8. Diseño de un Modelo Predictivo de Fuga de Clientes Utilizando Algoritmos Machine Learning

Design of a Predictive Model of Customer Leakage Using Machine Learning Algorithms

Diego Pinto[1*], José Fuentes[2**].

[1,2] Universidad ECCI, Bogotá D.C., Colombia.

* diegoa.pintog@ecci.edu.co
** jfuentesm@ecci.edu.co

RESUMEN

Las empresas prestadoras de servicios de telecomunicaciones móviles contribuyen al desarrollo económico del país, ya que son necesarias para mantener el flujo de información entre diversos canales y servicios. Actualmente, en Colombia hay varias empresas nacionales e internacionales que prestan servicios móviles y la alta penetración de las mismas en el mercado ha generado se incremente la competencia por los usuarios. Por consecuente, dichas empresas tienen el gran reto de mantener la satisfacción y fidelización de los mismos. Para lograr mantener dicha fidelización se hace pertinente analizar y entender el comportamiento de los clientes antes de que se fuguen a otra empresa, para anticiparse a este comportamiento se desarrollará en el presente artículo la descripción, análisis y modelamiento por medio de técnicas de Machine Learning y se escogerá la que mejor se ajuste a los comportamientos de los usuarios.

Palabras claves: Fidelización, Fuga de Clientes, Machine Learning, predicción.

Recibido: 21 de agosto de 2019. Aceptado: 04 de septiembre de 2019

ABSTRACT

The companies providing mobile telecommunications services contribute to the economic development of the country, as they are necessary to maintain the flow of information between various channels and services. Now in Colombia there are several national and international companies that provide mobile services and the high penetration of this market has generated that they train in competition for users. This means that companies have the great challenge of maintaining the satisfaction and loyalty of their users. In order to achieve user loyalty, it is necessary to understand and understand the behavior of users before they request portability or flight to another company, in order to anticipate this behavior, the analysis and modeling will be developed in this work through Machine Learning techniques and the choice of that best suits user behaviors.

Keywords: Loyalty, Customer Leak, Machine Learning, prediction.

Received: February 21, 2019 Accepted: September 04, 2019

1. INTRODUCCIÓN

Actualmente la era digital está girando en torno a las nuevas tecnologías, trayendo consigo cambios profundos y transformaciones de una sociedad que se mueve en un mundo globalizado. Por ende, se tiene la gran necesidad de estar activamente conectado a una red y estar al tanto de todo lo que pasa alrededor. En poco más de una década, se duplicaron sustancialmente los usuarios de Internet, que ya alcanzaban el 50,1% de la población en 2014; hoy existen más de 700 millones de conexiones a telefonía móvil, con más de 320 millones de usuarios únicos, y muchos países de Latinoamérica se encuentran entre los que más usan las redes sociales globales. Esta es la nueva era que impactó al mundo, pero también a los gobiernos, quienes han tenido que incorporar el uso de las tecnologías de la información y la comunicación en la gestión pública *[1]*.

En Colombia la estimación preliminar de habitantes según estadísticas entregadas por el DANE en el censo nacional de población y vivienda del año 2018 es de un poco más de cuarenta y ocho millones de habitantes *[2]*, y el número de líneas móviles para el tercer trimestre del año 2018 fue más de sesenta y tres millones, una cifra entregada por el Mintic (Ministerio de las Tecnologías y las Comunicaciones), la cual es superior en comparación con la cantidad de habitantes de Colombia *[3]*. Cabe recalcar que, aproximadamente un 80% de las líneas son prepago y el 20% son líneas pospago. Bajo estas estadísticas, donde es significativo el número de usuarios de servicios móviles, se hace relevante que las empresas de telecomunicaciones móviles generen estrategias de fidelización para que no exista el riesgo de fuga por parte del cliente a un mejor servicio, que alguna otra empresa de telecomunicaciones le pueda brindar.

Desde una perspectiva de inteligencia de negocios, el proceso de gestión de fuga de clientes se considera una de las tareas más importantes y se focaliza en dos actividades: la primera, en predecir una potencial fuga, y la segunda, aplicar medidas preventivas para evitar que se produzca la fuga en donde la gran cantidad de información recolectada y almacenada en las empresas, ligada a la necesidad de entender y satisfacer al cliente, ha traído consigo maneras eficientes de analizar la información, una de ella es el aprendizaje automático o más conocido en inglés como Machine Learning *(ML)*.

En este artículo se pretende encontrar un modelo que mejor se ajuste al comportamiento de los datos para la predicción de fuga de clientes prepago en una empresa prestadora de servicios de telecomunicaciones móviles.

El artículo se organiza en 3 partes: en una primera parte se realiza un breve acercamiento a la definición de ML y algunos de sus algoritmos más importantes e utilizados, los cuales se utilizaron para el desarrollo de la investigación propuesta. Además se realiza una descripción e implementación del Big data a partir de los datos considerados en el estudio. En la segunda parte se mostraran algunos resultados obtenidos como consecuencia de los algoritmos implementados en ML. Finalmente, se exponen las conclusiones a partir de la metodología utilizada y sus alcances.

