ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE CIENCIAS

MÉTODOS Y PROCESOS SOFISTICADOS DE REDUCCIÓN DE DIMENSIÓN Y SELECCIÓN DE VARIABLES PARA UN MODELO DE RESPUESTA BINARIA.

TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERA MATEMÁTICA

PROYECTO DE INVESTIGACIÓN

JOSSELYN LIZETH MORENO MANOBANDA
josselyn.moreno@epn.edu.ec

Directora: ADRIANA UQUILLAS ANDRADE, PHD
adriana.uquillas@epn.edu.ec

QUITO, MAYO, 2022

DECLARACIÓN

Yo JOSSELYN LIZETH MORENO MANOBANDA, declaro bajo juramento que el trabajo aquí escrito 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.

A través de la presente declaración cedo mis derechos de propiedad intelectual, correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su reglamento y por la normatividad institucional vigente.

Josselyn Lizeth Moreno Manobanda

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por JOSSELYN LIZETH MORENO MANOBANDA, bajo mi supervisión.

ADRIANA UQUILLAS ANDRADE, PhD
Director del Proyecto

AGRADECIMIENTOS

A mi padre Diego Moreno y mi madre Gloria Manobanda, por su apoyo incondicional, por su amor, paciencia, sus palabras de aliento y sobre todo por impulsarme a seguir adelante.

A mi hermano Henry que, con su ejemplo, me ha demostrado que también soy capaz de ser una persona igual de exitosa y a ver la vida de manera más sencilla.

A mi hermana Maribel, por ser mi confidente y mi amiga. Gracias por escucharme y apoyarme.

A Raul, mi mejor amigo, por su amor, ternura, soporte y amparo.

A mi directora de tesis, la Dra. Adriana Uquillas, por confiar en mí, por su apoyo absoluto y por ser una inspiración profesional.

A mis compañeros que tuve el agrado de conocer Nicolas, Karencita, Eve, Tefita y Mabe.

DEDICATORIA

A mi querida y estimada madre. Que Dios me la haga eterna.

Índice general

Índice de figuras

Índice de cuadros

Resumen

El desafío actual para investigadores y profesionales es la tasa de desarrollo respecto a datos recopilados. La manipulación de esta gran cantidad de datos es considerado actualmente un tema abierto y de gran auge. Parte de la información que se recolecta usualmente es irrelevante e innecesaria, lo que motiva a los analistas de datos a utilizar recursos tecnológicos que excluyan esta información, alcanzado resultados interpretables referentes algún fenómeno.

En el presente trabajo, se plantean métodos y técnicas de selección de variables y reducción de dimensiones para sobrellevar el problema del manejo de grandes cantidades de datos. Por un lado, se profundiza en algoritmos de aprendizaje automático y luego, en técnicas de selección poco comunes. Se aplican estos métodos y finalmente se realiza una validación de los resultados concluyendo un modelo propuesto adecuado. La base de datos con la que se trabaja en este proyecto proviene de una consultoría privada.

Palabras clave: Aprendizaje Automático, Big Data, modelos de respuesta binaria, MARS, Boruta, Lasso.

Abstract

The current challenge for researchers and professionals is the rate of development regarding data collected. The manipulation of this large amount of data is currently considered an open and booming topic. Part of the information that is collected is usually irrelevant and unnecessary, which motivates data analysts to use technological resources that exclude this information, achieving results referring to some phenomenon.

In this paper, methods and techniques for variable selection and dimension reduction are proposed to overcome the problem of handling large amounts of data. On the one hand, it goes deeper into machine learning algorithms and then into unusual selection techniques. These methods are applied and finally a validation of the results is carried out, concluding an adequate proposed model. The database used in this project comes from a private consultancy.

Keywords: Machine Learning, Big Data, binary response models, MARS, Boruta, Lasso.

Capítulo 1

Introducción

1.1. Antecedentes

En la actualidad, la humanidad genera una excesiva cantidad de datos diarios que se extraen de muchas fuentes tecnológicas modernas, tales como: Facebook, Twitter, Instagram Youtube, entre otros. De hecho, [33], expone que, en los últimos 4 años se ha generado el 90 % de los datos del mundo. Los números son extremadamente altos sin mencionar que, incluso si no se hace uso de redes sociales también se genera información a partir de otras herramientas de recolección como: web logs, sensores incorporados en dispositivos, teléfonos inteligentes, dispositivos GPS y búsquedas en internet, entre otros. Este fenómeno surge a partir de mediados de los años ochenta y mientras que las tecnologías y las computadoras mejoran su capacidad de velocidad, posibilidad de almacenamiento y procesamiento, seguirán generando nuevos problemas y serán difíciles de implementar sin nuevas herramientas analíticas que permitan ir orientando a los usuarios [45].

Un desafío importante para los investigadores y profesionales es que la tasa de crecimiento de datos recopilados está superando rápidamente su capacidad para diseñar sistemas apropiados para manejarlos, esto demanda establecer nuevos métodos de manipulación de datos para extraer información no banal, permitiendo simplificar la obtención de patrones y su análisis [4]. Sin embargo, es altamente complicado valorar esos millones de datos aprisionados en tiempo y espacio. Si bien en algunos casos es bueno tener muchos datos el analizar todas y cada una de las variables a nivel microscópico, se convierte en un desafío debido al tamaño [33]. Usualmente tener a disposición más características o más variables en una base de datos, implica un modelado predictivo más difícil, lo que generalmente se conoce

como la maldición de la dimensionalidad [9]. Es decir, se necesita una mejor manera de tratar con datos de gran dimensión para que se pueda extraer rápidamente patrones e información de ellos.

Parte de la información recolectada se convierte en irrelevante para el problema en cuestión por el hecho de que el almacenamiento y la recolección de datos es ahora mucho más accesible. Entonces, al excluir esta información, se logra explicar de una manera más adecuada el fenómeno [11]. A su vez, la excedencia de información podría genera colinealidad entre las variables, produciendo estimaciones erróneas. Además, variables predictoras innecesarios añaden ruido a la estimación ocultando las variables realmente importantes. Todo esto, motivan a la utilización de métodos de reducción de la dimensionalidad y selección de variables, que permiten abreviar la descripción del conjunto de datos y que sean capaces de cubrir altos volúmenes de información en tiempos prudentes.

Cabe mencionar que, las técnicas de reducción de dimensionalidad eficientes mapean el conjunto de datos a subespacios oriundo del espacio original a menor dimensión, en los que se encuentran todo el conglomerado de la información [48]. De manera análoga, los métodos de selección de variables se refieren a procedimientos en los que se selecciona un subconjunto de variables a partir de uno original, basados en algunos criterios de evaluación singular. Por lo que, el seleccionar y reducir cumplen con el objetivo de eliminar la redundancia de un conjunto de datos.

Adicionalmente, el problema de la selección y reducción de dimensionalidad es fundamental en tareas como: procesamiento de imágenes, minería de datos y clasificación, entre otros. No obstante, estos problemas siguen siendo en la actualidad problemas de investigación abierta dada su complejidad [11].

De manera general, existen varios métodos implementados como los métodos de mínimos cuadrados, análisis de componentes principales (ACP) y bosques aleatorios. Asimismo, se aplican estos métodos de manera puntual para distintos tipos de problemas multivariados [18]. Sin embargo, no se efectúa previamente un proceso metódico antes de la utilización de cualquiera de los métodos existentes para selección y reducción.

El principal problema que se aborda en este proyecto se centra en construir un proceso robusto de tratamiento y transformación de variables, seguido de reducción de dimensiones y selección de variables a ser aplicados en un modelo de respuesta binaria, utilizando el método de regresión logística. En consecuencia, buscamos llegar a un modelo parsimonioso (entre 10 a 15 variables), considerando una base

de partida con un volumen grande de datos, posiblemente alrededor de 2000 variables, disponibles en una empresa de consultoría privada. El planteamiento del mejor proceso se realizará en términos de algunos indicadores de correcta especificación y desempeño del modelo econométrico de regresión logística como: el estadístico de Kolmogorov - Smirnov, coeficiente Gini y curva roc, entre otros.

1.2. Justificación

En el aprendizaje automático, para captar información útil y obtener un resultado más preciso, se tiende a agregar funciones como sea posible. Sin embargo, después de cierto punto, el rendimiento del modelo disminuirá con el aumento de la cantidad de elementos. Esto conlleva a los problemas de sobreajuste y la maldición de la dimensionalidad [9]. Para asegurar un modelo de regresión logística adecuado, el problema en cuestión estudiará los mejores métodos de selección de variables y reducción de dimensiones. Una forma particular de realizar esto, consiste en elegir de entre diversos métodos el que mejor se ajuste a los datos, lo cual implica que se debe construir un proceso de selección utilizando técnicas matemáticas y estadísticas de manera que se analice la estructura interna de los datos. Si bien es cierto, estudios como los de [43] y [18], muestran resultados dónde se cumplen con los objetivos de reducir espacio de almacenamiento y acelerar tiempo necesario para realizar cálculos. Por otra parte, varios de ellos se basan únicamente en evaluar y aplicar diferentes métodos por separado, para posteriormente compararlos a través de estudios de simulación o aplicarlos en algún conjunto de datos reales, para concluir los distintos beneficios o utilidades. Para sobrellevar este problema surge la creación de un proceso de transformación, selección de métodos sofisticados de reducción de dimensiones y selección de variables finales, que incorpora varios de los métodos conforme al estado del arte de la investigación.

La importancia de la creación de este proceso de selección subyace en que, este proceso permitirá no solo comparar los distintos métodos si no disminuir tiempo en distinguir los mejores métodos acordes al modelo planteado, pues como se mencionó en el planteamiento del problema existen varios procesos ya implementados que son utilizados según su área de interés como pueden ser los modelos de aprendizaje supervisado como no supervisado. El estudio más relevante y similar que se ha publicado sobre la creación de un proceso de selección se describe en [25], dónde se realiza un enfoque práctico para modelos predictivos. Esta investigación ilustra una serie de estrategias y técnicas como los métodos de búsqueda global que

investigan un conjunto diverso de soluciones potenciales de forma que se pueden usar para seleccionar métodos de selección de variables y mejorar los modelos. Esta investigación nos permitirá ser la base de estudio. Trabajos relacionados son los que realiza Gonzáles Vidal (2015), donde se aplica una revisión de métodos existentes de selección en su estudio dónde se utiliza algoritmos, método lasso, métodos de mínimos cuadrados penalizados, método SCAD y método bridge. Para el caso métodos de reducción de dimensionalidad, [42] manifiestan en su investigación siete métodos, tales como: eliminación de columnas de datos con demasiados valores perdidos, filtro de baja varianza, reducción de columnas altamente correlacionadas, análisis de componentes principales (ACP), reducción de la dimensionalidad, (a través, de conjuntos de bosques aleatorios), eliminación de características hacia atrás y construcción de funciones hacia adelante. Se espera que el proceso de selección planteado mejore la eficacia de los algoritmos de Aprendizaje Automático. Realizar esta tarea en muchos dominios es extremadamente aconsejable porque los algoritmos escalan de una manera adecuada a medida que aumentan la magnitud de los datos y disminyen radicalmente su eficiencia en la resolución de problemas con estas características [50]. Así, dado que se contará con datos reales, la metodología planteada estará mejor alineada con la realidad.

La aplicación de este proceso robusto beneficiará a cualquier caso de modelización con respuesta binaria. Desde 1980 se ha incrementado el afecto por los modelos con respuesta binaria para estudiar como los individuos toman decisiones. La gracia de estos métodos se debe en su gran mayoría al sencillo acceso de datos microeconómicos, ya que permiten explicar su conducta y como su comportamiento afecta otras variables del mercado [12]. Un modelo con respuesta binaria se utiliza cuando queremos predecir un resultado binario, por ejemplo, quiebra vs no quiebra. Otro caso está en el área del Marketing, que se requiere conocer si en efecto un producto se compra o no. Casos más grandes como es el de la compañía de Netflix con aproximadamente 100 millones de usuarios, se basa en que la plataforma busca recomendar una película satisfaciendo los gustos de sus suscriptores. Del mismo modo, Amazon propone al cliente a través de la búsqueda realizada, productos relacionados que complementen o no su compra. Existen otros ejemplos que trabajan con modelos de respuesta binaria y al mismo tiempo trabajan con gran cantidad de datos que, de una u otra forma, harán uso del proceso a construir. En el tiempo actual, cualquier institución se propone entre sus objetivos exponer a sus usuarios calidad en sus servicios. En este marco, la utilidad de los modelos logit en valoración de programas comprende una gran variedad de contextos de intervención. De

esta forma, existen estudios llevados a cabo en el ámbito laboral como la de Roy y Wong (1999), sobre incidentes en el incumplimiento por regulación del seguro y paro. Otras investigaciones en el ámbito educativo como las de García, Alvarado y Jiménez (2000), que utilizan modelos logit para predecir el rendimiento académico en universitarios. Los resultados obtenidos en estos artículos, nos motivó el crear el proceso de selección de variables y reducción de dimensiones para una base de datos con un número de variables consideradamente grande (alrededor de 2000).

1.3. Objetivos

1.3.1. Objetivo General

Construir un proceso de selección de variables para modelos con variable respuesta dicotómica.

1.3.2. Objetivos Específicos

- Estructurar y homologar las bases de datos disponibles para el estudio.
- Analizar y entender la teoría matemática de los modelos de regresión logística. (método de modelización para respuesta binaria).
- Identificar y realizar un estudio del arte de los métodos existentes de reducción de la dimensionalidad y selección de variables estableciendo pros y contras de cada método.
- Simulación de distintos métodos en la base de datos y validación de los resultados.

Capítulo 2

Marco Teórico

En este capítulo se describe en primer lugar, conceptos sobre aprendizaje automático, algoritmos, minería de datos y técnicas de validación de modelos. Luego, se definen los métodos más importantes de reducción de dimensión y selección de variables que se encuentran detrás de la metodología planteada.

2.1. Conceptos previos

2.1.1. Minería de Datos

Pese a que la idea de Minería de Datos simula ser una invención tecnológica moderna, en realidad este término surgió cerca de los años sesenta. A pesar de ello, no fue hasta fines de la década de 1980 y principios de la de 1990 cuando inició su consolidación.

Acorde con [5], la minería de datos es el proceso de clasificar grandes conjuntos de datos que permitan identificar patrones y relaciones que ayuden a resolver problemas a través del análisis de datos. Por su parte, el minado de datos utiliza varias técnicas y tecnologías de análisis avanzadas para encontrar información útil a una magnitud más pequeña, que nos permita explorar grandes bases de datos.

Cabe mencionar que, los elementos centrales en la funcionalidad de la minería de datos incluyen el aprendizaje automático y el análisis estadístico, junto con las tareas de administración de datos realizadas para preparar los datos para el análisis. El uso de algoritmos de aprendizaje automático y herramientas de inteligencia artificial (IA) han automatizado más el proceso y han facilitado la extracción de conjuntos de datos masivos, como bases de datos de clientes, registros de transacciones

y archivos de registro de servidores web, aplicaciones móviles y sensores [5].

Basados en los descrito por [5] a continuación, se describe el proceso de minería de datos, el cual se puede dividir en cuatro etapas principales:

1. *Recopilación de datos:* Los datos pueden estar localizados en diferentes sistemas de origen, un almacén de datos o un lago de datos. Es por ello necesario tener un repositorio que contienen una combinación de datos estructurados y no estructurados. A su vez, también se puede utilizar fuentes de datos externas.

2. *Preparación de datos:* En esta etapa se incluyen varios pasos para preparar los datos para ser extraídos. Además, la transformación de datos también se realiza en esta etapa, para que los conjuntos de datos sean consistentes, sin embargo, no es necesaria si un científico de datos busca analizar datos sin procesar para una aplicación en particular.

3. *Minería de datos:* Aquí, se aplican técnicas de minería de datos adecuadas. De hecho, en las aplicaciones de aprendizaje automático, los algoritmos por lo regular deben entrenarse en conjuntos de datos de muestra para buscar la información antes de que ejecuten con el conjunto completo de datos.

4. *Análisis e interpretación de datos:* Los resultados de la etapa anterior se emplean para crear modelos analíticos que contribuyen a impulsar la toma de decisiones y otras acciones comerciales.

2.1.2. Aprendizaje Automático

Repasando un poco su historia, los comienzos del Aprendizaje Automático se hallan en los años 50s, cuando Arthur Samuel, personaje precursor en los juegos informáticos, compuso el primer programa de aprendizaje informático. Este programa era un juego de damas, que gracias a su estudio de movimientos que componían estrategias ganadoras, cooperó a que la computadora perfeccionara en el juego acorde más jugaba.

En esta misma década, Frank Rosenblatt inventó el Perceptro, un tipo de red neuronal que asemeja al cerebro humano. Perceptron conectaba una red de puntos donde se tomaban resoluciones simples.

Más tarde, en los años 60s, se creó el algoritmo "vecino más cercano", el cuál facultó a las computadoras emplear un reconocimiento de patrones básicos.

Mas adelante, a comienzos de los años 80, Gerald Dejong traza la noción de 'Aprendizaje Basado en Explicación'. Conocimiento en el que se analiza datos de entrenamiento en la computadora y se crea un patrón que sigue para descartar datos.

Para la década de los 90s, el Aprendizaje Automático cobró fama gracias al cruce de la informática y la estadística que dio lugar a una perspectiva probabilística en la inteligencia artificial.

Finalmente, en este nuevo milenio se han explotado nuevos términos, como es el término 'Aprendizaje profundo', con el que se exponen nuevas edificaciones de Redes Neuronales que permiten a las computadoras ver y distinguir texto en imágenes y objetos [47]. No obstante, empresas tecnológicas comenzaron sus propios desarrollos en Aprendizaje Automático como Google, Facebook y Amazon, entre otros.

El aprendizaje automático se ha convertido en una herramienta común en casi cualquier tarea que requiera la extracción de información de grandes conjuntos de datos. De manera que los algoritmos de aprendizaje automático permiten que las computadoras se comuniquen con humanos, conduzcan automóviles de manera autónoma, las cámaras digitales aprenden a detectar caras, los vehículos están equipados con sistemas de prevención de accidentes, entre otros. Su término se refiere a la detección automatizada, a partir de algoritmos informáticos, patrones en los datos. Así mismo, el aprendizaje automático ("Machine Learning"(ML)) es una de las áreas de la informática de crecimiento más rápido, con aplicaciones de gran alcance [19].

En este sentido, se comprende que los algoritmos del aprendizaje automático funcionan con más o menos intervención/refuerzo humano. Los cuatro modelos principales de aprendizaje automático son el aprendizaje supervisado, el aprendizaje no supervisado, el aprendizaje semisupervisado y el aprendizaje de refuerzo [23].

