del objetivo, pero que quizás si se construyera un solo árbol no se la tomaría en cuenta debido a su baja frecuencia con relación a la variable de salida. Otro modelo muy usado es el denominado "Extreme Gradient Boosting" (Breiman, 1997), la diferencia básica de este modelo es que mientras en el random forest los árboles son formados por conjuntos de variables independientes, el Extreme Gradient Boosting construye árboles de manera secuencial, donde cada nuevo árbol es creado de acuerdo al margen de error que dejan las variables peor clasificadas por el árbol anterior, hasta que el algoritmo llega a estabilizarse y el despeño de todos los árboles combinados alcanza un umbral máximo de ajuste.

En el presente trabajo se construyen estos tres modelos, basados en las técnicas previamente descritas (Modelo de árbol, Random Forest y Extreme Gradient Boosting) con el fin de comparar la calidad de predicción del nivel de satisfacción laboral de los jefes de hogar en Ecuador; para lo cual, se utilizó información recogida por el Instituto Nacional de Estadísticas y Censos (INEC) mediante la Encuesta Nacional de Empleo, Desempleo y Subempleo (ENEMDU) de diciembre del 2017.

El nivel de satisfacción laboral constituye un factor clave en el desempeño de la población económicamente activa, el cual afecta directamente al funcionamiento de una organización (Rowden, 2002) por lo que se recomienda incrementarlo (Friday & Friday, 2003). Un elemento clave para cumplir este objetivo es conocer el comportamiento que describe esta variable de interés; por ello, se busca definir un modelo que pueda caracterizarla con precisión.

Metodología de análisis.

Obtención de la base de datos.

ISSN: 2602-8085
Vol. 2, N°4.1., p. 42-54, octubre - diciembre, 2018

La base de datos con los individuos y variables de interés se importó utilizando el paquete base del software estadístico R (Team, 2016). Es importante señalar que, a partir de diciembre 2003, la ENEMDU se realiza bajo un esquema de panel de viviendas seleccionadas en una submuestra; la cual, se mantiene en la muestra durante dos trimestres consecutivos, seguido de un descanso de un semestre y finalmente entran en la muestra por dos últimos trimestres. En cada edición de la ENEMDU, la información se divide en módulos o subconjuntos con objetivos específicos de información. Particularmente, en diciembre del 2017 existen ocho módulos (15 años, ambiente, armonía, consumidor, financiero, hábitos, salud y seguridad, vivienda).

En total la encuesta es aplicada en 31092 viviendas, seleccionadas por muestreo probabilístico cuya población objetivo está conformada por personas de 5 años y más. Luego de un proceso de unión de módulos y filtro de variables e individuos se consiguió una base de datos formada por 27841 observaciones, correspondientes a todos los jefes de hogar encuestados que tienen un único trabajo, sobre los cuales se filtraron las variables mostradas en la Tabla 1, mismas que fueron seleccionadas tomando en cuenta su posible relación teórica (desde un punto de vista socio - económico) con la satisfacción laboral. Para esto, se creó una variable de identificación concatenando los valores de las variables: área, ciudad, conglomerado, zona, sector, panel, vivienda y hogar; en un solo vector de tipo carácter, la cual sirvió en el proceso de unión de los módulos 15 años y armonía; y en la discriminación de los individuos.

Tabla 1. Variables utilizadas para predecir la satisfacción laboral.

Código	Variable	Código	Variable
p59	¿Cómo se siente en su trabajo?	p47b	Número personas trabajan en el establecimiento
p02	Sexo	seg019	Seguridad en el trabajo
p03	Edad	p61a5	¿Ha recibido capacitación para prevenir accidentes y minimizar los riesgos de su trabajo actual?
p06	Estado civil	p61a61	Durante los últimos 12 meses, ¿ha sufrido algún accidente desempeñando su trabajo actual?
p15	¿Cómo se considera? (Indígena, Negro, Mulato, Mestizo, ...)	p61a7	¿Ha sufrido alguna enfermedad causada por las actividades de su trabajo actual?
p42	Categoría de ocupación (Empleado de gobierno, Empleado privado, Patrono, …)	p61a8	¿Cree que su seguridad o salud está expuesta a riesgos por lo que desempeña en su trabajo actual?
p61a2	¿Cree que conservar su trabajo actual durante los siguientes 12 meses es probable?	p71a	¿Recibió ingresos derivados del capital?
p45	¿Cuántos años trabaja?	ingrl	Ingreso del trabajo
p61a12	¿Desde hace cuánto tiempo empezó su trabajo actual, de manera continua? (Años)	SECEMP	Sectores de los Empleados (Sector Formal, Sector Informal, Sector Doméstico, ...)