2. MATERIALES Y MÉTODO

2.1. ¿Qué es el Machine Learning?

El ML es una disciplina científica en la cual se crean ciertos tipos de algoritmos capaces de aprender automáticamente, en otras palabras aprenden a identificar ciertos patrones que existen en el conjunto de datos y así ser capaces de predecir comportamientos futuros basándose en aquello que aprendió previamente.

El ML es un término que actualmente se escucha con frecuencia en la industria, debido a su gran cercanía con tecnologías como la inteligencia artificial *(AI)*, el Big Data, entre otros.

Los inicios del ML se dan en los años 50, cuando Arthur Samuel, un pionero en el campo de los juegos informáticos y AI, escribió el primer programa de aprendizaje automático que consistía en un juego en el cual la computadora iba mejorando por sí misma a medida que se jugaba más *[4]*. Los siguientes años se siguió adoptando las metodologías empleadas por Samuel y con ello el surgimiento de varios algoritmos posteriormente. Después del año 2010, los grandes "motores" tecnológicos como los son IBM, Google, Facebook, Amazon y Microsoft, comenzaron sus propios desarrollos en ML. El campo de la aplicación de estas técnicas depende de la necesidad, la imaginación y datos que se tengan disponibles.

Fig. 1. División del machine learning.

El ML se divide en dos partes especialmente: aprendizaje supervisado y aprendizaje no supervisado, tal como lo muestra la *Fig. 1.*

2.1.1. Aprendizaje supervisado

El aprendizaje supervisado se suele usar en problemas de clasificación: identificación numérica, problemas de regresión, predicción de algún fenómeno y/o comportamiento. Este tipo de aprendizaje por lo general se diferencia por el tipo de variable respuesta o variable objetivo, la cual es categórica o numérica *(previamente etiquetada).* Los algoritmos de este tipo de aprendizaje son: Árboles de decisión, Clasificación, Naive Bayes, Regresión por mínimos cuadrados, Regresión logística, Máquinas de soporte vectorial *(SVM)* y Métodos "emsemble".

Se reconocieron dos tipos de aprendizaje supervisado:

Regresión: Su propósito es establecer un modelo para la relación entre un cierto número de características y una respuesta objetivo continua. En la *Fig. 2*, se presentó un ejemplo de regresión lineal que ajusta unos datos a una variable de respuesta mediante una recta.

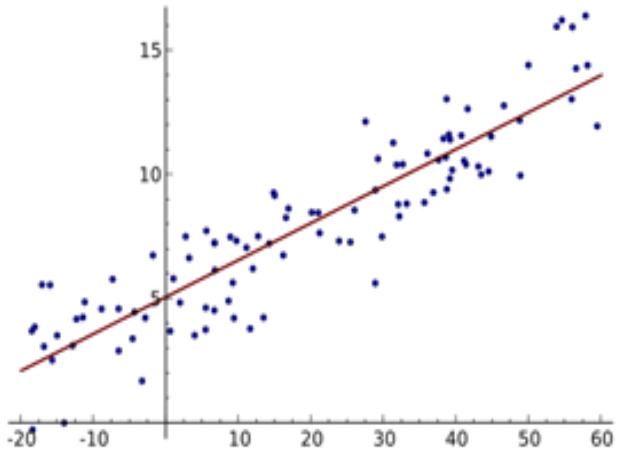

Fig. 2. Ejemplo de regresión lineal.

Clasificación: El objetivo es lograr clasificar los valores de respuesta categóricos en diferentes grupos en los que los datos se lograron separar en "clases" específicas, tal como se observa en la *Fig. 3.*

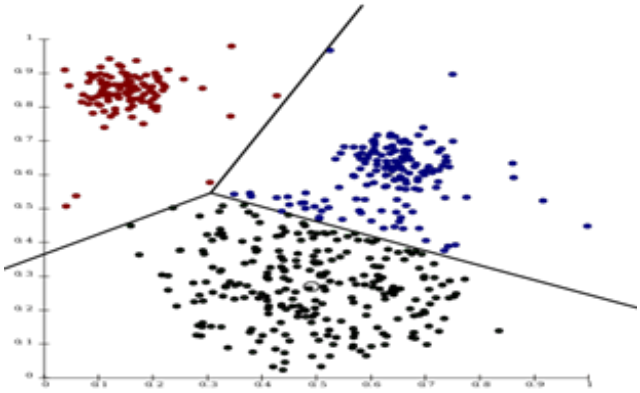

Fig. 3. Ejemplo de clasificación.

