ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE CIENCIAS

COMPARACIÓN DE UN MÉTODO CLÁSICO (LOGIT) Y DOS CON TÉCNICAS DE BOOSTING (ADABOOST, GRADIENT BOOSTING) PARA LA CLASIFICACIÓN CREDITICIA EN UNA ENTIDAD FINANCIERA ECUATORIANA

TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO MATEMÁTICO

PROYECTO DE INVESTIGACIÓN

EDISON JAVIER QUIZHPI SÁNCHEZ
Javier_edison@hotmail.es

DIRECTOR: MENTHOR OSWALDO URVINA MAYORGA

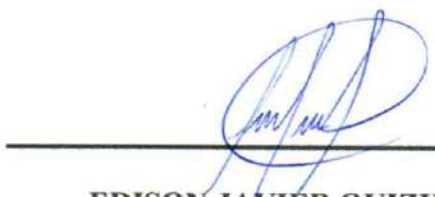
menthor.urvina@epn.edu.ec

QUITO, ABRIL 2023

DECLARACIÓN

Yo, Edison Javier Quizhpi Sánchez, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

La Escuela Politécnica Nacional puede hacer uso de los derechos correspondientes a este trabajo, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.



EDISON JAVIER QUIZHPI SÁNCHEZ

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Edison Javier Quizhpi Sánchez, bajo mi supervisión.

MENTHOR URVINA MAYORGA

DIRECTOR DE PROYECTO

AGRADECIMIENTOS

A mis padres, Mariana y Manuel, que con su amor, paciencia y trabajo siempre me han apoyado.

A mis hermanos Edwin y Mónica, por su tiempo, consejos y apoyo incondicional que siempre me han brindado.

A María José, que siempre estuvo a mi lado ayudándome y motivándome para cumplir mis metas alcanzadas.

A mi mejor amigo Carlos, al que considero como un hermano, por ser parte de este proceso y estar en cada paso del camino.

A mis amigos y familiares, por brindarme sus oportunas palabras de aliento.

A mi Director de trabajo de titulación, por su tiempo, paciencia y guía durante la realización de este proyecto.

A mis compañeros de trabajo, por sus consejos y ayuda.

DEDICATORIA

A mis padres Mariana y Manuel, por haberme forjado como la persona que soy, por sus consejos y apoyo incondicional.

Infinitas gracias porque todo lo que soy es gracias a ellos.

Tabla de contenidos

Lista de Tablas

Lista de Figuras

RESUMEN

En este proyecto de investigación se evalúa el comportamiento de tres métodos de clasificación, *regresión logística*, *adaboost* y *gradient boosting* en el proceso de modelización del *credit scoring* con base en un conjunto de datos suministrado por una institución financiera de Ecuador consistente en una cartera de crédito de tamaño desconocido.

Por motivos de confidencialidad de la institución financiera, no se pueden mostrar detalles del conjunto de datos; sin embargo, se puede mencionar que está compuesto por 29 variables independientes: 19 variables numéricas continuas y 10 variables categóricas que describen el comportamiento de pago y características demográficas asociadas a 21.492 clientes. Además, se incluye la variable dependiente, dicotómica, que especifica si un cliente es *bueno* o *malo* de acuerdo a la definición de la institución financiera. Este conjunto de datos constituye una muestra adecuada para el desarrollo de los modelos de *credit scoring* desarrollados en este trabajo.

Los tres modelos desarrollados fueron construidos con los programas R y *Answer Tree* v3.0. Una vez establecidos, se comparó su capacidad de predicción y clasificación de clientes catalogados como buenos y malos. Para la evaluación del desempeño de los modelos, se calcularon los estadísticos *Area Under Receiver Operator Characteristic Curve* (AUC-ROC), Kolmogorov-Smirnov (KS) y Gini, y se evaluó el error y la tasa de aciertos de cada modelo.

Atendiendo al desempeño mostrado, el mejor modelo de clasificación resultó ser *gradient boosting*, luego el *adaboost*, y por último regresión logística. El modelo *credit scoring* desarrollado con *gradient boosting* logra el mejor ajuste para el conjunto de datos disponible ya que el estadístico KS arrojó mayor valor que en los casos *adaboost* y regresión logística, aunque la diferencia con *adaboost* en términos prácticos es despreciable. Cuando se compara en base al estadístico AUC-ROC, *gradient boosting* también obtiene el mayor valor con una diferencia más notable cuando se compara con regresión logística. El mismo comportamiento descrito para los estadístico KS y AUC-ROC se observó para el coeficiente de Gini y tasa de aciertos. En lo que respecta al error, *gradient boosting* exhibe el menor error que los otros dos modelos.

Estos resultados demuestran que *gradient boosting* es el mejor modelo de *credit scoring*, sin embargo, el estadístico AUC-ROC debe evaluarse según el contexto, por lo cual queda a criterio del analista seleccionar el método de *credit scoring*.

Palabras clave: regresión logística, adaboost, gradient boosting, scoring, modelización, estadísticos, cliente bueno, cliente malo, árboles de decisión.

ABSTRACT

In this research project, referees of three classification methods: logistic regression, adaboost and gradient boosting. In the credit scoring modeling process is based on a data provided by a financial institution in Ecuador dedicated a credit portfolio of unknown size.

For confidentiality reasons of the financial institution, can't display details of the data; but it can be mentioned that it has 29 independent variables: 19 continuous numerical variables and 10 categorical variables. It describes the payment behavior and demographic characteristics associated with 21,492 clients. Also, it is included a dichotomous variable and dependent variable. That specifies if a customer is good or bad according to the definition of the financial institution. This data constitutes a sample for the development of the credit scoring models developed in this work.

The three developed models were built with the R and Answer Tree v3.0 programs. Once established, was compared its predictive ability and classify clients classified as good and bad. Model performances were calculated for the Area Under Receiver Operator Characteristic Curve (AUC-ROC), Kolmogorov-Smirnov (KS) and Gini statistics. the error and success rate of each model were evaluated.

The performance shown, the best classification model was gradient boosting, then adaboost, and finally logistic regression. The credit scoring model developed with gradient boosting achieves the best fit for the available data set, the KS statistic yielded a higher value than in the adaboost and logistic regression cases, although the KS statistic is more negligible with adaboost in practical terms.

When compared based on the AUC-ROC statistic, gradient boosting obtained the highest value with a more notable difference when compared to logistic regression. The same behavior described for the KS and AUC-ROC statistics was observed for the Gini coefficient and the correct answer rate. The gradient boosting exhibits the least error than the other two models.

These results show that gradient boosting is the best credit scoring model, Nevertheless, the AUC-ROC statistic must be evaluated according to the context, so each analyst to select the credit scoring method.

Keywords: logistic regression, adaboost, gradient boosting, scoring, modelling, statistics, good client, bad client, decision trees.

Capítulo Uno: Introducción

El Sistema Financiero Nacional (SFN) es el conjunto de instituciones financieras reguladas por la Superintendencia de Bancos (SB) y por la Superintendencia de Economía Popular y Solidaria (SEPS). En particular, las instituciones financieras canalizan los movimientos de dinero que realizan los ciudadanos; así como, los créditos que obtienen las personas, familias u organizaciones que requieren financiamiento. Por tanto, estas contribuyen al desarrollo del país, fortaleciendo la inversión productiva y el consumo responsable (BanEcuador, 2016).

La SB regula la actividad financiera de las instituciones adscritas y determina que estas posean metodologías que consideren una combinación de criterios cuantitativos y cualitativos, de acuerdo con la experiencia y las políticas estratégicas de la entidad, de tal manera que permitan monitorear y controlar la exposición crediticia al riesgo de los diferentes portafolios (Superintendencia de Bancos y Seguros del Ecuador, 2014).

En particular, si se consideran los créditos dirigidos hacia las personas naturales, estas metodologías deben servir para determinar si los individuos son aptos o no para adquirir un crédito; por lo tanto, se requiere tener un criterio confiable, lo más preciso y exacto posible, que diferencie a personas con un mal historial crediticio de aquellas que se esperaría sean aptas para recibir un crédito y que lo paguen de manera regular.

En la medida que se cuente con buenos modelos de clasificación, el riesgo de pérdidas y desequilibrios de liquidez para las instituciones financieras será lo más pequeño posible. En conclusión: un modelo de clasificación crediticia que no es fiable solo crea problemas en una entidad financiera.

Entre los principales problemas que se pueden mencionar son: por un lado, el otorgamiento de crédito a personas que en realidad no son buenos pagadores, lo que producirá pérdidas directas, pues el dinero probablemente no se logre recuperar; por otro lado, no otorgar créditos a personas que, si pagan regularmente, lo cual generará problemas internos para la institución financiera, como la pérdida de prestigio.

Tradicionalmente, las instituciones financieras ecuatorianas han utilizado técnicas de clasificación y creación de modelos de scoring con base principalmente en la regresión logística (Superintendencia de Bancos y Seguros, 2014). Este método ha funcionado con éxito en la tarea de modelar los *credit scoring*; sin embargo, en la estimación de los modelos paramétricos se necesita una función de distribución conocida, en este caso la *función logit* (Alan, 2002). Si un conjunto de datos al ser modelado no se ajusta a la distribución propuesta, va a ser un arduo trabajo encontrar un modelo válido y esto se complica más cuando se considera un número grande de variables.

Por lo tanto, es necesario plantear metodologías alternativas que sean más robustas teóricamente y que disminuyan el error de estimación; por esta razón, este trabajo plantea utilizar las técnicas de clasificación con base en técnicas de *machine learning*, denominadas *boosting* (en este caso, se considerarán los métodos *Adaboost* y *Gradient Boosting*) y se busca analizar si estos métodos logran un mejor desempeño y resultados en el modelamiento de *credit scoring* que la regresión logística. Es decir, se presenta una alternativa al momento de crear modelos de *scoring*.

Para poder verificar si los métodos de *boosting*, planteados en este trabajo, son mejores en términos estadísticos que el modelo de regresión logística, se van a comparar los resultados de cada uno de los métodos con respecto a la regresión logística con el fin de analizar las ventajas y desventajas de estos nuevos modelos. Según Gareth (2007), utilizar modelos de *scoring* creados con técnicas de *boosting* es más eficaz, sencillo, y tienen mayor robustez predictiva que los modelos creados con regresión logística, por lo que, se espera obtener un mejor desempeño al momento de clasificar clientes en una institución financiera.

Esta investigación es posible gracias al conjunto de datos suministrado por una institución financiera ecuatoriana, de la cual, sólo se conocerá ese conjunto de datos, ya que por cuestiones de confidencialidad, no se dará a conocer información sobre los segmentos a los cuales dirige su actividad, ni el volumen de las carteras de crédito que maneja. Entonces, la investigación se limitará a la construcción de los modelos de *scoring* con las metodologías mencionadas.

Para lograr este objetivo, esta investigación se divide en cinco capítulos, que tratan las siguientes temáticas: el Capítulo Uno es la introducción del trabajo. El Capítulo Dos define el *credit scoring* y presenta las bases teóricas de los modelos que se desarrollarán, se establece el contexto en el cual será utilizado el *credit scoring* y se definen las herramientas teóricas usadas para comparar los modelos. El Capítulo Tres presenta el análisis exploratorio de las variables que se incluirán en los modelos de *credit scoring* y las fases que comprende el modelamiento. El Capítulo Cuatro presenta el proceso de construcción de los modelos de *credit scoring* y el análisis de los resultados que arroja cada uno en función de los estadísticos que se utilizarán para compararlos. Finalmente, el capítulo cinco presenta las conclusiones y recomendaciones que se desprenden de este trabajo.

Capítulo Dos: Marco Teórico

En el presente capítulo se presentan las bases teóricas de los métodos utilizados en este trabajo con los que se construyen los modelos de *credit scoring*. También, se introducen los conceptos propios que aparecen en su construcción; de tal forma que, la discusión esté auto contenida y se puedan comprender los resultados arrojados por cada uno de ellos. También, se presentan conceptualmente los criterios de comparación.

2.1 Modelos de *credit scoring*

Las instituciones financieras se dotan de los instrumentos de análisis necesarios para manejar los recursos financieros que ostentan, procurando maximizar la ganancia y reducir las pérdidas en procesos de adjudicación de créditos los cuales llevan un riesgo asociado. Para esto, es necesario contar con metodologías que ayuden a clasificar a los clientes antes de otorgar un crédito.

Estos modelos de clasificación, en su conjunto denominados *credit scoring*, son métodos que tienen fundamento en la estadística y son empleados en la clasificación de clientes que optan a un crédito en una determinada institución financiera en dos grupos: buenos y malos. Estos modelos han venido a reemplazar con éxito a los modelos que se utilizaban hasta la década de los años 70, los cuales dependían del juicio de analistas expertos con base únicamente en las decisiones que habían sido tomadas anteriormente; es decir, se dependía del juicio de valoración humano (Hand y Henley, 1997).

El concepto de *credit scoring* ha evolucionado y se ha ampliado para cubrir muchos aspectos dentro de la labor financiera del otorgamiento del crédito. Es por ello que, en esta investigación se delimita el concepto al ámbito de obtener **la probabilidad de que un cliente cumpla o no con las obligaciones contraídas**, calculándose a partir de un método estadístico con el que se analiza cierta información recabada sobre el cliente. La información de valor la establece la institución y para hacer uso del modelo debe cuantificarse mediante la definición de variables significativas. De esta forma, mediante un procedimiento técnico (que será explicado más adelante para cada caso en estudio), es posible asignarle un valor numérico al riesgo que representa otorgar un crédito a un determinado cliente.

Son muchas las aproximaciones que se pueden seguir para construir un modelo de *credit scoring*; sin embargo, en este trabajo se construirán tres modelos diferentes con una metodología clásica (regresión logística) y dos metodologías de reciente introducción en el ámbito financiero (*adaboost* y *gradient boosting*), para comparar sus ventajas y desventajas,

y así presentar una propuesta que pueda ser considerada al momento de que una institución quiera escoger entre los métodos de clasificación.

2.2. Variable dependiente y variables independientes

Para los tres métodos de estimación de modelos de *scoring* que se utilizan en este trabajo es necesario definir las variables que categorizan a un cliente como bueno o malo y los factores que influyen para que un cliente tenga uno u otro comportamiento.

En este caso, se define a la variable dependiente o respuesta, para cada cliente, como:

$$y_i = \begin{cases} 1 & bueno \\ 0 & malo \end{cases} \tag{1}$$

De manera general, se supone que para n individuos, $y_1, y_2, \dots, y_n$ son variables aleatorias independientes con distribución de Bernoulli de parámetros $p_1, p_2, \dots, p_n$, respectivamente. Por otro lado, se tiene las variables $X_1, X_2, \dots, X_k$ como regresores (factores) que influyen sobre la probabilidad p_i; es decir, p_i es una función de $x_{i1}, x_{i2}, \dots, x_{ik}$.

2.2 Modelo de regresión logística

Considerando la definición de y_i, donde tomar el valor 1, significa que el cliente es **bueno**, se puede escribir:

$$p_i = \Pr(y_i = 1) \tag{2}$$

Como la probabilidad de que el cliente i sea bueno y si se supone que p_i depende de algunos factores relacionados con el individuo i; en este caso puntual, pueden ser: edad, sexo, saldo promedio en las cuentas, estado civil, motivo del préstamo, etc. Por tanto, es necesario encontrar una relación funcional entre los p_i y los factores.

Por tanto, se busca:

$$p_i = f(x_{i1}, x_{i2}, \dots, x_{ik}; \beta_0, \beta_1, \dots, \beta_k) \tag{3}$$

donde, los $\beta_0, \beta_1, \dots, \beta_k$ son constantes desconocidas y f es la función de distribución de probabilidades logística:

$$p_i = \frac{\exp(\alpha_i)}{1 + \exp(\alpha_i)}, \quad i = 1,2, \dots, n \tag{4}$$

donde,

$$\alpha_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_k x_{ik} = x_i^t \beta \tag{5}$$

con:

$$x_i^t = (1, x_{i1}, \dots, x_{ik}) \qquad \beta^t = (\beta_0, \beta_1, \dots, \beta_k) \tag{6}$$

La función logística (4), es una función de distribución de probabilidad y por tanto, toma sus valores en el intervalo [0,1]. Si se despeja α_i se obtiene la *función logit*:

$$\alpha_i = logit(p_i) = ln\left(\frac{p_i}{1 - p_i}\right) \tag{7}$$

Con los coeficientes del modelo logístico se puede cuantificar el riesgo. Estos coeficientes son llamados *odds ratio*. El *odds* se define como:

$$odds = \frac{p_i}{1 - p_i} \tag{8}$$

Este escalar establece el cociente entre la probabilidad de que un evento ocurra y la probabilidad de que no ocurra.

Para obtener los coeficientes $\beta^t = (\beta_0, \beta_1, \dots, \beta_k)$ se utiliza el método de máxima verosimilitud (*Credit scoring*, s.f). Que se puede sintetizar de la siguiente manera: se sabe que y_i adopta dos valores: 1 con una probabilidad p_i, y 0 con una probabilidad $1 - p_i$; por lo cual, y_i se distribuye de acuerdo con:

$$Pr(y_i) = p_i^{y_i}(1 - p_i)^{(1-y_i)}, \qquad con\ y_i = 0,1\ y\ n = 1,2, \dots, n \tag{9}$$

Ahora, es necesario definir la probabilidad conjunta de los y_i, asumiendo que estos son independientes e idénticamente distribuidos, de la siguiente manera:

$$Pr(y_1, \dots, y_n) = \prod_{i=1}^{n} {p_i}^{y_i} (1 - p_i)^{(1-y_i)} \tag{10}$$

Aplicando logaritmo a ambos lados de esta expresión se obtiene:

$$\log Pr(y_1, \dots, y_n) = \sum_{i=1}^{n} y_i \log p_i + \sum_{i=1}^{n} (1 - y_i) \log(1 - p_i) \quad (11)$$

Ahora, por las propiedades de los logaritmos queda:

$$\log Pr(y_1, \dots, y_n) = \sum_{i=1}^{n} y_i \log\left(\frac{p_i}{1 - p_i}\right) + \sum_{i=1}^{n} \log(1 - p_i) \quad (12)$$

Finalmente, reemplazando (4), (5) y (7) en (12), se puede expresar el logaritmo de la función de verosimilitud, que depende de los parámetros, como sigue:

$$L(\beta) = \log Pr(\beta | y_1, \dots, y_n) = \sum_{i=1}^{n} y_i \, x_i^t \beta - \sum_{i=1}^{n} \log\left(1 + e^{x_i^t \beta}\right) \quad (13)$$

Los coeficientes β se obtienen a partir de esta expresión al derivar $L(\beta)$ con respecto a cada uno de los coeficientes e igualar a cero; con esto, se forma un sistema no lineal de ecuaciones que debe resolverse mediante métodos numéricos.

2.3 Árboles de decisión o clasificación

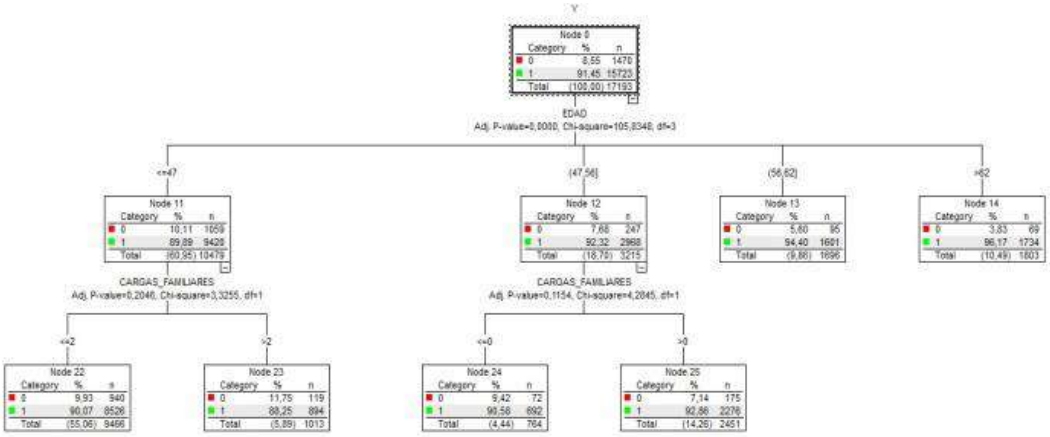
Para la estimación de los modelos de *credit scoring* en este trabajo, metodológicamente, es necesario definir un método con el que se obtengan "clasificadores débiles", que ayuden a categorizar las variables independientes; en este caso, se van a utilizar los árboles de decisión, que es una herramienta de clasificación muy popular y bastante útil para clasificar variables.

Los árboles de decisión son técnicas de exploración de datos que permiten identificar patrones dentro de un conjunto de datos. Representan un conjunto de reglas considerando algunas variables independientes según cómo se recojan en el árbol; para el desarrollo de este contenido teórico se ha adaptado el documento escrito por IBM Knowledge Center en el desarrollo de su herramienta Answer Tree 2015 y también a partir del trabajo denominado Arboles de Clasificación y Regresión por José Manuel Rojo (2006).

Rojo (2006), expone que los árboles se construyen a través de un algoritmo que divide la base de datos en grupos de manera recursiva, representados por nodos, de manera que con cada subdivisión las frecuencias relativas de las categorías de la variable dependiente vayan tendiendo a 0 o a 1.

Un árbol es un grafo conexo que tiene un nodo inicial (*Nodo Raíz*) y a partir de él se forman las aristas (ramas) que llegan a nuevos nodos y, estos a su vez, pueden formar otras aristas o ser nodos terminales.

Figura 1. Ejemplo del árbol de decisión.



Elaborado por: El autor

Por otro lado, es necesario exponer los algoritmos que sirven para desarrollar un árbol de decisión; a continuación, se describen los que se utilizarán en este trabajo.

2.3.1 Algoritmo CART

Es un algoritmo específico distribuido de manera comercial mediante *software*, resultado de la implementación de la propuesta de Breiman et al. (1984); este algoritmo está incluido en la librería *rpart* de Therneau et al (2015), en el ambiente estadístico desarrollado por *R core Team* (2017).