Desde luego, Aprendizaje Automático se considera un gran avance para ofrecer una mejor vida a las personas, a pesar de ello, muchos investigadores advierten peligros.

Etapas del aprendizaje

De acuerdo con [26], se describe las diferentes etapas a la hora de usar Aprendizaje Automático, que se siguen para solucionar un problema en la práctica y obtener buenos resultados. Además, se tendrá una orientación sobre qué es lo que debemos hacer en caso de que los resultados no sean tan buenos como se esperan.

Ilustramos estas etapas con la siguiente gráfica:

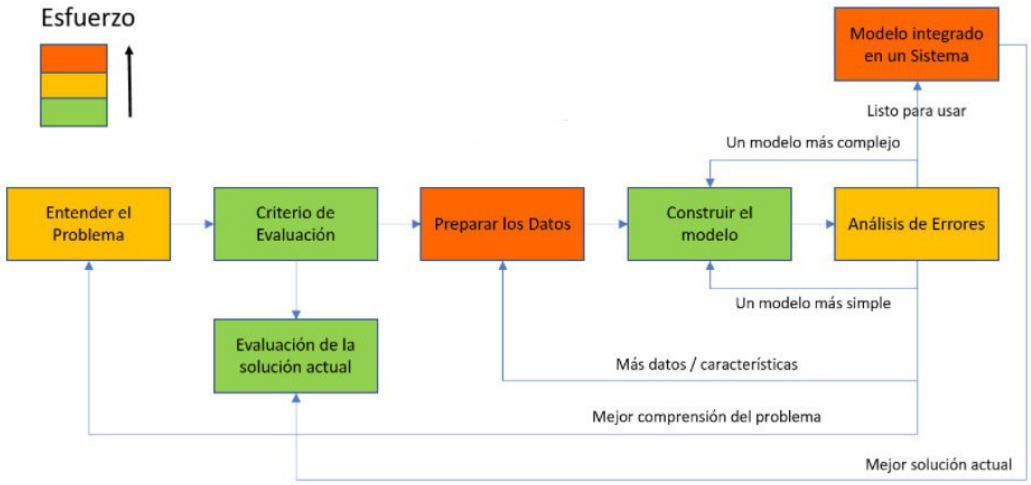

Figura 2.1: 7 etapas del Proceso de Aprendizaje Automático (Martinez, 2020.)

Se inicia por entender el problema que se tiene que resolver, saber qué tipo de problema es, si es un problema de regresión, clasificación o agrupación. En muchas situaciones resulta encontrarse una solución óptima para un problema diferente al que se tenía. Para ello plantearse preguntas ayudan a tener un mejor entendimiento del problema, conjuntamente con el entendimiento de los datos.

Acto seguido, es habitual realizar un análisis exploratorio de datos para familiarizarse con los datos. En este se suelen hacer correlaciones, gráficos y estadísticas descriptivas para comprender mejor qué suceso cuentan los datos. Posteriormente se define un criterio de evaluación para analizar si se obtienen o no, un modelo perfecto. La pauta de evaluación se trata normalmente de una medida de error, tipicamente se usa el error cuadrático medio para problemas de regresión. No obstante, existen ya diveros criterios de valoración estándar que son útiles para muchos problemas. Además, son tan generales que funcionan en casi todos los sectores e industrias.

Más adelante, se evalúa la solución actual, ya que probablemente el problema

que se quiere resolver con Aprendizaje Automático, esté resuelto de otra forma, sin embargo, se desea obtener mejores resultados o resultados parecidos de forma automática, sustituyendo quizás un trabajo manual tedioso.

Preparar los datos, por otro lado, es una de los aspectos del Aprendizaje Automático que supone un mayor esfuerzo, cumpliendo retos como; tener datos incompletos, combinar datos de varias fuentes, determinar un formato idóneo a los datos, normalización de datos y calcular características relevantes, además de la partición de manera aleatoria de los datos en entrenamiento y test. Nuestro conjunto de entrenamiento estará destinado a que el modelo realice el proceso de aprendizaje, mientras qué el conjunto test se usa con el objetivo de evaluar el desempeño del modelo.

Una vez que se preparan los datos se procede a construir el modelo, medir su error y finalmente integrarlo en un sistema.

2.2. Algoritmos de Aprendizaje Automático

Las técnicas de Aprendizaje Automático se fraccionan en cuatro grandes grupos, el aprendizaje supervisado, el aprendizaje no supervisado, el aprendizaje semisupervisado y el aprendizaje de refuerzo.

Dentro del aprendizaje supervisado, encontramos modelos de clasificación y modelos de regresión. En el aprendizaje no supervisado se basa en algoritmos de agrupación o clustering, en el que los datos no tienen etiquetas, no se sabe a qué categoría pertenecen. Para el aprendizaje sumisupervisado se utilizan datos de entrenamiento tanto etiquetados como no etiquetados y finalmente el aprendizaje por refuerzo se basa en un sistema de prueba y error [49]. La Figura 2.2 esquematiza lo mencionado anteriormente para mejor entendimiento.

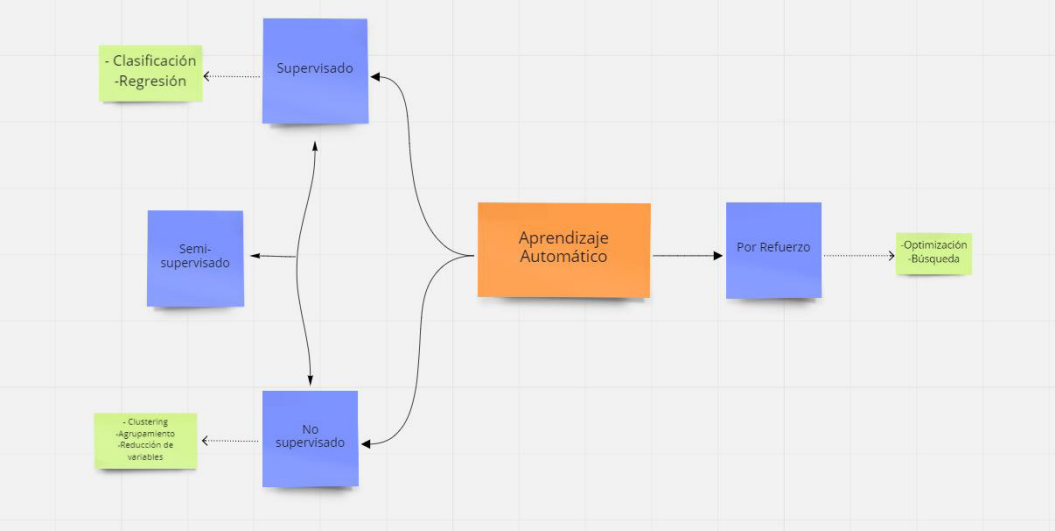

Figura 2.2: Esquema de modelos en Aprendizaje Automático. **Fuente:** Elaboración propia.

Ahora bien, este proyecto está centrado en el Aprendizaje Supervisado dónde se encuentran los problemas de clasificación como por ejemplo procesar solicitudes de préstamo en solventes o de riesgo, detección de fraudes, diferenciar mensajes de correo tipo spam o importantes, etc. Así mismo, se encuentran los problemas de regresión como por ejemplo predecir distancias, determinar la estatura de una persona en función de su edad, estimación de ventas de productos, predicciones meteorológicas, etc. A continuación, se presentan estos dos problemas.

2.2.1. Algoritmos de Regresión

El análisis de regresión establece un método para la relación entre una variable continua y un cierto número de características, donde el resultado de la técnica de aprendizaje automático que se use será un número, dentro de un conjunto entero de opciones. Algunos de los métodos de Aprendizaje Automático para problemas de regresión que se usan se destacan:

- Regresión lineal.
- Árboles de decisión.
- Aprendizaje profundo (Deep Learning).
- Redes neuronales.
- Bosques aleatorios.

- Regresión logística (se utiliza datos discretos).

Una aclaración importante es la confusión frecuente con la técnica de regresión logística. Su nombre semeja inclinarse a pensar que se puede usar en problemas de regresión. No obstante, la regresión logística solo funciona en problemas de clasificación [27].

2.2.2. Algoritmos de Clasificación

Cuando se usa clasificación se obtiene como resultado una clase, entre un número finito de clases. Además, en el Aprendizaje Supervisado este algoritmo se usa cuando el resultado es una etiqueta discreta, que es el caso de este proyecto.

En la actualidad, las técnicas de Aprendizaje Automático para problemas de clasificación más notables son:

- Regresión logística.
- Árboles de decisión.
- Redes neuronales.
- Aprendizaje profundo.
- Clasificación de Naive Bayes
- Bosques aleatorios.

Es necesario enfatizar en estos algoritmos dado su uso en este proyecto. Por ende, se presenta a continuación técnicas de Aprendizaje Automático utilizadas en este proyecto para mejor entendimiento.

Regresión Lineal Múltiple

La regresión es la técnica básica del análisis econométrico que trata de modelar la relación entre una o más variables independientes X_i (denominadas regresoras o opredictores) con la variable dependiente Y (también conocida como la respuesta). En este sentido, se desarrollan modelos de regresión lineal simple y múltiple. La regresión lineal múltiple se deriva de una regresión lineal simple. De esta, su ecuación general es:

$$Y = \beta_0 + \beta_1 X + e \tag{2.1}$$

Donde:

- Y: Variable dependiente.
- X: Variable independiente.
- β_0, β_1: Parámetros del modelo.
- e: Residuo o error de estimación.

Así, la regresión lineal múltiple se fundamenta en conseguir una relación lineal entre un conjunto de n variables regresoras con una variable dependiente. Este modelo sigue la siguiente ecuación:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_n X_n + e \tag{2.2}$$

Donde:

- Y es la variable respuesta.
- $X_1, X_2, X_3, ..., X_n$ son los regresores.
- $\beta_0, \beta_1, ..., \beta_n$ son los parámetros del modelo
- e_i representa el error.

Ahora, ¿Qué sucede si queremos usar la regresión múltiple para explicar un evento cualitativo? En este proyecto, el evento que se explica es un resultado binario, donde, la variable dependiente Y toma solo dos valores: 0 y 1. Entonces, es verdad que $P(Y = 1|X) = E(Y|X)$: la probabilidad de "éxito", es decir, la probabilidad de qué $Y = 1$ es igual al valor esperado de Y. Así, se obtiene la siguiente ecuación:

$$P(Y = 1|X) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_n X_n \tag{2.3}$$

Que manifiesta que, la probabilidad de éxito $P(Y = 1|X)$, es una función lineal de X_k, con $k = 1, ..., n$. Así, la ecuación (2.3) es un ejemplo de modelo de respuesta binaria, y $P(Y = 1|X)$ también se llama probabilidad de respuesta. $P(Y = 0|X) =$

$1 - P(Y = 1|X)$ también es una función lineal de X_k porque las probabilidades deben sumar 1. Sigue una distribución Bernoulli, diferenciándose de los modelos de regresión probit que sigue una distribución acumulada de la normal estandarizada y de los modelos de regresión logit, que siguen una distribución acumulada logística.

Al modelo de regresión lineal múltiple con variable dependiente binaria se denomina modelo de probabilidad lineal (LPM), dado que la probabilidad de respuesta es lineal en los parámetros β_k

Cabe mencionar que, las condiciones para la regresión lineal múltiple requieren de las mismas condiciones que los modelos lineales simples como la multicolinealidad, variabilidad constante de los residuos, no autocorrelación, etc. Más algunas complementarias.

Regresión Logística para Clasificación

La Regresión Logística es una técnica de Aprendizaje Automático para problemas de clasificación. Utiliza una función logística para modelar la variable respuesta. Esta variable es de naturaleza dicotómica. Como consecuencia, esta técnica se emplea al tratar con datos binarios.

La Figura 2.3 representa lo que hace la regresión logística.

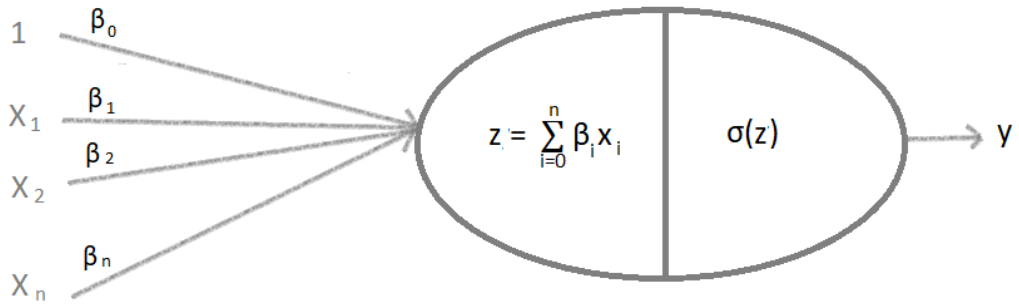

Figura 2.3: Esquema de Regresión Logística. **Fuente:** Elaboración propia.

De manera matemática se lo puede formular de la siguiente forma:

$$y = \sigma(z) = \sigma(\sum_{i=0}^{n} \beta_i X_i) = \sigma(\sum_{i=0}^{n} (\beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_n X_n)) \tag{2.4}$$

Es decir, tiene dos partes:

1. Una combinación lineal.

2. Aplicación de la función logística también llamada función sigmoidea.

Esta función logística se denomina precisamente a la forma de la propia función de distribución de probabilidad que se parece a una S, la cual presenta un crecimiento exponencial. A partir de esta función podemos interpretar sus resultados como probabilidades y está acotada entre 0 y 1. De forma matemática, la función sigmoide se expresa de la siguiente manera:

$$\sigma(z) = \frac{1}{1 + e^{(-z)}} \tag{2.5}$$

La Figura 2.4 representa la función logística o sigmoidea.

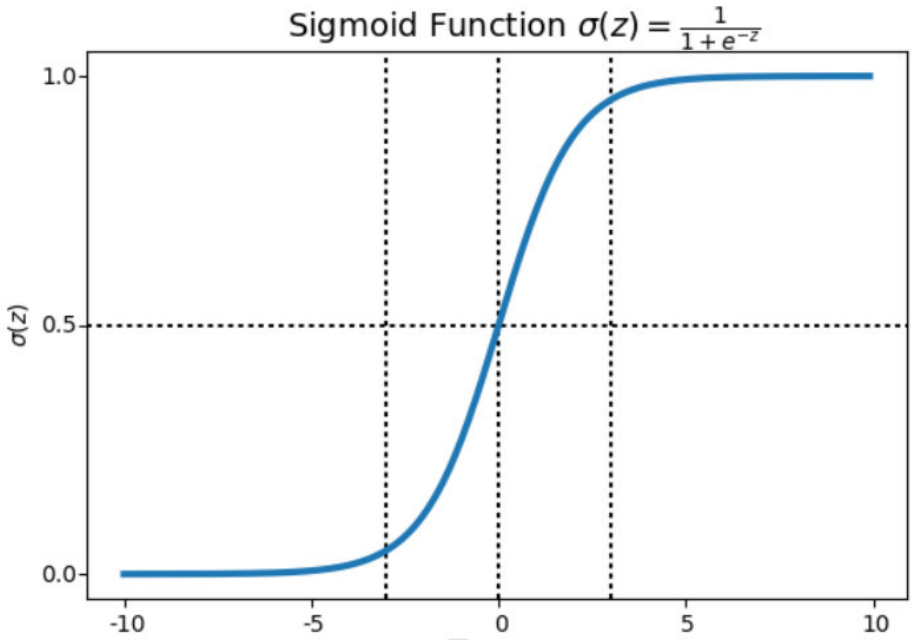

Figura 2.4: Gráfico de la función logística o sigmoidea

Como se ha expresado anteriormente, la parte de *regresión* en regresión logística da la impresión de que se trata de un método de regresión. No obstante, su nombre proviene de la aplicación de la función logística luego de la combinación lineal.

Como instancia final, es fundamental eliminar las características que muestran una gran multicolinealidad entre sí en la regresión logística. Por ello, una selección de las características es clave previo al entrenamiento del modelo [36].

Árboles de Decisión para Clasificación

Un árbol de decisión es beneficioso para asimilar la estructura de un conjunto de datos. Estos árboles son un tipo de algoritmo procedente de Aprendizaje Automático Supervisado y es uno de los más utilizados en la industria. Además de ser un algoritmo que se utiliza para tratar problemas de clasificación también se puede emplear para problemas de regresión. Adicionalmente, los árboles de decisión son conocidos como CART (Classification and Regression Trees) y su objetivo es dividir el espacio de variables regresoras o independientes en extensiones distintas. Esta división en conjuntos uniformes se lo realiza de acuerdo con ciertos parámetros y se basan en una variable de entrada considerada significativa [40].

Una de las ventajas más importante de estos árboles para este proyecto es que, beneficia la exploración de datos para identificar la importancia de variables a partir de cientos de ellas. Otras de las ventajas es la facilidad de interpretación, el poco requerimiento de preparación de datos, posibilidades de validar el modelo mediante pruebas estadísticas, entre otros. Por otro lado, una de las desventajas más notoria es el uso de un árbol de decisión super complejo dado que es probable que caiga en un escenario de sobreajuste [100].

En tal sentido, ahora es importante mencionar cómo funciona un árbol de decisión. El criterio de un árbol para ramificarse es diferente tanto para árboles de clasificación como de regresión. Recordemos que, en este proyecto se utiliza árboles de clasificación con variable dependiente categórica a diferencia de un árbol de regresión que utilizaría una variable dependiente continua.

Mencionamos primero sus elementos. Cada árbol consta de nodos, que es el punto dónde el árbol se divide, sus ramas unen los nodos, la raíz es el nodo donde se realiza la primera división y las hojas son nodos terminales que ayudan a predecir el resultado del árbol. Todo esto sigue un enfoque de arriba hacia abajo, es decir, el nodo raíz estará siempre en la parte superior de la estructura, en tanto que los resultados se desplazan hacia abajo.

Al crear un árbol, lo principal es la selección del mejor atributo del conjunto de datos para que sea este el nodo raíz y para los subnodos. La distinción de los mejores atributos se lo realiza mediante algunas técnicas comunes como: Ganancia de Información y Entropía, Índice de Gini, Chi Cuadrado y Reducción en la Varianza [31]. Mencionamos a continuación las más usadas en clasificación.