2.1.2. Aprendizaje no supervisado

El aprendizaje no supervisado tiene lugar cuando no se dispone de algún tipo de etiqueta para los datos, solamente se le otorgan las características de los mismos. Su principal función es lograr la agrupación, por lo que el algoritmo debe catalogar la información por similitud y poder crear grupos semejantes, sin tener la capacidad de definir la individualidad de algún integrante del grupo formado. Este tipo de aprendizaje se suele utilizar en problemas de clustering, perfilamiento y agrupamientos de co-ocurrencia.

Los tipos de algoritmos de aprendizaje no supervisado son: Algoritmos de clustering *(K-means por ejemplo)*, ACP *(Análisis de Componentes Principales)* como los más conocidos y utilizados.

En este artículo, se utilizaron dos tipos de algoritmos supervisados pertenecientes a la familia de árboles de decisión los cuales son el algoritmo Gradient Boosting Machine *(GBM)* y el algoritmo Distributed Random Forest *(DRF)*.

GBM

Es un algoritmo utilizado para regresión y clasificación, gracias a su gran capacidad de aprendizaje. Es un método de conjunto de aprendizaje avanzado en el cual se pueden obtener buenos resultados predictivos a través de aproximaciones cada vez más refinadas. El GBM construye secuencialmente árboles de regresión en todas las características del conjunto de datos de una manera totalmente distribuida, debido a que cada árbol de decisión se construye en paralelo *[5]*. El esquema de este algoritmo es el siguiente:

Inicia

$$f_{ko} = 0, k = 1,2, \dots, k \qquad 1]$$

Para $m = 1$ hasta M

Establece

$$p_x(x) = \frac{e^{fk(x)}}{\sum_{l=1}^{k} e^{fl(x)}} \quad k = 1,2 \dots K. \qquad 2]$$

Para

$m = 1$ hasta M

Calcular

$$r_{ikm} = y_{ik} - p_k(xi), xi = 1,2, \dots, N \qquad 3]$$

Ajustar los objetivos del árbol de regresión

$$R_{ijm}, j = 1,2, \dots., N \qquad 4]$$

Calcular

$$y_{jkm} = \frac{K-1}{K} \frac{\sum_{xeRjmk}(r_{ikm})}{\sum_{xeRjmk} |r_{ikm}|(1-|r_{ikm}|)} \qquad 5]$$

Actualizar

$$f_{km}(x) = f_{km-1}(x) + \sum_{j=1}^{Jm} y_{jkm} I(x) \qquad 6]$$

Salida

$$f_k(x) = f_{KM}(x), k = 1,2, \dots, K \qquad 7]$$

DRF

DRF es una combinación de árboles de predicción *[6]*, tal que cada árbol depende de los valores de un vector de muestreo de variables aleatorias independientemente e igualmente distribuidas y hace referencia a un algoritmo de inteligencia artificial para problemas de clasificación y regresión, el cual, genera un modelo predictivo que implementa árboles de decisión que se construyen en el modelo de forma escalonada y generalizada, permitiendo la optimización arbitraria de una función de pérdida *[7]*.

En todo modelo de regresión se debe asumir la relación de pérdida entre sesgo, varianza y la función objetivo compuesta por L que es la función de pérdida de entrenamiento, el cual

mide que tan bien se ajustan los datos al modelo y Ω la función de regularización midiendo la complejidad del modelo.

$$Obj(\theta)=L(\Theta)+\Omega(\Theta). \quad (8)$$

En donde se busca optimizar la función de pérdida de entrenamiento en función de los parámetros θ,Θ y de regularización. La función de pérdida de entrenamiento se optimiza a través de entrenamiento aditivo en el cual cada árbol f_i contiene la información correspondiente, así el árbol de aprendizaje tendrá una estructura mucho

$$y_i^{(0)} = 0, \quad (9)$$
$$y_i^{(1)} = f_1(x_1) = y_i^{(0)} + f_1(x_i), \quad (10)$$
$$y_i^{(2)} = f_1(x_i) + f_2(x_i) = y_i^{(0)} + f_2(x_i), \quad (11)$$
$$\vdots$$
$$y_i^{(t)} = \sum_{k=1}^{t} f_k(x_i) = y_i^{(t-1)} + f_t(x_i). \quad (12)$$

Existen varias maneras de definir la complejidad del modelo, sin embargo en este documento se expuso la anteriormente mencionada *[8]*, en la cual se le asignan pesos Wi a las hojas dependiendo de la estructura del árbol q así se tiene que

$$f_i(x) - w_{q(x)}, w \in R^T, q: R^d \longrightarrow \{1,2,\dots,T\}$$

y T el número de hojas, luego la función de regularización se definió como:

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2. \quad (13)$$

Actualmente existen varias herramientas y lenguajes de programación que permiten ensamblar o construir un modelo ML. La herramienta a seleccionar queda en manos de la persona que construye el modelo, ya que hay que tener en cuenta los conocimientos previos en algún tipo de lenguaje y formación o conocimiento de lo que se pretende lograr. Cabe aclarar que para grandes volúmenes de datos lo recomendado es trabajarla en Cloud, ya que un CPU estándar no tiene la capacidad de almacenamiento y procesamiento de este tipo de modelamientos.