Este algoritmo construye árboles binarios (cada nodo se divide exactamente en dos hijos (ramas)), para modelos de clasificación y regresión. El algoritmo crea divisiones binarias en las variables predictoras separando las categorías y reduciendo la heterogeneidad para crear grupos de individuos con valores similares de respuesta, respecto a las clases previstas. La división que más reduce la heterogeneidad se escoge como nodo y este proceso se realiza de manera recursiva hasta que no se pueda mejorar la diferencia entre los valores estimados y los valores reales de la variable dependiente.

Un problema que surge al aplicar esta metodología es que se pueden generar árboles que se "sobreajusten"; por lo que, algunas partes de la estructura del árbol no pueden ser utilizadas para generalizar muestras alternativas. Es por eso que el algoritmo CART viene dotado de un parámetro de complejidad α que "poda el árbol", controlando su tamaño de la siguiente forma:

$$R_\alpha(T_p) = R(T_p) + \alpha s_p \quad (14)$$

donde, α es el parámetro de complejidad o de costo, $R(T_p)$ es el error, y s_p es el número de hojas de T_p (Breiman et al, 1984). Entonces para escoger un modelo se prueban varios valores de α mediante validación cruzada. El resultado de esta estrategia es que se obtiene un árbol más pequeño que el inicial, se gana en interpretación, pero se pierde rendimiento cuando se compara con otros métodos de predicción.

2.3.2 Algoritmo CHAID

El algoritmo CHAID (*Chi-square automatic interaction detection*) es un algoritmo propuesto originalmente por Kass (1980) con base en el estadístico *chi-cuadrado*, para variables dependientes categóricas. La idea de este método es calcular una probabilidad en el que valores cercanos a cero indican que existe una diferencia entre dos categorías de una variable de análisis y valores cercanos a uno, sugieren que no existe diferencia significativa entre las dos categorías. Este algoritmo permite utilizar tanto variables categóricas como continuas.

El algoritmo CHAID consiste de los siguientes pasos, Ramzai (2019):

i. Se debe escoger una variable predictora, esta puede ser cuantitativa o cualitativa. Si la variable es cualitativa se avanza al paso ii, en el caso que se trate de una variable cuantitativa, se debe crear categorías con aproximadamente el mismo número de casos y con esta nueva variable se sigue al siguiente paso.
ii. Se itera de manera cíclica todas las categorías de la variable predictora, una a la vez, con la finalidad de determinar las dos categorías predictoras con menor diferencia significativa con respecto a la variable dependiente.
iii. Si la prueba para un par de categorías de predictores no es estadísticamente significativa según el nivel de significación (α) definido, entonces se fusionan las categorías predictoras y se repite el paso i.
iv. Si la prueba para un par de categorías de predictores es estadísticamente significativa, entonces se calcula un $valor\ p$ de Bonferroni para el conjunto de categorías de la respectiva variable predictora.
v. La variable predictora con él $valor\ p$ más pequeño; es decir, la variable predictora que produce la división más significativa se considera para la siguiente división del árbol.
vi. Si el $valor\ p$ más pequeño para cualquier variable predictora es mayor que algún valor α definido para dividir el árbol, no se realizan más divisiones y el nodo asociado se convierte en nodo terminal.

Se continúa el proceso de manera iterativa hasta que no se puedan realizar más divisiones dado el valor de α para fusionar y el valor de α para dividir.

2.3.3 Estimador OOB

Cuando se construyen clasificadores, es necesario realizar varias estimaciones con diferentes muestras, generalmente, obtenidas con métodos *bootstrap*. Esto produce un error de predicción o clasificación al momento de estimar los árboles de clasificación. El estimador OOB (*Out of the Bag*), permite aprovechar las observaciones no incluidas en cada muestra *bootstrap* y proporciona una estimación insesgada del error de generalización.
Este estimador se puede calcular, aunque no se disponga de un conjunto de prueba (*test*) o control.

El algoritmo para obtener este estimador es el siguiente:

Se parte de un conjunto de entrenamiento $\boldsymbol{D} = \{D_i = (x_i, y_i), i = 1,2, \dots, n\}$ en donde y_i denota la clase para cada caso.

Para $b = 1, \dots, B$. Donde, b indica el número de iteración; es decir, el proceso se realiza B veces.

i. Se genera una muestra *boostrap* D^* a partir de D.

ii. Se considera $D_b = \{D_i / D_i \notin D^*\} = D - D^*$
   a. Se construye el modelo A_b sobre $\boldsymbol{D}^*$
   b. Se aplica A_b a cada elemento de $\boldsymbol{D}_b$.

iii. Se toma el siguiente b.
   a. Este procedimiento genera las predicciones para cada caso D_i a partir de los modelos cuyos conjuntos de entrenamiento no incorporan D_i.
   b. De manera similar a la validación cruzada, se obtiene la predicción para D_i proporcionada por la clase donde ese caso es clasificado más veces:

$$V_i = arg \underbrace{max}_{j \in \{1,2,\dots,K\}} \{\#[A_b(x_i) = j / D_i \in \boldsymbol{D}_b]\} \tag{15}$$

iv. Se define OOB como:

$$OOB = \frac{1}{n} \sum_{i=1}^{n} I(y_i \neq V_i) \tag{16}$$

Donde $I(y_i \neq V_i)$, la función indicatriz, que toma el valor de uno cuando el valor observado (y_i) es diferente al valor predicho (V_i), cero en caso contrario.

Es decir, la tasa de error OOB establece la proporción de los casos clasificados que no coinciden con su clase real, entre los modelos ajustados sobre muestras *bootstrap* que no lo contienen.

2.3.4 Importancia de las variables

Dentro de la creación de los clasificadores, es importante tener en cuenta cuál o cuáles son las variables que más aportan al momento de realizar la clasificación; es así, que para este caso se define la siguiente medida de importancia para árboles de decisión (Breiman et al, 1984):

$$\widehat{I_j^2}(T) = \sum_{t=1}^{J-1} \widehat{\imath_t^2}\, 1(v_t = j) \tag{17}$$

La sumatoria se realiza sobre los nodos t no terminales del nodo terminal J del árbol T, v_t es la variable divisoria asociada al nodo t y $\hat{\imath}_t^2$ representa la mejora empírica en el cuadrado del error como resultado de la división. La expresión (17, es una medida de la influencia de cada variable en un árbol. Ahora, si se considera una colección de árboles de decisión $\{T_m\}_1^M$, a partir de un proceso de *boosting*, la medida de influencia se puede generalizar por su promedio sobre todos los árboles:

$$\hat{I}_j^2 = \frac{1}{M} \sum_{m=1}^{M} \hat{I}_j^2\,(T_m) \tag{18}$$

Cuando se enfrentan problemas de regresión logística con K-clases, existen K funciones de regresión logística $\{F_{KM}(\boldsymbol{x})\}_{k=1}^K$, cada una descrita por una secuencia de M árboles. En este caso la expresión (18), pasa a ser:

$$\hat{I}_{jk} = \frac{1}{M} \sum_{m=1}^{M} \hat{I}_j^2\,(T_{km}) \tag{19}$$

T_{km} es el árbol inducido por la *k*-ésima clase en la iteración *m*. La cantidad $\hat{I}_{jk}$ puede interpretarse como la relevancia de la variable predictora x_j en la separación de la clase *k* de las otras clases. La relevancia total de x_j se obtiene promediando sobre todas las clases:

$$\hat{I}_j = \frac{1}{K}\sum_{k=1}^{K} \hat{I}_{jk} \tag{20}$$

Para ilustrar las ecuaciones (17), y (18), se consideran, por ejemplo, dos árboles de decisión, cada uno formado con las variables independientes edad y cargas familiares, pero con diferentes muestras, para obtener entonces dos árboles distintos. El ejemplo de los árboles se visualiza en la Figura 1, entonces para el cálculo de las ecuaciones se sigue los siguientes pasos:

Se denomina a cada árbol como T_1 y T_2, respectivamente. Para cada árbol se realiza el siguiente procedimiento. Se tiene 2 variables dependientes por tanto j es igual a 1 o 2 y J es igual al número de nodos no terminales.

1. Calcular el error de clasificación; este es el valor de referencia y se denomina Err_0.
2. Se retira una a una las variables y se vuelve a construir el árbol de decisión.
3. Calcular el error de clasificación del nuevo árbol, al que se denominan Err_j, con j igual 1 o 2 (j igual al número de variables independientes).
4. Calcular la mejora empírica $\hat{\imath}_t^2$, como $(Err_j - Err_0)/Err_0$, con j igual 1 o 2.

Entonces la ecuación (17) para cada árbol T_1 y T_2, se expresa de la siguiente manera:

Para el árbol T_1:

$$\widehat{I_j^2}(T_1) = \sum_{t=1}^{J-1} \widehat{\imath_t^2}\, 1(v_t = j)$$

Para el árbol T_2:

$$\widehat{I_j^2}(T_2) = \sum_{t=1}^{J-1} \widehat{\imath_t^2}\, 1(v_t = j)$$

Donde v_t es la variable divisora asociada al nodo t, es decir, si j=1, entonces $1(v_t = 1)$ toma el valor de 1 para los nodos divididos por la variable uno (edad), si j=2 asocia a los nodos divididos por la variable dos (cargas familiares).

Y la ecuación (18) es el promedio de la mejora empírica en cada una de las variables, para la variable j igual a uno se tiene el siguiente resultado para la ecuación (18):

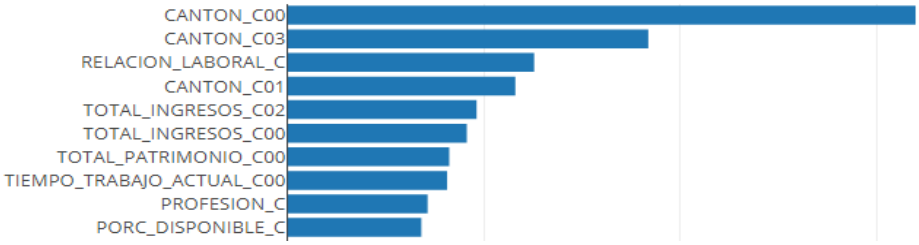
$$\hat{I}_1^2 = \frac{1}{2}\sum_{m=1}^{2} \hat{I}_1^2\,(T_m) = \frac{1}{2}\,(\widehat{I_j^2}(T_1) + \widehat{I_j^2}(T_2))$$

Y para la variable j igual a dos se tiene el siguiente resultado para la ecuación (18):

$$\hat{I}_2^2 = \frac{1}{2}\sum_{m=1}^{2} \hat{I}_2^2\,(T_m) = \frac{1}{2}\,(\widehat{I_2^2}(T_1) + \widehat{I_2^2}(T_2))$$

En la Figura 2, se muestra un ejemplo de cómo se visualiza gráficamente la importancia relativa calculada, descrita para 10 variables predictoras de este trabajo.

Figura 2. Importancia relativa de las variables predictoras.



Fuente: Elaboración propia, a partir de los datos reales

2.4 Métodos de *boosting*: aproximación basada en reglas

Es deseable dada una gran cantidad de información disponible, poder contar con mecanismos que permitan clasificarla y, sobre todo, conocer el proceso que la genera; de tal forma que, luego se puedan hacer predicciones o, en general, realizar un aprovechamiento consciente de esta.

Si esta información es tal que se puede expresar mediante cantidades medibles, se tiene un conjunto denominado datos de entrenamiento y en el caso de querer encontrar una regla para clasificar dicha información, regla que se puede entender como una asignación que relaciona números con ciertas propiedades (alto o bajo, frío o calor, mujer u hombre, pérdida o ganancia, etc), se está enfrentando un problema denominado problema de aprendizaje. Este tipo de problema se puede abordar mediante dos aproximaciones relevantes conocidas como enfoque basado en modelos y enfoque basado en reglas (Freund, 1998).

Para entender cómo funcionan estos enfoques, considérese que la información disponible se puede dividir en clases o categorías; estas se suelen etiquetar mediante el conjunto $Y = \{-1, +1\}$ para establecer la propiedad a clasificar correspondiente. Entonces, el enfoque con base en modelos comienza estimando un modelo estadístico que se le puede atribuir a cada clase.

Por otro lado, si se desea abordar el problema mediante el enfoque basado en reglas, ya no es necesario hacer estimaciones de modelos estadísticos, en cambio se proponen un conjunto de reglas denotadas con C, de las cuales se escoge la regla que cometa menos errores al momento de clasificar los datos de entrenamiento, en el proceso de relacionar los números con las propiedades en estudio.

Dentro del enfoque basado en reglas existen diversos métodos de aprendizaje entre los cuales está el *Boosting*. Este método de aprendizaje parte de un conjunto de reglas simples, las cuales son combinadas mediante un algoritmo, para formar una regla que se ajusta mejor al conjunto de datos. A continuación, se presenta el método de *Boosting* general y dos variantes conocidas como *Adaboost* y *Gradient Boosting*, en el contexto de *machine learning*.

2.4.1 Introducción al *boosting*

El poder de predicción de un modelo matemático requiere de técnicas de clasificación que aumenten la precisión en función del conocimiento disponible. Obtener una única regla clasificatoria con alta precisión es una tarea muy complicada; sin embargo, establecer reglas sencillas, cada una con precisión aceptable en el sentido de que mejoran el azar, es mucho más asequible. Si se cuenta con estas reglas sencillas, se puede considerar un método de clasificación conocido como *Boosting*.

La idea subyacente a este método es elegir en primera instancia un algoritmo que encuentre las reglas sencillas (clasificadores). Luego, el algoritmo *Boosting* prueba varias veces a estos clasificadores utilizando conjuntos de entrenamiento, considerando pesos sobre dicho conjunto; de tal manera que, se construya de manera artificial una distribución sobre el conjunto de datos. Cuando es requerido, el clasificador base genera una nueva regla y luego de muchas repeticiones de este proceso, el algoritmo *Boosting* combina las reglas sencillas en una regla que tiene mayor precisión (Alfaro et. al, s.f).

Durante este procedimiento se selecciona la distribución para cada iteración, asignándole más peso a las observaciones con mala clasificación que son resultado de iteraciones anteriores, esto con el objetivo de que el clasificador base opere sobre los casos más complicados. Al final del proceso, los clasificadores base se combinan ponderando las predicciones de cada uno de ellos, en un único clasificador final con mejor precisión que los clasificadores iniciales.

El primer algoritmo desarrollado para este fin, fue introducido por Schapire en 1990. En Alfaro et. al (s.f), se explica la idea de este algoritmo basada en la existencia de una regla por defecto con error asociado ϵ y la posibilidad de potenciar la precisión de un clasificador débil en el caso de que solo existan dos clases. El clasificador débil C_1 se construye sobre un conjunto de entrenamiento T de tamaño n y es superior a la regla por defecto ya que su error es menor; es decir, $\epsilon = 0{,}5 - \gamma$, siendo γ la ventaja que C_1 tiene sobre la regla por defecto.

El siguiente paso es tratar de eliminar la ventaja γ que tiene C_1; para esto, se construye un nuevo conjunto de entrenamiento T_2 a partir de T, del mismo tamaño, y se obliga a que en T_2 la mitad de las observaciones estén mal clasificadas por C_1. Esto se consigue mediante el lanzamiento de una moneda, dado que son dos clases. Si el resultado es cara, se extraen observaciones aleatorias del conjunto T hasta que se cumpla la condición $C_1(x_i) = y_i$, siendo x_i la observación e y_i la etiqueta; es decir, se clasificó correctamente. Si el resultado es cruz se extraen observaciones de T hasta que se cumpla $C_1(x_i) \neq y_i$, lo cual expresa una mala clasificación (Alfaro et. al, s.f).

Ahora, T_2 se utiliza para entrenar al clasificador C_2 bajo el mismo criterio que para C_1. El último paso del algoritmo es crear un T_3 del mismo tamaño de T y T_2, en el cual se eliminen las observaciones que cumplan $C_1(x_i) = C_2(x_i)$. Se entrena por tercera vez y se forma el clasificador C_3. En este caso, una observación x_i que cumpla $C_1(x_i) = C_2(x_i)$ le corresponde la clase asignada por C_1 y C_2; en caso contrario el clasificador final C_F le asigna la clase que indique $C_3(x_i)$.

Si bien el error del clasificador final es menor que el error inicial, este algoritmo es poco práctico en su implementación y poco eficiente computacionalmente. Cinco años después del primer algoritmo introducido por Schapire, Freund (1995) publicó un algoritmo que mejora la eficiencia y es más fácil de implementar. Se conservan elementos básicos del primer algoritmo, pero se modifica la estrategia de asignación de los pesos en cada iteración, esto incide en una reducción más rápida del error del clasificador final. Sin embargo, la estrategia de asignación de pesos tiene una dependencia en γ que ocasiona problemas en el caso de que esta ventaja sea muy pequeña, lo cual implica un aumento significativo del número de iteraciones. Tratando de salvar este inconveniente, una colaboración entre Freund y Schapire (1996) dio origen a un método de *Boosting* mejorado que logró eliminar esta dependencia, que se utiliza en la actualidad, conocido como *Adaboost* sobre el cual se ahondará en la siguiente sección.

2.4.2 *Adaboost*

Adaboost es un algoritmo que procura construir después de un proceso, un clasificador final fuerte. Las ideas que se presentan a continuación siguen de cerca el planteamiento original de Freund y Schapire (1997). Es importante destacar que el algoritmo a presentar es de modalidad de dos clases, ya que existen muchas variantes de *Adaboost*.

El problema de aprendizaje parte con el conjunto de entrenamiento:

$$M = \{(x_i, y_i): x_i \in \mathbb{R}^d, y_i \in \{-1, +1\}, i = 1, \dots, m\} \tag{21}$$

donde,

x_i: representa cada una de las observaciones o ejemplos
y_i: corresponde a las etiquetas

Nótese que, las observaciones pertenecen al espacio de las observaciones medibles, que es Euclídeo. Por citar un ejemplo de problema a resolver, x_i podría representar la edad de una persona e y_i si esta persona es mayor de edad o no.

Ahora, se define una distribución de probabilidades D sobre el conjunto de entrenamiento; de tal forma que, $\sum_i D(i) = 1$. Luego, se definen los clasificadores débiles como aplicaciones del espacio de las observaciones medibles en el conjunto de clase:

$$h_t: \mathbb{R}^d \rightarrow \{-1, +1\} \tag{22}$$

Ahora, es importante definir la tasa de error del clasificador débil $h_t(x)$ y se calcula de manera empírica sobre los datos del conjunto de entrenamiento de la siguiente forma:

$$\epsilon(h_t) = \frac{1}{m}\sum_{i=1}^{m} 1(h_t(x_i) \neq y_i) < \frac{1}{2} \tag{23}$$

Para conformar el clasificador final fuerte, se seleccionan T clasificadores débiles y T escalares α_t asociados a los clasificadores:

$$h = \{h_t: t = 1, \dots, T\} \;\; \alpha = \{\alpha_t: t = 1, \dots, T\} \tag{24}$$

Una combinación lineal de clasificadores débiles constituye el clasificador fuerte, con frecuencia, el conjunto de los clasificadores débiles $\mathcal{H} = \{h(x)\}$ es infinito. Entonces el clasificador fuerte está dado por:

$$f_T(x) = \sum_{t=1}^{T} \alpha_t \, h_t(x) \tag{25}$$

La respuesta de un clasificador fuerte se define como el signo de la combinación lineal presentada en (25):

$$H(x) = sign[f_T(x)] = sign\left[\sum_{t=1}^{T} \alpha_t h_t(x)\right] \tag{26}$$

donde, $sign(x)$, representa la función **signo**, que toma el valor de -1 si $x < 0$ y toma el valor de 1 en caso contrario.

El objetivo de *Adaboost* es escoger h y α que minimicen el error de clasificación empírico del clasificador fuerte, es decir:

$$(h, \alpha)^* = argminErr(H; D) = argmin \frac{1}{m} \sum_{i=1}^{m} 1(H(x_i) \neq y_i) \tag{27}$$

Con todos los elementos necesarios para establecer el algoritmo *Adaboost*, es momento de introducirlo. Se parte de $M = (x_i, y_i), \dots, (x_m, y_m)$ y se inicializa el algoritmo asignándole a todas las observaciones el mismo peso D_1, $D_1(i) = \frac{1}{m}$. Se repite para $t = 1, \dots, T$ y luego se entrena a los clasificadores débiles h_t con D_t. Luego se calcula el error pasado para cada clasificador débil:

$$\epsilon_t(h) = \sum_{i=1}^{m} D_t(i) 1(h(x_i) \neq y_i), \forall\, h \tag{28}$$

Se selecciona el clasificador débil con el mínimo error:

$$h_t = argmin_h \epsilon_t(h) \tag{29}$$

Se establecen los escalares α_t:

$$\alpha_t = \frac{1}{2} \ln\left(\frac{1 - \epsilon_t(h_t)}{\epsilon_t(h_t)}\right) \tag{30}$$

Se actualiza la distribución:

$$D_{t+1}(i) = \frac{1}{Z_t} D_t(i) exp[-\alpha_t y_i h_t(x_i)] \tag{31}$$

En esta expresión Z_t es un factor de normalización que se establece para mantener a D_{t+1} como una distribución. Para verificar la operación de una parte de este algoritmo, nótese que:

$$\begin{aligned} exp[-y_i \alpha_t h_t(x_i)] = exp[-\alpha_t] < 1 \; si \; h_t(x_i) = y_i \\ exp[-y_i \alpha_t h_t(x_i)] = exp[\alpha_t] > 1 \; si \; h_t(x_i) \neq y_i \end{aligned} \tag{32}$$

Esto significa que, los pesos de las observaciones clasificadas de manera correcta se reducen y los pesos de las observaciones clasificadas de manera incorrecta se incrementan; por tanto, recibirán más atención en la siguiente iteración. En la Figura 3, se muestra el algoritmo *Adaboost*.

Figura 3. Algoritmo *Adaboost*.