ISSN: 2602-8085
Vol. 2, N°4.1., p. 42-54, octubre - diciembre, 2018

p51a	Horas de trabajo principal	grupo1	Grupo de Ocupación (Empleados de Oficina, Fuerzas Armadas, Profesionales Científicos e Intelectuales, …)
AR02	¿Considera que su hogar es pobre?	rama1	Rama de actividad (Industrias manufactureras, Actividades financieras y de seguros, ...)
p27	¿Desea trabajar más horas?	p10a	Nivel de instrucción
p46	Sitio de trabajo	p14	Idioma que habla
p47a	Tamaño del establecimiento	p79	¿Practicó algún deporte la semana pasada?

La variable objetivo "Satisfacción laboral" presenta cuatro categorías (Contento, Poco contento, Descontento pero conforme y Totalmente descontento); pero, dado que por el interés del presente estudio, las dos últimas miden lo mismo y además ambas presentan una frecuencia relativa considerablemente baja (mostrado en figura 1), estas se unen en una sola categoría denominada "Descontento". En este punto, es preciso indicar que los resultados de los modelos son similares al unirse o no estas categorías; por efectos prácticos se trabajó con la opción que fusiona las frecuencias. Dado que la variable objetivo es de tipo ordinal, los modelos predictivos basados en árboles que se apliquen realizarán tareas de clasificación.

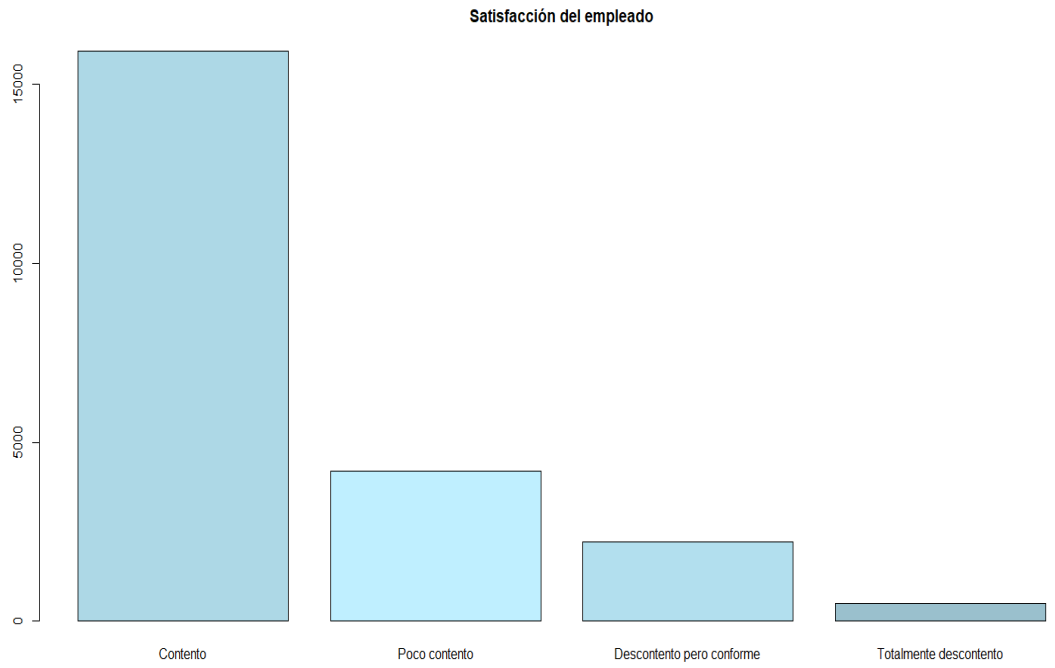

Figura 1. Frecuencia absoluta de la variable respuesta "Nivel de satisfacción laboral".