Fig. 4. Metodología CRISP-DM.

2.2.1. Comprensión del negocio

El entorno competido de las empresas de telecomunicaciones hace que el negocio se enfoque en el cuidado de los clientes, por eso la necesidad de anticiparse a la fuga de los mismos. Como indicador de cualquier compañía de telecomunicaciones se hace clave reducir esta tasa de fuga como uno de los objetivos fundamentales del negocio y como objetivo de ciertas áreas especializadas de analítica avanzada, en las cuales se plantean estrategias que conlleven a un beneficio dentro de la compañía. Desde estas áreas se diseña análisis de información, manejo de información, entre otras como los modelos ML de los que se han abarcado a lo largo de este artículo.

2.2.2. Comprensión de los datos

El conjunto de datos es una muestra aleatoria conformada por seis meses de registros históricos de los usuarios con líneas prepago, en la cual se evidenció el comportamiento de los clientes que se han fugado y los que siguen activos. Además, se cuenta con un gran número de variables como antigüedad de la línea, tipo de equipo, promedio de ingresos por usuario en voz y datos, el número de llamadas entradas y salientes, consumo de mensajes de texto, consumo de redes sociales, tiempo de permanencia en llamadas en otras redes,

entre otras; las cuales aportaron al aprendizaje del modelo.

Si se quiere controlar algo, debe ser observable, y para conseguir el éxito, es esencial definir lo que se considera un éxito: ¿será precisión?, ¿exactitud?, ¿tasa de retención de clientes? esta medida debe estar directamente alineada con los objetivos de mayor nivel del negocio, y también directamente relacionada con el tipo de problema que se presenta.

Así que, teniendo conformado el conjunto de datos, debe existir esa etiqueta la cual puede tener varios niveles o clases y es a la cual se debe dar respuesta. En este caso se contó con una etiqueta de interés en la que existen dos clases, "1" para referenciar al cliente que se fugó y "0" para el que no *(Cliente activo)*.

2.2.3. Preparación de los datos

Antes de comenzar a entrenar modelos, se debe transformar los datos de una manera que puedan alimentar el modelo ML. Para esto se procedió a realizar la limpieza del conjunto de datos donde se realizó el respectivo análisis de valores perdidos, los cuales generalmente se representan con los indicadores "NaN" o "Null". El debido tratamiento de estos es muy importante, ya que la mayoría de algoritmos no pueden manejar esos valores faltantes y por ende omitirlos. Preferiblemente antes de entrenar el modelo, se debe identificar los valores perdidos y tratarlos de la manera más conveniente.

Además, se realizó un tratamiento de datos categóricos donde es muy importante tener la misma estructura de los mismos, ya que el modelo solo aprenderá los datos que se tengan de entrada; entonces si en la predicción existe alguna otra categoría diferente, el modelo no podrá predecir y por ende la omitirá.

Es fundamental tener presente que el ML sólo puede ser usado para memorizar comportamientos o patrones que están presentes en los datos de entrenamiento, por lo tanto sólo podemos reconocer lo que hemos visto antes. Cuando usamos ML estamos asumiendo que el futuro se comportará como el pasado, lo que no siempre es cierto.

Teniendo limpia y tratada nuestra base de datos, se puede proceder a realizar las diferentes estadísticas descriptivas y exploratorias, ya que describen lo que tenemos en el conjunto de datos y exponen la forma en la que se distribuyen los datos, valores atípicos, discontinuidades, etc.

Por medio de un análisis descriptivo, se logró tener un previo conocimiento o hipótesis del porqué los usuarios deciden portarse a otras empresas y algunas de estas razones fueron:

- El monto de las recargas realizadas no satisfacía su necesidad y por ende necesitaban recargar más.

- El aumento en el promedio de sus consumos. *(Se empezó a tener más necesidades en cuanto a internet, voz o texto)*.

- La cantidad de llamadas a otra red era alta, por lo que se piensa en algún tipo de influencia, entre otras.

En la exploración del conjunto de datos se logró determinar que la etiqueta objetivo tiene un alto grado de desbalance, ya que tan sólo un 5% correspondió a los clientes que se fugaron y el 95% restante a los clientes activos, como se observa en la *Fig. 5*. Debido a que los algoritmos ML son deficientes en presencia de desbalance, por esta razón es pertinente balancear las clases existentes en la misma *[11]*.