Algoritmo *Adaboost*
Adaboost (m, T) $Di[i] \coloneqq 1/\|m\|$,para todo $i = 1,2, \dots, \|m\|$ para todo t=1, 2,…, T repetir $h[t] \coloneqq hipotesisPesadaInfima(m, Dt)$ $e[t] \coloneqq errorPesadoHipotesis(m, h[t], Dt)$ $alfa[t] \coloneqq 1/2 \ln(1 - e[t]/e[t])$ -Para todo para todo $i = 1,2, \dots, \|S\|$ $D_{t+1}[i] = D_t[i] * \exp(-alfa[t] * Y_i * h[t](x_i))/Z_t$ -Donde Z_t es tal que suma$(D_{t+1}[i]) = 1$ para todo i finalizar para todo t retornar $h[T]$ fin Adaboost

Fuente: elaboración propia a partir de Mendoza, R. (2013).

2.4.3 *Gradient Boosting*

El *Gradient boosting* es un algoritmo basado en una técnica conocida como gradiente funcional descendente, que puede ser interpretado como una aproximación a la regresión logística (Alfaro et. al, s.f). En este caso, la idea es realizar ajuste de funciones al conjunto de datos disponible.

El problema a resolver es: encontrar la dependencia funcional $x \overset{f}{\Rightarrow} y$; es decir, cómo se mapean los datos desde el conjunto $x = (x_1, \dots, x_d)$ (conjunto de las variables de entrada), a las etiquetas correspondientes (conjunto de las variables respuesta). Para esto, se estima una función $\hat{f}(x)$, de tal forma que otra función $\Psi(y, f)$ (función de pérdida), sea mínima:

$$\hat{f}(x) = y, \qquad \hat{f}(x) = arg \underbrace{min}_{f(x)} \Psi(y, f(x)) \tag{33}$$

Este problema se puede interpretar en términos del valor esperado; es decir, se trata de minimizar el valor esperado de la función de pérdida sobre la distribución conjunta de todos los valores (y, x); o, de manera equivalente, sobre la variable respuesta $E_y(\Psi[y, f(x)])$ condicionada sobre las variables de entrada x (Natekin y Knoll, 2013):

$$\hat{f}(x) = arg \underbrace{min}_{f(x)} E_x\left[E_y(\Psi[y, f(x)])|x\right] \tag{34}$$

Dependiendo de la naturaleza de la variable respuesta y, se especifica la función de pérdida (Ψ). Por ejemplo, si $y \in \mathbb{R}$ se utilizan las funciones de error cuadrático $(y-f)^2$ y error absoluto $|y-f|$. Si $y \in \{-1,1\}$ se utiliza la función $\ln(1+e^{-2yf})$ y si $y \in \{0,1\}$ la función de pérdida binomial (Friedman, 2001). Por lo general, se escoge f perteneciente a una familia de funciones paramétricas $f(x,\theta)$ con la finalidad de facilitar el problema. Por lo tanto:

$$\hat{f}(x) = f\left(x,\hat{\theta}\right), \qquad \hat{\theta} = arg\,\underbrace{min}_{\theta} E_x\left[E_y(\Psi[y,f(x,\theta)])|x\right] \tag{35}$$

Ahora, el problema pasa a ser uno de estimación de parámetros; por tanto, es necesario introducir procedimientos numéricos. Para estimar los parámetros se utiliza una técnica conocida como *gradiente descendente por pasos*. Dado el conjunto de partida $(x,y)_{i=1}^{N}$ y una función empírica de pérdida $J(\theta)$, se establece un procedimiento para decrementar dicha función, la que se define sobre este conjunto de partida de la siguiente forma:

$$J(\theta) = \sum_{i=1}^{N} \Psi(y_i, f(x_i,\hat{\theta})) \tag{36}$$

La idea subyacente es mejorar la optimización a lo largo de la dirección establecida por el gradiente de la función de pérdida; es decir, a lo largo de $\nabla J(\theta)$. Los parámetros se construyen de forma incremental:

$$\hat{\theta}^t = \sum_{i=1}^{M} \widehat{\theta_\iota} \tag{37}$$

donde,
$\widehat{\theta_\iota}$: representa el i-ésimo paso incremental del estimado $\hat{\theta}$
$\hat{\theta}^t$: representa la suma de todos los incrementos estimados desde el paso 1 hasta el paso t.

Ahora, se pueden estimar los parámetros mediante la siguiente secuencia de pasos. Primero se inicializa la estimación de parámetros $\widehat{\theta_0}$. Para cada iteración t se repite:

1. Obtener $\widehat{\theta^t}$ de las iteraciones anteriores $\widehat{\theta^t} = \sum_{i=0}^{t-1} \widehat{\theta_\iota}$.
2. Evaluar $\nabla J(\theta)$ con los parámetros estimados obtenidos $\nabla J(\theta) = \{\nabla J(\theta_i)\} = \left[\frac{\partial J(\theta)}{\partial J(\theta_i)}\right]_{\theta=\widehat{\theta^t}}$.
3. Calcular el nuevo $\widehat{\theta_t}$, $\widehat{\theta_t} \leftarrow \nabla J(\theta)$.

4. Sumar $\widehat{\theta_t}$ al ensamble.

Así, se estiman los parámetros; sin embargo, el problema se trata de estimar la función $\hat{f}(x) = f(x, \hat{\theta})$. Para ello se considera el funcional:

$$\hat{f}(x) = \widehat{f^M}(x) = \sum_{i=0}^{M} \widehat{f_i}(x) \tag{38}$$

Este tipo de expansión, tal como lo comenta Friedman (2001), está en el corazón de muchos métodos de aproximación de funciones; por ejemplo, las redes neuronales. En la ecuación (38), M es el número de iteraciones y $\widehat{f_0}$ corresponde a un estimado inicial.

La idea ahora es obtener $\hat{f}(x)$ mediante iteración:

$$\widehat{f_t} \leftarrow \widehat{f_{t-1}} + \rho_t h(x, \theta_t) \tag{39}$$

donde se ha introducido la función $h(x, \theta)$ conocida como la **función base de aprendizaje** de la misma forma como se aborda el problema de parametrizar familias de funciones (Natekin y Knoll, 2013). En la expresión (39), ρ representa el tamaño de paso óptimo y obedece la regla:

$$(\rho_t, \theta_t) = arg \underbrace{min}_{\rho,\theta} \sum_{i=1}^{N} \Psi\left(y_i, \hat{f}_{t-1}\right) + \rho h(x_i, \theta) \tag{40}$$

Solo falta especificar la función de pérdida Ψ ya que se conoce la variable respuesta y; además, es necesario establecer el modelo de aprendizaje base con la función $h(x, \theta)$, para poner a funcionar la maquinaria. Sin embargo, no es esto lo que se hace, ya que en la práctica es complicado obtener la estimación de parámetros. Lo que se hace es introducir una función $h(x, \theta_t)$ que sea lo más paralela posible al gradiente negativo $\{g_t(x_i)\}_{i=1}^{N}$ a lo largo de los datos observados:

$$g_t(x) = E_y \left[\frac{\partial \Psi(y, f(x))}{\partial f(x)}\right]_{f(x)=\hat{f}^{t-1}(x)} \tag{41}$$

Al final, lo que se hace es escoger la nueva función incrementada que esté más correlacionada con $-g_t(x)$, esto se reduce a un problema de aproximación por el método de los mínimos cuadrados:

$$(\rho_t, \theta_t) = arg \underbrace{min}_{\rho,\theta} \sum_{i=1}^{N} [-g_t(x_i) + \rho h(x_i, \theta)]^2 \quad (42)$$

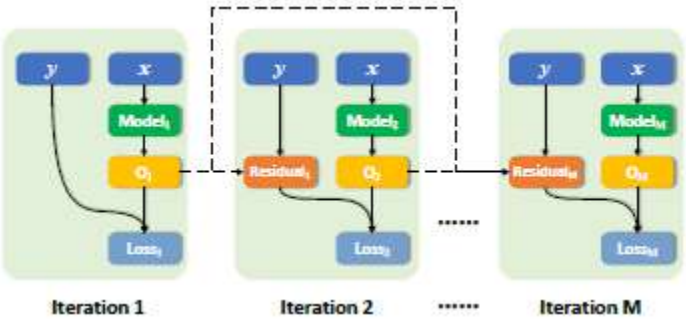
La Figura 4, condensa el procedimiento que se ha explicado y la Figura 5, ilustra su operación. Este procedimiento es una herramienta poderosa para resolver problemas de ajuste de datos y de clasificación. En particular, el modelo de redes neuronales de Fahlman y Lebiere (1989), es un caso de *gradient boosting*, ya que el modelo de aprendizaje base es una neurona y la función de pérdida es la de error cuadrático estándar (Natekin y Knoll, 2013).

Figura 4. Algoritmo *Gradient Boosting*.

Algoritmo Algoritmo *Gradient Boost* de Friedman
Entradas: -datos de entrada $(x, y)_{i=1}^{N}$ -número de iteraciones M -escoger la función de pérdida $\Psi(y, f)$ -escoger el modelo de aprendizaje base $h(x, \theta)$ **Algoritmo** 1. Inicializar $\hat{f}_0$ con una constante 2. **para** $t=1$ hasta M 3. Calcular el gradiente negativo $g_t(x)$ 4. Fijar una nueva función de aprendizaje base $h(x, \theta_t)$ 5. Encontrar el mejor tamaño de paso ρ_t $$\rho_t = \arg min_\rho \sum_{i=1}^{N} \Psi\left[y_i, \hat{f}_{t-1}(x_i) + \rho h(x_i, \theta_t)\right]$$ 6. Actualizar el estimado de la función $$\hat{f}_t \leftarrow \hat{f}_{t-1} + \rho_t h(x, \theta_t)$$ **terminar hacer**

Fuente: Elaboración propia a partir del algoritmo presentado en Natekin y Knoll (2013).

Figura 5. Procedimiento de cómo opera el algoritmo *Gradient Boosting*.



Fuente: Feng, Xu, Jiang y Zhou (2020).

2.4.4 *Ventajas y desventajas de los métodos con técnicas de boosting.*

Las principales ventajas de este algoritmo son (Corso (s.f)):

- Es rápido de evaluar por su naturaleza lineal, en lo que respecta a la construcción del clasificador fuerte.
- Puede ser rápido de entrenar si se escogen adecuadamente los clasificadores débiles.
- Solo tiene un parámetro ajustable T; es decir, el número de clasificadores débiles.
- Es efectivo si se puede encontrar de manera consistente a los clasificadores débiles.

Como desventajas se pueden señalar:

- El desempeño depende del conjunto de datos disponible y de los clasificadores débiles.
- Puede fallar si los clasificadores débiles son complejos y/o muy débiles; es decir, si están correlacionados y su error es tan cercano a 50% que prácticamente no añaden información.
- Fácilmente estas metodologías caen en un problema de sobreajuste, por tanto, se debe incorporar técnicas que minimicen o eliminen este problema.

2.5. Conceptos clave de la Modelación

Es necesario introducir algunos conceptos importantes que se utilizan al construir los modelos estadísticos de este trabajo:

2.5.1 Muestreo aleatorio simple sin reposición

En Capa (2015), se define el muestreo aleatorio simple sin reposición (MASSR), al procedimiento que consiste en seleccionar una muestra de tamaño n de una variable aleatoria X, que pertenece a una población de tamaño N; tal que, se obtiene un conjunto de datos $x_1, \dots, x_n$ que tienen la misma probabilidad de ser seleccionados. Si se desea estimar el tamaño de muestra *n*, correspondiente a una proporción poblacional *p*, con un error de estimación *B*, entonces se puede utilizar la siguiente aproximación:

$$n = \frac{Npq}{(N-1)D + pq}; \quad q = 1 - p;\ D \approx \frac{B^2}{4} \tag{43}$$

La expresión (43) se utiliza para realizar particiones de muestras grandes, manteniendo la representatividad en cada una de ellas.

2.5.2. Valor de información (IV) en *credit scoring*

Dentro de la formulación de un modelo de *credit scoring*, es útil contar con herramientas que permitan reducir la cantidad de variables al inicio del modelamiento, en particular de la regresión logística. Una de estas herramientas es el valor numérico conocido como Valor de Información (IV, por sus siglas en inglés). Típicamente, en un problema donde se busca establecer la dependencia de una variable binaria (y) y de una variable continua (x), el IV permite medir el poder de predicción de la variable independiente continua (Zeng, 2013). Para ello hace uso de la expresión:

$$IV = \sum_{i=1}^{n} \left(\frac{g_i}{g} - \frac{b_i}{b} \right) \times \ln \left(\frac{\frac{g_i}{g}}{\frac{b_i}{b}} \right) \tag{44}$$

donde,
g_i: denota el número de buenos clientes en la categoría i
b_i: es el número de malos clientes en la categoría i
n: es el número de categorías en los que se divide la variable independiente
g: es el número de clientes buenos en la población
b: es la cantidad total malos clientes en la población

Nótese que, el cálculo de IV depende de cómo se haga la categorización y del número de categorías que se utilicen; por tanto, se pueden obtener valores diferentes de IV para distintas formas de agrupar los datos. Es por ello que Zeng (2013), recomienda ordenar primero los datos y luego dividirlos en categorías que tengan la misma cantidad de observaciones. Aun así, se podría tener problema en el caso de que haya valores que se repitan. Es por ello que, se establece una regla para cuantificar IV en el trabajo de Siddiqi (2006), en el cual se explica el establecimiento de los rangos que caracterizan el IV, de la siguiente manera:

- $IV < 0{,}02$, no se puede predecir
- $IV \in (0{,}02;\ 0.1]$, la predicción es débil
- $IV \in (0{,}1;\ 0.3]$, la predicción es medianamente fuerte
- $IV > 0{,}3$, la predicción es fuerte

2.5.3. Criterio de información de Akaike

Este criterio desarrollado en el marco de la Teoría de la Información por Akaike (1974), calcula el Criterio de Información de Akaike (AIC), para cada modelo estimado de tal manera que el modelo óptimo es aquel con menor AIC. La expresión para el AIC está dada por:

$$AIC = -2\ln(máxima\ verosimilitud) + 2(n°\ parámetros\ independientes) \tag{45}$$

En Montalván (2019), se presenta la expresión para la máxima verosimilitud denotada con la letra L:

$$L = -\frac{n}{2}\ln 2\pi - \frac{n}{2}\ln \sigma^2 - \frac{1}{2}\sum_{t=1}^{n}\frac{e_t^2}{\sigma^2} \tag{46}$$

donde,
n: denota el número de datos
e_t: corresponde a los residuales
σ^2: representa el promedio de los residuales al cuadrado.

2.5.4. La prueba de Wald

El estadístico de Wald se define como el cociente (Montalván, 2019):

$$\omega_i = \frac{\hat{\beta}_i - \beta_i}{Var(\hat{\beta}_i)} \tag{47}$$

donde,
β_i: es el coeficiente estimado asociado a la variable independiente x_i
$\hat{\beta}_i$: denota el estimador de β_i
$Var(\hat{\beta}_i)$: es la varianza de $\hat{\beta}_i$

Este estadístico contrasta la hipótesis nula H_0: $\hat{\beta}_i = \beta_i$. Cuando ω_i es un valor lejano de cero, H_0 es falsa, entonces el estadístico sigue una distribución t – Student con $n - p - 1$ grados de libertad, siendo n el número de datos y p la probabilidad. En el caso de muestras grandes, la prueba de Wald sigue una distribución normal.

2.5.5. Generalización del factor de inflación de la varianza

Considere un modelo lineal:

$$y = \beta_1 \mathbf{1} + \beta_2 \boldsymbol{X_2} + \beta_3 \boldsymbol{X_3} + \epsilon \tag{48}$$

donde,
$\beta_1, \beta_2, \beta_3$: son constantes de regresión
1: corresponde a un vector de unos de tamaño n
X_2: es la matriz que contiene las columnas del efecto debido a la existencia de múltiples grados de libertad
X_3: es la matriz que contiene el resto de columnas del modelo matricial
ϵ: es el error

Se define el factor de inflación de varianza generalizada (GVIF) como sigue:

$$GVIF = \frac{\det(\boldsymbol{R}_{22})\,\det(\boldsymbol{R}_{33})}{\det(\boldsymbol{R})} \tag{49}$$

donde,
R_{22}: es la matriz de correlación entre las columnas de X_2
R_{33}: es la matriz de correlación entre las columnas de X_3
R: es la matriz de correlación para $[X_2, X_3]$
det: denota determinante.

GVIF es una medida que cuantifica la inflación de la varianza, de cada uno de los coeficientes de regresión, la cual se genera por la existencia de correlación en las variables del modelo (no se incluye la constante). El GVIF es utilizado comúnmente con variables del tipo categóricas con dos o más posibles valores y con variables polinomiales (Fox, 2003).

La colinealidad es un problema dentro de un modelo ya que expresa que existe información redundante; es decir, lo que un coeficiente de regresión explica sobre la respuesta del modelo, se puede superponer con la explicación de otro coeficiente de regresión u otro conjunto de regresores. Cuando el número de coeficientes de cada variable es uno, o que es lo mismo que la variable cuente con solo dos categorías, el GVIF coincide con el Factor de inflación de la varianza $(VIF) := \frac{1}{1-{R_i}^2}$, siendo ${R_i}^2$ el coeficiente de determinación, que representa la proporción de la varianza de la variable independiente i, que se asocia con las otras variables independientes en el modelo VIF (Yoo et al. 2014).

En este trabajo, se toma la recomendación de Yoo et al. (2014), de no utilizar valores de GVIF superiores a 10, pues estos pueden ser perjudiciales en el modelo de regresión.

2.5.6. El estadístico de Kolmogorov-Smirnov

Este estadístico permite contrastar las distribuciones empíricas de buenos y malos clientes. Se parte del supuesto de que el *score* (puntaje) está disponible para cada cliente en la cartera crediticia. Para establecer el estadístico de Kolmogorov-Smirnov (KS) se seguirán las ideas de *Rezac* y *Řezáč* (2011). Se inicia considerando lo siguiente:

$$D_k = \begin{cases} 1, buen\ cliente \\ 0, mal\ cliente \end{cases} \tag{50}$$

Se pueden definir las funciones de distribución para los buenos y malos clientes como sigue:

$$\begin{aligned} P_g(a) &= \frac{1}{n}\sum_{i=1}^{n} I(s_i \le a \wedge D_k = 1) \\ P_b(a) &= \frac{1}{m}\sum_{i=1}^{m} I(s_i \le a \wedge D_k = 0) \qquad a \in [L, H] \end{aligned} \tag{51}$$

donde,
s_i: es el puntaje del cliente i-ésimo
n: es el número de buenos clientes
m: corresponde al número de malos clientes
I: es un indicador tal que: $I(verdadero) = 1, I(falso) = 0$
L y H: son el mínimo y máximo de los puntajes de los clientes, respectivamente

Se introducen las proporciones de buenos y malos clientes; también, la función de distribución empírica de todos los clientes:

$$\begin{aligned} p_g &= \frac{n}{n+m} \\ p_b &= \frac{m}{n+m} \\ P_{todos}(a) &= \frac{1}{N}\sum_{i=1}^{N} I(s_i \leq a) \quad a \in [L, H] \end{aligned} \tag{52}$$

siendo $N = n + m$. El estadístico KS es una medida de la diferencia de las dos distribuciones empíricas:

$$KS = \underbrace{max}_{a\in[L,H]} \left|P_b(a) - P_g(a)\right| \tag{53}$$

Si las muestras son grandes, se puede calcular el valor crítico del KS bajo el supuesto de que las distribuciones son normales, para un valor de significancia α, de la siguiente manera (Rodó, s.f):

$$VC_\alpha = vc_\alpha \sqrt{\frac{n+m}{n.m}} \tag{54}$$

Entonces,

$$Si\ KS > VC_\alpha\ las\ distribuciones\ empíricas\ son\ diferentes$$

$$Si\ KS < VC_\alpha\ las\ distribuciones\ empíricas\ son\ iguales$$

2.5.7. Coeficiente de Gini

Este estadístico es útil cuando se pretende distinguir entre buenos y malos clientes en un modelo de *credit scoring*. Adopta valores entre 0 y 1, entendiéndose que si el valor es 1, el modelo es capaz de separar de manera perfecta a los buenos y malos clientes (Anderson, 2007).

El coeficiente de Gini se calcula con la expresión:

$$G = \left| 1 - \sum_{i=L}^{H} [p_b(i+1) - p_b(i)]\left[p_g(i+1) + p_g(i)\right] \right|, \quad i \in [L, H] \tag{55}$$

donde,
i: es el índice de score
L: es el score mínimo
H: es el score máximo en la población de clientes en estudio.
$p_b(i)$: es la proporción de clientes malos con puntaje menor o igual a i
$p_g(i)$: es la proporción de clientes buenos con puntaje menor o igual a i.

2.5.8. Matriz de confusión

La idea de la matriz de confusión es contrastar los valores reales disponibles en el conjunto de datos con las predicciones del modelo que se escoja para clasificar clientes. Una vez escogido el modelo, el analista establece un punto de corte dentro del conjunto de los *scores* disponibles. Entonces, los *scores* por debajo del punto de corte se clasifican como correspondientes a malos clientes y en caso contrario se clasifican como buenos clientes. Ahora, se construye una matriz 2x2 cuyas entradas serán las siguientes:

- Mal cliente en el conjunto de datos y mal cliente predicho por el modelo
- Buen cliente en el conjunto de datos y buen cliente predicho por el modelo
- Mal cliente en el conjunto de datos y buen cliente predicho por el modelo
- Buen cliente en el conjunto de datos y mal cliente predicho por el modelo

Según Montalván (2019), los casos correctamente clasificados se denominan verdaderos positivos (buenos) y verdaderos negativos (malos). Si no son bien clasificados se tiene a los falsos positivos (malos que son clasificados como buenos) y a los falsos negativos (buenos que son clasificados como malos).

En la Figura 6, se presenta la forma de la matriz de confusión y en la Figura 7, los indicadores que se utilizan para comparar modelos con los datos reales disponibles.