- **Ganancia de información y entropía**: Para comenzar, la entropía se entiende

como la medida de incertidumbre de un conjunto de datos, aquella que controla cómo un árbol de decisión decide dividir los datos y su valor expone el grado de aleatoriedad de un nodo en particular. Se prefiere una entropía baja al construir un árbol de decisión dado que cuanto mayor sea este, mayor será la aleatoriedad en el conjunto de datos. La expresión para su cálculo tiene la siguiente forma:

Sea la distribución de probabilidad de X

$$P(X = X_i) = p_i, i = 1, 2, ..., n \tag{2.6}$$

Entonces, se define a la entropía de X como:

$$\text{Entropía}(S) = \sum_{i=1}^{n} p_i \log \frac{1}{p_i} = -\sum_{i=1}^{n} p_i \log p_i \tag{2.7}$$

Aquí, p_i es la probabilidad de éxito respectivamente en el nodo y S el conjunto de ejemplo.

Finalmente, la ganancia de información es la medida de los cambios en el valor de la entropía después de la fragmentación del conjunto de datos. Según su valor se realiza la división del nodo y la construcción del árbol de decisión. Generalmente la ganancia de información en los árboles de decisión se describen mediante la siguiente ecuación:

$$\text{Ganancia}(S, A) = \text{Entropía}(S) - \sum_{v \in Valores(A)} \frac{|S_v|}{|S|} \text{Entropía}(S_v) \tag{2.8}$$

Donde;

- A: un atributo o variable
- Valores(A): valor posible del atributo A.
- S_v: subconjunto de S para el cual el atributo A tiene un valor v.

- **Indice Gini:** Medida de impureza. La impureza se refiere a cómo están mezcladas las clases en cada nodo.

  Para su cálculo se utiliza la siguiente ecuación:

$$I_G = 1 \sum_{i=1}^{n} p_i^2 \tag{2.9}$$

Con un valor mayor de índice Gini, mayor será su homogeneidad. CART usa el método Gini para la división binaria

Bosques Aleatorios

Un Bosque aleatorio es uno de los métodos de Aprendizaje Automático Supervisado más utilizado dado su flexibilidad y su fácil uso. Se utiliza tanto para la regresión como la clasificación. Un Bosque Aleatorio o también llamado Random Forest, está formado por un conjunto de árboles de decisión combinados con *bagging*. La idea general de utilizar el método baggin es que se ajustan los árboles de decisión a partir de B muestras *bootstrap*, las cuales son muestras extraídas aleatoriamente con reemplazo de los datos de entrenamiento. Estas muestras bootstrap que se generan al hacer bagging insertan un elemento aleatorio provocando que todos los árboles sean distintos, sin embargo, no siempre son los suficientemente distintos. Así, el bosque tendrá árboles con estructuras muy semejantes, singularmente en la parte alta. Esta peculiaridad se describe como correlación entre árboles. En resumidas cuentas, cada vez que un árbol se divida, en lugar de considerar todo el conjunto de variables regresoras, solo se tomarán en cuenta un subconjunto aleatorio de este, reduciendo la correlación existente entre los árboles.

La cuantía de variables regresoras considerados para la división de árboles elatorios es distinta para problemas de regresión como de clasificación. De un conjunto de p variables regresoras, se escoge el hiperparámetro de los bosques aleatorios m mediante:

- Clasificación: $m = \sqrt{p}$
- Regresión: $m = \frac{p}{3}$

En pocas palabras, un bosque aleatorio crea múltiples árboles de decisión a partir de bagging. Además, se evita la dependencia con la selección al azar en cada nodo de los posibles predictores con $m = \sqrt{p}$ en este proyecto, dado que se trabaja con bosques aleatorios para clasificación.

Entre sus ventajas está que se considera un método robusto y preciso debido al número de árboles de decisión que intervienen en el proceso, igualmente no sufre el problema de sobreajuste. Por otro lado, una de sus desventajas es la dificultad de interpretación comparado con un árbol de decisión, incluyendo que computacio-

nalmente no es tan rápido porque demora en la generación de predicciones dado la cantidad de árboles de decisión que maneja.

2.3. Técnicas de selección de variables y reducción de dimensiones

De manera general, cada vez que se desea reducir la dimensionalidad de los datos, se encuentra con métodos convencionales utilizados en la industria como la Descomposición de Valores, Análisis de Componentes Principales, Regresión Paso a Paso, Regresión de todos los Subconjuntos, Eliminación de características hacia atrás, entre otros. Pese a que estos sostienen diversas ventajas, este proyecto nos lleva a preguntarnos por qué existe la necesidad de utilizar otros métodos de selección de características. El hecho reside en que, muchos de estos métodos tienen defectos como la inestabilidad, realizan predicciones imprecisas, no ejecutan una selección correcta de predictores, correlación alta, contienen predictores que superan el número de observaciones que no se realizan ajuste y la gran mayoría no trabaja bien para un volumen grande de datos [21].

La Figura 2.3 nos ayuda a identificar la clasificación de las técnicas de selección de variables dentro del Aprendizaje Automático para su mejor entendimiento.

Figura 2.5: Clasificación de algoritmos del Aprendizaje Automático (Dobilas, 2020).

A continuación, se exponen los métodos de selección de variables no convencionales propuestos en este documento.

2.3.1. Splines de Regresión Adaptativa Multivariante (MARS)

La regresión spline adaptativa multivariante, en inglés multivariate adaptative regression splines [MARS; Friedman (1991)], es un método adaptativo de regresión que puede verse como una generalización tanto de los árboles de decisión CART como de la regresión lineal por pasos (stepwise linear regression) [10].

MARS pertenece a la categoría de aprendizaje automático supervisado que utiliza datos etiquetados para modelar la relación entre las entradas de datos y las salidas.

El modelo MARS es un spline multivariante lineal de la forma:

$$y_t = f(x_t) = \beta_0 + \sum_{i=1}^{k} \beta_i h(x_{it})$$

El modelo es una suma ponderada de funciones bases, donde y_t es la variable dependiente en el instante t y β_i son los parámetros del modelo para las respectivas variables x_{it}, que van de $i = 1, ..., k$. El valor β_0 representa el intercepto, las funciones bases $h(x_{it})$ son funciones que dependen de las respectivas variables x_{it}, en donde cada $h(x_{it})$ es una transformación que se construye de forma adaptativa empleando funciones bisagras [46].

Funciones de bisagra

Las funciones de bisagra toman la siguiente forma:

$$h(x) = \begin{cases} x, & \text{si x es verdadero,} \\ 0, & \text{Caso contrario} \end{cases}$$

Una función de bisagra es cero para parte de su rango, por lo que se puede usar para dividir los datos en regiones separadas, cada una de las cuales se puede tratar de forma independiente. MARS selecciona automáticamente las variables y los valores de esas variables para los nudos (valores observados de los predictores) de las funciones de bisagra [10]. A estas divisiones de datos las llamaremos variables transformadas y en función de este número de funciones bases resultantes se indicará la complejidad del modelo.

El resultado de combinar funciones de bisagra lineales lo podemos ver de ejemplo en la Figura 2.4, donde los puntos negros son las observaciones y la línea roja es una predicción dada por el modelo MARS:

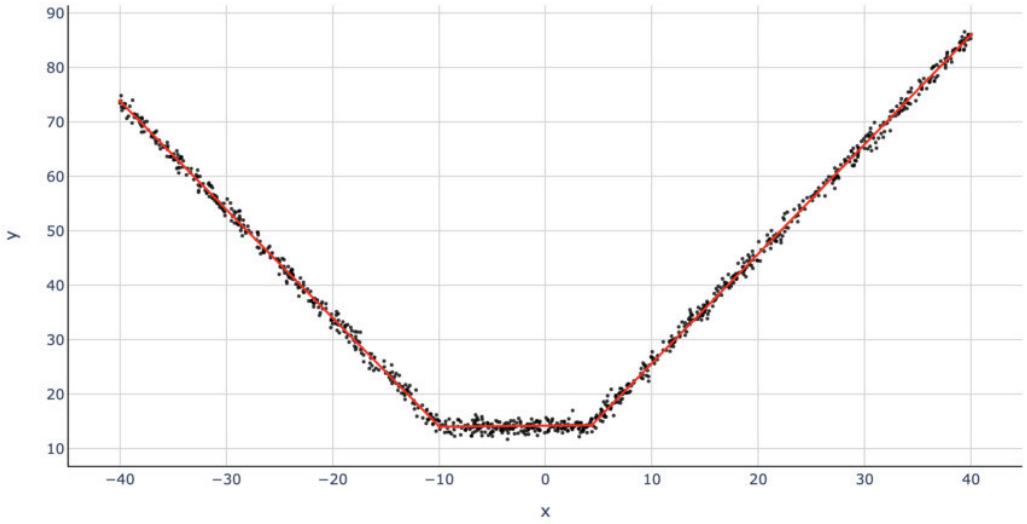

Figura 2.6: MARS para predecir valores de y dados x (Dobilas, 2020).

Proceso de construcción del modelo

Como se mencionaba, MARS produce puntos de corte para las distintas variables. Los puntos son identificados a través de las funciones de base, las que indican el inicio y el término de una región. En cada región en que se va fraccionando el espacio se ajusta una función base de una variable, la cua es lineal. El modelo final se establece como una combinación de las funciones base generadas.

Para determinar los puntos de corte se usa el algoritmo forward y backward stepwise por etapas. Con el mismo criterio funciona los árboles de partición recursivos, el cual crea un árbol de decisiones que clasifica miembros de una población fraccionandolos en subpoblaciones en función de variables independientes dicotómicas [7].

- Forward: El proceso de fabricación del modelo es un proceso repetitivo forward que empieza con el modelo

$$\hat{y}(x_{it}) = \hat{\beta}_0$$

donde $\hat{\beta}_0$ es la media de todas las respuestas, para a continuación considerar todos los puntos de corte posibles x_{ij} con $i = 1, 2, ..., n$, $t = 1, 2, ..., p$, es decir, todas las observaciones de todas las variables explicativas. Para cada punto de corte x_{it} se consideran dos bases:

$$h_1(x) = h(x_j - x_{ti})$$

$$h_2(x) = h(x_{ti} - x_t)$$

con lo que se construye el nuevo modelo

$$\hat{y}(x_{it}) = \hat{\beta}_0 + \hat{\beta}_1 h_1(x_{it}) + \hat{\beta}_2 h_2(x_{it})$$

La estimación de los parámetros $\beta_0, \beta_1, \beta_2$ se ejecuta de la forma estándar en regresión lineal. De esta forma se componen varios modelos alternativos y entre ellos se escoge aquel que tenga un menor error de entrenamiento. En la siguiente iteración se conservan $h_1(x)$ y $h_2(x)$ y se añade una pareja de términos nuevos siguiendo el mismo procedimiento. El proceso continua así sucesivamente, agregando en cada vez dos nuevos términos.

El proceso iterativo se detiene cuando se logra un modelo de tamaño k, que se alcanza después de incorporar $\frac{k}{2}$ cortes. Este modelo depende de $k+1$ parámetros β_i con $i = 0, 1, ..., k$.

La finalidad es lograr un modelo lo suficientemente bueno, para a continuación conducirse a su poda en un proceso de supresión de variables hacia atrás (backward).

- Backward: Este proceso consiste en la supresión de las variables de una en una. En cada paso de poda se excluye el término que produce el menor incremento en error o las que no agregan rendimiento material al modelo. Así, para cada tamaño $\lambda = 0, 1, ..., k$ se logra el mejor modelo estimado. Para la elección del valor del hiperparámetro λ óptimo se utiliza los procedimientos usuales de tipo validación cruzada y mucho más rápido si se usa validación cruzada generalizada (GCV) por medio de la siguiente fórmula:

$$GCV(\lambda) = \frac{RSS}{(\frac{1-k(\lambda)}{n})^2} \tag{2.10}$$

donde, $k(\lambda)$ representa el número de parámetros efectivos del modelo dependiente del número de términos más el número de puntos de corte utilizados penalizado por un factor

El algoritmo para cuando la aproximación construida incluye un número máximo de funciones fijadas por el investigador [46].

Ventajas

A pesar de que el procedimiento de construcción del modelo realiza búsquedas exhaustivas lo que conllevaría a consecuencias como un esfuerzo computacional bastante grade, en la práctica se realiza de forma moderadamente rápida.

Sus principales ventajas de MARS, basados en [10] y [7] son las siguientes:

- MARS es ideal para problemas con alta dimensionalidad.
  A manera de ejemplo se puede construir un modelo MARS a partir de una base de 100 predictores y 1000 individuos. Dicho modelo se construirá en aproximadamente un minuto en una máquina de 1 GHz en el caso de que nuestro grado máximo de iteración de los términos MARS lo limitemos a uno (es decir, sólo términos aditivos). Por otro lado, un modelo de grado dos con los mismos datos en la misma máquina de 1GHz tardaría aproximadamente 12 minutos.
- No es necesario un preprocesado de los datos, tampoco es necesario una transformación de ellas. Las funciones bisagra fraccionan automáticamente los datos de entrada, por lo que se contiene el efecto de los valores atípicos.
- No afecta a la construcción del modelo la presencia de predictores con correlaciones altas, sin embargo, podría dificultar su interpretación.
- Realiza una selección automática de las variables predictoras, lo que significa que incluye variables importantes en el modelo y excluye las que no lo son. Sin embargo, puede existir cierta arbitrariedad en la selección, en particular cuando hay predictores correlacionados, y esto puede afectar la interpretabilidad.
- Se puede trabajar con variables predictoras tanto numéricas como cualitativas.
- Se puede llevar a cabo una cuantificación de la importancia de las variables.
- Los modelos MARS son fáciles de entender e interpretar.
- Estos modelos pueden hacer predicciones rápidamente.
- Robustos a valores atípicos.

Desventajas

- Susceptible de sobreajuste

- Por lo general, son más lentos de entrenar. Ya que el algoritmo escanea cada valor de cada regresor en busca de puntos de corte potenciales, la productividad computacional se ve afectada.

- La correlación dificulta la interpretación del modelo a pesar de que no impide el rendimiento del modelo. El algoritmo escoge la primera característica que encuentre al escanear, entre dos que sean casi perfectas. Posterior a esto, es muy probable que esa característica no se incluya dado que no agrega poder explicativo gracias a su arbitrariedad al ser escogida [17] .

Cabe mencionar que, a pesar de que formalmente MARS se deriva de una regresión. Éste se utilizó como método de selección de variables dado que resulta ser una herramienta fuerte.

2.3.2. Boruta

Boruta es un algoritmo de selección de características utilizando bosques aleatorios como algoritmo subyacente.

Dato aparte, su nombre tan extraño deriva de un demonio en mitología eslava que habitaba en bosques de pinos [14].

Proceso de construcción del modelo

En base a [7].

- Primero, se crean copias duplicadas de todas las variables regresoras a las cuales denominaremos características de sombra.

- Luego, se combinan las variables originales con su copia formando un nuevo conjunto de datos. Se ejecuta un clasificador de Bosque Aleatorio en esta combinación y aplica una medida de importancia de características. Al hacerlo, se asegura tener la importancia para cada una de las características de nuestro conjunto de datos. Un puntaje alto significará mayor importancia.

  El valor predeterminado para la medida de importancia de características es la media de la pérdida de precisión, que básicamente este indicador permite ordenar las variables según su capacidad discriminatoria.

- A continuación, el algoritmo verifica si una característica real tiene una mayor importancia que la mejor de sus características de sombra. Así, para empezar, se calcula la puntuación Z de la siguiente forma:

$$Z = \frac{x - \bar{x}}{S} \tag{2.11}$$

Donde:

- $\bar{x}$: Es la media de la población, en este caso es la media de la pérdida de precisión.
- S : Desviación estándar de la pérdida de precisión.

Si la característica tiene una puntuación Z más alta que la puntuación Z máxima de sus características sombra, entonces se registra un vector de variables importantes.

- En cada iteración, el algoritmo compara las puntuaciones Z de las copias mexcladas de las funciones y las funciones originales con el objetivo de ver si las últimas funcionaron mejor que las primeras. Si es así, el algoritmo marcará la función como importante.

- Por último, Boruta se detiene luego de un número determinado de iteraciones, cuando todas las características se confirman o rechazan o alcanzan un límite específico de ejecuciones aleatorias del bosque.

Ventajas

- El uso de Boruta en lugar de otros algoritmos de selección de características tradicionales se fundamente en su método de selección, el cuál es completamente relevante, en el que se captura todas las variables explicativas que, en ciertas particularidades, son significativas para la variable resultado. Mientras que, la mayoría de algoritmos de selección tradicionales siguen un método mínimo óptimo que se basa en un subconjunto pequeño de variables explicativas que produce un error mínimo en un clasificador elegido.

- Otra ventaja de Boruta es el hecho de que encuentra todas las variables explicativas que son débilmente o fuertemente relevantes para la variable respuesta. Esto resulta conveniente para aplicaciones biomédicas en las que se podría

intersar en determinar genes humanos conectados de alguna manera a una condición médica particular [15].

Desventajas

- Boruta es aplicable en casi cualquier conjunto de datos. Sin embargo, el uso de árboles complejos, o incluso de menor complejidad, su tiempo de ejecución está medido en horas.
- El tiempo de ejecución del método hace que sea difícil de ajustar, dado que cada ajuste de parámetro requiere un tiempo de CPU adicional sumamente alto [8].
- Boruta se puede usar en cualquier problema de clasificación o regresión para generar un subconjunto de características relevantes.
- Boruta no funciona con un conjunto de datos con valores faltantes.
- Boruta considera variables altamente correlacionadas, lo que implica que no trata colinealidad al seleccionar las variables importantes. Esto se debe a la forma en que funciona el algoritmo. Sin embargo, es importante manejar la colinealidad luego de obtener las variables importantes.
- Es importante asegurarse imputar valores faltantes o en blanco antes de utilizar el método. Caso contario, éste lanzará errores de manera descarada.

2.3.3. Operador de selección y contracción mínima absoluta (Lasso)

Para evitar el impacto de problemas como la incorporación de predictores correlacionados, selección de predictores inútiles, inestabilidad y algunos otros, existen también algunas estrategias mucho más modernas, como los métodos de regularización. De este modo, Lasso pertenece al grupo de los métodos más empleados en regularización, junto con Ridge y Elastic Net [2].

Una regularización se fundamenta en ajustar un modelo incorporando todas las variables independientes, pero aplicando una penalización que fuerce a que los estimadores de los coeficientes de regresión tiendan a cero. Con esto se elude reducir la varianza, minorar la influencia de los predictores menos relevantes en el modelo y

minimiza el efecto de la correlación entre predictores. Con la aplicación del método de regularización se consigue modelos con un alto poder predictivo [2].

Ahora bien, de la misma forma, Lasso (Least Absolute Shrinkage and Selection Operator), introducido por Tibshirani (1996), es un método que acopla un modelo de regresión con un procedimiento de contracción de algunos parámetros hacia cero y selección de variables, infringiendo una penalización o una restricción sobre los coeficientes de regresión [35].