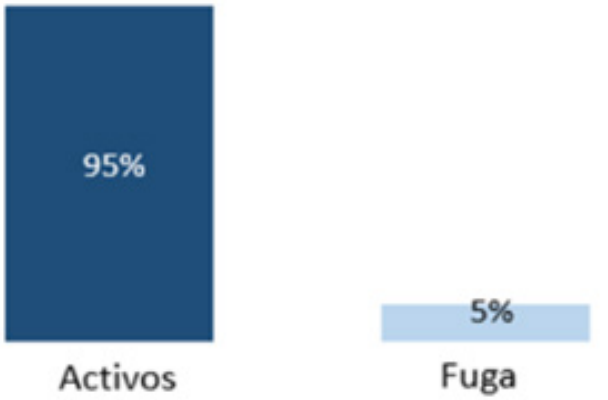

Fig. 5. Desbalanceo del conjunto de datos.

Dada esta situación se procedió a utilizar técnicas de balanceo por medio de una librería en R llamada "ROSE" en la cual se encuentran funciones que generan muestras sintéticas balanceadas y por lo tanto, permiten fortalecer la estimación posterior de cualquier clasificador binario, manejando datos continuos y categóricos a partir de una estimación de densidad condicional de las dos clases. De esta librería se utilizó la función "Ovun.sample" la cual permite balancear las clases de tres diferentes maneras:

Undersampling

Es el proceso en el que se elimina al azar algunas de las observaciones de la clase mayoritaria para hacer coincidir los números con la clase minoritaria. Se manejaron tres escenarios diferentes en esta técnica tal como lo muestra la *Fig. 6.*

Oversampling

Es el proceso de generar datos sintéticos aleatoriamente para una muestra de los atributos a partir de observaciones en la clase minoritaria. De la misma manera se simularon tres escenarios para el modelo, tal como lo muestra la *Fig. 7.*

Balanceo con igual probabilidad

En este proceso se implementan ambos métodos anteriormente mencionados en las respectivas clases, tal como lo muestra la *Fig. 8.* Esta metodología será denotada por "Both sampling", ya que de acuerdo a la probabilidad aplicada se reduce la clase mayoritaria y se

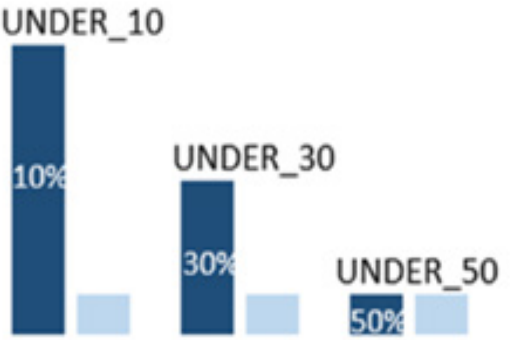

Fig. 6. Escenarios implementados con undersampling.

aumenta la clase minoritaria al mismo tiempo, hasta quedar en un punto medio alcanzando la paridad.

Los escenarios antes mencionados corresponden a los diferentes pesos en probabilidades para balancear la clase.

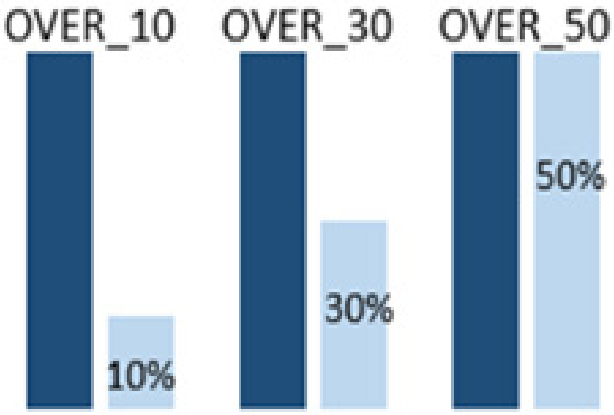

Fig. 7. Escenarios implementados con oversampling.

2.2.4. Modelado

El objetivo en este paso del proceso fue desarrollar un modelo de comparación que sirviera de referencia, sobre el que se medirá el rendimiento del algoritmo mejor y más ajustado a los datos. Así que, como se mencionó anteriormente se utilizaron los algoritmos GBM y DRF ya que el rendimiento de ambos algoritmos en algunos escenarios aleatorios de balanceo fue similar. Se tuvo en cuenta el criterio AUC *(Área Bajo la Curva "ROC") [12]*, para medir que tan buena clasificación de las clases existentes tuvo

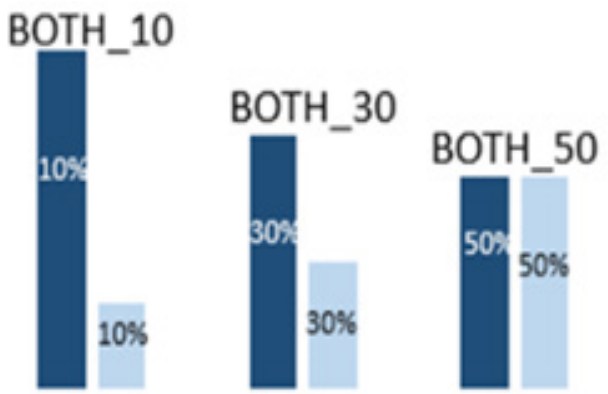

Fig. 8. Ambas técnicas de balanceo al tiempo.

el modelo. Además de eso se utilizó un método común para encontrar un buen modelo por medio de la validación cruzada, en la cual se establece n número de subdivisiones en las que se fragmentará los datos, los cuales se irán puntuando en la validación del mismo.