Figura 6. Matriz de confusión para comparar valores reales con valores estimados por un determinado modelo.

Matriz de Confusión		**Real**	
		0: Malo	**1: Bueno**
Estimado	**0: Malo**	A	B
	1:Bueno	C	D

Fuente: Montalván (2019).

Figura 7. Indicadores de eficiencia utilizados para comparar el poder de predicción entre modelos.

Indicador de eficiencia	**Definición**	**Fórmula**
Tasa de aciertos	Razón entre las predicciones correctas y el total	$\frac{A+D}{A+B+C+D}$
Error	Razón entre las predicciones incorrectas y el total	$\frac{B+C}{A+B+C+D}$
Sensibilidad	Razón entre los clasificados correctamente y el total de buenos	$\frac{D}{B+D}$
Especificidad	Razón entre los malos clasificados correctamente y el total de malos	$\frac{A}{A+C}$

Fuente: Elaboración propia con datos de Montalván (2019).

2.5.9. La curva ROC

El objetivo de un *credit scoring* es diferenciar los buenos clientes de los malos clientes. Para lograr esto, se cuenta con una herramienta estadística conocida como *la curva ROC* (Valle, s.f). Una curva ROC (Receiver Operator Characteristic) se grafica considerando la *sensibilidad* y con el cálculo de (1-*especificidad*). La sensibilidad (56) y la especificidad (57) vienen dadas por:

$$S = P(y = 1|D = 1) = \frac{P(y = 1 \cap D = 1)}{P(D = 1)} \tag{56}$$

$$E = P(y = 0|D = 0) = \frac{P(y = 0 \cap D = 0)}{P(D = 0)} \tag{57}$$

donde,
y: es una variable aleatoria de Bernoulli, cuando $y = 1$ el caso es positivo.
D: representa la variable aleatoria *estado*, cuando $D = 1$ el individuo es mal cliente.

Entonces la curva ROC viene dada por:

$$ROC(c) = \begin{cases} y = S(c) \\ x = 1 - E(c) \end{cases} \tag{58}$$

donde, c representa el punto de corte establecido para diferenciar los casos positivos y negativos, es decir:

$$\begin{cases} Positivo \equiv y = 1, si\ x \geq c \\ Negativo \equiv y = 0, si\ x < c \end{cases} \tag{59}$$

Ahora, se puede calcular el área bajo la curva (AUC) mediante la expresión:

$$AUC = \int_0^1 ROC(t)\delta(t) \tag{60}$$

El valor de esta integral permite discriminar a los buenos de los malos clientes. Nótese que los límites de integración se explican por estar integrando una curva de probabilidad. Cuando $AUC = 0{,}5$ la prueba no discrimina entre buenos y malos clientes, si $AUC = 1$ se discrimina de manera perfecta. Para valores intermedios existen diversos criterios de interpretación; por ejemplo: Swets (1988), establece los puntos $0{,}7$ y $0{,}9$ como puntos relevantes. Si $AUC < 0{,}7$ la exactitud es baja, si $0{,}7 < AUC < 0{,}9$ este valor puede ser útil según el contexto y si $AUC > 0{,}9$ la exactitud es alta.

2.5.10 Método ROSE

Es inherente a los métodos de clasificación que su desempeño se vea afectado en la medida que exista sesgo en la distribución de las clases; también, se afecta cuando se escogen estimadores de baja calidad de la medida de exactitud, ya que se dificulta el entendimiento del proceso de aprendizaje. Estos problemas se pueden superar mediante la aplicación de una estrategia que consiste en alterar la distribución de las clases con el objetivo de crear balance en la muestra. Esta estrategia es aplicable a cualquier método de clasificación y se conoce como *Random Over Sampling Examples* (ROSE); su éxito radica en la creación de datos de manera artificial a partir del enfoque de *boostrap* suavizado (Menardi y Torelli, 2010).

Para aplicar esta estrategia, se consideran inicialmente $\mathcal{X}$ dominios pertenecientes a R^d, también se considera que $f(x) = P(x)$ es una función de densidad de probabilidad tal que $f(x) \in \mathcal{X}$. Si $n_j < n$ es el tamaño de $\mathcal{Y}_j, j = 0,1$. Los pasos para generar los datos de manera artificial mediante ROSE son los siguientes:

1) Se escoge $y = \mathcal{Y}_j \in \mathcal{Y}$ con probabilidad $\frac{1}{2}$

2) Se escoge (x_i, y_i) en el conjunto de entrenamiento, $y_i = y$ con probabilidad $p_i = \frac{1}{n_j}$

3) Se escoge $\boldsymbol{x}$ de $K_{\mathbf{H}_j}(., x_i)$ con $K_{\mathbf{H}_j}$una distribución de probabilidad con centro en x_i y $\mathbf{H}_j$ representa una matriz de parámetros.

Lo que se hace es sacar una observación del conjunto de entrenamiento perteneciente a una de las dos clases y se genera un nuevo ejemplo en la vecindad con ancho determinado por $\mathbf{H}_j$. Repetir los pasos 1)-3) genera un conjunto de entrenamiento artificial con tamaño m con una cantidad de ejemplos similares para ambas clases; de tal forma que, el clasificador estime mejor la regla de clasificación, ya que se presta la misma atención a las dos clases (Menardi y Torelli, 2010).

Capítulo Tres: Selección de variables

Para realizar los modelos considerados en este trabajo, se dispone de un conjunto de datos generado durante tres años (tiempo conocido como período de modelización), consistente en la información de 21.492 clientes. El periodo de modelización fue previamente determinado por la institución financiera; es decir, la institución financiera al proporcionar la base de datos, esta ya tenía la variable dependiente y las variables independientes con las que contaba la institución financiera en el período especificado.

Metodológicamente, se realiza una partición de la muestra en dos partes: la primera corresponde al conjunto de datos de "entrenamiento" y, la segunda, se utilizará como un conjunto de datos de validación. El objetivo de esta partición es utilizar el conjunto de entrenamiento para realizar el ajuste de los modelos y el conjunto de validación para evaluar la complejidad de los mismos (Siddiqi, 2006).

Generalmente, se toma un porcentaje del período de modelación en el rango 70% - 80% para obtener el conjunto de entrenamiento y el restante 30% - 20% para la validación del modelo. Ahora, utilizando muestreo aleatorio simple sin reposición (MASSR) con distintos valores del error de estimación (B) (ver Tabla 1), se determina que el tamaño de la muestra de modelamiento es 17.193 (80% de la base, aproximadamente) y los restantes 4.299 clientes (20%, aproximadamente) forman la base de validación o *testing*.

Tabla 1. Tamaño de la muestra según el error de representatividad B.

B	n	% Muestra
0,45%	8.988	41,8%
0,41%	9.974	46,4%
0,37%	11.076	51,5%
0,33%	12.294	57,2%
0,29%	13.622	63,4%
0,25%	15.036	70,0%
0,21%	16.495	76,7%
0,19%	**17.193**	**80,0%**
0,18%	17.221	80,1%
0,17%	17.932	83,4%
0,13%	19.256	89,6%
0,09%	20.359	94,7%
0,05%	21.129	98,3%
0,00%	21.492	100,0%

Fuente: elaboración propia a partir del conjunto de datos disponible.

La Tabla 1, muestra el error de estimación (B) en la primera columna; la segunda columna, corresponde a los valores del tamaño de la muestra (n) resultante para un determinado B y;

la tercera columna, corresponde al porcentaje de participación de la muestra en la muestra total (21.492 clientes).

Una vez dividida la base de datos, es necesario analizar la cantidad de clientes buenos y malos que existen en cada uno de los conjuntos de datos. En la Tabla 2, se muestra la composición de los conjuntos de datos:

Tabla 2. Composición de los conjuntos de datos

	0:Malo	**1:Bueno**	**Total**	**% Total**
Modelamiento	1.470	15.723	17.193	80%
Testing	368	3.931	4.299	20%
Total	1.838	19.654	21.492	100%

Fuente: elaboración propia a partir del conjunto de datos disponible.

Nótese que, la proporción entre malos y buenos clientes es de 1:9, tanto en el conjunto de modelamiento como en el de *testing*; es decir, por cada cliente malo hay nueve buenos. Esta es una proporción adecuada, ya que un 10% de clientes malos es un porcentaje considerable que puede comprometer la sostenibilidad del negocio crediticio en el tiempo (Siddiqi, 2006). Adicionalmente, la variable dependiente está dada y que no existen valores indeterminados (ver Tabla 2).

Ahora, es necesario definir las variables independientes que se utilizan en el desarrollo de los modelos.

3.1 Variables independientes

Se denomina variable independiente a la característica del cliente que permitirá clasificarlo como bueno o malo. Esta característica puede ser de tipo demográfica, de comportamiento evaluado en entornos internos o externos a la institución, producto de la operación desembolsada (Bambino, 2005).

Las variables en estudio pueden ser cualitativas (género, ocupación, estudios, etc.), o cuantitativas (nivel de ingresos, edad, número de cargas familiares, etc.). Estas variables, sean cualitativas o cuantitativas, para incorporarlas al modelo, primero se realiza una categorización de las mismas y, luego, cada una de las categorías da lugar a una nueva variable dicotómica, de acuerdo a las metodologías de *credit scoring*. Con las variables categorizadas los modelos son más estables, porque exhiben menor variabilidad y se dicotomizan para asegurar tener en cada categoría una variable binaria (Bambino, 2005). Para categorizar las variables se emplearán los árboles de decisión.

Previo categorizar las variables, es necesario realizar un análisis de tipo exploratorio que ayude a comprender cómo está conformada la base de datos. Dentro del análisis se realizan los siguientes pasos:

- **Verificar valores perdidos:** Un porcentaje superior al 5% de valores perdidos dentro de una determinada variable es criterio para no considerarla en el modelo (Weldon, 2011).
- **Inconsistencia en las variables**: Se revisa la naturaleza propia de la variable, se excluye la posibilidad que adopte ciertos valores; por ejemplo, la edad de un cliente no puede tener un valor negativo.

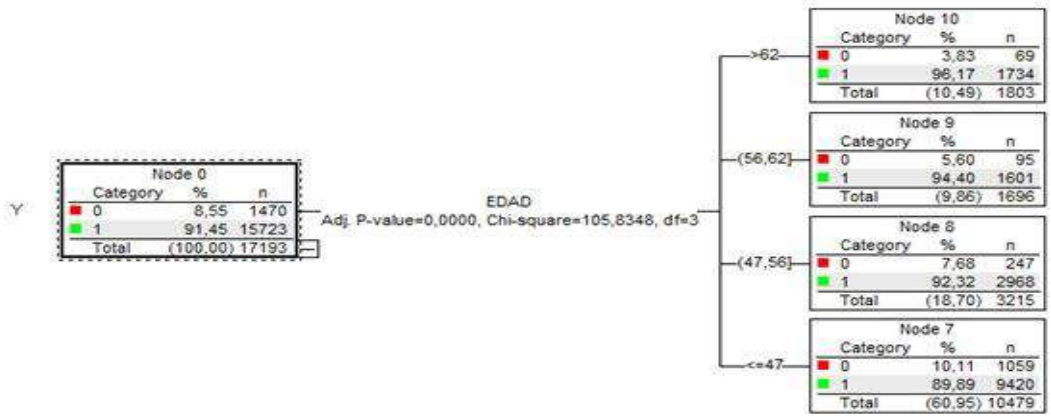
  También, se estudia el dominio de la variable, para que quede correctamente definido; por ejemplo, si la variable es de tipo porcentual debe estar comprendida en el dominio de 0 a 100.

3.2.1 Categorización de las variables

Como ya se dijo antes, para categorizar las variables se utiliza los árboles de decisión, se procura crear grupos con un mínimo del 5% del total de los registros (Weldon, 2011). En la Figura 8, se presenta el árbol de decisión creado para la variable EDAD como ejemplo y en el Anexo 2.1 los árboles de decisión para las demás variables consideradas en este trabajo. Estos árboles fueron creados mediante el programa estadístico SPSS *AnswerTree* v3.0 con algoritmo CHAID.

A continuación, se describirá brevemente el proceso de categorización para una variable con el algoritmo CHAID. Se toma una variable cuantitativa, por ejemplo: la EDAD. Primero se procede dividir en *N* categorías de esta variable, donde cada categoría tiene el mismo número o aproximadamente el mismo número de individuos en cada categoría. En una segunda fase el algoritmo empieza a iterar cíclicamente todas categorías y se busca una pareja de categorías que su diferencia no sea estadísticamente significativas a un nivel de significancia (α), para que así el algoritmo procede a unir estas categorías y este proceso se repite hasta que se logre el "mejor resultado".
Se debe considerar que si la prueba de significancia se la realiza a un grupo de categorías, se utiliza un *p* valor de Bonferroni el cual consiste en dividir el nivel de significancia (α), entre el número de comparaciones dos a dos realizadas. Con esta corrección se asegura que la probabilidad de obtener al menos un falso positivo entre todas las comparaciones es $\leq \alpha$ y con este *p valor* se evalúa si el grupo de categorías presenta o no diferencias estadísticamente significativas a un nivel de significancia (α).

Figura 8. Árbol de decisión para la variable EDAD.



Fuente: elaboración propia a partir de la base de datos suministrada por la institución financiera.

Este árbol establece cuatro categorías para la variable EDAD, permitiendo explicar la variable dependiente Y, ya que se rechaza hipótesis nula de independencia de estas dos variables, con nivel de confianza del 95%. Así, se determina que la categorización de la variable EDAD es estadísticamente significativa a un nivel de significancia (α).

Ahora, se debe hacer un análisis para saber si esta variable es apta económicamente para continuar al siguiente análisis, como ejemplo en la Figura 8, se puede ver el árbol de decisión para la variable EDAD es el resultado de aplicar el algoritmo CHAID con el programa estadístico SPSS *AnswerTree* v3.0.

La estructura del árbol es la siguiente: se tiene un nodo raíz y cuatro nodos terminales, y cada nodo contiene una tabla resumen, el nodo raíz corresponde al resumen de la base y los otros cuatro nodos corresponden a cada categoría respectivamente. Las tablas resumen de los nodos están formadas por 3 columnas, que son las siguientes: la columna ***Category*** muestra las categorías de la variable dependiente malos (0) y buenos (1), la columna ***%*** muestra el porcentaje de malos (0) y buenos (1) en el nodo **y** la columna ***n*** que muestra el número de casos malos (0) y buenos (1) en el nodo. En un análisis simple al árbol de decisión de la EDAD se ve que en la categoría ≤ 47, el porcentaje de malos (0) es 10,11%, mientras que en el resto de las categorías los porcentajes de malos van disminuyendo de la siguiente forma: 7.68% en la categoría (47; 56], 5.60% en la categoría (47; 56] y 3.83% en la categoría ≥ 62, con esto se puede notar que esta variable ordena a los individuos malos pagadores según la edad, de tal forma que a mayor edad, el riesgo de otorgarle un crédito es menor y esto tiene un sentido económico correcto, pues se esperaría que a mayor edad la persona sea más

responsable y por tanto sea un cliente menos riesgoso. Con este análisis esta variable puede continuar a las siguientes pruebas.

3.3 Variables predictoras

Las variables independientes con poder de predicción que se usarán para construir los modelos se eligen a partir del estadístico IV calculado con el programa R. A manera de ejemplo, se presenta el cálculo de este estadístico en el caso de la variable EDAD (ver Tabla 3), en la que se determina que el IV es superior al 0,3 % e inferior al 10% para esta variable; por tanto, es un predictor débil de la variable dependiente Y. En este trabajo, se consideran variables con poder de predicción débil o superior.

Tabla 3. Cálculo del IV para la variable EDAD.

Categorías	**Malos**	**Buenos**	$\boldsymbol{c_i = \%b_i - \%m_i}$	$\boldsymbol{d_i = \ln(\%b_i) / \ln(\%m_i)}$	$\boldsymbol{IV = c_i * d_i}$
0	1.334	11.809	-12,49%	-18,89%	2,4%
1	303	3.726	2,47%	13,98%	0,3%
2	111	1.948	3,87%	49,54%	1,9%
3	90	2.171	6,15%	81,35%	5,0%
Total	**1.838**	**19.654**			**9,6%**

Fuente: elaboración propia a partir del conjunto de datos disponible.

En la Tabla 4, se presentan los porcentajes del IV para las 15 variables que resultaron con un poder predictivo de débil o superior.

Tabla 4. Variables con un IV superior a débil.

N°	Variable	IV
1	CANTON_C	41.52%
2	ESTADO_CIVIL_EDAD_C	10.12%
3	CARGAS_FAMILIARES_EDAD_C	10.11%
4	EDUCACION_EDAD_C	9.90%
5	Edad_C	9.58%
6	TOTAL_INGRESOS_C	8.42%
7	TIPO_VIVIENDA_TIEMPO_VIVIENDA_C	5.69%
8	TOTAL_PATRIMONIO_C	5.43%
9	TIEMPO_TRABAJO_ACTUAL_C	5.26%
10	TIPO_VIVIENDA_C	4.78%
11	PROFESION_C	4.31%
12	ESTADO_CIVIL_C	2.75%
13	RELACION_LABORAL_C	2.55%
14	VALOR_VIVIENDA_C	2.53%
15	PORC_DISPONIBLE_C	2%

Fuente: elaboración propia a partir del conjunto de datos disponible.

En total, se cuenta con 52 nodos totales para las 15 variables, cada uno corresponde a una variable dicotómica. Con fin de evitar un problema de combinación lineal se retiran 15 variables dicotómicas. Finalmente, se obtiene como resultado 37 variables dicotómicas con las que se procede a crear los tres modelos de *score* en el siguiente capítulo.

Capítulo Cuatro: Análisis y Construcción de Modelos

En este capítulo, se construyen los modelos de *score* con las metodologías de regresión logística, *Adaboost* y *Gradient boosting*. Luego de realizar la construcción de los modelos, se hace una evaluación estadística de los mismos. Los tres modelos se desarrollaron con el programa estadístico R.

4.1 Regresión logística

El modelo de regresión logística debe trabajar con variables significativas con un nivel de confianza superior al 95%, los signos de los coeficientes de las variables deben ser consistentes y el modelo resultante debe cumplir las pruebas de significancia de los coeficientes y de multicolinealidad.

4.1.1 Variables del modelo de regresión logística

En esta sección se confirma que los estimados de los parámetros tienen coherencia en la interpretación de signos. El signo negativo de un coeficiente implica que la variable penaliza al cliente, elevando el riesgo de no pago (*default*). A continuación, se presentan las variables consideradas en el modelo:

- CANTON_C00: Variable explicativa que adopta el valor 1 si el cantón de residencia del cliente está dentro del grupo de cantones con riesgo bajo y el valor 0 en el caso contrario.
- CANTON_C01: Variable explicativa que adopta el valor 1 si el cantón de residencia del cliente está dentro del grupo de cantones con riesgo medio y el valor 0 en el caso contrario.
- CANTON_C03: Variable explicativa que adopta el valor 1 si el cantón de residencia del cliente está dentro del grupo de cantones con riesgo alto y el valor 0 en el caso contrario.
- EDAD_C04: Variable explicativa que adopta el valor 1 si el cliente es mayor a 62 años y el valor 0 en el caso contrario.
- EDUCACION_EDAD_C00: Variable explicativa que adopta el valor 1 si la educación de la persona es: no aplica, secundaria, primaria o técnica con edad menor o igual a 47 años y adopta el valor 0 en el caso contrario.
- EDUCACION_EDAD_C04: Variable explicativa que adopta el valor de 1 si la educación de la persona es: universitaria o posgrado, con edad menor o igual a 51 años y adopta el valor 0 en el caso contrario.
- PORC_DISPONIBLE_C: Variable explicativa que se obtiene al realizar el siguiente calculo: se resta a los ingresos totales los egresos totales, dividiendo el resultado entre los ingresos totales, si este valor es mayor a 0,4999942, la variable adopta el valor 1 y el valor 0 en el caso contrario.

- PROFESION_C: Variable explicativa que adopta el valor 1 cuando la profesión del cliente pertenece al grupo de profesiones de riesgo alto y el valor 0 en el caso contrario.
- RELACION_LABORAL_C: Variable explicativa que adopta el valor 1 cuando la persona está en relación de dependencia y el valor 0 en el caso contrario.
- TIPO_VIVIENDA_C02: Variable explicativa que adopta el valor 1 si el tipo de vivienda del cliente es arrendada y el valor 0 en el caso contrario.
- TOTAL_INGRESOS_C00: Variable explicativa que adopta el valor 1 si el cliente tiene ingresos totales anuales inferiores a $79.571,98 y el valor 0 en el caso contrario.
- TOTAL_INGRESOS_C02: Variable explicativa que adopta el valor 1 si el cliente tiene ingresos totales mayores a $520.000 y el valor 0 en el caso contrario.
- TOTAL_PATRIMONIO_C02: Variable explicativa que adopta el valor 1 si el patrimonio total del cliente es menor o igual a $6.500 y el valor 0 en el caso contrario.
- TOTAL_PATRIMONIO_C03: Variable explicativa que toma el valor 1 si el patrimonio total del cliente es mayor a $111.000 y el valor 0 en el caso contrario.
- NUMERO_CHEQUES_PROTESTADOS_C: Variable explicativa que toma el valor de 1 si el número de cheques con protestos son más de uno y el valor 0 en el caso contrario.

En la Tabla 5 se presentan los p-valor determinados mediante la prueba de Wald (columna $Pr(>|z|)$); se observa que todas las variables son significativas con un nivel de significación del 5% ($p - valor < 0{,}05$). También, se puede observar que el GVIF para todas las variables independientes del modelo de regresión logística. Nótese, que en todos los casos el valor es inferior a 10; por tanto, el modelo no presenta problemas de correlación entre las variables.