Proceso de construcción del modelo

La regresión de Lasso es similar a la de Ridge. Explicaremos a breves rasgos como funciona la regresión Rigde.

Ridge penaliza la suma de los coeficientes elevados al cuadrado.

$$||\beta||_2^2 = \sum_{j=1}^{p} \beta_j^2 \tag{2.12}$$

La ecuación (2.1) se le conoce como l_2, una penalización que tiene el impacto de reducir de forma proporcional el valor de todos los coeficientes, pero sin que estos lleguen a 0 en el modelo. Para controlar el grado de penalización se utiliza el parámetro λ que se muestra en la ecuación (2.14). Esta penalización será nula cuando $\lambda = 0$ y su resultado es semejante al de un modelo lineal por mínimos cuadrados ordinarios. Menor es el valor de los predictores y mayor es la penalización cuando λ aumenta [2].

$$RSS_{ridge} = \sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij})^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \tag{2.13}$$

$$= sumaresiduoscuadrados + \lambda \sum_{j=1}^{p} \beta_j^2 \tag{2.14}$$

Ahora, de manera similar, Lasso penaliza la suma del valor absoluto de los coeficientes de regresión.

$$||\beta||_1 = \sum_{j=1}^{p} \beta_j \tag{2.15}$$

A esta penalización se le conoce como l_1 y tiene el impacto de forzar a que los coeficientes de los predictores se inclinen a 0. Lasso consigue excluir las variables independientes menos relevantes gracias a que un predictor con coeficientes de regresión 0 no predomina en el modelo. De la misma forma que en ridge, el grado de penalización está moderado por el parámetro λ. El resultado es semejante al de un modelo lineal por mínimos cuadrados ordinarios cuando $\lambda = 0$ y mayor será la penalización y más predictores quedarán excluidos a medida que aumente λ.

$$RSS_{ridge} = \sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij})^2 + \lambda \sum_{j=1}^{p}|\beta_j| \tag{2.16}$$

$$= sumaresiduoscuadrados + \lambda \sum_{j=1}^{p}|\beta_j| \tag{2.17}$$

El uso de Lasso en lugar de Ridge en este proyecto se da gracias a la ventaja en escenarios donde no todas las variables independientes son importantes en el modelo y se desea que se excluyan las menos influyentes, esto se da porque Lasso consigue que algunos coeficientes sean exactamente cero por consiguiente realiza la selección de predictores [2].

Ventajas

- Como cualquier método de regularización, consigue evitar el sobreajuste. Se aplica aun cuando el número de variables regresoras es mayor que el número de datos.
- Rápido en términos de inferencia y ajuste.
- Selecciona variables reduciendo el coeficiente hacia 0.

Desventajas

- Cuando se encuentren características correlacionadas, Lasso selecciona una de ellas de manera aleatoria. Esta selección resulta ser de naturaleza arbitraria.
- Las variables seleccionadas estarán muy sesgadas.

2.4. Técnicas de Validación

2.4.1. Multicolinealidad

El problema de la multicolinealidad hace referencia a la relación de dependencia lineal que existe entre dos o más variables regresoras en un modelo de regresión múltiple o lo que es lo mismo, la multicolinealidad es la alta correlación entre variables regresoras. Si el nivel de correlación entre las variables es lo bastante alto, podría causar problemas al interpretar los resultados y ajustar el modelo. Además, si la multicolinealidad es severa puede incrementar la varianza de los coeficientes de la regresión, lo que los convierte en inestables.

¿Cómo se mide la existencia de multicolinealidad?

El economista Ragnar Frisch incrustó este término y su explicación matemática es la siguiente:

Consideramos un modelo de regresión con n-variables explicativas $X_1, X_2, ..., X_n$. Se dice que, existe relación lineal exacta si se cumple la siguiente cualidad:

$$\beta_1 X_1 + \beta_2 x_2 + ... + \beta_n X_n = 0 \tag{2.18}$$

Donde, $\beta_1, \beta_2, ..., \beta_n$ son constantes y diferentes de 0.

No obstante, el término multicolinealidad se utiliza también para el caso dónde la colinealidad no es perfecta, es decir, las variables X están interrelacionadas, pero no de manera perfecta. Así,

$$\beta_1 X_1 + \beta_2 x_2 + ... + \beta_n X_n + \lambda_i \tag{2.19}$$

Donde λ_i es un término estocástico.

Ahora, para detectar la multicolinealidad, también existen varios métodos:

1. Examinar la estructura de correlación de las variables explicativas. Un valor alto indica problema de multicolinealidad.

2. Determinar el R-cuadrado de un modelo en el análisis de regresión, si este es alto y tiene pocas razones t significativas, esto indica multicolinealidad.

3. También se puede indagar los factores de inflación de la varianza (FIV). Los FIV miden la correlación entre variables independientes en modelos de regre-

sión, es decir, qué tanto crece la varianza de un coeficiente de regresión. Si los FIV son 1, significa que no hay multicolinealidad, de otra forma, si los FIV son mayores a 1 significa que los regresores están correlacionados.

Formas de corregir la multicolinealidad

Varios factores afectan la multicolinealidad como; exceso de variables regresoras, métodos de recopilación de datos que no se pueden manejar, al crear nuevas variables predictoras por el investigador, datos insuficientes, variables ficticias, incluir variables idénticas, entre otros. Existen varias situaciones dónde no sea necesario lidiar con este problema de multicolinealidad. Sin embargo, cuando este sea grave o severa en sus datos existe diversos métodos que se pueden utilizar según su conveniencia para subsanar este problema. Enlistamos algunos de ellos en base a [101].

1. Combinar las variables independientes de manera lineal.
2. Realizar un análisis de regresión de mínimos cuadrados parciales o componentes principales. Estos métodos disminuyen el número de predictores a un conjunto mucho más pequeño donde sus componentes están no correlacionados.
3. Eliminar las variables que estén altamente correlacionadas.
4. Utilizar formas avanzadas de análisis que manejan la multicolinealidad como el método LASSO y la regresión Ridge.

2.4.2. Validación Cruzada (GCV)

La validación cruzada es un instrumento común para acoplar modelos de minería de datos. Luego de haber creado una estructura de minería de datos se utiliza validación cruzada a fin de validar el modelo [40]. La validación cruzada tiene la siguiente práctica:

- En un modelo de minería ayuda a validar su robustez.
- A partir de una sola declaración se evalúan múltiples modelos.
- Luego de la construcción de modelos, ayuda a identificar el mejor, apoyado en estadística.

La fórmula para el GCV es:

$$GCV = \frac{\text{RSS}}{(N(1 - \text{número de parámetros})/N)^2} \tag{2.20}$$

Donde RSS es la suma de cuadrados residuales y N es el número de observaciones, es decir el número de variables x.

2.4.3. Curva ROC - AUC

La curva ROC - AUC es una magnitud de rendimiento para los problemas de clasificación en varias configuraciones de umbral. Por un lado, ROC es una curva de probabilidad y AUC representa el grado o medida de separabilidad. Indica cuánto es capaz el modelo de discrepar entre clases. Por analogía, cuanto mayor sea el AUC, mejor será el modelo. El análisis ROC se basa en graficar la curva con la Sensibilidad (TPR) contra Especificidad (FPR), donde TPR está en el eje y y FPR está en el eje x como lo podemos ver en la siguiente gráfica.

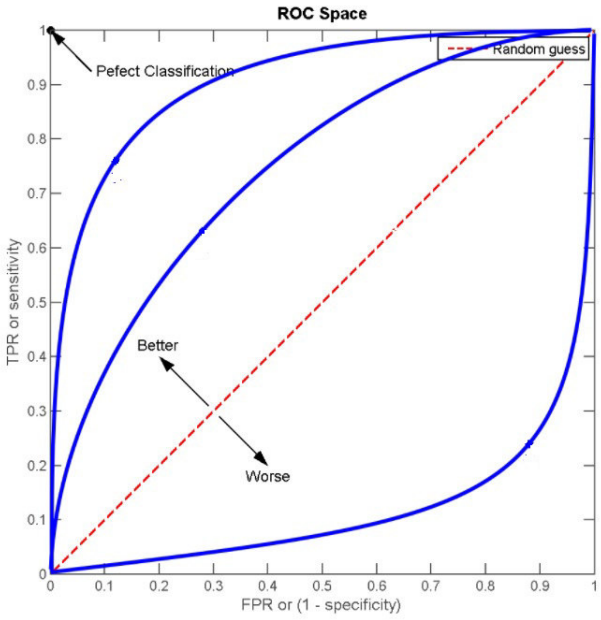

Figura 2.7: Curva Característica Operativa del Receptor (ROC) (Melillanca, 2018).

Por una parte, el término TPR también conocido como Tasa de Verdaderos Positivos (True Positive Rate), recuperación o sensibilidad, es la probabilidad de que la prueba dé como resultado positivo, lo que puntualmente se reduce a calcular la

probabilidad condicionada:

$$sensibilidad = \frac{TP}{TP + FN}$$

donde,
$TP =$ verdaderos positivos
$FN =$ falsos negativos

La especificidad o tasa de verdaderos negativos, es la probabilidad de que la prueba dé como resultado negativo, lo que se reduce a calcular:

$$especificidad = \frac{TN}{TN + FP}$$

donde,
$TN =$ verdaderos negativos
$FP =$ falsos positivos

Finalmente FPR o valor predictivo de una prueba positiva.

$$FPR = 1 - especificidad$$

En resumen, el área bajo la curva ROC (AUC) es una métrica que avalúa qué tan bien clasifica los resultados positivos y negativos un modelo de regresión logística. Así pues, un modelo óptimo tiene un valor AUC cerca de 1, lo que implica que tiene una medida buena de separabilidad. Al contrario, un mal modelo tiene un AUC cercano a 0, lo que implica que tiene la peor medida de separabilidad. Y cuando AUC es 0,5 denota que el modelo no tiene capacidad de división de clases [29].
De manera más representativa las afirmaciones anteriores se muestran en la siguiente gráfica.

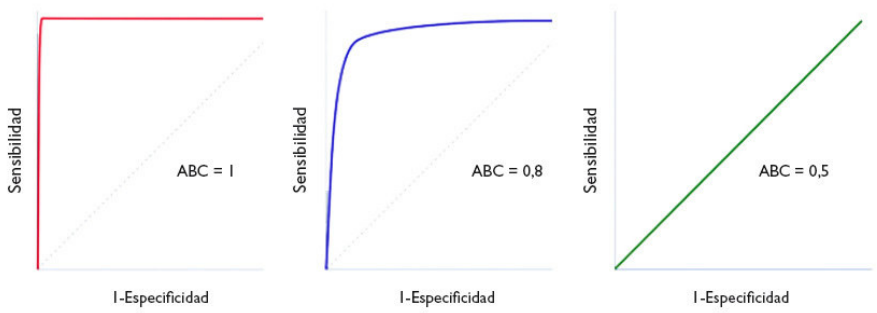

Figura 2.8: Comparación de los tres casos hipotéticos de las curvas ROC.

En definitiva y a modo de guía, para interpretar las curvas ROC se establecen los siguientes intervalos de valores AUC:

- $[0{,}5]$: Similar al lanzamiento de una moneda.
- $[0{,}5, 0{,}6)$: Test malo.
- $[0{,}6, 0{,}75)$: Test regular.
- $[0{,}75, 0{,}9)$: Test bueno.
- $[0{,}9, 0{,}97)$: Test muy bueno.
- $[0{,}97, 1)$: Test excelente.

2.4.4. Coeficiente Gini

El coeficiente de Gini es una magnitud de la desigualdad de una distribución. Se define como una relación que toma valores entre 0 y 1. En su numerador se encuentra el área entre la curva de Lorenz de la distribución y la línea uniforme de distribución y, por otro lado, en su denominador se encuentra el área bajo la línea de distribución uniforme.

Este coeficiente fue desarrollado por el estadístico italiano Corrado Gini y publicado en su artículo de 1912.

Gini puede utilizarse a menudo para medir la desigualdad de ingresos, no obstante, comúnmente se utiliza para medir el dominio discriminatorio de los sistemas de calificación en el riesgo de crédito.

Su cálculo se simplifica y se deriva del cálculo del valor AUC;

$$Gini = 2AUC$$

Capítulo 3

Implementación

En este capítulo se presenta una sección de procesamiento de datos a una base proporcionada previamente. A su vez, esta sección se incluye una descripción, limpieza y transformación de datos. Acto seguido se evalúa los métodos de selección de variables con su aplicación.

Con las variables recopiladas luego de la aplicación de los métodos de selección, se procede a una validación con su respectiva comparación respecto a métodos comunes utilizados en la industria.

3.1. Preprocesamiento de los datos

Una base de datos actual, suele estar sujeta a datos ruidosos, inconsistentes y ausentes debido a su gran tamaño y a su origen de múltiples fuentes usualmente heterogéneas. De esta manera, los datos de baja calidad presidirán a resultados de baja calidad. Al procesarlos y analizarlos, mejoramos su calidad y eficiencia.

Se presenta a continuación un análisis descriptivo de la base de datos con la que se cuenta para esta investigación, proporcionada por una consultoría privada. Dónde además, se incluyen técnicas de procesamiento de datos como una limpieza, integración, reducción y transformación.

3.1.1. Descripción de la base de datos

La base de datos contiene información correspondiente a 23489 registros de individuos con acceso a crédito de consumo y 2129 características correspondientes a variables regresoras, recopilados en el periodo 2017-2019. Varias de estas variables

tienen mucha o poca importancia, sin embargo, se describen las variables, a priori, más relevantes según criterio de autor.

a. **Variable Dependiente:**

   Para el presente estudio, la variable dependiente es la "Var_ Objetivo"(Buen o mal pagador).

Variable	**Descripción**	**Tipo**	**Escala de Medición**	**Criterios de Medición**	**Porcentaje de buen y mal pagador en la base**
Var_Objetivo	Buen o mal pagador	Cualitativa	Nominal	Si: 1 No: 0	Si: 70.7 % No: 29.3 %

Cuadro 3.1: Descripción de la variable dependiente o respuesta.

b. **Variables independientes**

   - *IdCliente:* Lista enumerada que permite identificar distintos clientes.
   - *UsoCupoPorcentaje:* Cupo usado de crédito del cliente, medido en porcentaje.
   - *Antiguedad:* Antigüedad del cliente consumiendo productos.
   - *Género:* Masculino / Femenino.

Mas adelante se describe con más detalle estas variables independientes.

3.1.2. Descripción Estadística de los Datos

El objetivo de esta sección, es describir las variables de manera estadística para identificar el comportamiento de la distribución de estas. Se aplica ciertas funciones para procesar datos y reducirlos a valores indicativos, para luego interpretarlos sobre varios aspectos importantes de la distribución de diversas variables. A partir de estas funciones identificamos aspectos básicos de la distribución como: dispersión de los datos y centralidad.

Variables Categóricas o Cualitativas

Se cuenta con 15 variables cualitativas de las cuales algunas de ellas se presentan a continuación:

Variables Cualitativas	**Categorías**
Género	▪ Femenino ▪ Masculino
Banco Peor Calificación	▪ Pichincha ▪ Internacional ▪ Pacífico ▪ Guayaquil ▪ Bolivariano ▪ Diners ▪ Cooperativas ▪ etc.
Nacionalidad	▪ Ecuador ▪ Chile ▪ Colombia ▪ Cuba ▪ Nicaragua ▪ Venezuela
Estado Civil	▪ Casado ▪ Divorciado ▪ Soltero ▪ Union Libre ▪ Viudo
Ciudad	▪ Quito ▪ Guayaquil ▪ Cuenca ▪ Loja ▪ Manabí ▪ etc.
Central de Riesgos	▪ 1 (Si) ▪ 0 (No)

Cuadro 3.2: Descripción de variables cualitativas.

Variables Numéricas o Cuantitativas

Son 2114 variables cuantitativas

Variables Cuantitativas	Datos que toman
Edad	0 - 97 años
Cupo	0 - 13600 dólares
Saldo de Tarjeta de Crédito	0 - 232853.81 dólares
Score Aval	0 - 998 puntos de calificación
Días de Atraso en Pagos	1 - 720 días
Cartera Castigada	0 - 30865.28 dólares
Deuda Total	0 - 5.00E+05 dólares
Saldo Vencidos	0 - 24746.44 dólares
Saldo para Consumo máximo	0 - 135168.1 dólares

Cuadro 3.3: Descripción de variables cuantitativas.

3.1.3. Limpieza de datos

Depurar datos es una tarea fundamental para consolidar la calidad en datos que serán procesados.

Así como la selección de variables propuesta en este proyecto, la limpieza nos ayuda a evitar información errónea corrigiendo inconsistencias en la base. De la misma forma, podemos ahorrar notablemente costos de espacio en el almacenamiento en disco (la base de datos pesa 148,235 kiloByte (kB)), dado que se elimina información duplicada. Sin embargo, existe el método MARS de selección que variables que, como se explicó en el capítulo anterior no exige un preprocesamiento de datos, se lo realiza para obtener resultados mucho más exactos y tomar decisiones estratégicas correctas.

El detalle de la limpieza de datos se lo realizó en el siguiente orden:

- **Valores Perdidos:** El primer paso fue el análisis de valores perdidos. La base de datos tiene varias variables que no tienen valores registrados en sus columnas, como *comPeorCalificacionU3M_Conyuge*, que posee el 100 % de valores en

blanco en su columna, o como *comPeorCalificacionU12M* que posee un 80 % de valores faltantes.

El paso usual es rellenar estos valores faltantes mediante alguna medida de tendencia central del atributo. Sin embargo, para aquellas variables que poseen un porcentaje alto, mayor o igual al (80 %) de valores NAs se procedió a retirarlos. Este método de ignorar variables no es muy eficaz si la variable contiene pocos valores faltantes, pero a criterio de autor se procedió a retirarlos con un porcentaje considerablemente alto de NAs.
Así, se obtuvo un total de 27 variables que no contenían registros en su columna o que a su vez superaban el 80 % de valores sin registro o NAs.

Posterior al retiro de estas 27 variables, se continuó trabajando con las 2102 sobrantes. Entonces, para aquellas variables que quedaron y que aún tenían valores perdidos se conduce a rellenarlas mediante una medida de tendencia central. Concretamente para datos numéricos se remplazó con el valor medio de su columna y para datos categóricos con la moda.

- **Multicolinealidad** La redundancia también es cuestión crucial en el procesamiento de datos. Las redundancias de la base en este proyecto se detectaron con el análisis de correlación que, a su vez, resuelve el problema de multicolinealidad con la detección de la alta correlación entre las variables explicativas. Para los atributos numéricos se utilizó el coeficiente de correlación que permite acceder a la forma en que los valores de un atributo varían de los de otro y para datos nominales se utilizó la prueba chi-cuadrado.