Un algoritmo ML tiene dos tipos de parámetros, el primer tipo son los parámetros que se aprenden a través de la fase de aprendizaje, y el segundo tipo son los hiperparámetros que se transfieren a los modelos, los cuales son combinación de parámetros óptimos o de mejor ajuste. Una vez identificado el modelo que se manejará, el siguiente paso es ajustar sus hiperparámetros para obtener el mayor poder predictivo posible. La forma más común de encontrar la mejor combinación de hiperparámetros se llama "Grid Search". Esto se realizó en todos los escenarios planteados anteriormente los cuales son nueve, es decir, se realizaron 18 modelos donde 9 fueron modelados bajo el algoritmo GBM y los otros 9 bajo el algoritmo DRF, respectivamente a su escenario de balanceo.
Para esto se siguió el siguiente proceso:

- Se establece la matriz de parámetros que se evaluarán. Los parámetros fueron secuencias establecidas de acuerdo a los requerimientos de cada uno.

- Se establece el número de subdivisiones de los datos, el estado aleatorio, criterios de parada del algoritmo, máximo número de modelos que se planean se entrenen por el algoritmo o consecuentemente un tiempo límite de entrenamiento.

- Se determina una estrategia de búsqueda de cuadrícula la cual fue aleatoria discreta.

Cuando finalizó el entrenamiento, se observó la cantidad de modelos que se entrenaron con sus respectivos parámetros establecidos anteriormente. Se seleccionó el mejor de cada escenario por medio del criterio AUC, donde se estableció que el modelo que tuviera mayor AUC fuera seleccionado como el mejor.

2.2.5. Evaluación

Terminados los modelos y seleccionados los mejores de cada escenario, se procedió a mirar los comportamientos de los mismos con un conjunto de datos que se usará para predicción con un millón de registros, el cual contiene la misma estructura del conjunto de datos de entrenamiento pero con diferentes registros, esto con el fin de medir que tan bien se comporta o predice el modelo con datos que no conoce y además a eso para lograr comparar los diferentes resultados por medio del análisis de las curvas ROC *[12]*, Análisis de precisión por observación.

3. RESULTADOS Y DISCUSIÓN

Las curvas ROC nos permite conocer que tan bien fueron clasificadas las clases, además indicaron el área de cobertura en los diferentes escenarios con el algoritmo GBM se observan en la *Fig. 9*, donde los escenarios de under sampling se destacaron

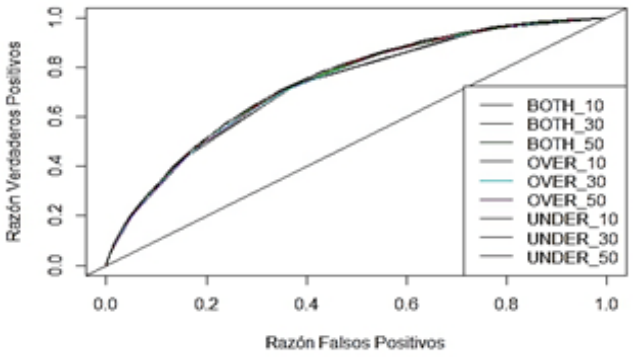

Fig. 9. Curvas ROC GBM.

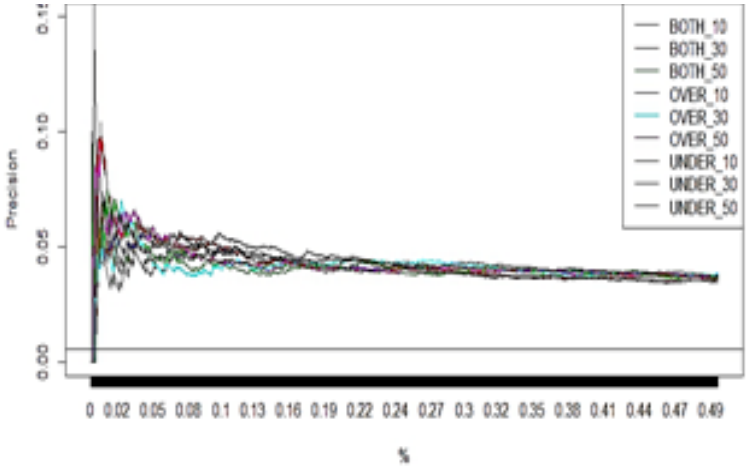

Fig. 10. Análisis de Precisión GBM.