Tabla 5. Modelo *scoring* originación – Metodología Logit

Variables	Definición	Grupo-Árbol	B	Pr(>|z|)	Odds Ratio	VIF
Constante			*2,035	0,00		
CANTON_C00	Cantón	Riesgo Bajo	1,090	0,00	2,97	1,531
CANTON_C01		Riesgo medio	0,616	0,00	1,85	1,262
CANTON_C03		Riesgo Alto	-0,540	0,00	0,58	1,420
EDAD_C04	Edad	EDAD>62	0,488	0,00	1,63	1,105
EDUCACION_EDAD_C00	Educación y Edad	NO APICA, S-Secundaria P-Primaria, T -Formación intermedia(técnica) **and** <=47	-0,165	0,01	0,85	1,187
EDUCACION_EDAD_C04		U-universitaria, G-POSTGRADO **and** <=51	0,386	0,00	1,47	1,114
PORC_DISPONIBLE_C	(T. Ingresos - T. Egresos) / T. Egresos	>0,4999942	0,286	0,00	1,33	1,190
PROFESION_C	Profesión	Riesgo Alto	-0,334	0,00	0,72	1,041
RELACION_LABORAL_C	Relación Laboral	Dependiente	0,394	0,00	1,48	1,403
TIPO_VIVIENDA_C02	Tipo de vivienda	Arrendada	-0,374	0,00	0,69	1,031
TOTAL_INGRESOS_C00	Ingresos totales	<=79571,98	-0,503	0,00	0,60	1,583
TOTAL_INGRESOS_C02		>520000	1,277	0,00	3,59	1,192
TOTAL_PATRIMONIO_C02	Patrimonio Total	<=6500	0,302	0,00	1,35	1,507
TOTAL_PATRIMONIO_C03		>111000	0,422	0,00	1,53	1,860
NUMERO_CHEQUES_PROTESTADOS_C	Número de cheques con protestas	>1	-0,772	0,00	0,46	1,123

Fuente y elaboración propias

*Si se trabaja con muestras desproporcionadas se debe realizar una corrección al valor de la constante (Maddala, 1992). Se resta a la constante el término $(p_1) - (p_2)$, siendo p_1 y p_2 las proporciones de las observaciones escogidas de los grupos de buenos y malos clientes, respectivamente.

Los *odds ratio*, ayudan a interpretar el modelo; en este caso, dado que se deja una categoría de referencia en cada variable, se puede realizar el siguiente análisis. En el caso de la variable CANTON, se encontraron cuatro categorías de riesgo: bajo, medio, medio-alto y alto; Se dejó de referencia a la variable dicotómica correspondiente a los cantones de riesgo medio-alto (CANTON_C02). Por tanto, el valor de *odds ratio* para CANTON_C00 es 2,97 e indica que una persona dentro del grupo de cantones de riesgo bajo, es 2,97 veces mejor que una persona que esté en riesgo medio-alto. Este mismo análisis se puede realizar con cada variable. Las categorías de cada variable se encuentran en el Anexo 2.

4.1.2 Validación del modelo de Regresión logística

Los estadísticos calculados para el modelo de regresión logística se presentan. Ahora, para realizar el cálculo de los indicadores asociados a las matrices de confusión de cada uno de los tres modelos que se implementará, es necesario escoger un punto de corte, es decir, seleccionar un umbral en el score, de tal manera que a los sujetos que superen este umbral se los clasificará en el grupo de buenos y a los que no superan este umbral en el grupo de malos. En este trabajo para los tres modelos, se utilizará como punto de corte, el punto donde el porcentaje de buenos correctamente identificados (Sensibilidad) y el porcentaje de malos correctamente identificados (Especificidad) alcanzan el equilibrio, como se muestra en la Figura 11, para el modelo de regresión logística el punto de corte es 0.91697.

En la Tabla 6, se constata, tanto para el conjunto de modelamiento como para el de *testing*, el estadístico KS está en el nivel *satisfactorio* según los estándares utilizados por la empresa TransUnion (ver Figura 9) y el coeficiente de Gini (ver Figura 10), para este modelo es cercano a 50%, por lo cual se puede considerar satisfactorio (Anderson, 2007).

Por otro lado, la curva ROC para el modelo de regresión logística (ver Figura 12), indica que el estadístico AUC-ROC es de 73,75% para el grupo de modelamiento y de 73,43% para el de *testing*; son superiores a 70%; por tanto, se considera adecuado (Siddiqi, 2006).

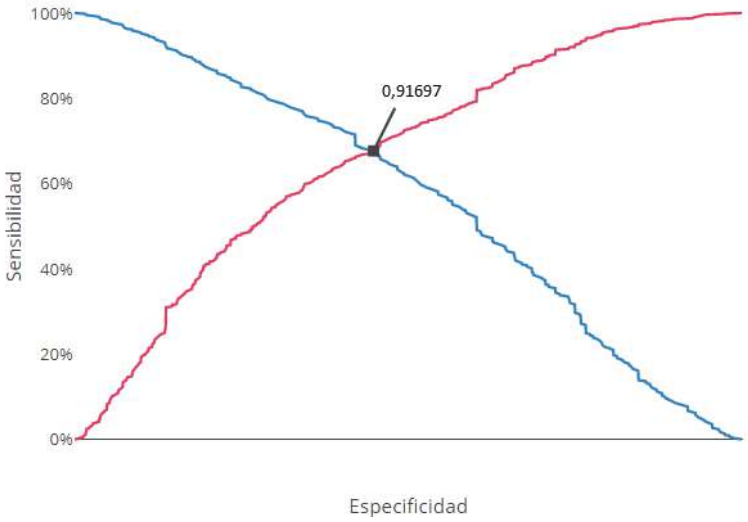
Figura 9. Estándar internacional para el estadístico KS seguido por la empresa TransUnion.

Fuente: TransUnion 2012.

Figura 10. Límites del coeficiente de Gini

Fuente: Elaboración propia a partir de los umbrales propuestos por Anderson (2007).

Figura 11. Punto de Equilibrio entre Sensibilidad y Especificidad.



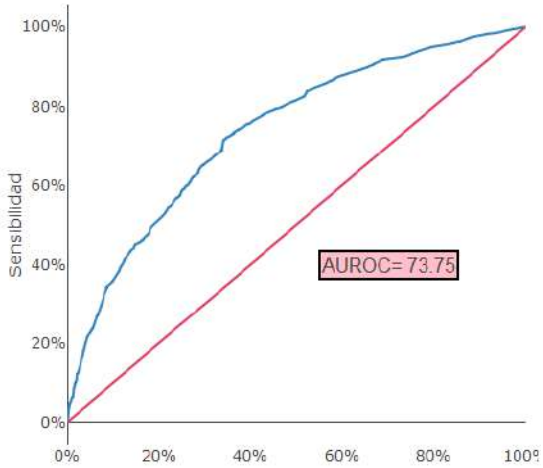
Fuente: elaboración propia.

Tabla 6. Estadísticos del modelo de regresión logística.

	Conjunto	
Estadísticos	**Modelamiento**	***Testing***
KS	37,44%	38,09%
ROC	73,75%	73,43%
Gini	47,50%	46,86%
Error	32,35%	31,57%
Aciertos	67,65%	68,43%

Fuente: elaboración propia.

Figura 12. Curva ROC del modelo Regresión Logística.



Fuente: elaboración propia.

Ahora, en la Tabla 7, se presentan los porcentajes de aciertos del modelo, para el conjunto de modelación. Los datos presentados son una comprobación de que el modelo de regresión logística predice con buen nivel y discrimina de manera adecuada los clientes buenos y malos. En general, el conjunto de los estadísticos presentados para este modelo están en un rango que permite concluir que el modelo es fuerte y eficiente.

Tabla 7. Matriz de confusión para el modelo Regresión Logística.

Regresión Logística		Real	
		0:Malo	1:Bueno
Estimada	0:Malo	996	5.086
	1:Bueno	474	10.637

Error	32,35%
Aciertos	67,65%
Sensibilidad	67,64%
Especificidad	67,76%

Fuente: elaboración propia.

4.2 Análisis previo para los modelos de *boosting*

Para realizar la construcción de los modelos de *credit scoring* haciendo uso de las dos metodologías *boosting*, *adaboost* y *gradient boosting*, es necesario cumplir con ciertos

requisitos y se deben tener ciertas precauciones; una de ellas es, considerar que el proceso no es automático, a pesar de que el analista no modifica los pasos internos de los algoritmos (Di Cellio et al, 2018).

Los pasos dados para construir los modelos en este trabajo son los siguientes:

1. Se realiza un análisis de los datos desproporcionados en la muestra del modelo, y se realiza una corrección mediante remuestreo con el método de ROSE.
2. Para evitar que los modelos sobreajusten, se calcula el número de clasificadores débiles, en este caso árboles de decisión utilizando el estimador OOB.
3. Se procede a eliminar las variables cuya importancia relativa sea cero estadísticamente.

El primer paso se realiza para los dos métodos de *boosting* desarrollados en este trabajo. Los pasos 2 y 3, se explican dentro de la construcción de cada método y se muestran en las siguientes secciones.

4.2.1 Remuestreo

En este caso, se aplica el procedimiento para el conjunto de modelación, la muestra de *testing* permanece inalterada. Lo que se busca es lograr el equilibrio en el conjunto de datos mediante la técnica de remuestreo con el método de ROSE; para esto, se generan datos artificiales en la clase minoritaria, manteniendo la precisión alta de la clase mayoritaria (King y Zeng, 2001). En la Tabla 8, se muestra la composición inicial del conjunto de modelamiento; con el método de ROSE, se aumentó el número de datos artificiales en la clase minoritaria (clientes malos) hasta que las proporciones de las clases quedaran en 20% - 80% (ver Tabla 9), pues esta proporción es una de las recomendadas por (King y Zeng, 2001).

Tabla 8. Composición inicial del conjunto de modelamiento

Clase	Descripción	Clientes	Porc_Cliente
0	Malo	1.470	8,55%
1	Bueno	15.723	91,45%
Total		17.193	100%

Fuente: elaboración propia a partir del conjunto de datos disponible.

Tabla 9. Composición del conjunto de modelamiento luego del remuestreo.

Clase	Descripción	Clientes	Porc_Cliente
0	Malo	3.995	20%
1	Bueno	15.723	80%
Total		19.718	100%

Fuente: elaboración propia a partir del conjunto de datos disponible.

4.3 *Adaboost*

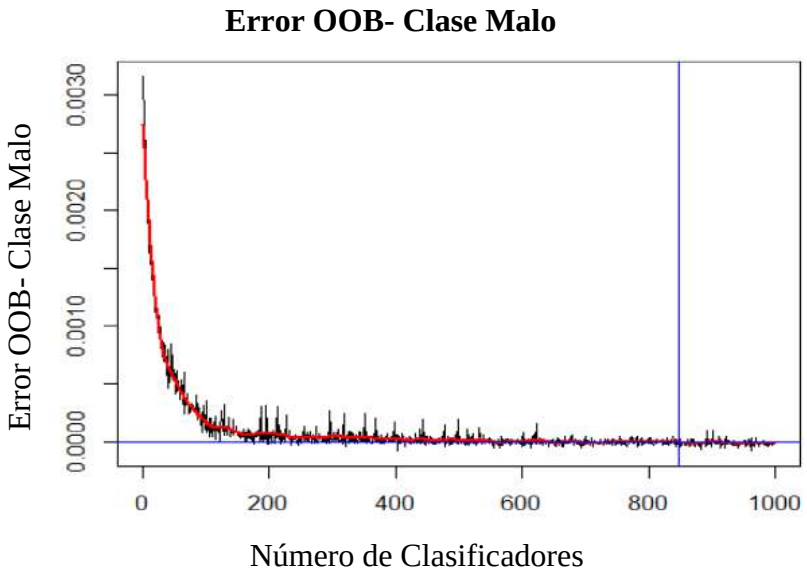
A continuación, se muestra el proceso realizado para la construcción de *Adaboost.*

4.3.1 Simulación para número de clasificadores con el error OOB

Como ya se ha mencionado anteriormente, para seleccionar el número de clasificadores débiles del modelo óptimo, sin llegar a sobreajustar el modelo, se utiliza el error OOB que se calcula a partir de simular una estimación para generalizar el error del modelo.

El cálculo del error OOB que consiste en calcular el error del modelo en una muestra generada por varias sub-muestras que no se consideraron en la adición de un nuevo árbol de decisión, en la construcción del modelo *Adaboost*, de manera más detalla el algoritmo para el cálculo del error OOB se encuentra en la sección 2.3.3.

Figura 13. Evolución del OOB para la clase malo en función al número de clasificadores débiles.



Fuente: elaboración propia.

En la Figura 13, la recta azul horizontal es una medida del error OOB para el cual ocurre la estabilización y la recta azul vertical representa el número óptimo de clasificadores resultante; en este caso 847. En este nivel se cuenta con la muestra de modelamiento equilibrada y el número óptimo de clasificadores débiles a ser entrenados. Ahora, es necesario explicar cómo escoger las variables predictoras, lo que se mostrará en la siguiente sección.

4.3.2 Selección de variables para el *Adaboost*

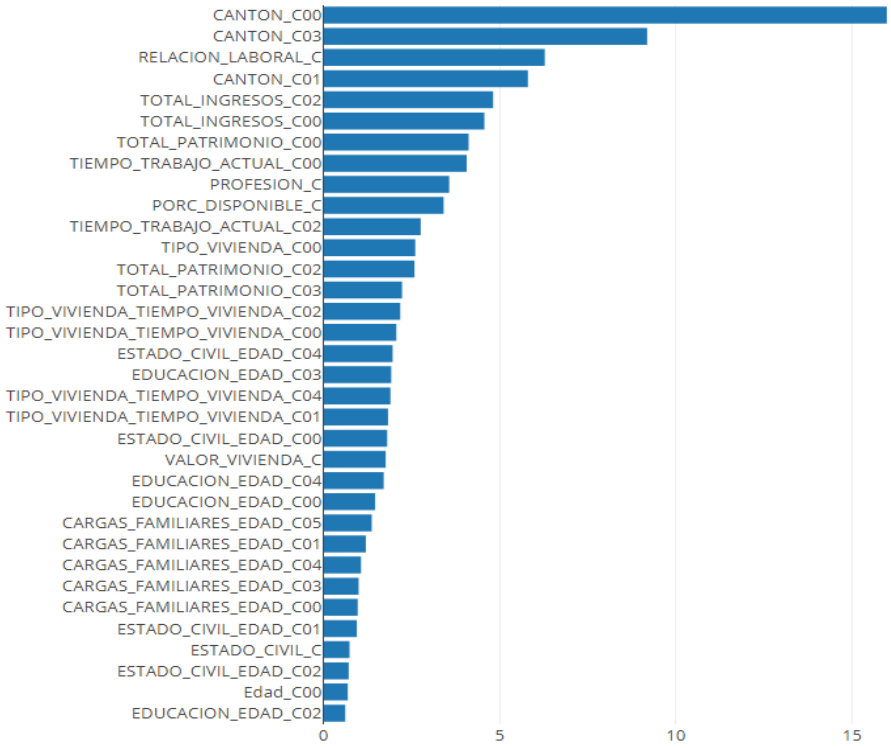
Una vez aplicado el *Adaboost* con el uso programa estadístico R, se encontraron que 34 variables tenían importancia estadísticamente distinta de cero (ver **Tabla 10**); además, se determina que las variables con mayor importancia dentro del modelo son: CANTON_C00, CANTON_C03 y RELACIÓN_LABORAL (ver Figura 14).

Tabla 10. Variables independientes utilizadas en el modelo de *credit scoring* mediante *Adaboost*.

Variables	Grupo/árbol
CANTON_C00	Riesgo bajo
CANTON_C03	Riesgo alto
RELACION_LABORAL_C	Dependiente
CANTON_C01	Riesgo medio
TOTAL_INGRESOS_C02	>520000
TOTAL_INGRESOS_C00	<=79571,9799
TOTAL_PATRIMONIO_C00	<=6500
TIEMPO_TRABAJO_ACTUAL_C00	<=59
PROFESION_C	Riesgo alto
PORC_DISPONIBLE_C	>0,49999418935722667
TIEMPO_TRABAJO_ACTUAL_C02	>173
TIPO_VIVIENDA_C00	Propia hipotecada
TOTAL_PATRIMONIO_C02	(24051,84,111000)
TOTAL_PATRIMONIO_C03	>111000
TIPO_VIVIENDA_TIEMPO _VIVIENDA_C02	Vive con familiares+<=17
TIPO_VIVIENDA_TIEMPO_ VIVIENDA_C00	Propia hipotecada;Propia no hipotecada;Prestada+<=10
ESTADO_CIVIL_EDAD_C04	C;V+>56
EDUCACION_EDAD_C03	U-universitaria,G-POSTGRADO <=51
TIPO_VIVIENDA_TIEMPO_ VIVIENDA_C04	Arrendada
TIPO_VIVIENDA_TIEMPO_ VIVIENDA_C01	Propia hipotecada; Propia no hipotecada;Prestada+>10
ESTADO_CIVIL_EDAD_C00	S;D;U+<=51
VALOR_VIVIENDA_C	>49500
EDUCACION_EDAD_C04	U-universitaria,G-POSTGRADO >51
EDUCACION_EDAD_C00	NO APICA, S-Secundaria,P-Primaria,T -Formacion intermedia(técnica)+EDAD<=47
CARGAS_FAMILIARES_EDAD_C05	EDAD>62
CARGAS_FAMILIARES_EDAD_C01	EDAD<=47 y CARGAS>2
CARGAS_FAMILIARES_EDAD_C04	EDAD (56,62]
CARGAS_FAMILIARES_EDAD_C03	EDAD (47,56] + CARGAS>0
CARGAS_FAMILIARES_EDAD_C00	EDAD<=47 + CARGAS<=2
ESTADO_CIVIL_EDAD_C01	S;D;U+>51
ESTADO_CIVIL_C	C
ESTADO_CIVIL_EDAD_C02	C;V+<=47
Edad_C00	<=47
EDUCACION_EDAD_C02	NO APICA,S-Secundaria,P-Primaria,T -Formacion intermedia(técnica)+ >62

Fuente: elaboración propia.

Figura 14. Orden de la importancia variables utilizadas en el modelo *Adaboost*.



Fuente: elaboración propia.

4.2.4 Validación del modelo *Adaboost*

En este caso, se calculó los mismos estadísticos para los conjuntos de modelamiento y *testing*. En la Tabla 11, se puede observar que el valor del estadístico KS en el conjunto de modelamiento resultó ser 48,86%, que se cataloga como **bueno** y se obtuvo un valor **satisfactorio** de 38,56% para el conjunto de *testing*, según los estándares internacionales de TransUnion. Por otro lado, el coeficiente de Gini tiene un de valor 62,72% para el conjunto de modelamiento y 49,14% para el de *testing*.

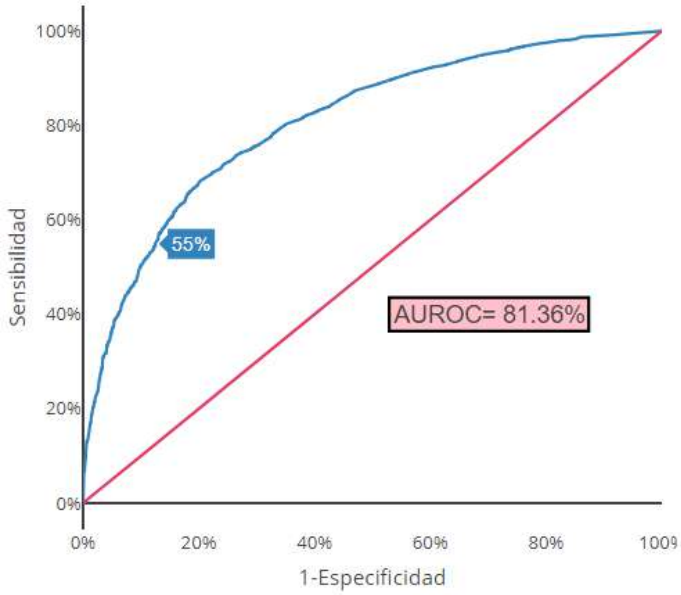
Tabla 11. Estadísticos empleados en esta investigación para medir la capacidad predictiva del modelo *Adaboost*.

	Valores del período de	
Estadísticos	**Modelamiento**	***Testing***
KS	48,86%	38,56%
ROC	81,36%	74,57%
Gini	62,72%	49,14%
Error	26,25%	27,36%
Aciertos	73,75%	72,64%

Fuente: elaboración propia.

En la Figura 15, se muestra la curva ROC del modelo *Adaboost*, con un estadístico AUC-ROC superior al 80% para el conjunto de modelamiento. El estadístico AUC-ROC tiene un valor de 74,57% para el conjunto de *testing*; en ambos casos se logra discriminar correctamente los clientes malos de los buenos.

Figura 15. Curva ROC del modelo *Adaboost*.



Fuente: elaboración propia.

Adicionalmente, el porcentaje de aciertos para el período de modelo es de 73,75%, obtenido a partir de la matriz de confusión (ver Tabla 12), en la cual se utilizó como punto de corte el equilibrio entre la especificidad y la sensibilidad, de manera análoga al modelo *Logit* (ver Figura 11), cuyo valor es de 0,8167.

Los resultados expuestos son evidencia de que el modelo *Adaboost* tiene buen nivel de predicción, ya que discrimina adecuadamente a buenos y malos clientes. Además, los valores para cada estadístico están en un rango adecuado para el conjunto de *testing*.

Tabla 12. Matriz de confusión para el modelo *Adaboost*.

Adaboost		Real	
		0:Malo	1:Bueno
Estimada	0:Malo	2.923	4.104
	1:Bueno	1.072	11.619

Error	26,25%
Aciertos	73,75%
Sensibilidad	73,32%
Especificidad	74,29%

Fuente: elaboración propia.