  Acto seguido, con una división de datos en: numéricos 2087 y categóricos 15, se resuelve el problema de multicolinealidad. En el caso de los atributos numéricos, se evalua la correlación entre dos atributos, digamos A y B calculando su coeficiente de correlación. Si este valor es mayor a 0, entonces A y B estarán positivamente correlacionadas. Cuanto más alto es el valor, más fuerte será la correlación. En consecuencia, un valor más alto indicará que podemos eliminar A o B como variable redundante. En este proyecto se utiliza un valor igual o mayor a 0,7 (usualmente usado en la industria) como valor referencia para retirar las variables correlacionadas. Por lo tanto, si A y B están correlacionadas positivamente, entonces se elimina una de ellas y permanece aquella que mejor explica a la variable respuesta. En última instancia, resultaron 394 variables numéricas explicativas eliminadas conforme al proceso anterior. Por otra parte, para las 15 variables categóricas la estadística X^2 probó la hipótesis de

que todas las variables eran independientes, es decir, no hay correlación entre ellos.

Por otro lado, el análisis de algunas variables en el cálculo del coeficiente Gini, nos muestran que existen algunas variables predictoras repetidas, que afectan el modelo de regresión y se procedió a retirarlas (scoreAval, maximoDiasAtraso12M, Cupo, maximoDiasAtraso3M, maximoDiasAtraso6M).

En resumidas cuentas, la limpieza de datos finaliza con 1702 variables.

3.1.4. Transformación de Datos

Tras la limpieza de datos procedemos a consolidar algunas variables de la forma más apropiada mediante la realización de operaciones. Estas transformaciones ayudan a mejorar la precisión y la eficiencia de los algoritmos de minería, como el caso de las variables *antigedad* y $fechaPeorCalf$ que son transformaciones de tipo date, que tiene un formato común, a años, respecto al año actual 2019.
En el cuadro 3.4 y 3.5 se muestra un resumen de las primeras observaciones de la variable antigüedad y fechaPeorCalif Vs su transformación.

Antigüedad / Fecha	**Años de Antigüedad**
12/7/1995	24
1/6/2006	13
1/6/2006	13
1/7/2006	13
1/12/2013	6
23/3/2016	3
1/11/2003	16
1/4/2012	7
1/6/2010	9
1/6/2006	13

Cuadro 3.4: Transformación de la variable Antigüedad

fechaPeorCalf / Fecha	Años de Peor Calificación
30/11/2017	2
31/5/2018	1
28/2/2018	1
30/11/2016	3
31/1/2015	4
30/9/2018	1
31/8/2015	4
31/3/2018	1
31/8/2016	3
31/5/2016	3

Cuadro 3.5: Transformación de la variable fechaPeorCalf

3.2. Evaluación de los métodos de selección de variables propuestos a la base de datos

La base en cuestión posee aún 1702 variables que pueden contribuir o no al correcto desempeño de algún modelo planteado. Ciertas variables proporcionan información útil, y otras no. Por esta razón, es necesario realizar aún más, una busqueda minuciosa de variables que serán utilizadas posteriormente.

A partir de una base limpia, se particiona el conjunto de datos en entrenamiento y test, 75 % y 30 % respectivamente y utilizamos los siguientes métodos de selección de variables presentados anteriormente; Mars, Boruta y Lasso a correspondencia.

3.2.1. Aplicación de Splines de regresión adaptativa multivariante (MARS)

MARS realizará una selección automática de predictores y gracias a las funcionalidades de paquetes en Rstudio, el paquete earth nos permite incorporar variables no solo numéricas si no también cualitativas siguiendo los procedimientos estándar mencionados en el marco teórico. Cabe considerar, por otra parte que, se puede realizar una cuantificación de la importancia de las variables de forma similar a como se realiza árboles de clasificación y de regresión.

Función en el Software R para Splines de regresión adaptativa multivariada

(Earth)

Earth crea modelos de regresión utilizando técnicas mencionadas por Friedman (1991), y dado que está implementado en R, se debe instalar el paquete *earth*. A continuación, se detallan los parámetros utilizados en la función earth en el Cuadro 3.6.

Argumentos	Descripción
x	Matriz de datos que contiene las variables independientes
y	Vector que contiene la variable respuesta o, a su vez, contenga múltiples respuestas.
glm	Si la respuesta es binaria se considera usar el argumento glm. Sin embargo, este viene por defencto NULL (default).
degree	Máximo grado de iteración. El valor predeterminado es 1, lo que significa construir un modelo aditivo.
nk	Número máximo de términos creados para forward pass. Sin embargo, este se calcula semiautomáticamente a partir del número de predictores.

Cuadro 3.6: Argumentos de la función earth(). **Fuente:** RDocumentation. Paquete stats.

La construcción del modelo MARS realiza búsquedas exhaustivas, lo que conlleva a parecer un problema computacionalmente intratable. Para evitar este problema, se ha particionada la base de datos de manera que el tiempo computacional se reduzca. Así, 1702 variables fueron separadas en bases de 150 variables.

El paquete de R earth() nos muestra la selección de términos y selección de predictores para cada uno de las bases creadas. La tabla del Anexo B ilustra las 145 variables que se consideraron importantes, donde se incluyen tanto variables originales pero importantes, como las variables que son transformación de los predictores originales. Sin embargo, earth simplemente muestra los nombres de estas variables transformadas asumiendo la teoría MARS para crearlas. De modo que, a partir de la teoría spline multivariante lineal, se procede a construir las variables.

Recordemos que nuestro modelo spline multivariante lineal es de la forma;

$$m(x) = \beta_0 + \sum_{m=1}^{M} \beta_m h_m(x)$$

Es decir, un modelo lineal con transformaciones $h_m(x)$, creados de nuestros predictores originales. Estas bases $h_m(x)$ serán creadas empleando funciones de bisagra de la forma;

$$h(x) = \begin{cases} 1, & x > 0 \\ 0, & x <= 0 \end{cases}$$

Así, se prueba que se deben considerar 145 variables importantes repartidas entre variables ya existentes (125) y variables que son transformación de otras (20), las cuales se muestran en el Cuadro 3.7.

Variable Original	Descripción / Categoría	Variable Transformada
menorCalifTc	▪ O1, O2, O3 ▪ R ▪ T ▪ V1, V2 ▪ Y1, Y2	menorCalifTc R
mayorValorVencido	0 - 120918.41	mayorValorVencido _Conyuge
peorCalfActual	▪ O1, O2, O3 ▪ R ▪ T ▪ V1, V2 ▪ Y1, Y2	▪ peorCalfActual V2 ▪ peorCalfActual O3 ▪ peorCalfActual V1 ▪ peorCalfActual O2 ▪ peorCalfActual Y1 ▪ peorCalfActual R ▪ peorCalfActual T
bancoPeorCalf	▪ Amazonas ▪ Pacifico ▪ BanEcuador ▪ etc.	▪ bancoPeorCalf PACIFICO ▪ bancoPeorCalf COOP 11 DE JUNIO ▪ bancoPeorCalf BELCORP
MinTotalSaldo ActualTC9M	0 - 74379.39	MinTotalSaldoActualTC9M _Conyuge
AcrescimoTotalSaldo ActualTC9M	0 - 8	AcrescimoTotalSaldo ActualTC9M _Conyuge
DecrescimoTotalSaldo ActualTC6M	1 - 5	DecrescimoTotalSaldo ActualTC6M_Conyuge
MaxDiasAtraso Totales3M	0 - 720	MaxDiasAtrasoTotales3M _Conyuge
MaxDiasAtraso Totales6M	0 - 720	MaxDiasAtrasoTotales6M _Conyuge
RazonDiasAtraso Totales3M_6M	0 - 1	RazonDiasAtrasoTotales3M _6M _Conyuge
peorCalfHistorica	▪ O1, O2, O3 ▪ R ▪ T ▪ V1, V2 ▪ Y1, Y2	▪ peorCalfHistorica O3 ▪ peorCalfHistorica Y

Cuadro 3.7: Variables transformadas del MARS

3.2.2. Aplicación de Boruta

Boruta es una función del paquete 'Boruta' en el Software R de selección de características relevantes. Este usa por defecto el algoritmo wrapper de Bosques aleatorios (Random Forest), que realiza búsquedas de tipo Backwards. El método, al igual que la teoría, ejecuta un chequeo de arriba hacia abajo de características o variables significativas al compararlas con la importancia de las cualidades originales con la importancia alcanzable al azar. Al mismo tiempo elimina gradualmente características intrascendentes para estabilizar.

A continuación, se detalla la función boruta () del paquete Boruta y sus principales argumentos que tiene, en el cuadro 4.6

Argumentos	Descripción
X	Marco de datos de predictores.
y	Vector de respuesta.
maxRuns	Número máximo de ejecuciones. Puede examinar aumentar este parámetro si se dejan atributos provisionales. Su valor predeterminado es 100. Ayuda a optimizar tiempo de cómputo
doTrace	Hace referencia al nivel de verbosidad. Es decir, 0 significa que no hay seguimiento, 1 informa la decisión del atributo tan pronto cómo se borre y 2 significa todo de 1 más informar cada iteración de manera adicional. Su valor predeterminado es 0.
holdHistory	Historial completo de ejecuciones importantes. Este se almacena si se establece TRUE, que es su forma predeterminada. Además, proporciona un gráfico de la ejecución del clasificadorfrente a la importancia si, llamamos la función plotImpHistory.
pValor	Nivel de confianza. Usualmente se utiliza el predeterminado
obtenerImp	Función que permite obtener la importancia de un atributo. Tiene un valor predeterminado de getImpRfZ, el cual ejecuta un Random Forest y recopila puntajes Z de la medida de precisión de disminución de medios. Devuelve un vector numérico de un tamaño igual al número de columnas de su primer argumento, que contiene la medida de importancia de sus respectivos atributos.
fórmula	Fórmula que describe el modelo a analizar.

Cuadro 3.8: Argumentos de la función boruta(). **Fuente:** RDocumentation. Paquete stats.

Por defecto se usa un $p - valor$ del 0,01 que confirma cuándo una variable es estadísticamente importante o no.

Ahora, trazamos una gráfica que nos muestra una representación visual de dicha selección. Sin embargo, debido a la cantidad de variables se dificulta la observación de los nombres de variables.

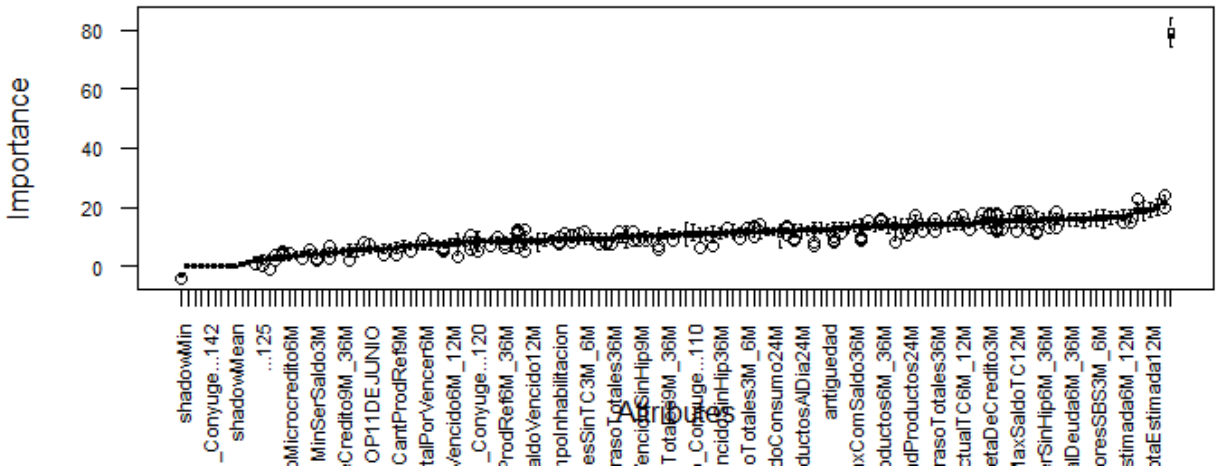

Figura 3.1: Variables regresoras importantes y no importantes según Boruta.

La Figura 4.1 se crea de forma determinada según la función de trazado de Boruta, agregando las variables importantes o atributos al eje x de manera vertical. Estos diagramas de caja corresponden a puntuaciones Z de atributos de sombra, rechazados, confirmados y tentativos.

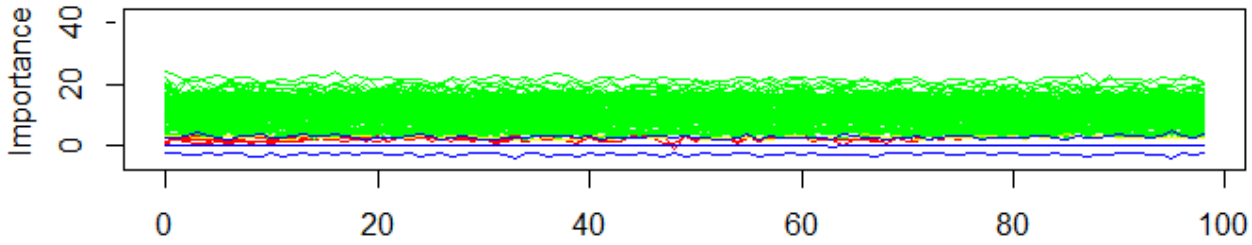

Figura 3.2: Historial de importancia de los atributos.

Esta última Figura es el gráfico de historial de importancia, donde se puede visualizar que los atributos verdes tienen mayor importancia, estos son aquellos que se consideraron atributos importantes confirmados. Siguen los atributos rojos que son aquellos que no son importantes. Luego están los atributos de sombra representados en azul y finalmente está un solo atributo amarillo considerado provisional, lo que significa que el algoritmo no logró llegar a una conclusión sobre su importancia.

En esta perspectiva, se corre el modelo y se verifica un marco de datos de resultados finales dónde evidentemente nos muestra 99 iteraciones, con 132 atributos

confirmados importantes y 12 atributos confirmados irrelevantes y 1 atributo considerado tentativo o provisional. Finalmente, para aquellos atributos provisionales se necesita tomar una decisión para clasificarlos como confirmados o rechazados comparando la puntuación Z media del mejor atributo de sombra con la puntuación Z mediana de los atributos. Sin embargo, dado que se tiene únicamente un atributo provisional se procedió a retirarlo.

La lista de atributos confirmados importantes son 132 y se agrupan en un nuevo dataframe para su próximo análisis.

3.2.3. Aplicación de Operador de selección y contracción mínima absoluta (Lasso)

Recordamos que Lasso realiza regularización al modelo para mejorar la exactitud e interpretabilidad.

El parámetro α nos indicará con qué tipo de regularización o modelo vamos a trabajar. Este es útil al evaluar cómo se aproxima a 0 los coeficientes a medida que se incrementa α así como el desarrollo del error de validación cruzada en función del α empleado.

- Ridge: $\alpha = 0$
- Lasso: $\alpha = 1$
- Elastic Net: $0 < \alpha < 1$

Para la regularización en rstudio, es usual la utilización de la librería *glmnet*. que trabaja tanto para modelos Ridge, Elastic Net y para lasso. La función a utilizar lleva el mismo nombre glmnet() y cv.glmnet().

A continuación se presenta los principales argumentos de la función glmnet().

glmnet

Argumentos	Descripción
x	Matriz de variables x.
y	vector de la variable a predecir.
pesos	Peso de observación; el valor predeterminado es 1 por observación.
type.measure	Función de error/pérdida que se va a utilizar en validación cruzada. El valor predeterminado es type.measure="deviance", que usa error cuadrático .
alpha	Tipo de modelo.
Pliega	Vector opcional de valores entre 1 y nfold que identifica en qué pliegue se encuentra cada observación.

Cuadro 3.9: Argumentos de la función glmnet(). **Fuente:** RDocumentation. Paquete stats.

Corremos nuestro modelo y realizamos un análisis en base a las siguientes gráficas. Observamos que en la Figura 4.3 de coeficientes en función de λ y en función de la norma de penalización respectivamente, algunos de ellos son cercanos a cero y otros exactamente cero.

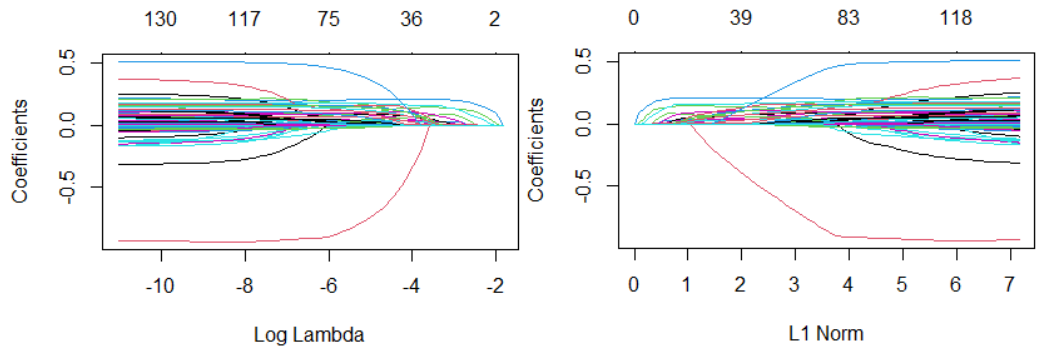

Figura 3.3: Coeficientes en función de lambda y de la norma de penalización.

Ahora, presentamos un gráfico para los valores de lambda en ggplot.

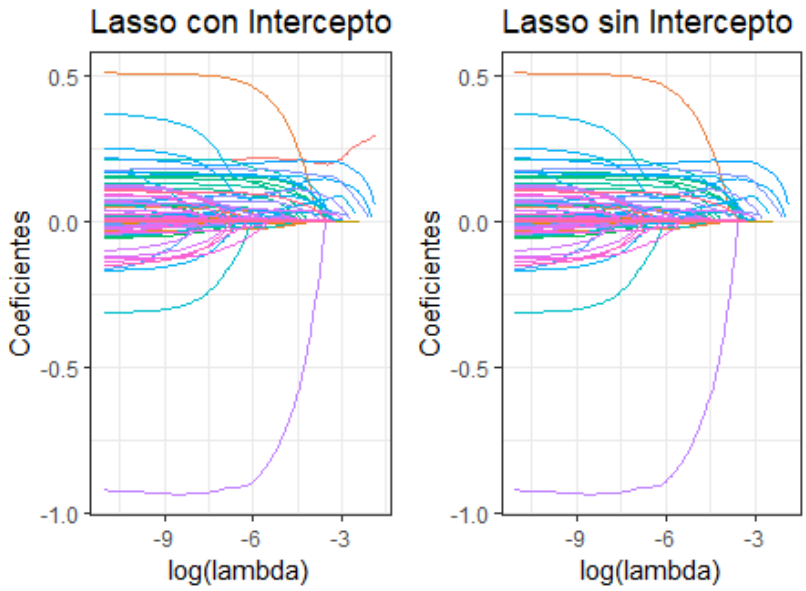

Figura 3.4: Coeficientes en función de lambda.