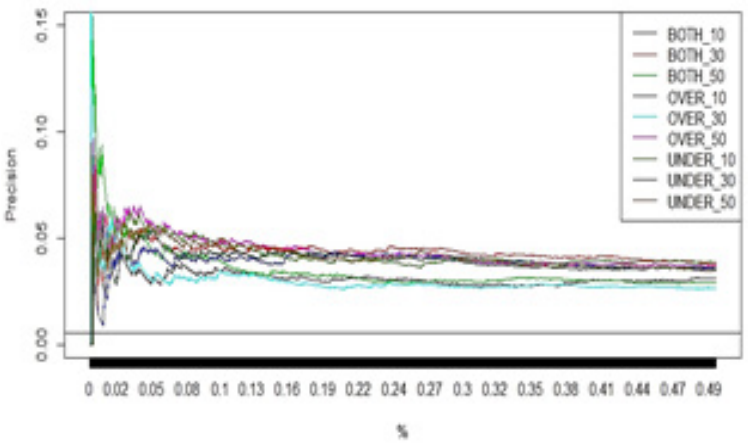

Fig. 12. Análisis de Precisión DRF.

con un AUC mayor a 0,74.

De igual manera el análisis de precisión *Fig. 10* en las observaciones, nos permite conocer que tan precisa fue la predicción para los individuos del conjunto de datos nuevo. Este análisis va muy de la mano con la capacidad de gestión que se tenga o la cantidad de personas que se puedan gestionar, ya que cada modelo proporciona diferentes maneras de hacerlo. Los picos máximos reflejan la precisión máxima alcanzada por cada modelo *Fig 11*.

De igual manera lo anterior con el algoritmo DRF *Fig 12*.

Donde de igual manera, los escenarios "under sampling" se destacan de los demás. Para conocer qué modelo es el mejor, se realizó el análisis "lift" el cual es una medida de la efectividad de un modelo predictivo calculado como la relación entre los resultados obtenidos con y sin el modelo predictivo, donde se evidenció que el modelo con el algoritmo GBM bajo el escenario Under_50 tuvo un buen desempeño y por ende es el mejor y el seleccionado.

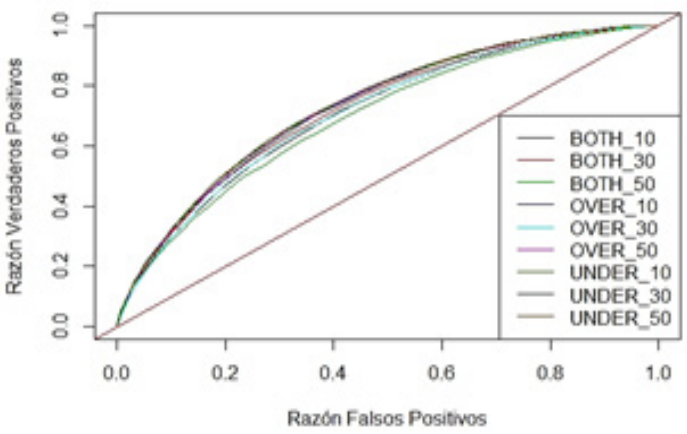

Fig. 11. Curvas ROC DRF.

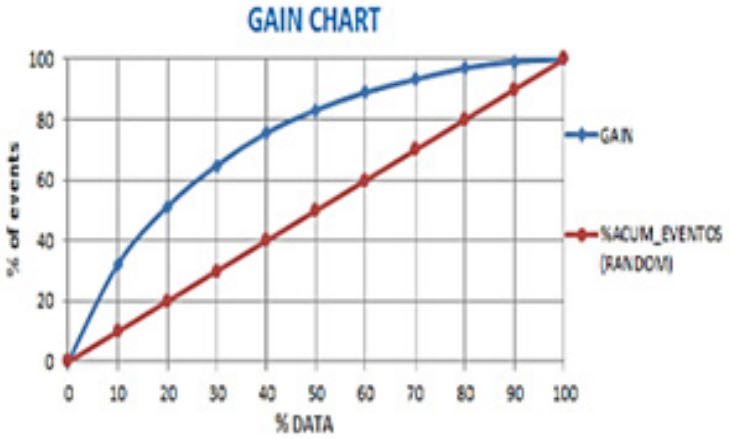

Fig. 13. Ganancia del modelo.