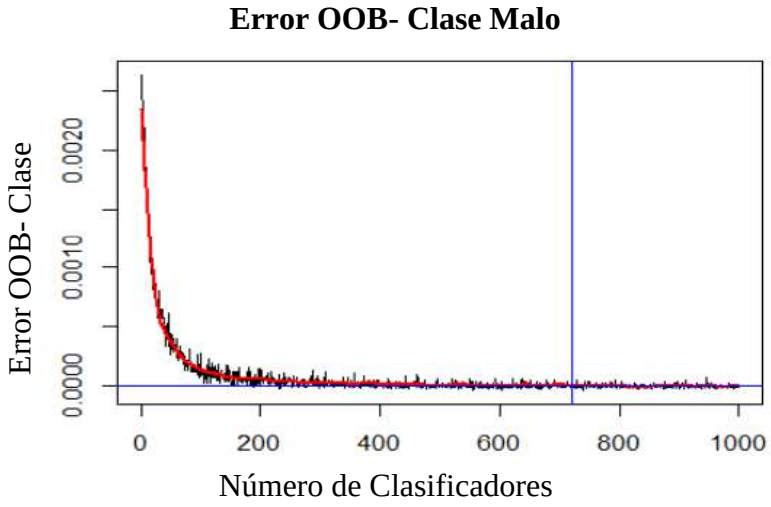
4.4 Gradient Boosting

Ahora, se realiza el mismo proceso que se expuso para el *Adaboost.*

4.4.1 Simulación para número de clasificadores con el error OOB, para *Gradient Boosting*

De manera análoga al caso *Adaboost*, se selecciona el número de clasificadores débiles para el modelo óptimo, utilizando una simulación del error de generalización del modelo, esto mediante el estimador OOB. En la gráfica de la Figura 16, se observa que el estimador OOB se estabiliza en 721 clasificadores, que se los utiliza para realizar el modelo.

Figura 16. Evolución del OOB para la clase malo con el *Gradient Boosting*.



Fuente: elaboración propia.

4.2.6 Selección de variables en el caso *Gradient Boosting*

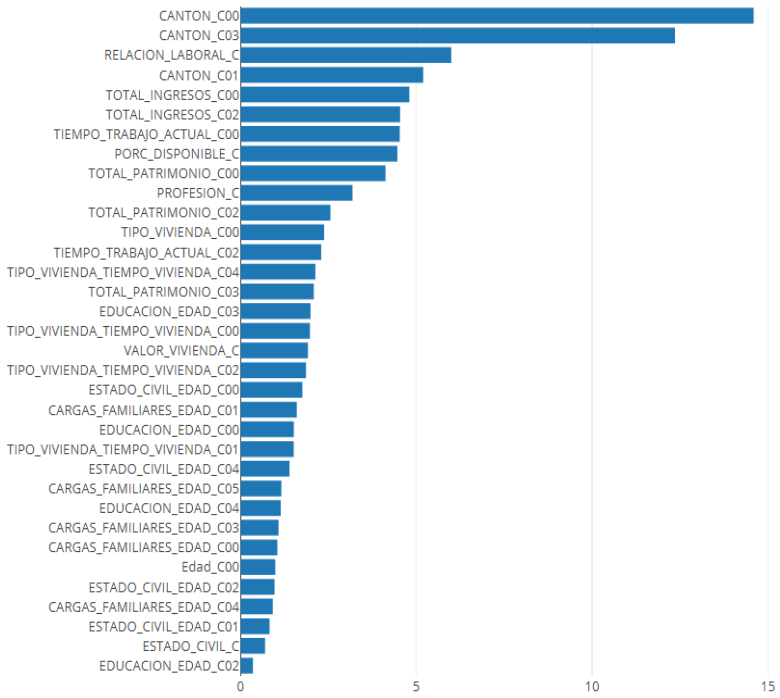
Al igual que en el caso de *Adaboost*, se lleva a cabo el proceso de entrenamiento de *Gradient Boosting* con el programa estadístico R, encontrándose que 34 variables explicativas del modelo. Así, se obtiene:

Tabla 13. Variables independientes utilizadas en la construcción del modelo de *credit scoring* mediante *Gradient Boosting*.

Variables	Grupo/árbol
CANTON_C00	Riesgo bajo
CANTON_C03	Riesgo alto
RELACION_LABORAL_C	Dependiente
CANTON_C01	Riesgo medio
TOTAL_INGRESOS_C00	<=79571,9799
TOTAL_INGRESOS_C02	>520000
PORC_DISPONIBLE_C	>0,4999941
TIEMPO_TRABAJO_ACTUAL_C00	<=59
TOTAL_PATRIMONIO_C00	<=6500
PROFESION_C	Riesgo alto
TOTAL_PATRIMONIO_C02	(24051,84,111000]
TIPO_VIVIENDA_C00	Propia hipotecada
TIPO_VIVIENDA_TIEMPO_VIVIENDA_C04	Arrendada
ESTADO_CIVIL_EDAD_C00	S; D; U+<=51
TIEMPO_TRABAJO_ACTUAL_C02	>173
EDUCACION_EDAD_C03	U-universitaria-POSTGRADO <=51
TIPO_VIVIENDA_TIEMPO_VIVIENDA_C02	Vive con familiares+<=17
TOTAL_PATRIMONIO_C03	>111000
VALOR_VIVIENDA_C	>49500
CARGAS_FAMILIARES_EDAD_C01	EDAD<=47 y CARGAS>2
ESTADO_CIVIL_EDAD_C04	C; V+>56
TIPO_VIVIENDA_TIEMPO_VIVIENDA_C00	Propia hipotecada; Propia no hipotecada; Prestada+<=10
EDUCACION_EDAD_C04	U-universitaria-POSTGRADO EDAD>51
EDUCACION_EDAD_C00	NO APICA, S-Secundaria, P-Primaria, T - Formacion intermedia(técnica)+ EDAD <=47
TIPO_VIVIENDA_TIEMPO_VIVIENDA_C01	Propia hipotecada; Propia no hipotecada; Prestada+>10
CARGAS_FAMILIARES_EDAD_C05	EDAD>62
CARGAS_FAMILIARES_EDAD_C03	EDAD (47,56] y CARGAS>0
CARGAS_FAMILIARES_EDAD_C00	EDAD<=47 y CARGAS<=2
Edad_C00	<=47
ESTADO_CIVIL_C	C
ESTADO_CIVIL_EDAD_C01	S; D; U+>51
ESTADO_CIVIL_EDAD_C02	C; V+<=47
CARGAS_FAMILIARES_EDAD_C04	EDAD (56,62]
EDUCACION_EDAD_C02	NO APICA, S-Secundaria-Primaria - Formación intermedia(técnica)+ >62
Edad_C03	>62
TIPO_VIVIENDA_C02	Arrendada
Edad_C02	(56,62]

Fuente: elaboración propia.

Figura 17. Ranking de importancia para las variables usadas en el modelo *Gradient Boosting*.



Fuente: elaboración propia.

4.2.7 Validación del modelo *Gradient Boosting*

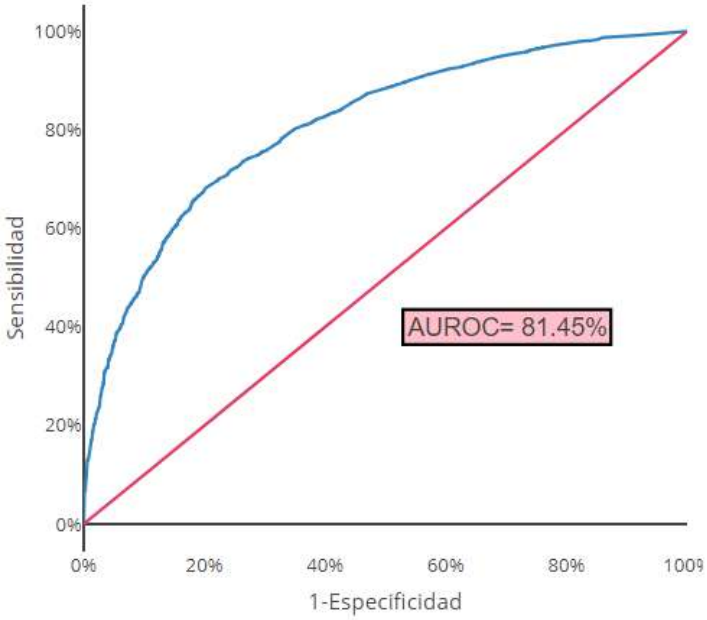
El poder de predicción del modelo *Gradient Boosting* está determinado por el valor de sus estadísticos. En la Tabla 14, se muestran los valores del estadístico KS: 48,91% para el conjunto de modelamiento (clasificado como **bueno**) y 37,98% para el conjunto de *testing* (**satisfactorio**). El coeficiente de Gini evidencia que el modelo *Gradient boosting* es más que satisfactorio al tomar valores por encima del 50% en ambos conjuntos.

Tabla 14. Estadísticos empleados en esta investigación para medir la capacidad predictiva del modelo *Gradient Boosting*.

	Valores del período de	
Estadísticos	**Modelamiento**	***Testing***
KS	48,91%	37,98%
ROC	81,45%	74,66%
Gini	62,90%	50,32%
Error	26,20%	27,66%
Aciertos	73,80%	72,34%

Fuente: elaboración propia.

Figura 18. Curva ROC del modelo *Gradient Boosting*.



Fuente: Elaboración propia.

Nótese que en la Figura 18, la curva ROC para el modelo *Gradient boosting*, muestra un AUC-ROC superior al 80% para el conjunto de modelamiento y en la se muestra que tiene un valor de 74,66% para el *testing*, con lo cual se alcanza una correcta capacidad de discriminación de clientes.

Ahora, en la Tabla 15, se puede observar que el porcentaje de aciertos es 73,80% para el conjunto de modelamiento; mientras que, para el conjunto de *testing* es de 72,34%. La matriz de confusión se construyó con el punto de corte hallado de manera similar al punto de corte para el modelo *Logit* (ver Figura 11) cuando la sensibilidad y especificidad del modelo alcanzan su punto de equilibrio, cuyo valor es de 0,78469.

Tabla 15. Matriz de confusión del modelo *Gradient Boosting*.

Gradient boosting		Real	
		0:Malo	1:Bueno
Estimada	0:Malo	2.940	4.150
	1:Bueno	1.055	11.573

Error	26,20%
Aciertos	73,80%
Sensibilidad	73,80%
Especificidad	73,87%

Los resultados presentados son evidencia de que el modelo *Gradient boosting* tiene un buen nivel predictivo y; por tanto, discrimina adecuadamente clientes buenos y malos.

4.5 Comparación de los modelos *scoring*

Una vez construidos los modelos con los tres métodos, es necesario compararlos para determinar cuál de ellos resulta mejor en términos estadísticos; para este fin, se hace uso de los estadísticos: KS, Coeficiente de Gini, Matriz de Confusión y AUC-ROC. Un resumen de los estadísticos para el conjunto de modelamiento y los criterios para escoger cuál es el mejor modelo se muestran en la Tabla 16 (se ha sombreado en verde el mejor resultado):

Tabla 16. Comparación de los estadísticos de los modelos de *scoring* construidos

Estadísticos	**Regresión Logística**	***Adaboost***	***Gradient boosting***	**¿Cuál es mejor Estadístico?**	**Mejor Modelo**
KS	37,44%	48,86%	48,91%	Mayor valor	*Gradient boosting*
ROC	73,75%	81,36%	81,45%	Mayor valor	
Gini	47,50%	62,72%	62,90%	Mayor valor	
Error	32,35%	26,25%	26,20%	Menor valor	
Aciertos	67,65%	73,75%	73,80%	Mayor valor	
Sensibilidad	67,64%	73,32%	73,80%	Mayor valor	
Especificidad	67,76%	74,29%	73,87%	Mayor valor	

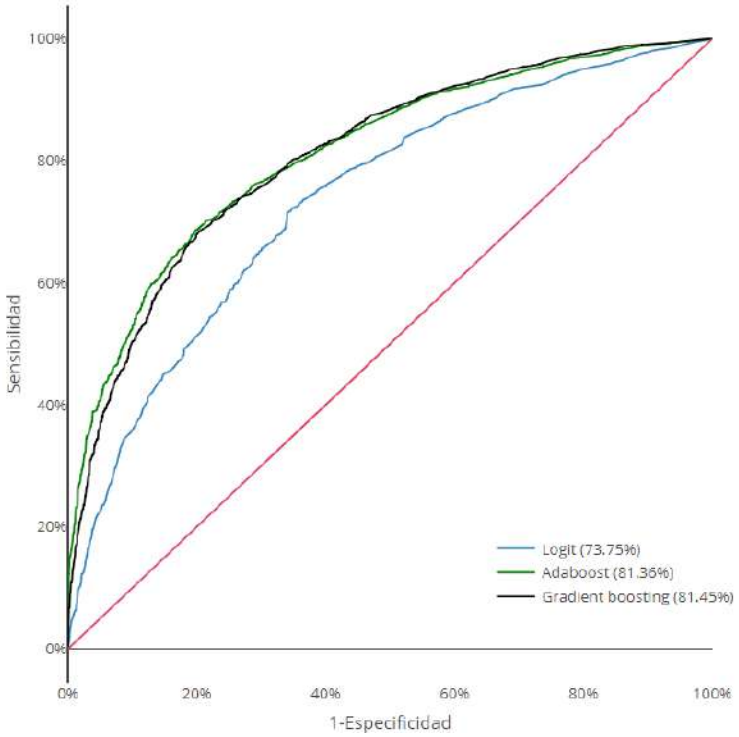
Fuente: elaboración propia.

Es decir:

- Con respecto al estadístico KS, el modelo de *Gradient boosting* tiene el mayor valor (48,91%) en el período de modelamiento, valor que supera al de regresión logística en 11,47 puntos porcentuales y solamente 0,05 al *Adaboost*. Los tres valores se encuentran dentro del rango recomendado según TransUnion.
- En cuanto al coeficiente de Gini, el modelo de *Gradient boosting* supera al de regresión logística en 15,4 puntos porcentuales y al modelo *Adaboost* lo supera por 0,18 puntos porcentuales. Para los tres modelos, el coeficiente supera el 50%; por lo que, todos son satisfactorios.
- Los puntos de corte óptimos, calculados a partir del equilibrio entre la sensibilidad y la especificidad son: 0,91697, 0,81669 y 0,78469 para los modelos de regresión logística, *Adaboost* y *Gradient boosting*, respectivamente. Con esto, se puede observar que el *Gradient boosting* tiene un mayor porcentaje de aciertos (73,8%), que los otros dos modelos; aunque, con una diferencia de 6 puntos porcentuales respecto a la regresión logística y solo con unos 0,5 puntos porcentuales si se lo compara con el *Adaboost*.

Adicionalmente, se muestra la comparación entre las curvas ROC (ver Figura 19), donde se puede observar la diferencia entre la regresión logística y los métodos de boosting, mostrando una mejora respecto al método clásico.

Figura 19. Comparación de las Curvas ROC de los tres modelos: regresión logística (azul), *Adaboost* (verde) y *Gradient Boosting* (negro).



Fuente: elaboración propia.

Como se puede apreciar en la Tabla 16, los modelos creados mediante técnicas de boosting, tienen claramente un mejor desempeño con respecto al modelo creado con la metodología de regresión logística, sin embargo, como se pudo ver en las debilidades de los modelos con técnicas de *boosting*, estos se suelen sobreajustar con facilidad a los datos de entrenamiento, por tanto, se utilizó una muestra de prueba, en donde se evaluará el comportamiento real de estos modelos en una base de datos totalmente diferente.

4.3.1 Poder predictivo: conjunto de prueba o *testing*

Para medir el poder de predicción de los modelos se hace uso del conjunto de *testing*, que corresponde al 20% de la base (4.299 registros), los cuales están distribuidos de la siguiente forma (ver Tabla 17):

Tabla 17. Datos correspondientes al período de *testing*.

Clase	Descripción	Clientes	Porc_Cliente
0	Malo	368	8,56%
1	Bueno	3.931	91,44%
Total		4,299	100%

Fuente: elaboración propia

En la Tabla 18, se muestran los resultados obtenidos para los estadísticos AUC-ROC, KS, Coeficiente de Gini, Error y Aciertos para los tres modelos evaluados en el período de *testing* (se marcan de color verde el valor que corresponde al mejor modelo en cada estadístico). Aquí, se puede observar, que el *Adaboost* tiene el mayor número de estadísticos con el mejor resultado. Es importante señalar, que aunque los métodos de *boosting* tienen un mejor rendimiento que la regresión logística, la diferencia en conjunto de *testing* se reduce respecto a lo visto en el conjunto de modelamiento.

Tabla 18. Estadísticos calculados para los tres modelos en estudio en el período de *testing.*

Estadísticos *Testing*	**Regresión Logística**	***Adaboost***	***Gradient Boosting***
KS	38,09%	38,56%	37,98%
ROC	73,43%	74,57%	74,66%
Gini	46,86%	49.14%	49,32%
Error	31,57%	27,36%	27,66%
Aciertos	68,43%	72,64%	72,34%

Fuente: elaboración propia

En la Tabla 18, se puede apreciar que la mejora entre los indicadores de los modelos creados con técnicas de *boosting* y el modelo creado con regresión logística no es relativamente grande, por tanto, se tienen las siguientes observaciones:

- En el indicador KS la diferencia es relativamente pequeña ya que el indicador KS de la regresión logística es aproximadamente 0,5% menor que el indicador KS del modelo creado con *Adaboost*, mientras que el indicador KS del modelo creado con *Gradient boosting* es aproximadamente de 0,9% mejor que la regresión logística.
- En los indicadores ROC y Gini se aprecia una diferencia mayor al 1% a favor de los modelos creados con técnicas de *boosting.*
- En el porcentaje de acierto se nota una mejoría considerable, aproximadamente de tres puntos porcentuales a favor de los modelos con técnicas de *boosting*

Aunque la ganancia en los indicadores estadísticos en la base de prueba no fue tan grande como en la base de entrenamiento, los modelos creados con técnicas de *boosting* tienen un

buen desempeño en la base de prueba, incluso superior a los modelos creados con la metodología de regresión logística.

4.3.2 Costo y rendimiento computacional

Un punto importante en la creación de estos modelos es el costo y rendimiento computacional, por lo que, para generar los tres tipos de modelos se utilizó una computadora con procesador Intel core i5 de cuarta generación, con una base de entrenamiento de tamaño 17.193, con la cual se obtuvo los siguientes resultados:

- Primero, para la creación de los modelos con técnicas de *boosting* se procede con una primera ejecución de 1.000 árboles de decisión con cada metodología, donde el tiempo promedio aproximado fue de 270 segundos.
- Una vez conocido el número óptimo de árboles, los tiempos de ejecución son los siguientes: con 847 árboles de decisión y utilizando la técnica *Adaboost*, toma la creación del modelo aproximadamente 225 segundos, mientras que el modelo creado con la técnica de *Gradient boosting*, tomó aproximadamente 195 segundos.
- Por otro lado, crear un modelo con regresión logística no tiene un coste computacional demasiado grande, ya que para el modelo final tomó aproximadamente 15 segundos; sin embargo, se debe considerar que para llegar a ese modelo final se tuvo que pasar por varios modelos preliminares, que garantizan la eficiencia del modelo final; por lo que, aunque generar el modelo no conlleva un coste computacional alto, realizar el análisis para obtener el modelo final, puede tomar igual o más tiempo que con las otras metodologías.

Capítulo Cinco: Conclusiones y Recomendaciones

La idea central de este trabajo es realizar la comparación en cuanto a desempeño de la metodología de regresión logística contra dos metodologías de *boosting* (*adaboost* y *gradient boosting*, en este caso), en la modelización del *credit scoring* de una cartera de crédito desconocida en cuanto a la institución financiera, pero apropiada para este fin. El conjunto de datos fue suministrado por una institución financiera ecuatoriana y por motivos de confidencialidad no se pueden mostrar detalles del conjunto de datos; sin embargo, este conjunto se evaluó como apropiado porque dentro de la información proporcionada se pudo reconocer lo siguiente:

- Los 21.492 clientes que comprende la muestra, cumplen con el requisito de ser clientes bancarizados; es decir, presentan al menos un registro en central de riesgos en el período de los últimos 36 meses.
- El conjunto de datos suministrado fue generado durante tres años, siguiendo las recomendaciones de la Superintendencia de Bancos. Se le dividió dos partes: la primera parte correspondiente al entrenamiento para realizar el ajuste de los modelos y la otra parte para validar y evaluar la complejidad de los mismos. La parte más grande, denominada conjunto de modelamiento, alcanzó el 80% del total y fue obtenida mediante muestreo aleatorio simple sin reposición para asegurar representatividad. El conjunto de *testing*, con el restante 20%, se utilizó para validar los modelos.

Luego de fijar los conjuntos de datos para el análisis, se seleccionó el conjunto de variables independientes; la variable dependiente ya estaba fijada por la definición que la institución financiera sobre un cliente bueno o malo. En la selección de las variables independientes se utilizó el estadístico valor de la información (*IV*), con el objetivo de tomar las variables que tienen mayor poder de predicción, luego de ser categorizadas mediante árboles de decisión.

Una vez cumplidos los pasos previos se procedió a aplicar las tres metodologías desarrollando los modelos con el programa estadístico R. Después de realizada la construcción de los modelos, se procedió a comparar su desempeño, evaluando su capacidad de predicción con los estadísticos AUC-ROC, KS, Gini y Matriz de Confusión; tanto en el conjunto de modelación, como en el de *testing*.

Una vez listos los modelos y comparaciones en el desempeño estadísticos, se enumeran las principales conclusiones:

1. Uno de los primeros resultados que se puede apreciar es que con métricas como KS, Gini y ROC, los modelos creados con técnicas de *Boosting* tienden a interpretarse como un problema de sobreajuste; pues, como se pudo ver el KS tuvo una caída de más de 10 puntos porcentuales entre la base de modelamiento y la base de prueba. De igual manera, el coeficiente de Gini con una caída de 12 puntos porcentuales y el

ROC con una caída de más de 5 puntos porcentuales. Sin embargo, los modelos de *machine learning* buscan mantener el porcentaje de aciertos en el cambio de bases; es decir, la diferencia entre el proceso de modelamiento contra el de prueba, es muy pequeña, solo un poco mayor a un punto porcentual, para ambos modelos de *boosting*. Considerando estos antecedentes se puede concluir que el modelo no se sobreajusta.