En las bases de datos proporcionadas por el INEC, se identifican valores perdidos con diferentes designaciones: 99, 999, 9999, 999999; etiquetados como "No informan" o

ISSN: 2602-8085
Vol. 2, N°4.1., p. 42-54, octubre - diciembre, 2018

simplemente se dejan los espacios sin llenar; por ello, en todos estos casos se asignó el valor "NA".

Los individuos que presentaron NA's en todas las variables medidas fueron eliminados, mientras que aquellos con una cantidad considerablemente baja de datos perdidos fueron imputados utilizando medidas de tendencia central y un modelo de regresión lineal múltiple. La técnica de tendencia central consistió básicamente en imputar los datos perdidos mediante la moda en el caso de variables cualitativas y mediante la media recortada en el caso de variables cuantitativas. La técnica de regresión lineal múltiple consistió fundamentalmente en encontrar el mejor modelo (basado en buscar las variables explicativas que provoquen el mayor coeficiente de determinación corregido) que estime la variable cuantitativa a ser imputada. Luego de realizar dicho procedimiento se imputaron 19.30% de datos de la variable p47b, 13.95% de p61a12, 4,05% de ingrl y 0,84% de p27; se eliminaron 17.98% de individuos con NA's totales sobre todas las variables y finalmente se eliminaron 14 individuos que respondieron "No sabe, no responde" en la variable respuesta; dejando una muestra de 22821 observaciones.

Elección de la muestra de entrenamiento y validación de modelo.

Mediante un proceso de muestreo aleatorio simple sin reemplazo, se dividió a la muestra en dos grupos de observaciones. El primer grupo denominado muestra de entrenamiento contiene 20000 individuos (87.64%), sobre los cuales se aplican los tres modelos basados en árboles para obtener una descripción del comportamiento de las variables predictoras; luego, el proceso de predicción se realiza con los 2821 individuos restantes (12.36%), los cuales se denominan muestra de prueba. Esto con el propósito de realizar una comparación de los valores reales y los valores predichos por los modelos y poder obtener medidas sobre la precisión de la predicción.

Dado que el presente estudio tiene como objetivo construir modelos de clasificación; la validación respectiva, en cada caso, se realizó mediante la denominada matriz de confusión. Las columnas de esta matriz cuadrada representan el número de predicciones en cada clase de la variable respuesta, mientras que las filas representan la frecuencia absoluta de las observaciones en cada clase donde realmente se encuentran los individuos de la muestra de prueba. La diagonal principal permitió observar el número de individuos que fueron clasificados correctamente después de la aplicación del modelo. En este sentido, la proporción de aciertos se calculó como el cociente entre la suma de la diagonal principal y la suma total de la matriz; el respectivo complemento puede ser visto como una tasa de error de la predicción.

Además, utilizando la función "peakRAM( )"(Quinn, n.d.) se comparó el tiempo en segundos y el uso de la memoria RAM en mebibytes que cada uno de estos modelos ocupa. Si bien es cierto, este no es un índice de validación del modelo pero podría servir para tener una idea

ISSN: 2602-8085
Vol. 2, N°4.1., p. 42-54, octubre - diciembre, 2018

general del tiempo de procesamiento de cada modelo que, en la práctica, podría resultar de utilidad para aquellos que vayan a aplicarlo. La Tabla 2 muestra esta información para los tres modelos, donde se observa que la técnica Gradient Boosting, a pesar de ser un modelo más complejo gasta menos memoria RAM y se ejecuta más rápido.

Tabla 2 . Medición del uso de la memoria RAM cuando se aplican los tres modelos.

Técnica	Tiempo (s)	Uso total de RAM (MiB)	Pico de RAM (MiB)
Classification Trees	34,45	0,20	79,80
Random Forest	52,43	17,70	159,90
Gradient Boosting	12,00	0,50	0,90

Aplicación de los modelos.