Podemos visualizar aquellas variables que sobreviven para valores de λ mayores. Sin embargo, no se logra identificar un valor de λ que sea óptimo. Para encontrar un lambda óptimo lo común es realizar validación cruzada. Para ello, *cv.glmnet* es la función que nos permite realizar lo de manera sencilla.

La salida de este modelo nos brinda muchísima información:

- lambda
- cvm (Cross-validation mean): media del MSE.
- cvsd (Cross-validation Standar Error): Desviación estándar del MSE.
- nzero: Coeficientes distintos de cero .
- lambda.min: lambda para el cual el MSE (error) es mínimo = 0,001009
- lambda.1se: lambda que se encuentra a 1 desviación estándar de lambda.min = 0,009411

La figura 4.5 nos muestra la media del MSE con su límite inferior y superior y la cantidad de variables que se consideran importantes para cada valor de λ.

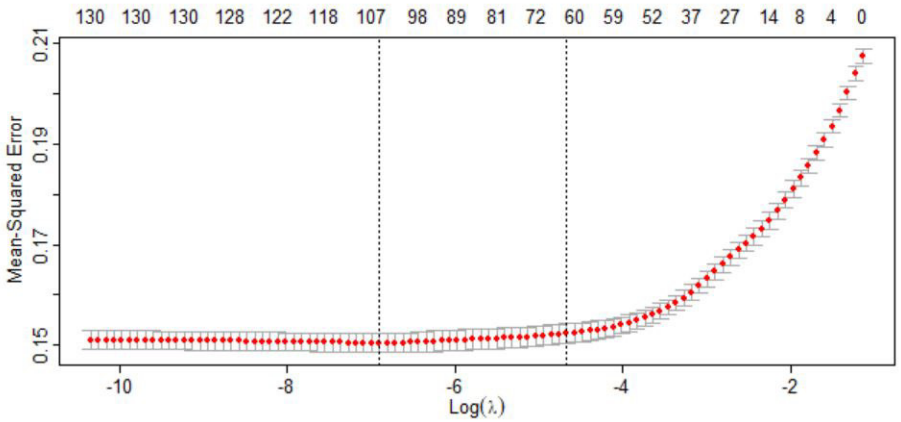

Figura 3.5: Media del MSE.

Seleccionamos el lambda óptimo para crear el modelo final. Según [22], en cuanto a la mejor lambda para glmnet, la regla general es usar lambda.1se en lugar de lambda.min. $\lambda.min$ es el valor de λ,1se que da el error de validación cruzada media mínima, mientras que λ,1se es el valor de λ que nos da para un modelo más regularizado tal que el error de validación cruzada está dentro de un error estándar del mínimo.

Con ese λ hacemos un llamado nuevamente al modelo final y se obtiene que, de los 132 predictores disponibles en Boruta, 106 predictores son relevantes para Lasso.

3.3. Validación del Modelo

Una vez hemos seleccionado las variables más relevantes, es necesario comprobrar resultados mediante técnicas de validación.

Se calcula un modelo de regresión logística para tres bases de datos. Una de ellas contiene todas las variables seleccionadas mediante el proceso de selección de variables planteado en este proyecto, dónde se utilizó el método MARS, seguido de Boruta y finalmente Lasso. La otra, es la base con las 100 primeras variables según coeficiente Gini y la última es una base con las 100 primeras variables según coeficiente gini y árboles de decisión. Es preciso decir que, en la industria frecuentemente se aplican estos dos últimos criterios para selección de variables.

Regresión Logística con las primeras 100 variables, según coeficiente Gini

Ahora, con un cálculo de coeficientes gini para cada una de las variables de nuestra base de datos inicial limpia de la sección 3.1, se ordenan de forma ascendente, de

manera que se obtienen las primeras 100 variables con coeficiente gini alto. Estos se almacenan en una nueva base de datos con la misma cantidad de individuos iniciales y procederemos a correr nuestro modelo de regresión logística para identificar su desempeño.

La función glm() disponible en el software estadístico R, será la función que nos permite obtener una regresión logística dónde se predice el resultado de la variable categórica 'Var_Objetivo' de nuestra base. El cuadro a continuación presenta los argumentos de esta función para mejor entendimiento.

Argumentos	**Descripción**
formula	Un objeto de clase.
family	Descripción de la distribución de errores y la función de enlace que se utilizará en el modelo.
data	Marco de datos que contiene las variables en el modelo.

Cuadro 3.10: Argumentos de la función glm(). **Fuente:** RDocumentation. Paquete stats.

3.3.1. Curva ROC 1

El modelo de regresión logística de este apartado arroja un valor de:

$$AUC = 0{,}8215 \tag{3.1}$$

Así,

$$\begin{aligned} Gini &= 2AUC - 1 \\ &= 2(0{,}8215) - 1 \\ &= 0{,}6430 \end{aligned} \tag{3.2}$$

La Curva ROC de la Figura 3.6 muestra evidencias de los valor anteriores, dónde el eje de la abscisa representa el ratio de falsos positivos o 1- especificidad y el eje de ordenadas, el ratio de verdaderos positivos o sensibilidad.

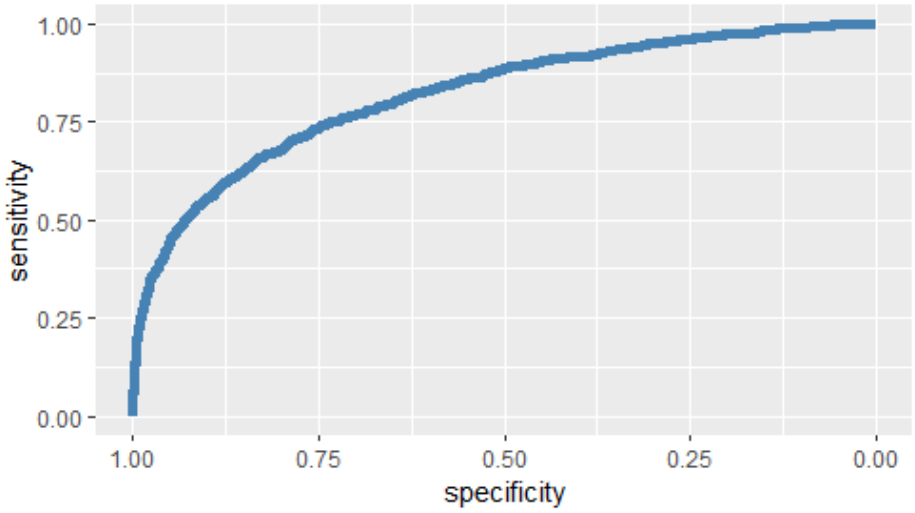

Figura 3.6: Curva ROC para el primer modelo.

Los resultados muestran un buen modelo. En general, si un valor AUC se encuentra entre el intervalo [0.75, 0.9) se considera un test bueno

Regresión logística luego de aplicar árbol de decisión con las primeras 100 variables, según coeficiente Gini.

Uno de los métodos más usuales en la industria, es la categorización de variables mediante árboles de decisión. Utilizaremos este método para categorizar las 100 primeras variables según coeficiente Gini para observar resultados.

En Aprendizaje Automático existen diversas formas de obtener árboles de decisión, en este proyecto se usa la conocida como árboles de clasificación y regresión CART (Classification and Regresssion Tress). Como su nombre lo indica, esta técnica nos ayuda a obtener árboles de clasificación y de regresión. No obstante, usaremos clasificación dado que nuestra variable objetivo es discreta.

La implementación de CART que se usará es conocida como Árboles de regresión y partición recursiva con sus siglas en inglés Recursive Partitioning and Regression Tress o RPART". A continuación se presenta la función *rpart()* usando el paquete que lleva el mismo nombre para mejor entendimiento.

Argumentos	Descripción
formula	una fórmula , con una respuesta pero sin términos de interacción.
data	un marco de datos opcional en el que interpretar las variables nombradas en la fórmula.
method	Se especifica un método para luego agregar más criterios; ejem: method="poisson", method=çlass", method=.ªnova."
control	Lista de opciones que controlan la complejidad del árbol. Ayuda a cortar, limitar y optimizar el tamaño y la forma de los árboles.

Cuadro 3.11: Argumentos de la función rpart(). **Fuente:** RDocumentation. Paquete stats.

Por lo general, este algoritmo busca la variable regresora que mejor separa en grupo los datos, que corresponden con las categorías de la variable objetivo. Esta separación es expresada con una regla. Un nodo es correspondiente a cada regla.

Para este proyecto se supone la variable Objetivo como ya lo habíamos mencionado "Var_Objetivo", que tiene dos grados, buen pagador y mal pagador. El algoritmo busca la variable que mejor separa nuestros datos, con ello se marca una regla. Así, los datos para los que la regla es verdadera significará que tendrá más probabilidad de pertenecer a un grupo, que al otro. Este proceso se realiza de manera recursiva hasta que no es imposible obtener una separación mejor y se obtiene un nodo terminal.

Ahora bien, construimos un primer modelo con la función rpart(), que por defecto utiliza la medida de Gini mencionada en el capítulo 2 para la división de los nodos.

Medimos el rendimiento del modelo de nuestro primer árbol y se obtiene un valor de exactitud (Accuracy) de:

$$Accuracy = 0{,}9038 \tag{3.3}$$

La precisión en este caso se consiera buena. Sin embargo, recordamos que un árbol de decisión suele ser inestable y puede no garantizar que se genere el óptimo. En consecuencia, mejoramos aún más su precisión realizando un ajuste de los hyperparámetros, incluyendo restricciones definidas por el autor utilizando el comando cp de la librería rpart que ayuda a la poda del árbol y ayuda a controlar su complejidad. Además, se utilizan los parámetros $maxdepth = 30$ indicando un número máximo de ramas y $minisplit = 10$ indicando un número mínimo de ocurrencias.

El resultado nos muestra un rendiento mejorado de :

$$Accuracy = 0{,}9043 \tag{3.4}$$

3.3.2. Curva ROC 2

Dentro de este marco se procede a calcular el valor AUC de un modelo de regresión, utilizando las variables categorizadas de nuestro último árbol de decisiones.

$$AUC = 0{,}8190 \tag{3.5}$$

Así,

$$\begin{aligned} Gini &= 2AUC - 1 \\ &= 2(0{,}8190) - 1 \\ &= 0{,}6380 \end{aligned} \tag{3.6}$$

Es posible seguir probando varias combinaciones para encontrar el óptimo. Sin embargo, su valor AUC no aumentará con notoriedad.

La Figura 3.7 evidencia que también es un buen modelo.

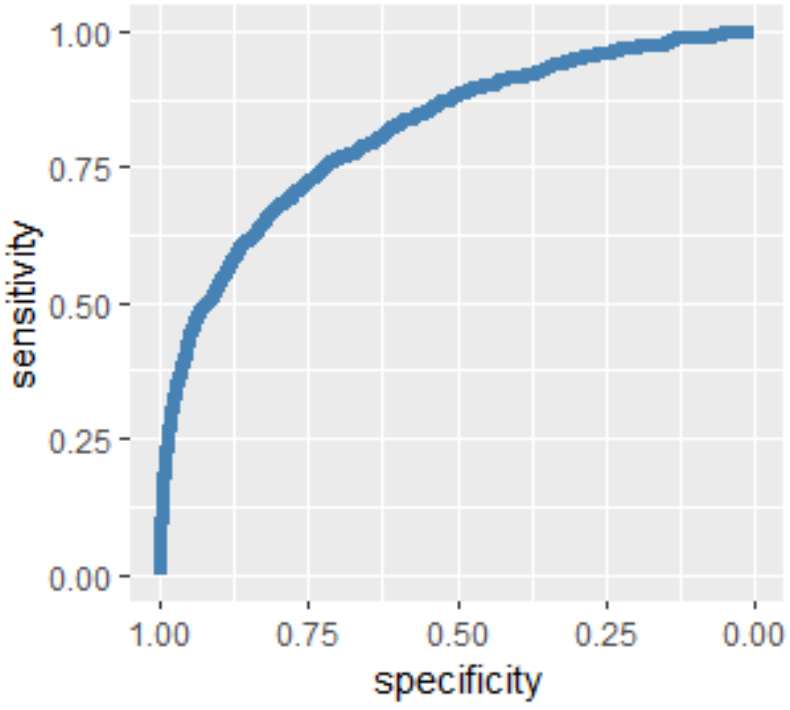

Figura 3.7: Curva ROC para el segundo modelo.

Regresión logística con las 107 variables procedentes del proceso de selección de variables

Para nuestra base de datos final, de la sección 3.2, calculamos su valor AUC y visualizamos su gráfica.

3.3.3. Curva ROC 3

$$AUC = 0{,}8246 \tag{3.7}$$

Así,

$$\begin{aligned} Gini &= 2AUC - 1 \\ &= 2(0{,}8246) - 1 \\ &= 0{,}6492 \end{aligned} \tag{3.8}$$

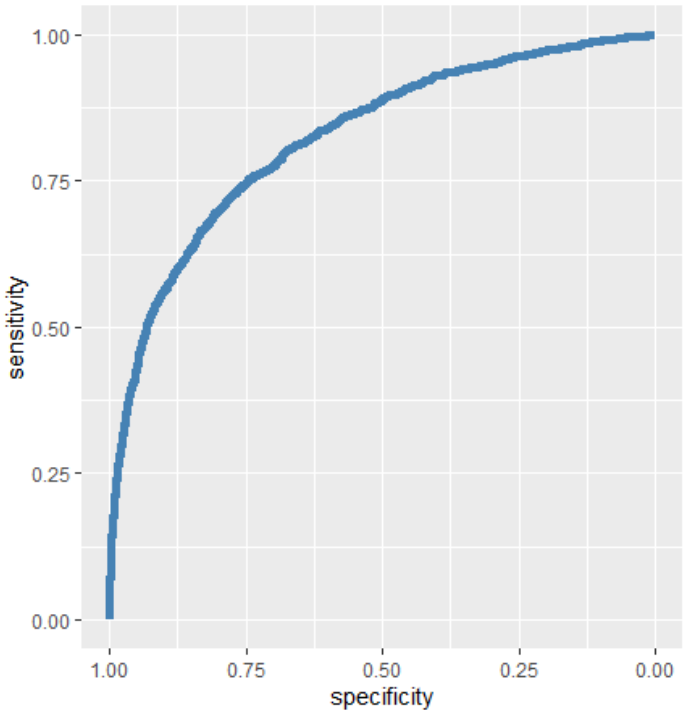

Figura 3.8: Curva ROC para el último modelo planteado en validación.

Esto verifica que nuestro modelo tiene un 82 % de probabilidad de que pueda distinguir entre clase positiva y clase negativa.

En concreto, este modelo obtiene una pequeña mejora comparado con los mode-

los anteriores.

A continuación, se detalla los resultados obtenidos de estos tres métodos.

3.4. Análisis de resultados

Otra representación de resultados de este estudio, es con el cálculo del p-valor. Un p-valor o valor de probabilidad, es aquel que manifiesta qué relaciones son significativas en un modelo de regresión y la naturaleza de esas relaciones.

Se muestra a continuación una tabla de p-valores correspondiente a los 3 procesos probados anteriormente.

3.4.1. Proceso con variables según Coeficiente Gini

En este proceso, luego de aplicar regresión logística con los 100 valores más alto según coeficiente gini, se obtuvieron los siguientes resultados:

Variables Significativas	41
Variables No Significativas	59
Total de Variables	100

Cuadro 3.12: Tabla de resultados del primer proceso planteado.

41 variables significativas, es decir, variables con un p-valor<0.05 y 59 de no significativas. Cabe mencionar que, estas variables fueron tomadas con coeficiente gini alto y que además se obtuvo un valor AUC bueno. La tabla de p-valores se encuentra anexada en el Apéndice C.

3.4.2. Proceso con variables según Coeficiente Gini y Árbol de Decisión

Este proceso, como ya se había mencionado, es considerado el más usual en la industria, gracias a la categorización que realiza el árbol de decisión. Aplicando nuestro árbol de decisión se obtuvieron 85 variables importantes con los siguientes resultados:

Variables Significativas	28
Variables No Significativas	57
Total de Variables	85

Cuadro 3.13: Tabla de resultados del segundo proceso.

28 variables significativas y 57 no significativas. Asimismo, se obtuvo un valor AUC bueno en su modelo de regresión. La tabla de p-valores se encuentra Anexada en el apéndice D.

3.4.3. Proceso con variables seleccionadas del proceso planteado en el estudio

Tenemos en cuenta que, los p-valores calculados en este apartado proceden del análisis de regresión con las 107 variables finales según procesos de selección que se planteó en este proyecto. Adicionalmente, hemos realizado una comparación con los valores de coeficiente Gini de las variables. Los resultados se muestran a continuación:

	Nro de Variables	**Total de Variables**
Variables Significativas	44	107
Variables No Significativas	63	
Variables con Coeficiente Gini > 0.5	90	107
Variables con Coeficiente Gini < 0.5	17	

Cuadro 3.14: Tabla de resultados del tercer proceso.

El Cuadro 3.14 nos indica que, nuestro modelo de regresión tiene 44 variables significativas y 63 no significativas. De la misma forma, posee 90 variables con coeficiente gini alto y 17 con coeficiente bajo. Con lo que se concluye que, muchas de las variables que seleccionó el método planteado en este proyecto resultan ser tomadas con coeficiente Gini alto y el 41 % de ellas son significativas. Además, la comparación de estos tres métodos manifiesta que, el 17,76 % variables seleccionadas del primer método coinciden con el método planteado en este estudio y el 14,95 % cocinciden con el segundo método.

La tabla de p-valores y coeficientes se encuentra anexada en el Apéndice E.

Capítulo 4

Conclusiones y Recomendaciones

En la realización de este trabajo se ha podido demostrar la importancia de conocimientos como el Aprendizaje Automático, Minería de Datos, Algoritmos de Regresión y lo rápido que avanzan estos últimos para el análisis de datos. Dado que es un área aún en desarrollo, constantemente se actualizan nuevos métodos y es gratificamente estudiarlos ya que cada vez son mucho más fáciles de entender y de aplicarlos en algún software estadístico.