2. Se evidenció que los indicadores en la base de prueba caen considerablemente con respecto a los indicadores de la base de entrenamiento, para los modelos con técnicas de *boosting;* sin embargo, estos modelos al ser evaluados con la base de prueba tienen una capacidad de predicción igual o mejor que los indicadores de la regresión logística. Este orden demuestra la hipótesis planteada en esta investigación: que los modelos de *boosting* logran mejor desempeño que el modelo de regresión logística en la modelización del *credit scoring* de una cartera de crédito.

3. Los tres modelos en estudio tienen buena capacidad predictiva, exhiben un ajuste correcto a los datos con los que se construyeron y logran clasificar a los clientes como buenos o malos de manera adecuada. Dado que se probó su funcionamiento en un conjunto de datos distinto al del desarrollo, los tres modelos se pueden generalizar a bases nuevas.

4. Continuando con la comparación, a favor del modelo de *credit scoring* con regresión logística, se presenta la interpretación directa de los parámetros del modelo, la fácil aplicación del mismo y la gran estabilidad que muestra en el tiempo; todavía reina en el mercado por su visión orientada al negocio y la confianza que se ha ganado. A favor de las metodologías *boosting*, está la mayor facilidad en la obtención de los resultados, ya que no necesitan un análisis estadístico muy amplio y tienen un buen desempeño predictivo.

5. En lo que concierne a las dificultades, el modelamiento con regresión logística requiere mucho esfuerzo y ajuste manual; en el caso del modelamiento con métodos de *boosting* no se puede realizar una interpretación de las variables y todavía el mercado discute la validez de su exactitud y su poder de predicción.

6. Con respecto al tiempo computacional, para una base de 17.193 individuos, los tres modelos tienen un coste de tiempo aceptable, aunque con una significativa ganancia para el modelo creado con la metodología de regresión logística.

En la elaboración de este trabajo también se pueden realizar varias recomendaciones que ayudarán en futuros proyectos o trabajos:

1. Con el fin de disminuir el sobreajuste en los modelos creados con técnicas de *boosting* se puede optar por probar otros tipos de técnicas que ayuden en la reducción o solución de este problema, por ejemplo: simplificando más lo modelos si fuese posible o utilizar técnicas para calcular el número de árboles óptimo como validación cruzada o parada temprana (*Early stopping*).

2. Empezar a trabajar de la mano con las metodologías no tradicionales y las tradicionales, de tal forma que se ganará mayor comprensión de este tipo de modelos y esto abrirá un mayor horizonte en el problema de clasificación crediticia.
3. Al utilizar estas metodologías nuevas o la clásica, es importante siempre controlar y monitorear de manera continua la estabilidad de los modelos.

Referencias

Akaike, H.(1974). A new look at the statistical model identification. IEEE Transactions on Automatic Control (Volume: 19, Issue: 6 , Dec 1974).

Alfaro, E., Gámez, M., García, N., Alfaro, J., y Mondéjar, J. (s.f). *Historia de la Probabilidad y la Estadística*. pp. 457-468. Recuperado de https://idus.us.es/bitstream/handle/11441/84129/Breve_historia_de_la_familia_de_clasificadores_boosting.pdf;jsessionid=C896223AABEF49B59E6CBF00D554D4AE?sequence=1&isAllowed=y.

Anderson, R. (2007). *The credit scoring toolkit: theory and practice for retail credit risk management and decision automation*. Oxford: Oxford University Press.

Bambino, C. (2005). Prestar como locos y obtener beneficios: ¿es realmente posible? (Un análisis logit multinomial para los determinantes del comportamiento de pago de una cartera de consumo). Quito: FLACSO. Recuperado de http://repositorio.flacsoandes.edu.ec/handle/10469/61.

BanEcuador. (2016). Programa de educación financiera. *Educación Finaciera*, 28.

Bauer, E., y Kohavi, R. (1999). An Empirical Comparison of Voting Classification Algorithms: Bagging, Boosting, and Variants. Machine Learning 36, 105–139 (1999).

Breiman, L., Friedman, J., Olshen, R., y Stone, C. (1984). Classification and Regression Trees.Wadsworth International Group, Belmont, CA.

Capa, H., (2015). Investigación por muestreo: Fundamentos y Aplicaciones.

Corso, J. (s.f). Boosting and Adaboost. Suny at Buffalo. Recuperado de https://cse.buffalo.edu/~jcorso/t/CSE555/files/lecture_boosting.pdf.

Credit Scoring. (s.f). Recuperado de http://sgpwe.izt.uam.mx/files/users/uami/blanca/capitulo4a1.pdf el 31 de julio de 2020.

Di Cellio, P., Forti, M., y Witarsa M. (2018). A comparison of Gradient Boosting with Logistic Regression in Practical Cases. Recuperado de https://www.sas.com/content/dam/SAS/support/en/sas-global-forumproceedings/2018/1857-2018.pdf el 11 de agosto de 2020.

Ding, C., Wang, D., Ma, X., & Li, H. (2016). Predicting short-term subway ridership and prioritizing its influential factors using gradient boosting decision trees. *Sustainability (Switzerland)*, *8*(11), 1–16. https://doi.org/10.3390/su8111100

Drucker, H., y Cortes C. (s.f) Boosting Decision Trees. AT&T Bell Laboratories. Recuperado de http://papers.nips.cc/paper/1059-boosting-decision-trees.pdf

Emer, E. (2005). AdaBoost Algorithm. *Science*, 2–5.

Fahlman, S., y Lebiere, C. (1989). The Cascade-Correlation Learning Architecture. Technical Report, Carnegie Mellon University, Pittsburgh, PA.

Feng, J., Xu, Y., Jiang, Y., y Zhou, Z. (2020). Soft Gradient Boosting Machine. arXiv:2006.04059v1 $[cs.LG]$ 7 Jun 2020.

Fox, J. (2003). Linear Models, Problems. Canada: McMaster University. Recuperado de https://socialsciences.mcmaster.ca/jfox/papers/linear-models-problems.pdf el 31 de julio de 2020.

Freund, Y. (1995). *Boosting a weak learning algorithm by majority*. Information and Computation, 121(2):256-285.

Freund, Y. (1998). *An introduction to boosting based classification*. AT&T Labs 180 Park Avenue Florham Park, NJ 07932. Recuperado de https://perun.pmf.uns.ac.rs/radovanovic/dmsem/cd/install/Weka/doc/classifiers-papers/meta/AdaBoostM1/QUAC2000.pdf.

Freund, Y. y Schapire, R. (1997). A Decision-Theoretic Generalization of On-lineLearning and an Application to Boosting. Journal of Computer and System Science, 55(1):119139, 1997.

Freund, Y., & Schapire, R. E. (1997). A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *Journal of Computer and System Sciences*, *55*(1), 119–139. https://doi.org/10.1006/jcss.1997.1504

Freund, Y., y Schapire, R. (1996). *Experiments with a New Boosting Algorithm*. *Proceedings of the Thirteenth International Conference on Machine Learning,* pp. 148- 156, Morgan Kaufmann.

Friedman, J. (2001).Greedy boosting approximation:a gradient boosting machine. *Ann.Stat.* 29, 1189–1232.doi:10.1214/aos/1013203451.

Gareth, J., Witten, D., Trevor, H., & Tibshirani, R. (2007). An Introduction to Statistical Learning with Applications in R. In *Performance Evaluation* (Vol. 64, Issues 9–12). https://doi.org/10.1016/j.peva.2007.06.006

Hand, D., y Henley, W. (1997). Statistical Classification Methods in Consumer Credit Scoring: A Review. *Journal of the Royal Statistical Society. Series A (Statistics in Society)* 160 (3): 523–41.

Jacobucci, R. (s.f). Decision Tree Stability and its Effect on Interpretation. Recuperado de https://mfr.osf.io/export?format=pdf&url=https%3A//files.osf.io/v1/resources/f2utw/providers/osfstorage/5a61182594ef6d000f60b5c5%3Fformat%3Dpdf%26action%3Ddownload%26direct%26version%3D2

Kearns, M. (1988). Thoughts on hypothesis boosting. *Unpublished Manuscript*, *45*, 105. https://doi.org/citeulike-article-id:5980850

Kearns, M., & Valiant, L. (1989). Cryptographic Limitations on Learning Boolean Formulae and Finite Automata. *Journal of the ACM (JACM)*, *41*(1), 67–95. https://doi.org/10.1145/174644.174647

Menardi, G., y Torelli, N. (2010). Training and assessing classification rules with unbalanced data. Working Paper Series, N. 2, 2010. Recuperado de https://www.openstarts.units.it/bitstream/10077/4002/1/Menardi%20Torelli%20DEAMS%20WPS2.pdf

Mendoza, M. (s.f). Econometría aplicada utilizando R. Recuperado de http://saree.com.mx/econometriaR/sites/default/files/Cap9_teoria.pdf el 31 de julio de 2020.

Mendoza, R. (2013). Boosting en el modelo de aprendizaje PAC. Revista Elementos, Número 3, Junio de 2013, pp. 37-48.

Modelización de un proceso SETAR. (s.f). Recuperado de https://www.tdx.cat/bitstream/handle/10803/6503/05CAPITULO4.pdf?sequence=5&isAllowed=y el 31 de julio de 2020.

Montalván, C. (2019). Credit scoring, aplicando técnicas de regresión logística y redes neuronales, para una cartera de microcrédito. (Tesis de Maestría). Universidad Andina Simón Bolívar, Quito, Ecuador.

Natekin, A., y Knoll, A. (2013). Gradient boosting machine, a tutorial. Frontiers in Neurorobotics. Volume 7, Article 21, pp. 1-21.

R Core Team. (2017). R: A language and environment for statistical computing [Computer

Ramzai, J. (2019). Simple guide for Top 2 types of Decision Trees: CHAID & CART. Recuperado de https://towardsdatascience.com/clearly-explained-top-2-types-of-decision-trees-chaid-cart-8695e441e73e el 24 de agosto de 2020.

Rayo, S., Lara, J., & Camino, D. (2010). Un Modelo de Credit Scoring para instituciones de microfinanzas en el marco de Basilea II / A Credit Scoring Model for Institutions of Microfinance under the Basel II Normative. *Journal of Economics, Finance and Administrative Science VO - 15*, *28*, 89. http://ezproxy.unal.edu.co/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=edssci&AN=edssci.S2077.18862010000100005&lang=es&site=eds-live

Rezac, M., y Řezáč, F. (2011). How to Measure the Quality of Credit Scoring Models. *Czech Journal of Economics and Finance (Finance a uver)* 61 (enero):486–507.

Ripley, B. D. (1994). Neural Networks and Related Methods for Classification. *Journal of the Royal Statistical Society: Series B (Methodological)*, *56*(3), 409–437. https://doi.org/10.1111/j.2517-6161.1994.tb01990.x

Rodó, P. (s.f). Prueba de Kolmogorov-Smirnov (KS). Economipedia. Recuperado de https://economipedia.com/definiciones/prueba-de-kolmogorov-smirnoff-k-s.html el 1 de agosto de 2020.

Schapire, R. E. (1990). The Strength of Weak Learnability. *Machine Learning*, *5*(2), 197–227. https://doi.org/10.1023/A:1022648800760

Schapire, R. E. (2013). Boosting: Foundations and Algorithms. In *Kybernetes* (Vol. 42, Issue 1). https://doi.org/10.1108/03684921311295547

Scheaffer, R., Mendenhall, W., y Ott, R. (2013). *Elementos de muestreo*. Madrid (España): Paraninfo.

Siddiqi, N. (2006). *Credit Risk Scorecards—Developing and Implementing Intelligent Credit Scoring*, John Wiley & Sons, 2006.

software manual]. Vienna, Austria. Recuperado de http://www.R-project.org/

Superintendencia de Bancos y Seguros del Ecuador. (2014). Gestion de Riesgo de Credito. *Libroi.-Normas Generales Para Las Instituciones Del Sistema Financiero*, 626–659.

Superintendencia de Bancos y Seguros. (2014). *LIBRO I.- NORMAS GENERALES PARA LAS INSTITUCIONES DEL SISTEMA FINANCIERO TITULO*. 626–659.

Swets, J.(1988). Measuring the accuracy of diagnostic systems. Science 240: 1.285-1.293.

Valle, A. (s.f). Curvas ROC (Receiver-Operating-Characteristic) y sus aplicaciones. (Tesis de pregrado). Universidad de Sevilla, España. Recuperado de https://idus.us.es/bitstream/handle/11441/63201/Valle%20Benavides%20Ana%20Roc%C3%ADo%20del%20TFG.pdf?sequence=1 el 31 de julio de 2020.

Vapnik, V., y Chervonenkis, A. (1971). On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its applications*, XVI(2):264–280.

Yoo, W., Mayberry, R., Sejong B., Singh, K., Qinghua H., y Lillard, J. (2014). A Study of Effects of MultiCollinearity in the Multivariable Analysis. *International journal of applied science and technology* 4 (5): 9–19. Recuperado de https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4318006/ el 31 de julio de 2020.

Yu, L. (2008). *Bio-inspired credit risk analysis: computational intelligence with support vector machines*. Berlin: Springer Verlag, ed. 2008.

Zeng, G. (2013). Metric Divergence Measures and Information Value in Credit Scoring. Journal of Mathematics. https://doi.org/10.1155/2013/848271

Zhang, T., & Yu, B. (2005). Boosting with early stopping: Convergence and consistency. *Annals of Statistics*, *33*(4), 1538–1579. https://doi.org/10.1214/009053605000000255

Anexos

Anexo 1 Descripción del total de variables

***Tabla 1*:** Descripción de las variables

Nombre de variables	Descripción
RELACION_LABORAL	Indica la relación que existe entre el empleador y el trabajador puede ser independiente o dependiente.
PROVINCIA	Provincia donde reside.
CANTON	Cantón donde reside.
PARROQUIA	Parroquia donde reside.
EDAD	Edad del cliente a la fecha de desembolso.
TOTAL_INGRESOS	Total de ingresos del cliente.
TOTAL_EGRESOS	Total de egresos del cliente.
TOTAL_PATRIMONIO	Indica el total de bienes propios que tiene el cliente.
CARGO	Cargo que ocupa en el trabajo
ESTADO_CIVIL	Estado civil que se encuentra el cliente
TIPO_VIVIENDA	Tipo de vivienda en donde se encuentra: hipotecada, propia, prestada, etc.
VALOR_VIVIENDA	Valor estimado de la vivienda
TIEMPO_RESIDENCIA_ACTUAL	Indica el tiempo donde reside actualmente.
MONTO_ORIGINAL_OPERACION	Indica el monto original.
NUMERO_CHEQUES_PROTESTADOS	Número de cheques.
NIVEL_EDUCATIVO	Indica el nivel de educación del cliente: Primaria, secundaria, universidad o posgrado
CARGAS_FAMILIARES	Número de cargas familiares del cliente
PROFESION	Indica la actividad u ocupación que realiza el cliente.
TIEMPO_TRABAJO_ACTUAL	Indica el tiempo de trabajo actual
PLAZO_OPERACION	Indica el plazo que durara la operación
INHABILITADO	Si un cliente se encuentra inhabilitado en el sistema crediticio
CUOTA_MENSUAL	Cuota que se pagara mensual
Cuota_fracc_ing_dispo	Cuota como fracción del ingreso disponible
PORC_DISPONIBLE	Total de Ingresos menos el total de egresos sobre el total de ingresos
TIPO_VIVIENDA_TIEMPO_ RESIDENCIA_ACTUAL_C	Las variables tipo de vivienda y el tiempo de residencia actual categorizadas por arboles de decisión
ESTADO_CIVIL_EDAD_C	Las variables estado civil y edad categorizadas por arboles de decisión
CARGAS_FAMILIARES_EDAD_C	Las variables cargas familiar y edad por arboles de decisión
EDUCACION_EDAD_C	Las variables educación y edad por arboles de decisión
LIQUIDEZ_C	Total de la deuda en riesgo sobre el total de patrimonio

Fuente: Elaboración propia a partir de los datos suministrados por la institución financiera

Anexo 2 Categorización de variables independientes-árboles

***Tabla 2*:** Categorización con árboles de decisión y valor de información.

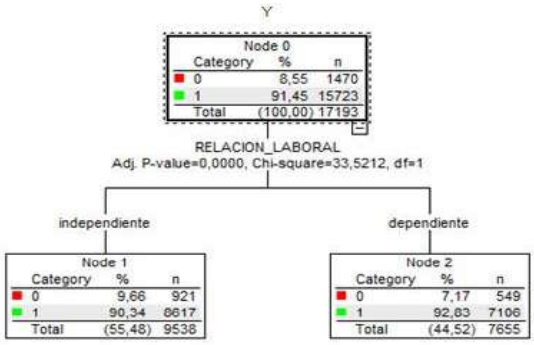
Variable	Grupos- árbol	Nueva categorización	IV
CANTON_C	Riesgo bajo, riesgo medio, riesgo medio-alto, riesgo alto	0; 1; 2; 3	41.52%
ESTADO_CIVIL_EDAD_C	S; D; U+<=51; S; D; U+>51; C; V+<=47; C; V+(47,56); C; V+>56	0; 1; 2; 3; 4	10.12%
CARGAS_FAMILIARES_EDAD_C	EDAD<=47 y CARGAS<=2; EDAD<=47 y CARGAS>2; EDAD (47,56] y CARGAS<=0; EDAD (47,56] y CARGAS>0; EDAD (56,62]; EDAD>62	0;1;2;3;4;5	10.11%
EDUCACION_EDAD_C	NO APICA, S-Secundaria-Primaria -Formación intermedia(técnica)+<=47; NO APICA, S-Secundaria-Primaria -Formación intermedia(técnica)+ (47,62]; NO APICA,S-Secundaria-Primaria -Formación intermedia(técnica)+ >62; universitaria, POSTGRADO <=51;U-universitaria,G-POSTGRADO >51	0;1;2;3;4;5	9.9%
Edad_C	<=47;(47,56];(56,62];>62	0; 1; 2; 3	9.58%
TOTAL_INGRESOS_C	<=79571,9799; (79571,979999999996,520000); >520000	0; 1; 2	8.42%
TIPO_VIVIENDA_TIEMPO_VIVIENDA_C	Propia hipotecada; Propia no hipotecada; Prestada+<=10; Propia hipotecada; Propia no hipotecada; Prestada+>10; Vive con familiares+<=17; Vive con familiares+>17; Arrendada	0; 1; 2; 3; 4	5.69%
TOTAL_PATRIMONIO_C	<=6500;(6500,24051,84) ;(24051,84,111000) ;>111000	0; 1; 2; 3	5.43%
TIEMPO_TRABAJO_ACTUAL_C	<=59;(59,173) ;>173	0; 1; 2	5.26%
TIPO_VIVIENDA_C	Propia hipotecada; Propia no hipotecada; Prestada - Vive con familiares - Arrendada	0; 1; 2	4.78%
PROFESION_C	Riesgo bajo; Riesgo alto	0; 1	4.31%
ESTADO_CIVIL_C	S; D; U - C- V	0; 1; 2	2.75%
RELACION_LABORAL_C	independiente; dependiente	0; 1	2.55%

VALOR_VIVIENDA_C	<=49500; >49500	0; 1	2.53%
PORC_DISPONIBLE_C	<=0,49999418935722667; >0,49999418935722667	0; 1	2%
INHABILITADO_C	S; N	0; 1	1.73%
TIEMPO_RESIDENCIA_ACTUAL_C	<=10; >10	0; 1	1.15%
CUOTA_FRACC_ING_DISPO_C	<=0,011544303797468354; >0,011544303797468354	0; 1	0.59%
CARGAS_FAMILIARES_C	<=0; >0	0; 1	0.54%
NUMERO_CHEQUES_PROTESTADOS_C	<=1; >1	0; 1	0.43%
NIVEL_EDUCATIVO_C	NO APICA; Secundaria; Primaria; Formación intermedia(técnica); U-universitaria; POSTGRADO	0; 1	0.39%
LIQUIDEZ_C	<=0,0088348408710217756; >0,0088348408710217756	0; 1	0.29%
PROVINCIA_C	Riesgo bajo, riesgo medio, riesgo medio-alto, riesgo alto, riesgo muy alto	0; 1; 2; 3; 4	0
TOTAL_EGRESOS	<=4850.299999; >4850.299999	0; 1	0
CARGO	Porcentaje de nulos mayor al 10%		
PARROQUIA			
MONTO_ORIGINAL_OPERACION	Variables no relevantes		
PLAZO_OPERACION			
CUOTA_MENSUAL			

Fuente: Elaboración propia a partir de los datos suministrados por la institución financiera.