Para aplicar el algoritmo del modelo basado en árboles, se utilizó la función "rpartXse( )" del paquete "DMwR2" (Torgo, 2003) que integra los procesos realizados por las funciones "rpart( )" y "prune.rpart( )" del paquete "rpart" (Therneau & Beth, 2018), las cuales se encargan respectivamente de extender y podar un árbol tanto como se les indique. En este caso, no se podaron los árboles formados con el fin de obtener la mayor cantidad de variables predictoras en el modelo y poder realizar una exploración del espacio de variables lo más amplia posible, evitando de esta forma, los problemas descritos en la introducción. El árbol que se generó en el proceso se muestra en el siguiente enlace: https://1drv.ms/u/s!AuxDwRXuZsjEjHHZUeqDRuV6Wriq

El algoritmo utilizado para implementar el modelo Random Forest se aplicó mediante la función "randomForest( )" del paquete que lleva el mismo nombre (Liaw & Wiener, 2002), mientras que la función "xgb.train( )" del paquete "xgboost" (Li et al., 2018) se usó para aplicar el algoritmo correspondiente al modelo Gradient Boosting. En este caso, la función requiere de una estructura de datos denominada "sparse matrix", la cual se consiguió a través de la función "sparse.model.matrix( )" del paquete "Matrix" (Bates & Maechler, 2017).

Resultados y discusión.

La Tabla 3 muestra las variables más y menos influyentes sobre la variable objetivo. Se puede observar que la mayoría de estas variables se repiten en los tres modelos, lo cual indica que al parecer las variables más significativas para predecir el Nivel de Satisfacción Laboral están claramente definidas, no hace falta generar demasiados árboles ya que difícilmente se encontrará una variable que quizás no se tomó en cuenta debido a su baja frecuencia. En el análisis mediante "Classification Trees", las variables que se distinguen como más influyentes (en su orden de aparición con la Tabla 3) tienen que ver con: El tiempo que trabaja el empleado, el ingreso económico que percibe, el sitio donde trabaja, su espacio o tamaño

ISSN: 2602-8085
Vol. 2, N°4.1., p. 42-54, octubre - diciembre, 2018

de trabajo, la categoría laboral que tiene, la estabilidad laboral, la rama a la que pertenece, el grupo al cual pertenece (empleado, intelectual, militar, …), las horas de trabajo y el sector (Formal, Informal, …) en el cual se desenvuelve.

Tabla 3. Variables más y menos influyentes sobre la variable objetivo "Nivel de Satisfacción Laboral".

Classification Trees		Random Forest		Gradient Boosting	
Más influyentes	**Menos influyentes**	**Más influyentes**	**Menos influyentes**	**Más influyentes**	**Menos influyentes**
p27	p15	ingrl	p61a8	p27	p61a8
ingrl	p14	p46	p61a5	ingrl	p06
p46	p61a7	p27	p61a7	p51a	p15
p47a	p06	rama1	p06	AR02	p61a7
p42	p61a5	p61a12	p14	p45	p61a5
p61a2	p61a8	p45	p15	p03	p14
rama1	p02	p03	p02	p61a12	p02
grupo1	p61a61	grupo1	p61a61	p61a2	p61a61
p51a	p79	p42	p79	p46	p79
SECEMP	p71a	SECEMP	p71a	p42	p71a

Las matrices de confusión generadas con cada uno de los modelos son resumidas en el cálculo de las tasas de error de clasificación mostradas en la Tabla 4. Como se puede observar estas tasas no son significativamente bajas, lo cual nos indica que la eficiencia de predicción de los tres modelos para clasificar a un individuo no resulta ser totalmente confiable. Es posible que otro(s) modelo(s) podría(n) arrojar mejores resultados. Indagar al respecto resulta una clara propuesta para investigaciones futuras.

Tabla 4. Tasas de error de clasificación en los tres modelos comparados.