4.1. Conclusiones

1. Una de las preguntas en este proyecto fue, ¿Qué pasaría si usamos otros algoritmos tradicionales de selección de características en el mismo conjunto de datos dado? La respuesta a esta pregunta la encontramos en la sección de validación. Utilizamos tres diferentes técnicas para seleccionar variables dónde se seleccionan distintas variables consideradas importantes en cada método. Los tres procesos de selección resultaron ser considerablemente buenos.

2. Este estudio se lo ha realizado con datos de una consultoría privada, se podría profundizar la investigación utilizando más métodos de selección o ajustando parámetros de los mismos. Este proceso creado tiene una gran aplicación dentro del campo de Aprendizaje Automático dado el manejo de una base de datos oportunamente grande y de este modo se puede utilizar como ayuda tanto para empresas con manejo de grandes volúmenes de datos.

3. El uso del primer método MARS fue más influyente a la hora de reducción de variables porque, además de automatizar los aspectos de una modelación

de regresión clásica, resumir mejor la versión de la regresión lineal, estimar valores perdidos, transformar variables, entre otros, ayudó a una selección de predictores mucho más amplia en comparación con los otros métodos. Este redujo de 1702 variables a 145 incluida las variables transformadas.

4. El algoritmo Boruta resulta ser una técnica no convencional que arroja resultados claros de importancia de variables, además que es fácil de interpretar ya que no se consideran varios para sintonizar.

5. Una de las desventajas más notable de Boruta es su alta complejidad en tiempo computacional. A pesar de ser sencillo de interpretar y clasificar sus variables en importantes y no importantes. Boruta demoró en este proyecto 14,98594 horas en tiempo de cómputo.

4.2. Recomendaciones

- Es importante mantenerse informado de nuevos métodos de selección de variables ya que pueden resultar mejores que los utilizados habitualmente en la industria.

- Para el estudio de estos métodos es recomendable utilizar paquetes ya implementados en R.

- El uso del software estadístico Rstudio fue de gran utilidad. A pesar de ello, existen también otras herramientas factibles de estudio como la de Python que, trabaja de manera más intuitiva para métodos de Aprendizaje Automático.

Bibliografía

[1] Amat, J. *Regresión logística simple y múltiple.* Recuperado de https://www.cienciadedatos.net, 2016.

[2] Amat, J. *Selección de predictores, regularización ridge, lasso, elastic net y reducción de dimensionalidad.* Recuperado de https://www.cienciadedatos.net/documentos/31_seleccion_de_predictores_subset_selection_ridge_lasso_dimension_reduction, 2016.

[3] Amat, J. *Regularización Ridge, Lasso y Elastic Net con Python.* Recuperado de https://www.cienciadedatos.net/documentos/py14-ridge-lasso-elastic-net-python.html, 2020.

[4] Arroyo-Hernández, J. Métodos de reducción de dimensionalidad: Análisis comparativo de los métodos apc, acpp y acpk. *Volumen 30. 115p*, 2016.

[5] Bello, E. *¿Qué es el minado de Datos o Data Mininig? Técnicas y pasos a seguir.* Recuperado de https://www.iebschool.com/blog/data-mining-mineria-datos-big-data/, 2021.

[6] Bernard, M. *Una breve historia del aprendizaje automático.* Recuperado de https://www.forbes.com/sites/bernardmarr/2016/02/19/a-short-history-of-machine-learning-every-manager-should-read, 2016.

[7] Bhalla, D. *Selección de Variables con el paquete Boruta.* Recuperado de https://www.listendata.com/2017/05/feature-selection-boruta-package.html, 2017.

[8] Bilogur, A. *Selección automática de características con boruta.* Recuperado de https://www.kaggle.com/residentmario/automated-feature-selection-with-boruta, 2018.

[9] Brownlee, J. *Introducción a la reducción de dimensionalidad para el aprendizaje automático.* Recuperado de https://machinelearningmastery.com/, 2020.

[10] Costa, J. *Aprendizaje Estadístico.* Recuperado de https://rubenfcasal.github.io/aprendizaje_estadistico/, 2021.

[11] Data., P. *Big Data ¿En qué consiste? Su importancia, desafíos y gobernabilidad.* Recuperado de https://www.powerdata.es/, 2018.

[12] Dau, M, A. K. A. L. . Aplicación de los modelos de respuesta binaria a los determinantes de la demanda de postgrado. *Caribe. Vol 14.*, 2016.

[13] Dobilas, S. *MARS: Splines de regresión adaptativa multivariante: ¿cómo mejorar la regresión lineal?.* Recuperado de https://towardsdatascience.com/mars-multivariate-adaptive-regression-splines-how-to-improve-on-linear-regress 2020.

[14] Dutta, D. *¿Cómo realizar la selección de funciones (es decir, elegir variables importantes) usando el paquete Boruta en R?.* Recuperado de https://www.analyticsvidhya.com/blog/2016/03/select-important-variables-boruta-package/, 2016.

[15] Dutta, D. *¿Cómo realizar la selección de funciones (es decir, elegir variables importantes) usando el paquete Boruta en R?.* Recuperado de https://www.analyticsvidhya.com/blog/2016/03/select-important-variables-boruta-package/#:~:text=Boruta, 2016.

[16] España., R. Qué son regresión y clasificación en machine learning. *B12admark*, 2020.

[17] Friedman, J. . Splines de regresión adaptativa multicariante. *The Annals of Statistics*, 1991.

[18] Gimenez, Y. Selección de variables para datos multivariados y datos funcionales. *Argenita Estadìstica de San Andrés*, 2015.

[19] Gonzales, L. Curvas roc y Área bajo la curva (auc). *AprendeIA*, 2019.

[20] Guerrero, J. El problema de la dimensionalidad. *CEO Datrik Intelligence, S.A.*, 2016.

[21] Gupta, A. *Técnicas de selección de funciones en el aprendizaje automático*. Recuperado de https://www.analyticsvidhya.com/blog/2020/10/feature-selection-techniques-in-machine-learning/, 2020.

[22] Hastie, T, Q. J. y. T. K. *Una introducción aglmnet.* Recuperado de https://glmnet.stanford.edu/articles/glmnet.html, 2021.

[23] Hewlett, P. *¿Qué es el aprendizaje automático.* Recuperado de https://www.hpe.com/lamerica/es/what-is/machine-learning.html, 2020.

[24] Kjell, J. Modelado predictivo aplicado. *Nueva York, NY: Springer New York. doi : 10.1007*, 2013.

[25] Kuhn M, J. K. . Feature enginnering and selection: A practical approach for predctive models. *Chapman Hall/ CRC Data Science Series.*, 2019.

[26] Martinez, J. *Las 7 Fases del Proceso de Machine Learning.* Recuperado de https://www.iartificial.net/fases-del-proceso-de-machine-learning/, 2020.

[27] Martinez, J. *Regresión Logística para Clasificación.* Recuperado de https://www.iartificial.net/regresion-logistica-para-clasificacion/#Curiosidades, 2020.

[28] Martinez, J. *Árboles de Decisión con ejemplos en Python.* Recuperado de https://www.iartificial.net/arboles-de-decision-con-ejemplos-en-python/#Arboles_de_Decision_para_Clasificacion, 2020.

[29] Melillanca, . *Evaluación de modelos de clasificación: Matriz de Confusión y Curva ROC.* Recuperado de http://ericmelillanca.cl/content/evaluaci-n-modelos-clasificaci-n-matriz-confusi-n-y-curva-roc, 2018.

[30] Miron, B. Boruta para los que tienen prisa. 2020.

[31] Orellana, J. *Arboles de decision y Random Forest.* Recuperado de https://bookdown.org/content/2031/arboles-de-decision-parte-i.html, 2018.

[32] Pathak, M. *Selección de funciones en R con el paquete Boruta R.* Recuperado de https://www.datacamp.com/community/tutorials/feature-selection-R-boruta, 2018.

[33] Pulkit, S. *La guía definitiva para 12 técnicas de reducción de dimensionalidad*. Recuperado de https://www.analyticsvidhya.com/, 2018.

[34] Raj, T. *A beginner's guide to dimensionality reduction in Machine Learning.* Recuperado de https://towardsdatascience.com/, 2019.

[35] Ramos, L. Regresión lasso. *Departamento de Estadística e Investigación Operativa*, 2018.

[36] Rodríguez, D. *La regresión logística.* Recuperado de https://www.analyticslane.com/2018/07/23/la-regresion-logistica/, 2018.

[37] Rodó, P. Multicolinealidad. *Economipedia.com*, 2019.

[38] Salcedo, P y Mercedes, C. Estimación de la ocurrencia de incidencias en declaraciones de pólizas de importanción. 2002.

[39] Salinas, C. *4 Pasos para preparar tus bases de datos para análisis.* Recuperado de https://blogdatlas.wordpress.com/, 2020.

[40] Sherer, T Y Coulter, D. *Validación cruzada (Servicios de análisis - Minería de datos).* Recuperado de https://docs.microsoft.com/en-us/analysis-services/data-mining/cross-validation-analysis-services-data-mining?view=asallproducts-allversions, 2020.

[41] Shwartz, S. y. B. Comprender el aprendizaje automático: de la teoría a los algoritmos. *Prensa de la Universidad de Cambridge*, 2014.

[42] Silipo, R, A. I. H. A. B. M. Seven techniques for data dimensionality reduction. 2009.

[43] Solorio, S. Selección de variables para clasificación no supervisada utilizado un enfoque híbrido filter-wrapper. *Instituto Nacional de Astrofísica óptica y electrónica.*, 2010.

[44] Stedman, C y Hughes, A. *¿Qué es la ciencia de datos? La ultima guia.* Recuperado de https://searchbusinessanalytics.techtarget.com/definition/data-mining, 2020.

[45] Stephen K., Armour F., A. E. M. W. Big data: Issues and challenges moving forward. *Conferencia del sistema (HICSS)*, 2013.

[46] Vanegas, J y Vásquez, F. Multivariate adaptative regression splines (mars), una alternativa para el análisis de series de tiempo. 2017.

[47] Velazques, M. *Historia y evolución del Machine Learning.* Recuperado de https://recluit.com/historia-y-evolucion-del-machine-learning/#.YhRjD-jMJPZ, 2018.

[48] Wolfgang, M. . *Dimension Reduction vs. Variable Selection.* Recuperado de , 2016.

[49] Zambrano, R. *Técnicas de Machine Learning.* Recuperado de https://openwebinars.net/blog/modelos-de-machine-learning/, 2019.

[50] Álvaro, S y Mur, R. Estudio de técnicas supervisadas de reducción de dimensionalidad para problemas de clasificación. 2017.

Anexos

Apéndice A

Código de implementación de los métodos de selección en Rstudio

A.1. Algoritmo Splines de Regresión Adaptativa Multivariante (MARS)

```
#Separacion de Datos
datos_150 <- train[0:150]
datos_300<- select(train, 151:300,starts_with("Var_Objetivo"))
datos_450<- select(train, starts_with("Var_Objetivo"), 301:450)
datos_600<- select(train, starts_with("Var_Objetivo"), 450:600)
datos_750<- select(train, starts_with("Var_Objetivo"), 600:750)
datos_900<- select(train, starts_with("Var_Objetivo"), 750:900)
datos_1050<- select(train, starts_with("Var_Objetivo"), 900:1050)
datos_1200<- select(train, starts_with("Var_Objetivo"), 1050:1200)
datos_1350<- select(train, starts_with("Var_Objetivo"), 1200:1350)
datos_1500<- select(train, starts_with("Var_Objetivo"), 1350:1500)
datos_1650<- select(train, starts_with("Var_Objetivo"), 1500:1650)
datos_1704<- select(train, starts_with("Var_Objetivo"), 1650:1702)

#Aplicacion del modelo MARS con earth()
mars4<- earth(Var_Objetivo ~ . ,data=datos_150)
mars5<- earth(Var_Objetivo ~ ., data=datos_300)
mars6<- earth(Var_Objetivo ~ ., data=datos_450)
mars7<- earth(Var_Objetivo ~ ., data=datos_600)
```

```
mars8<- earth(Var_Objetivo ~ ., data=datos_750)
mars9<- earth(Var_Objetivo ~ ., data=datos_900)
mars10<- earth(Var_Objetivo ~ ., data=datos_1050)
mars11<- earth(Var_Objetivo ~ ., data=datos_1200)
mars12<- earth(Var_Objetivo ~ ., data=datos_1350)
mars13<- earth(Var_Objetivo ~ ., data=datos_1500)
mars14<- earth(Var_Objetivo ~ ., data=datos_1650)
mars15<- earth(Var_Objetivo ~ ., data=datos_1704)

#ExtracciOn de las variables importantes

gbmImp4 <- varImp(mars4, scale = FALSE)
gbmImp5 <- varImp(mars5, scale = FALSE)
gbmImp6 <- varImp(mars6, scale = FALSE)
gbmImp7 <- varImp(mars7, scale = FALSE)
gbmImp8 <- varImp(mars8, scale = FALSE)
gbmImp9 <- varImp(mars9, scale = FALSE)
gbmImp10 <- varImp(mars10, scale = FALSE)
gbmImp11 <- varImp(mars11, scale = FALSE)
gbmImp12 <- varImp(mars12, scale = FALSE)
gbmImp13 <- varImp(mars13, scale = FALSE)
gbmImp14 <- varImp(mars14, scale = FALSE)
gbmImp15 <- varImp(mars15, scale = FALSE)
```

A.2. Función para crear las variables que sugiere transformación del MARS

```
#vector con variables importantes del modelo MARS
TablaOriginal<-as.vector(names(train))# variables train ORIGINAL
dimensionOriginal<-length(TablaOriginal)
TablaMars<-as.vector(row.names(tablaVariablesMars))
dimensionMars<-nrow(tablaVariablesMars)

#funcion que permite separar variables del train original
#con las creadas del mars
```

```
resultado<-NULL
for (i in 1:dimensionOriginal) {
  str<- TablaOriginal[i]
  for (j in 2:dimensionMars) {
    chars<- TablaMars[j]
    EstaConteniendo<-grepl(chars, str, fixed = TRUE)
    if (EstaConteniendo){
      resultado<-append(resultado,str)
      break

    }
  }
}

#Matriz binaria de variables transformadas del MARS

sukasa<-as.vector(names(train))
LongitudSukasa<-length(sukasa)
sukasaMars<-as.vector(row.names(tablaVariablesMars))
LongitudSukasaMars<-length(sukasaMars)
vectorGrande<-NULL

for (i in 1:LongitudSukasa) {
  almacen<- sukasa[i]
  for (j in 1:LongitudSukasaMars) {
    almacenmars<-sukasaMars[j]
    isContained<-grepl(almacen, almacenmars, fixed = TRUE)
    esIgual<-almacen==almacenmars
    if (isContained & esIgual==FALSE){
      longitudAlmacen<-nchar(almacen)
      longitudAlmacenMars<-nchar(almacenmars)
      resultadtoAlmacen<- substr(almacenmars,1,longitudAlmacen)
      resultadoSobrante<-substr(almacenmars,longitudAlmacen+1,
          longitudAlmacenMars)
      vectorSmall<-c(resultadtoAlmacen,resultadoSobrante)
```

```
      vectorGrande<-rbind(vectorGrande,vectorSmall)
    }
  }
}

longitudmatrizGrande<-nrow(vectorGrande)

matrizBinaria<-matrix()
for (i in 1:longitudmatrizGrande) {
  listaBinaria<-c()
  nameAlmacen<-vectorGrande[i,1]
  nameSobrante<-vectorGrande[i,2]
  valoresOriginales<-select(train,nameAlmacen)
  nombreCompuesto<-paste(nameAlmacen,nameSobrante)
  listaBinaria<-append(listaBinaria,nombreCompuesto)
  for (j in 1:nrow(valoresOriginales)) {
    value<-valoresOriginales[j,1]
    if(value==nameSobrante){
      listaBinaria<-append(listaBinaria,1)
    }
    else{
      listaBinaria<-append(listaBinaria,0)
    }
  }
  matrizBinaria<-cbind.data.frame(matrizBinaria,listaBinaria)
}
matrizBinaria <- matrizBinaria[, -1]
oldnames <-names(matrizBinaria)
newnames <-gsub("␣", "", names(matrizBinaria))
matrizBinaria<-setnames(matrizBinaria, old = oldnames, new = newnames)
```

A.3. Algoritmo Boruta

```
#Utilizacion de la funcion Boruta()
BorutaTestMarsBinary<-Boruta(data.frame(matrizFinalMars),
```

```
    matrizFinalMars$Var_Objetivo,doTrace = 2)
#Extracci n de variables importantes
boruta_significativas <- names(BorutaTestMarsBinary$finalDecision
    [BorutaTestMarsBinary$finalDecision %in% "Confirmed"])
```

A.4. Algoritmo Lasso

```
# Seleccion lambda optimo
lasso_lambda_opt = lasso_cv$lambda.min
lasso_opt = glmnet(x=xB, # Matriz de regresores
                   y=matrizFinalBoruta$Var_Objetivo,
                   alpha=1, # Indicador del tipo de regularizacion
                   standardize = TRUE,  # Estandarizamos
                   lambda = lasso_lambda_opt)

#Variables importantes
variablesLassoImport<-lasso_opt %>% tidy()
```

Apéndice B

Tabla de variables MARS

Variable	AUC	GINI
RazonCobSaldo6M_36M	0,50117487	0,00234973
bancoPeorCalf COOP 11 DE JUNIO	0,69917688	0,39835377
SumaMicTotalVencido6M	0,5040248	0,00804961
SumaTotalCarteraCastigadaSinHip3M	0,50220448	0,00440897
MaxSaldoVencidoMicrocredito6M	0,51027253	0,02054507
RazonMicTotalPorVencer3M_9M	0,5030158	0,0060316
peorCalfActual T	0,64679269	0,29358538
RazonTotalCarteraCastigada3M_6M	0,50221554	0,00443108
RazonTotalCarteraCastigadaSinHip3M_6M	0,50221554	0,00443108
tiempoInhabilitacion	0,51105247	0,02210494
peorCalfActual Y1	0,64579927	0,29159855
MinSerSaldo3M	0,49940651	-0,00118698
MinComSaldo12M	0,52049309	0,04098618
MaxTotalCarteraCastigadaSinHip36M	0,49285136	-0,01429728
menorCalifTc R	0,69460718	0,38921435
RazonTotalDeudaSinTC6M_36M__1	0,50004327	8,6538E-05
MaxSerTotalPorVencer6M	0,50004604	9,208E-05
PromedioSerTotalPorVencer6M	0,50004061	8,1228E-05
RazonCantProdRef6M_36M	0,51812772	0,03625544
peorCalfActual V2	0,69338669	0,38677339
SumaCantProdRef9M	0,51797182	0,03594364
SumaCantProdRef12M	0,52015258	0,04030517