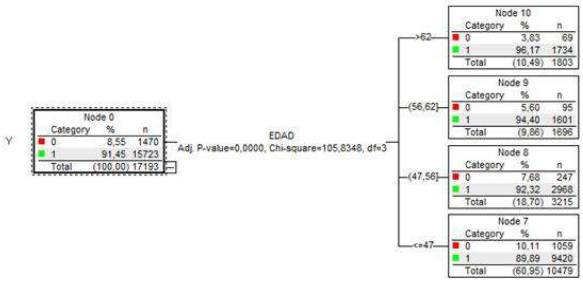
Anexo 2.1 Árboles de decisión

Figura 1
RELACION_LABORAL



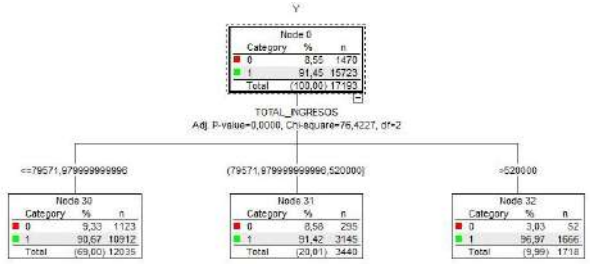
Fuente y elaboración propias

Figura 2
EDAD



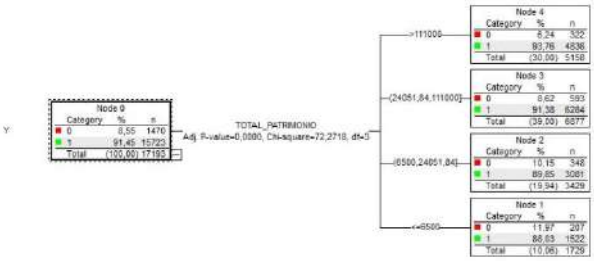
Fuente y elaboración propias

Figura 3
TOTAL_INGRESOS



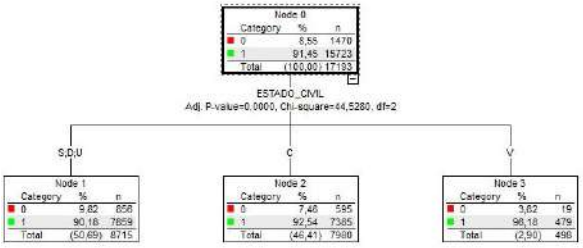
Fuente y elaboración propias

Figura 4
TOTAL_PATRIMONIO



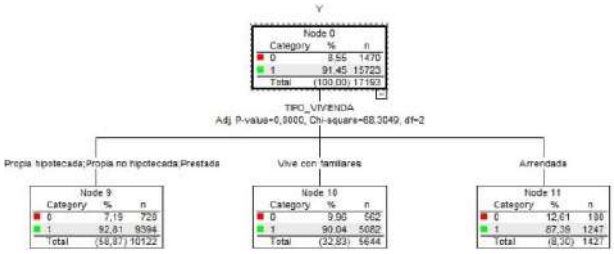
Fuente y elaboración propias

Figura 5
ESTADO_CIVIL



Fuente y elaboración propias

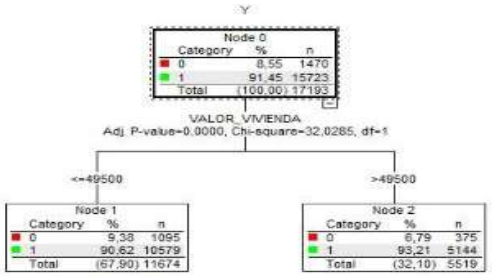
Figura 6
TIPO_VIVIENDA



Fuente y elaboración propias

Figura 7

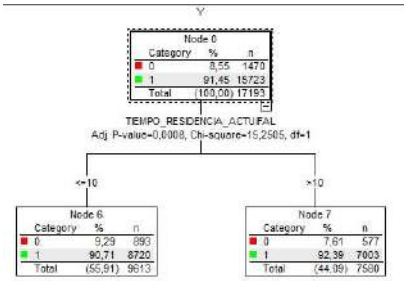
VALOR_VIVIENDA



Fuente y elaboración propias

Figura 8

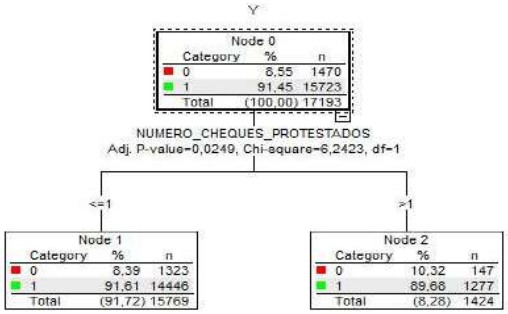
TIEMPO_RESIDENCIA_ACTUAL



Fuente y elaboración propias

Figura 9

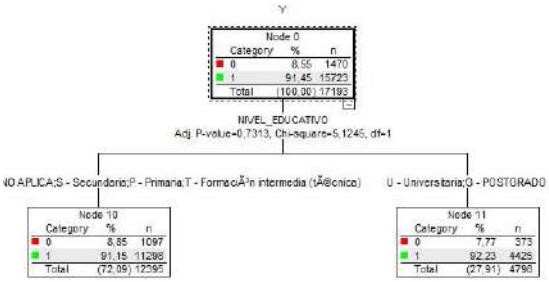
NUMERO_CHEQUES_PROTESTADOS



Fuente y elaboración propias

Figura 10

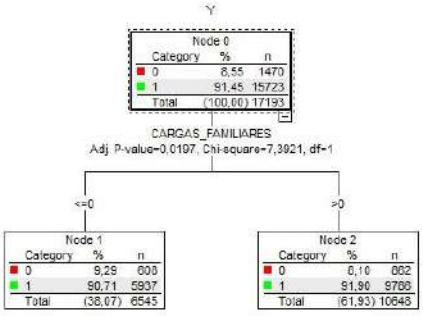
NIVEL_EDUCATIVO



Fuente y elaboración propias

Figura 11

CARGAS_FAMILIARES



Fuente y elaboración propias

Figura 12

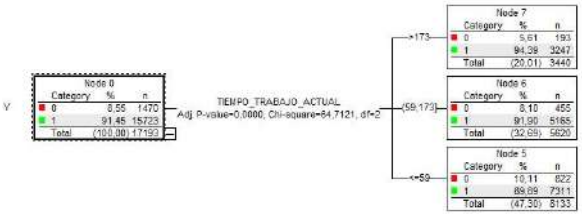
PROFESION

%Malos Por Categoria	
6.64%	**9.87%**
0	**1**
E - Ciencias de la educacion	A - Arquitectos y afines
M - Medicos, Biologos, Veterinarios y otros profesionales de la salud	C - Ciencias Administrativas y Economicas
	D - Derecho
	F - Policias, militares
	I - Ingeniero-a y Ciencias exactas
	P - Periodistas
	S - Ciencias sociales
	NULL
	NO APLICA

Fuente y elaboración propias

Figura 13

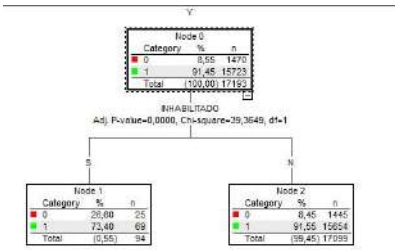
TIEMPO_TRABAJO_ACTUAL



Fuente y elaboración propias

Figura 14

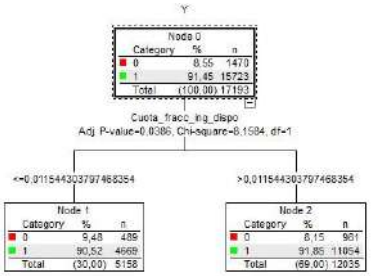
INHABILITADO



Fuente y elaboración propias

Figura 15

CUOTA_FRACC_ING_DISPO



Fuente y elaboración propias

Figura 16

PROVINCIA

%Malos Por Categoria				
19.74%	**13.60%**	**10.20%**	**7.14%**	**4.93%**
0	**1**	**2**	**3**	**4**
04 DEL CARCHI	06 DEL CHIMBORAZO	02 DE BOLIVAR	01 DEL AZUAY	07 DE EL ORO
05 DE COTOPAXI	12 DE LOS RIOS	09 DEL GUAYAS	03 DE CAÑAR	08 DE ESMERALDAS
10 DE IMBABURA	13 DE MANABI	18 DEL TUNGURAHUA		14 DE MORONA SANTIAGO
11 DE LOJA				15 DE NAPO
17 DE PICHINCHA				16 DE PASTAZA
19 DE ZAMORA CHINCHIPE				21 DE SUCUMBIOS
24 SANTA ELENA				22 DE ORELLANA
				23 SANTO DOMINGO DE LOS TSACHILAS

Fuente y elaboración propias

Figura 17

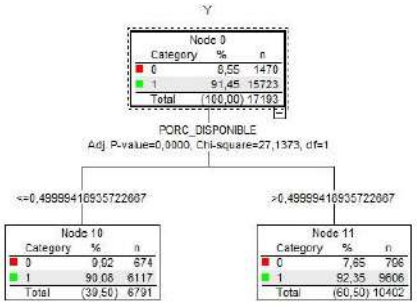
CANTON

%Malos Por Categoria			
4.44%	7.43%	11.51%	19.75%
0	1	2	3
01 AZOGUES	01 CUENCA	01 AMBATO	01 BABAHOYO
01 ESMERALDAS	04 BALSAS	01 GUAYAQUIL	01 GUARANDA
01 IBARRA	05 CHAGUARAPAMBA	04 LA TRONCAL	01 LATACUNGA
01 LAGO AGRIO	07 HUAQUILLAS	05 QUEVEDO	01 LOJA
01 MACHALA	08 MARCABELI	09 PALTAS	01 PORTOVIEJO
01 PASTAZA	09 MONTECRISTI	11 SARAGURO	01 QUITO
01 SANTA ELENA		11 VALENCIA	01 RIOBAMBA
01 SANTO DOMINGO		16 SAMBORONDON	01 ZAMORA
01 TENA			02 ANTONIO ANTE
01 TULCAN			02 BOLIVAR
02 ALAUSI			02 CALVAS
02 ALFREDO BAQUERIZO MORENO			02 LA LIBERTAD
02 ARENILLAS			03 CATAMAYO
02 BAÑOS,BAÑOS DE AGUA SANTA			03 CHONE
02 CAYAMBE			03 PANGUA
02 CHILLANES			04 BALZAR
02 CHINCHIPE			04 CELICA
02 GIRON			04 PEDRO MONCAYO
02 LA MANA			04 PUEBLOVIEJO
03 ATAHUALPA			05 MONTUFAR
03 BALAO			05 PAUTE
03 CAÑAR			05 RUMIÑAHUI
03 CEVALLOS			05 SALCEDO
03 CHIMBO			06 DAULE
03 COLTA			06 EL PANGUI
03 GUALACEO			06 JIPIJAPA
03 LA JOYA DELOS SACHAS			06 SAN MIGUEL DE URCUQUI
03 MEJIA			07 DURAN
03 MONTALVO			07 SAN FERNANDO
03 SALINAS			07 SAN MIGUEL DE LOS BANCOS
04 ECHEANDIA			08 EMPALME
04 EL CARMEN			08 MACARA
04 MOCHA			08 MANTA
04 NABON			10 BUENA FE
04 OTAVALO			10 PUYANGO
04 PUJILI			14 LAS LAJAS
04 QUININDE			15 TOSAGUA
04 YACUAMBI			16 OLMEDO
05 CHILLA			20 SAN JACINTO DE YAGUACHI
05 CHUNCHI			21 JARAMILLO
05 PATATE			
05 SANTIAGO			
06 CALUMA			
06 DELEG			
06 EL GUABO			
06 GUAMOTE			
06 PUCARA			
06 QUERO			
06 SAQUISILI			
07 GONZANAMA			
07 GUANO			
07 JUNIN			
07 SAN PEDRO DE PELILEO			
07 VENTANAS			
08 PALANDA			
08 SANTA ISABEL			
08 SANTIAGO DE PILLARO			
08 VINCES			
09 EL TRIUNFO			

Fuente y elaboración propias

Figura 18

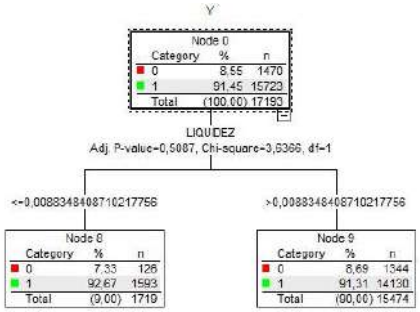
PORC_DISPONIBLE



Fuente y elaboración propias

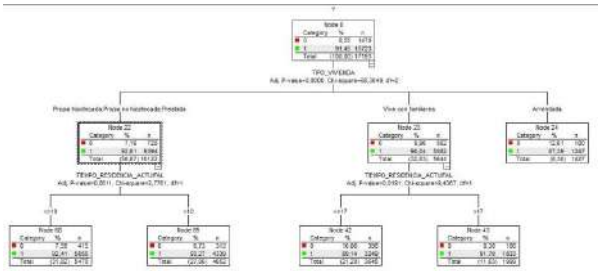
Figura 19

LIQUIDEZ



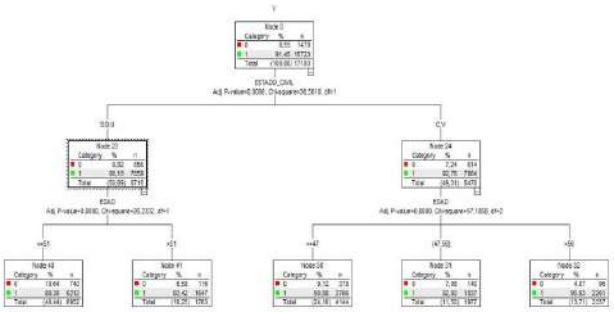
Fuente y elaboración propias

Figura 20
TIPO_VIVIENDA_TIEMPO_VIVIENDA



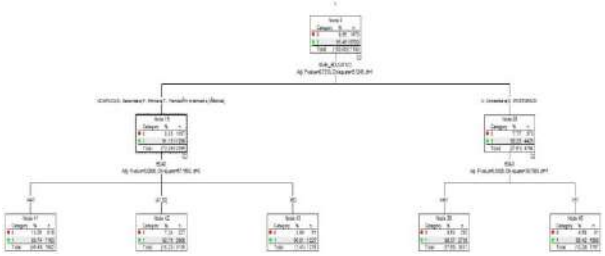
Fuente y elaboración propias

Figura 21
ESTADO_CIVIL_EDAD



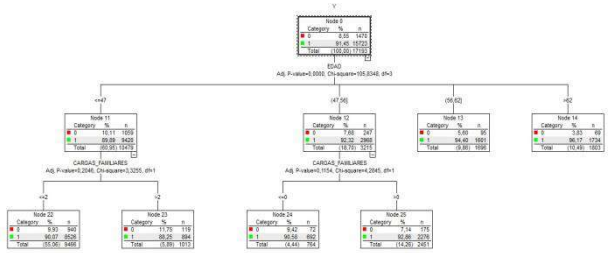
Fuente y elaboración propias

Figura 22
EDUCACION_EDAD



Fuente y elaboración propias

Figura 23
CARGAS_FAMILIARES_EDAD



Fuente y elaboración propias

Anexo 3 CÓDIGO R

SCRIPT MODELO SCORE LOGIT

```
## Construcción del modelo Logit

set.seed(7)

Modelo_Logistico <- step(glm(reg,data = Matriz_Datos,

 family=binomial(link="logit")), trace=0)

### Prueba GVIF

vif(Modelo_Logistico)

##Cálculo del z, valor de probabilidad y Score

Matriz_Modelo_F2=df6_Dicotomica_Modelo

Matriz_Modelo_F2$Z = predict.glm(Modelo_F, type='link',
Matriz_Modelo_F)

Matriz_Modelo_F2$prob <- predict(Modelo_F,
Matriz_Modelo_F,type = 'response')

Matriz_Modelo_F2$Score = round(1000 * Matriz_Modelo_F$prob,
0)

m1_pred <- prediction(Matriz_Modelo_F2$prob ,
Matriz_Modelo_F2$Y)

m1_perf <- performance(m1_pred,"tpr","fpr")

## KS, ROC, GINI

KS <- round(max(abs(attr(m1_perf,'y.values')[[1]]-
attr(m1_perf,'x.values')[[1]])*100), 2)

ROC <- round(performance(m1_pred, measure =
"auc")@y.values[[1]]*100, 2)

Gini <- (2*ROC - 100)

##Punto de corte y matriz de confusión

optCutOff <- optimalCutoff(Matriz_Modelo_F1$Y,
Matriz_Modelo_F1$prob1, optimiseFor = "Both",
returnDiagnostics = TRUE)

PuntoCorte <- optCutOff$optimalCutoff
```

```
a=confusionMatrix(Matriz_Modelo_F1$Y,
Matriz_Modelo_F1$prob1,threshold = PuntoCorte1)

Error=100*(a[2,1]+a[1,2])/sum(a)

Matriz_Modelo_F1$Y_estimado <-
ifelse(Matriz_Modelo_F1$prob1>PuntoCorte1,1,0)

MatrizConfusion <-
confusionMatrix(Matriz_Modelo_F1$Y,Matriz_Modelo_F1$Y_estimad
o)

Sensibilidad_M  <-
100*sensitivity(Matriz_Modelo_F1$Y,Matriz_Modelo_F1$Y_estimad
o, threshold = PuntoCorte1)

Especificidad_M <-
100*specificity(Matriz_Modelo_F1$Y,Matriz_Modelo_F1$Y_estimad
o, threshold = PuntoCorte1)

Aciertos=100-Error

##Remuestreo ROSE

base_modelo=df6_Dicotomica_Modelo


base_modelo=base_modelo[,c("Y",df5_Significancia_var[,1])]

base_modelo0<-ovun.sample (Y~.,
                           data = base_modelo,
                           method = "over",p = 0.20, # buenos
80%- malos 20%
                           seed=1)$data


table(base_modelo0$Y) %>% prop.table()

##  Characteres a Factores

base_modelo0=base_modelo0 %>%
mutate_if(is.character,as.factor)
```

SCRIPT MODELO SCORE GRADIENT BOOSTING

```
Matriz_Modelo_F1=base_modelo0

set.seed(1000)

gbm_simple=gbm(Y~. ,distribution = "bernoulli",

               data = Matriz_Modelo_F1,

               n.trees = 1000,

               interaction.depth =5,

               shrinkage = 0.03,cv.folds = 20,n.cores = 8)

## Gráfico importancia relativa de variables

a=summary(gbm_simple)

fig <- plot_ly(data = a,y =~reorder(var,rel.inf) , x
=~rel.inf , type = 'bar', orientation = 'h')

##Calculo de número de clasificadores débiles error OOB

ntree_opt_oob=gbm.perf(gbm_simple,method = "OOB",oobag.curve
= T)

##Cálculo de probabilidad

Matriz_Modelo_F1$prob1 <- predict(object =gbm_simple,
newdata = Matriz_Modelo_F1,
n.trees = ntree_opt_oob, type = "response")

m1_pred <- prediction(Matriz_Modelo_F1$prob1 ,
Matriz_Modelo_F1$Y)

m1_perf <- performance(m1_pred,"tpr","fpr")

## KS, ROC, GINI

KS <- round(max(abs(attr(m1_perf,'y.values')[[1]]-
attr(m1_perf,'x.values')[[1]])*100), 2)

ROC <- round(performance(m1_pred, measure =
"auc")@y.values[[1]]*100, 2)

Gini <- (2*ROC - 100)

### Cálculo punto de Corte y matriz de confusión

optCutOff <- optimalCutoff(Matriz_Modelo_F1$Y,
Matriz_Modelo_F1$prob1, optimiseFor = "Both",
returnDiagnostics = TRUE)
```

```
PuntoCorte <- optCutOff$optimalCutoff

a=confusionMatrix(Matriz_Modelo_F1$Y,
Matriz_Modelo_F1$prob1,threshold = PuntoCorte1)

Error=100*(a[2,1]+a[1,2])/sum(a)


Matriz_Modelo_F1$Y_estimado <-
ifelse(Matriz_Modelo_F1$prob1>PuntoCorte1,1,0)

MatrizConfusion <-
confusionMatrix(Matriz_Modelo_F1$Y,Matriz_Modelo_F1$Y_estimad
o)


Sensibilidad_M  <-
100*sensitivity(Matriz_Modelo_F1$Y,Matriz_Modelo_F1$Y_estimad
o, threshold = PuntoCorte1)

Especificidad_M <-
100*specificity(Matriz_Modelo_F1$Y,Matriz_Modelo_F1$Y_estimad
o, threshold = PuntoCorte1)

Aciertos=100-Error
```

SCRIPT MODELO SCORE ADABOOST

```
Matriz_Modelo_F1=base_modelo0

set.seed(1000)

gbm_simple=gbm(Y~. ,distribution = "adaboost",

               data = Matriz_Modelo_F1,

               n.trees = 1000,

               interaction.depth =5,

               shrinkage = 0.03,cv.folds = 20,n.cores = 8)

## Gráfico importancia relativa de variables

a=summary(gbm_simple)

fig <- plot_ly(data = a,y =~reorder(var,rel.inf) , x
=~rel.inf , type = 'bar', orientation = 'h')
```

```
fig


##Calculo de número de clasificadores débiles error OOB

ntree_opt_oob=gbm.perf(gbm_simple,method = "OOB",oobag.curve
= T)



Matriz_Modelo_F1$prob1 <- predict(object =gbm_simple,
                                  newdata = Matriz_Modelo_F1,
                                  n.trees = ntree_opt_oob,
                                  type = "response")


m1_pred <- prediction(Matriz_Modelo_F1$prob1 ,
Matriz_Modelo_F1$Y)

m1_perf <- performance(m1_pred,"tpr","fpr")


## KS, ROC, GINI

KS <- round(max(abs(attr(m1_perf,'y.values')[[1]]-
attr(m1_perf,'x.values')[[1]])*100), 2)

ROC <- round(performance(m1_pred, measure =
"auc")@y.values[[1]]*100, 2)

Gini <- (2*ROC - 100)


## Cálculo punto de Corte y matriz de confusión

optCutOff <- optimalCutoff(Matriz_Modelo_F1$Y,
Matriz_Modelo_F1$prob1, optimiseFor = "Both",
returnDiagnostics = TRUE)

PuntoCorte <- optCutOff$optimalCutoff

a=confusionMatrix(Matriz_Modelo_F1$Y,
Matriz_Modelo_F1$prob1,threshold = PuntoCorte1)
```

```
Error=100*(a[2,1]+a[1,2])/sum(a)

Matriz_Modelo_F1$Y_estimado <-
ifelse(Matriz_Modelo_F1$prob1>PuntoCorte1,1,0)

MatrizConfusion <-
confusionMatrix(Matriz_Modelo_F1$Y,Matriz_Modelo_F1$Y_estimad
o)

Sensibilidad_M  <-
100*sensitivity(Matriz_Modelo_F1$Y,Matriz_Modelo_F1$Y_estimad
o, threshold = PuntoCorte1)

Especificidad_M <-
100*specificity(Matriz_Modelo_F1$Y,Matriz_Modelo_F1$Y_estimad
o, threshold = PuntoCorte1)

Aciertos=100-Error
```