Técnica	Classification Trees	Random Forest	Gradient Boosting
Tasas de error de clasificación	30.70%	30.20%	29.95%

Aun cuando, "Random Forest" y "Gradient Boosting" son modelos más complejos que "Classification Trees", desarrollados con el objetivo de mejorar el nivel de predicción; en el presente análisis se observa que, las tasas de error de clasificación en todos los modelos son muy similares. Al parecer, las variables más influyentes están claramente definidas y sin importar el número o el tipo de árboles que se corran, éstas seguirán siendo las que mejor

ISSN: 2602-8085
Vol. 2, N°4.1., p. 42-54, octubre - diciembre, 2018

expliquen el Nivel de Satisfacción Laboral. Este hecho permite concluir que no necesariamente un modelo más complejo obtiene resultados más precisos.

Anexos.

Código en R

```
# Training set and Sample set ####
set.seed(1234)
sample <- sample(1:nrow(data), 2821)
train <- data[-sample, ]
test <- data[sample, ]

# TREE MODEL ####
library(DMwR2)
set.seed(1234)
model.t <- rpartXse(p59 ~ ., train, se = 0)
summary(model.t)
# Variables importance
vars.imp <- as.data.frame(model.t$variable.importance)
rm(vars.imp)
# Plot
library(rpart.plot)
prp(model.t, type = 0, extra = 104)
# Predicting
set.seed(1234)
pred.t <- predict(model.t, test, type = "class")
# Estimated Error clasification Rate
cm <- table(pred.t, test$p59)
100*(1 - sum(diag(cm)) / sum(cm))

# RANDOM FOREST ####
library(randomForest)
set.seed(1234)
model.rf <- randomForest(p59 ~ ., train, ntree = 69, importance = T)
# Variables importance
model.rf$importance
var.imp.rf <- as.data.frame(model.rf$importance)
var.imp.rf <- var.imp.rf[, ]
# Predicting
set.seed(1234)
pred.rf <- predict(model.rf, test, type = "class")
# Estimated Error clasification Rate
cm <- table(pred.rf, test$p59)
100*(1 - sum(diag(cm)) / sum(cm))

# XGBOOSTING ####
library(Matrix)
library(xgboost)
```

ISSN: 2602-8085
Vol. 2, N°4.1., p. 42-54, octubre - diciembre, 2018

```
library(magrittr)
# Sparse matrix (response variable)
p59.xg <- vector()
for (i in 1:nrow(data)) {
  if (data[i, "p59"] == "Contento") {
    p59.xg[i] <- 2
  }
  else {
    if (data[i, "p59"] == "Poco contento") {
      p59.xg[i] <- 1
    }
    else {
      if (data[i, "p59"] == "Descontento") {
        p59.xg[i] <- 0
      }
    }
  }
}
# Data with response variable
data.xgb <- cbind(data, p59.xg)
data.xgb <- select(data.xgb, p59.xg, everything())
data.xgb <- data.xgb[, -2]
# Training set and Sample set
set.seed(1234)
sample.xg <- sample(1:nrow(data.xgb), 2821)
train.xg <- data.xgb[-sample.xg, ]
test.xg <- data.xgb[sample.xg, ]
test.xg.1 <- data.xgb[sample.xg, ]
# Training sample
train.p59.xg = train.xg[, "p59.xg"]
train.xg <- sparse.model.matrix(p59.xg ~ .-1, data = train.xg)
train.matrix <- xgb.DMatrix(data = as.matrix(train.xg),
                            label = train.p59.xg)
# Test sample
test.p59.xg = test.xg[, "p59.xg"]
test.xg <- sparse.model.matrix(p59.xg ~ .-1, data = test.xg)
test.matrix <- xgb.DMatrix(data = as.matrix(test.xg),
                           label = test.p59.xg)
# Model parameters
params <- list("objective" = "multi:softmax",
               "eval_metric" = "mlogloss",
               "num_class" = length(unique(train.p59.xg)))
watchlist <- list(train = train.matrix, test = test.matrix)
# Model construction
model.xg <- xgb.train(params = params,
                      data = train.matrix,
                      nrounds = 40,
                      watchlist = watchlist,
                      eta = 0.2,
                      max.depth = 6, # profundidad de los arboles (podaje
```