La ganancia reflejada por este modelo se presentó en la *Fig. 13*, donde la línea roja muestra los eventos aleatorios sin un modelo y la azul muestra la ganancia que se tendría con el modelo seleccionado. *(Fig 14)*

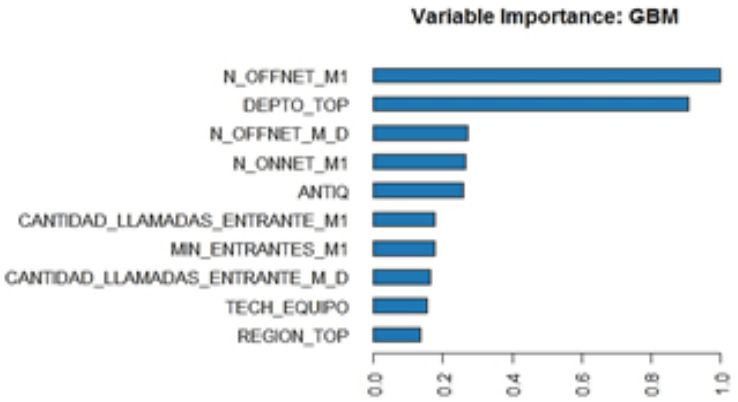

Fig. 14. Variables de importancia del modelo.

4. CONCLUSIÓN

El modelo seleccionado como el mejor de todos los escenarios modelados en los dos algoritmos es el que mejor se ajusta a los datos y posteriormente tuvo buen desempeño prediciendo un conjunto de datos. Además, este modelo otorga la posibilidad de poder conocer las variables de importancia para el mismo en el cual se determina calculando la influencia relativa de cada variable, si esa variable se seleccionó para dividirse durante el proceso de construcción del árbol y cuánto mejoró *(disminuyó)* el error al cuadrado *(sobre todos los árboles)*.

En adición, se logra observar que una de las variables importantes definidas por el modelo, previamente fue un hallazgo encontrado mediante el análisis descriptivo del conjunto de entrenamiento, la cual era la cantidad del servicio de llamadas fuera de la red y por ende se puede pensar en algún tipo de influencia por personas cercanas al cliente como anteriormente se mencionó.

La efectividad del modelo para la gestión de usuarios y la efectividad del mismo será reflejada en los estudios posteriores de algún tipo de campaña de retención o fidelización que se realice. Para futuros proyectos con el mismo, se buscará mejorar la ganancia del modelo a la obtenida hasta ahora. Se implementa de igual manera aprendizajes supervisados pero con diferentes algoritmos como el XGBoosT o modelos "ensemble" los cuales son combinaciones de más de un algoritmo, además contando con algún tipo de herramienta o equipo que reduzca el tiempo de entrenamiento de los modelos, ya que estos modelos son un poco más complejos.

5. AGRADECIMIENTOS

En primer lugar deseo expresar mi agradecimiento a Juan Peñaloza- coordinador minería de datos y José Fuentes- docente Universidad ECCI, por su apoyo constante en la construcción de este artículo, por el respeto a mis sugerencias e ideas y por la dirección y el rigor que ha facilitado las mismas, además de estar profundamente agradecido con el área de analítica avanzada de la empresa en la cual estoy realizando mis prácticas empresariales, ya que gracias a ellos, mi formación profesional tomo un sendero. A mi madre y abuela por apoyarme día a día en mis metas y sueños propuestos.

6. REFERENCIAS BIBLIOGRÁFICAS

[1] Naser. A, La nueva era digital. Disponible en: https://www.tec.ac.cr/pensis/articulos/nueva-era-digital. [consultado el 3 de julio de 2019].
[2] Colombia. Departamento Administrativo Nacional de Estadística -DANE-. Censo Nacional de Población y Vivienda 2018 -CNPV 2018. Bogota D.C. Colombia. 2019.
[3] Colombia alcanzó los 62,2 millones de líneas de celular habilitadas. Disponible en: https://www.mintic.gov.co/portal/604/w3-article-72964.html. [Consultado el 23 de abril de 2018].
[4] J. McCarthy and E. A. Feigenbaum, "*In Memoriam: Arthur Samuel: Pioneer in Machine Learning*", AIMag, vol. 11, no. 3, p. 10, 1990.
[5] Drucker, H., Robert S. y Patrice S. *Boosting performance in neural networks. Advances in Pattern Recognition Systems using Neural Network Technologies.*61-75, 1993.
[6] L. *Breiman. Random forests. Machine Learning*, 45(1): 5–32, 2001.
[7] Amit, Y., y Geman, D. *Shape Quantization and Recognition with Randomized Trees. Neural Computation,* 9, 1545-1588. 1997.
[8] Chen, Tianqi. *Introduction to boosted trees. University of Washington Computer Science* 22, 115. 2014.
[9] Aiello, S., Eckstrand, E., Fu, A., Landry, M., y Aboyoun,

P. *Machine Learning with R and H2O. H2O booklet*. 2016.
[10] Azevedo, A. I. R. L., y Santos, M. F. KDD, SEMMA and CRISP-DM: *a parallel overview*. IADS-DM. 2008.
[11] Estabrooks, A., Jo, T., y Japkowicz, N. *A multiple resampling method for learning from imbalanced data sets. Computational intelligence,* 20(1), 18-36. 2004.
[12] Davis, J., y Goadrich, M.. *The relationship between Precision-Recall and ROC curves. In Proceedings of the 23rd international conference on Machine learning.* 233-240. 2006.