ISSN: 2602-8085
Vol. 2, N°4.1., p. 42-54, octubre - diciembre, 2018

```
)
                      gamma = 0, # evita el overfitting
                      subsample = 1, # prevent overfitting [0, 1]
                      colsample_bytree = 0.3, # [0, 1]
                      seed = 1234)
# Plots
e <- data.frame(model.xg$evaluation_log)
min(e$test_mlogloss)
e[e$test_mlogloss == 0.798937, ]
plot(e$iter, e$train_mlogloss, col = 'blue')
lines(e$iter, e$test_mlogloss, col = 'red')
# Variables importance
imp <- as.data.frame(xgb.importance(model = model.xg))
xgb.plot.importance(imp)
# Predicting
set.seed(1234)
pred.xg <- predict(model.xg, test.matrix, type = "class")
# Estimated Error clasification Rate
cm <- table(pred.xg, test.xg.1$p59.xg)
100*(1 - sum(diag(cm)) / sum(cm))

# RAM used ####
library(peakRAM)
set.seed(1234)
peakRAM(rpartXse(p59 ~ ., train, se = 1))
set.seed(1234)
peakRAM(randomForest(p59 ~ ., train, ntree = 69))
set.seed(1234)
peakRAM(xgb.train(params = params,
                  data = train.matrix,
                  nrounds = 40,
                  watchlist = watchlist,
                  eta = 0.2,
                  max.depth = 6, # profundidad de los arboles (podaje)
                  gamma = 0, # evita el overfitting
                  subsample = 1, # prevent overfitting [0, 1]
                  colsample_bytree = 0.3, # [0, 1]
                  seed = 1234)

)
```

ISSN: 2602-8085
Vol. 2, N°4.1., p. 42-54, octubre - diciembre, 2018

Referencias bibliográficas.

Bates, D., & Maechler, M. (2017). Matrix: Sparse and Dense Matrix Classes and Methods.

Breiman, L. (1996). Bagging predictors. *Machine Learning*, *24*(2), 123 140.

Breiman, L. (1997). *Arcing the edge*. California.

Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5–32.

Friday, S. S., & Friday, E. (2003). Racioethnic perceptions of job characteristics and job satisfaction. *Journal of Management Development*, *22*(5–6), 426–442. https://doi.org/10.1108/02621710310474778

Kam, H. T. (1995). Random decision forest. In *Proc. of the 3rd Int'l Conf. on Document Analysis and Recognition* (pp. 14–18). Montreal, Canada.

Leo, B., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). Classification and regression trees. *Wadsworth International Group*.

Li, T. C., He, T., Benesty, M., Yutian, V. K. and Y. T., Cho, H., Chen, K., … Geng, Y. (2018). xgboost: Extreme Gradient Boosting.

Liaw, A., & Wiener, M. (2002). Classification and Regression by randomForest. *R News*, *2*(3), 18–22.

Quinlan, J. R. (2014). *C4. 5: programs for machine learning*. Elsevier B.V.

Quinn, T. (n.d.). Monitor the Total and Peak RAM Used by an Expression or Function [R package peakRAM version 1.0.2]. Retrieved from https://cran.r-project.org/web/packages/peakRAM/index.html

Rokach, L., & Maimon, O. Z. (2008). *Data mining with decision trees: theory and applications*. World scientific.

Rowden, R. W. (2002). The relationship between workplace learning and job satisfaction in U.S. small to midsize businesses. *Human Resource Development Quarterly*, *13*(4), 407–425. https://doi.org/10.1002/hrdq.1041

Team, R. C. (2016). R: A Language and Environment for Statistical Computing. Vienna, Austria: R Foundation for Statistical Computing.

Therneau, T., & Beth, A. (2018). rpart: Recursive Partitioning and Regression Trees. R package.

Torgo, L. (1999). *Inductive learning of tree-based regression models*. Universidade do Porto. Reitoria.

Torgo, L. (2003). *Data mining with R*. University of Porto: LIACC-FEP.

Para citar el artículo indexado.

Cumbicos J., Jiménez L., Luna M., Valdivieso Á. & Barona L. . (2018). Comparativa del avance en desarrollo en las telecomunicaciones entre ecuador y Bolivia. *Revista electrónica Ciencia Digital* ***2***(4.1.), 42-54. Recuperado desde: http://cienciadigital.org/revistacienciadigital2/index.php/CienciaDigital/article/view/189/167