Cuadro B.1: Tabla de variable Mars

Variable	AUC	GINI
peorCalfActual V1	0,69210945	0,38421889
bancoPeorCalf BELCORP	0,64023615	0,2804723
SumaTotalVencidoSinHip12M	0,67323661	0,34647322
MinProductosSaldoVencido12M	0,52539275	0,0507855
MaxSaldoVencidoTarjetaDeCredito12M	0,58149057	0,16298113
PromedioComTotalVencido36M	0,57842259	0,15684518
MaxTotalVencidoSinHip9M	0,66009982	0,32019963
peorCalfHistorica Y2	0,68583674	0,37167348
SumaDiasAtrasoTotales3M	0,62481368	0,24962735
peorCalfActual O2	0,63001817	0,26003633
PromedioDiasAtrasoTotales36M	0,68477066	0,36954133
peorCalfActual O3	0,67984787	0,35969573
peorCalfActual R	0,62981948	0,25963896
MaxSaldoVencidoConsumo24M	0,6739047	0,3478094
RazonComTotalVencido9M_36M	0,56463699	0,12927398
MaxSaldoVencidoTarjetaDeCredito36M	0,58872159	0,17744318
MaxTotalVencidoSinHip24M	0,67896939	0,35793879
AcrescimoComSaldo6M	0,53127986	0,06255972
RazonSaldoVencidoTarjetaDeCredito9M_36M	0,56936708	0,13873416
MaxDiasAtrasoTotales12M	0,66916657	0,33833313
SumaTotalVencido36M	0,68573955	0,3714791
PromedioTotalVencido36M	0,68512102	0,37024203
DecrescimoComSaldo3M	0,5405013	0,08100261
valorPeorCalf	0,66194608	0,32389216
MaxSaldoVencidoConsumo36M	0,67568542	0,35137083
mayorValorVencido	0,68244304	0,36488609
MaxTotalVencidoSinHip36M	0,682144	0,36428801
SumaProductosSaldoVencido12M	0,64379491	0,28758982
RazonTotalVencido3M_9M	0,62452859	0,24905719
RazonDiasAtrasoTotales3M_6M	0,62269847	0,24539694
RazonSaldoAlDiaMicrocredito9M_36M	0,52520705	0,0504141
RazonTotalVencidoSinHip3M_6M	0,62257206	0,24514411
MaxComSaldo3M	0,60181927	0,20363855

Cuadro B.2: Continuación tabla de variables MARS

Variable	AUC	GINI
RazonTotalVencido6M_36M	0,64734882	0,29469765
RazonProductosSaldoVencido6M_12M	0,62386661	0,24773323
MaxTotalVencido36M	0,68029773	0,36059547
MaxDiasAtrasoTotales36M	0,68181994	0,36363989
RazonDiasAtrasoTotales9M_36M	0,65302171	0,30604342
RazonTotalVencidoSinHip9M_36M	0,65483557	0,30967114
PromedioProductosSaldoVencido36M	0,66032479	0,32064957
RazonTotalVencido9M_36M	0,65442716	0,30885433
RazonComTotalPorVencer6M_36M	0,58322223	0,16644446
RazonComSaldo3M_9M	0,59927671	0,19855343
MaxComSaldo36M	0,59866202	0,19732404
RazonComSaldo3M_6M	0,59914719	0,19829437
RazonNumAcreedoresSICOM3M_6M	0,5988609	0,1977218
RazonComSaldo9M_36M	0,5951985	0,19039699
bancoPeorCalf PACIFICO	0,62288261	0,24576521
peorCalfHistorica O3	0,6684832	0,33696639
SumaTotalDeudaSinTC24M	0,46672491	-0,06655019
SumaVFAC_12	0,58004642	0,16009283
MinTotalDeudaRotativos3M	0,52411695	0,04823389
MinSaldoAlDia9M	0,50623162	0,01246325
PromedioTotalDeudaRotativos3M	0,52415671	0,04831343
DecrescimoTotalDeudaSinHip9M	0,51613258	0,03226517
DecrescimoSaldoAlDiaConsumo9M	0,52057794	0,04115587
MinSaldoAlDiaTarjetaDeCredito3M	0,49750288	-0,00499424
SumaDiasAtrasoTotales3M_Conyuge	0,57600748	0,15201495
MaxDiasAtrasoTotales3M_Conyuge	0,57599034	0,15198069
MaxDiasAtrasoTotales6M_Conyuge	0,58159214	0,16318428
totalCarteraCastigada1M_Conyuge	0,55167436	0,10334871
MaxTotalDeudaRotativos12M	0,5055371	0,0110742
RazonDiasAtrasoTotales3M_6M_Conyuge	0,57862436	0,15724871
PromedioTotalSaldoActualTC12M	0,48972323	-0,02055355
MaxTotalDeudaRotativos24M	0,49706467	-0,00587067
PromedioSaldoTC3M	0,50889262	0,01778524

Cuadro B.3: Continuación tabla de variables MARS

Variable	AUC	GINI
MaxTotalSaldoActualTC3M	0,50549053	0,01098107
MaxCuotaEstimada12M	0,53925855	0,07851709
AcrescimoTotalDeudaSinTC6M	0,48866046	-0,02267908
RazonTotalDeudaSinTC6M_36M	0,61312362	0,22624725
mayorValorVencido_Conyuge	0,5829099	0,1658198
MaxSaldoTC12M	0,51408968	0,02817936
MaxSaldoAlDiaTarjetaDeCredito24M	0,52276949	0,04553899
MaxSaldoAlDiaTarjetaDeCredito36M	0,52575855	0,05151711
PromedioTCAbiertas24M	0,5352186	0,0704372
Rentacomprobada	0,57842642	0,15685284
AcrescimoTotalSaldoActualTC9M_Conyuge	0,49846545	-0,00306911
DecrescimoTotalSaldoActualTC6M_Conyuge	0,52959475	0,0591895
cupoMaxTarjetas	0,58846013	0,17692026
RazonTotalDeudaSinTC3M_9M	0,6061586	0,21231719
MinTotalSaldoActualTC9M_Conyuge	0,53629531	0,07259063
usoCupoPorcentaje	0,64228598	0,28457196
PromedioTotalSaldoActualTC12M_Conyuge	0,52997155	0,05994309
MaxTotalSaldoActualTC3M_Conyuge	0,5277754	0,0555508
RazonTotalDeudaRotativos9M_36M	0,53767258	0,07534516
RazonSaldoAlDiaConsumo6M_36M	0,5736108	0,1472216
RazonTotalPorVencerSinHip6M_36M	0,57240304	0,14480608
RazonSaldoAlDia6M_36M	0,59300133	0,18600266
RazonTotalDeuda6M_36M	0,59092102	0,18184204
RazonCantidadOperacionesSinTC3M_6M	0,60095631	0,20191262
SumaProductosAlDia24M	0,52900028	0,05800056
RazonSaldoAlDiaTarjetaDeCredito9M_36M	0,55417201	0,10834402
antiguedad	0,59064381	0,18128762
MaxProductosAlDia3M	0,53447998	0,06895995
RazonCantidadProductos6M_36M	0,59372045	0,18744091
RazonTotalSaldoActualTC6M_12M_Conyuge	0,50673229	0,01346458
PromedioNentidadesSinHip9M	0,53844159	0,07688319
MaxProductosAlDia24M	0,52996728	0,05993456
RazonProductosAlDia9M_36M	0,56641864	0,13283729

Cuadro B.4: Continuación tabla de variables MARS

Variable	AUC	GINI
RazonCantidadProductos9M_36M	0,59037839	0,18075678
RazonTotalSaldoActualTC9M_36M	0,51854753	0,03709506
SumaCantidadProductos24M	0,51314076	0,02628152
RazonTotalSaldoActualTC6M_12M	0,53098994	0,06197988
fechaPeorCalf	0,54677904	0,09355807
RazonSaldoAlDiaTarjetaDeCredito3M_6M	0,5297527	0,05950541
RazonSaldoTC3M_6M	0,52853379	0,05706757
RazonTotalDeuda6M_12M	0,57443509	0,14887019
RazonCuotaEstimada6M_12M	0,57862349	0,15724698
MaxCantidadProductos3M	0,56635722	0,13271445
PromedioNumeroMesesCentralDeRiesgos36M	0,55687228	0,11374456
RazonNumeroDeAcreedoresSBS3M_6M	0,53614014	0,07228028
cantPagosConse12M	0,55582257	0,11164514
RazonCantidadProductos6M_12M	0,57267072	0,14534144

Cuadro B.5: Continuación tabla de variables MARS

Apéndice C

Tabla de p-valores del primer proceso

Variable	AUC	GINI	Pvalue
peorCalfHistorica	0,7057998	0,41159961	0,1631727
SumaTotalVencidoSinHip36M	0,68613882	0,37227764	0,638929343
SumaTotalVencido36M	0,68573955	0,3714791	6,10268E-05
PromedioTotalVencidoSinHip36M	0,6851358	0,37027161	0,638734398
PromedioTotalVencido36M	0,68512102	0,37024203	3,61037E-05
SumaDiasAtrasoTotales36M	0,68477066	0,36954133	0,340260174
PromedioDiasAtrasoTotales36M	0,68477066	0,36954133	0,340068821
mayorValorVencido	0,68244304	0,36488609	0,000478566
MaxTotalVencidoSinHip36M	0,682144	0,36428801	0,761965614
MaxDiasAtrasoTotales36M	0,68181994	0,36363989	0,169071606
SumaTotalVencidoSinHip24M	0,6809983	0,3619966	0,450215523
PromedioTotalVencido24M	0,68088536	0,36177071	0,024816275
SumaTotalVencido24M	0,6808226	0,36164521	0,024814695
PromedioTotalVencidoSinHip24M	0,68050072	0,36100144	0,45028926
MaxTotalVencido36M	0,68029773	0,36059547	0,539135622
SumaSaldoVencidoConsumo36M	0,67966475	0,35932949	0,133955065
MaxTotalVencidoSinHip24M	0,67896939	0,35793879	0,569580389
PromedioSaldoVencidoConsumo36M	0,6786586	0,3573172	0,133888553
SumaDiasAtrasoTotales24M	0,67781623	0,35563246	0,422109791
PromedioDiasAtrasoTotales24M	0,67781623	0,35563246	0,421668432
MaxTotalVencido24M	0,67769198	0,35538397	0,011983645

Cuadro C.1: Tabla de p-valores del primer proceso

Variable	AUC	GINI	Pvalue
MaxDiasAtrasoTotales24M	0,67619735	0,35239469	0,005130637
SumaSaldoVencidoConsumo24M	0,67588752	0,35177505	0,080494459
MaxSaldoVencidoConsumo36M	0,67568542	0,35137083	0,228861144
PromedioSaldoVencidoConsumo24M	0,67538992	0,35077983	0,080510519
SumaTotalVencido12M	0,67419294	0,34838589	0,125290497
MaxSaldoVencidoConsumo24M	0,6739047	0,3478094	0,494597332
PromedioTotalVencido12M	0,67366299	0,34732598	0,125299203
SumaTotalVencidoSinHip12M	0,67323661	0,34647322	0,429847157
MaxTotalVencido12M	0,67298221	0,34596441	0,011366138
MaxTotalVencidoSinHip12M	0,67241798	0,34483596	0,803530332
PromedioTotalVencidoSinHip12M	0,6719841	0,34396819	0,429746931
SumaDiasAtrasoTotales12M	0,66969582	0,33939163	0,793929787
PromedioDiasAtrasoTotales12M	0,66969582	0,33939163	0,793390076
MaxDiasAtrasoTotales12M	0,66916657	0,33833313	0,178210177
bancoPeorCalf	0,66790832	0,33581665	0,106404938
SumaSaldoVencidoConsumo12M	0,66615321	0,33230642	0,578125035
MaxSaldoVencidoConsumo12M	0,66530676	0,33061352	0,88824157
PromedioSaldoVencidoConsumo12M	0,66489565	0,3297913	0,578013558
valorPeorCalf	0,66194608	0,32389216	0,000307335
PromedioTotalVencido9M	0,66104416	0,32208831	0,024371842
SumaTotalVencido9M	0,6610435	0,32208701	0,024371457
SumaTotalVencidorSinHip9M	0,66079519	0,32159038	0,030095753
SumaProductosSaldoVencido36M	0,66032479	0,32064957	0,225496757
PromedioProductosSaldoVencido36M	0,66032479	0,32064957	0,272728816
MaxTotalVencidoSinHip9M	0,66009982	0,32019963	0,424214634
PromedioTotalVencidoSinHip9M	0,66007036	0,32014072	0,030113156
MaxTotalVencido9M	0,66006357	0,32012713	0,505597584
SumaDiasAtrasoTotales9M	0,6579625	0,31592499	0,05897066
PromedioDiasAtrasoTotales9M	0,6579625	0,31592499	0,059180313
MaxDiasAtrasoTotales9M	0,65762078	0,31524157	0,004154295
RazonTotalVencidoSinHip9M_36M	0,65483557	0,30967114	0,142508347

Cuadro C.2: Continuación tabla de p-valores del primer proceso

Variable	AUC	GINI	Pvalue
RazonTotalVencido9M_36M	0,65442716	0,30885433	0,005096395
SumaSaldoVencidoConsumo9M	0,65355725	0,30711451	0,006883853
RazonDiasAtrasoTotales9M_36M	0,65302171	0,30604342	0,014865936
MaxSaldoVencidoConsumo9M	0,65286076	0,30572152	0,459408856
PromedioSaldoVencidoConsumo9M	0,65282983	0,30565965	0,006888238
PromedioTotalVencido6M	0,65265709	0,30531418	0,000114251
SumaTotalVencido6M	0,65265708	0,30531415	0,000114273
SumaProductosSaldoVencido24M	0,65247754	0,30495509	0,007419668
PromedioProductosSaldoVencido24M	0,65247754	0,30495509	0,006294137
MaxTotalVencido6M	0,65198999	0,30397998	0,060870904
MaxProductosSaldoVencido36M	0,65130532	0,30261063	0,563964949
SumaTotalVencidoSinHip6M	0,65107807	0,30215615	0,001651813
MaxTotalVencidoSinHip6M	0,6505743	0,30114861	0,675208701
PromedioTotalVencidoSinHip6M	0,65034484	0,30068968	0,001652504
SumaDiasAtrasoTotales6M	0,64896512	0,29793024	0,037662281
PromedioDiasAtrasoTotales6M	0,64896512	0,29793024	0,037752257
RazonTotalVencido6M_12M	0,64863754	0,29727509	0,006364516
MaxDiasAtrasoTotales6M	0,64862717	0,29725433	0,007256755
MaxProductosSaldoVencido24M	0,64800981	0,29601962	0,026410903
RazonSaldoVencidoConsumo9M_36M	0,64749073	0,29498147	0,19229669
RazonTotalVencido6M_36M	0,64734882	0,29469765	3,23047E-05
RazonTotalVencidoSinHip6M_36M	0,6452667	0,29053339	0,370740314
RazonTotalVencidoSinHip6M_12M	0,64522735	0,29045469	0,108041478
RazonDiasAtrasoTotales6M_12M	0,64483723	0,28967446	5,02469E-05
SumaProductosSaldoVencido12M	0,64379491	0,28758982	0,5821968
PromedioProductosSaldoVencido12M	0,64379491	0,28758982	0,611165604
SumaSaldoVencidoConsumo6M	0,64370124	0,28740248	0,008974471
RazonDiasAtrasoTotales6M_36M	0,6432143	0,28642859	1,1589E-05
MaxSaldoVencidoConsumo6M	0,64321059	0,28642119	0,660076671
PromedioSaldoVencidoConsumo6M	0,64296592	0,28593185	0,008977693
MaxProductosSaldoVencido12M	0,64231247	0,28462495	0,143184416
usoCupoPorcentaje	0,64228598	0,28457196	0,627403539

Cuadro C.3: Continuación tabla de p-valores del primer proceso

Variable	AUC	GINI	Pvalue
peorCalfActual	0,63908503	0,27817006	0,006360649
RazonSaldoVencidoConsumo6M_12M	0,63792399	0,27584798	0,372472124
RazonSaldoVencidoConsumo6M_36M	0,637859	0,27571799	0,39964009
SumaProductosSaldoVencido9M	0,63375944	0,26751888	6,30385E-05
PromedioProductosSaldoVencido9M	0,63375944	0,26751888	4,44307E-05
MaxProductosSaldoVencido9M	0,63288388	0,26576777	0,079129771
RazonProductosSaldoVencido9M_36M	0,63284967	0,26569933	0,023533605
PromedioTotalVencido3M	0,62837485	0,2567497	0,290181555
SumaTotalVencido3M	0,62837478	0,25674956	0,290172201
MaxTotalVencido3M	0,62807067	0,25614134	0,342934216
Ciudad	0,62567363	0,25134725	0,520977212
SumaTotalVencidoSinHip3M	0,6253722	0,25074439	0,354845804
RazonTotalVencido3M_6M	0,62533992	0,25067984	0,042274337
MaxTotalVencidoSinHip3M	0,62512115	0,25024231	0,927583167
SumaDiasAtrasoTotales3M	0,62481368	0,24962735	2,46698E-05

Cuadro C.4: Continuación tabla de p-valores del primer proceso

Apéndice D

Tabla de p-valores del segundo proceso

Variable	AUC	GINI	Pvalue
bancoPeorCalf	0,66790832	0,33581665	0.995450
Ciudad	0,62567363	0,25134725	1.09e-07
MaxDiasAtrasoTotales12M	0,66916657	0,33833313	3.01e-05
MaxDiasAtrasoTotales24M	0,67619735	0,35239469	0.047632
MaxDiasAtrasoTotales36M	0,68181994	0,36363989	0.029214
MaxDiasAtrasoTotales9M	0,65762078	0,31524157	0.837733
MaxProductosSaldoVencido36M	0,65130532	0,30261063	0.002408
MaxSaldoVencidoConsumo12M	0,66530676	0,33061352	0.998163
MaxSaldoVencidoConsumo24M	0,6739047	0,3478094	0.000237
MaxSaldoVencidoConsumo36M	0,67568542	0,35137083	0.989965
MaxSaldoVencidoConsumo6M	0,64321059	0,28642119	0.106789
MaxSaldoVencidoConsumo9M	0,65286076	0,30572152	0.387966
MaxTotalVencido12M	0,67298221	0,34596441	0.078867
MaxTotalVencido24M	0,67769198	0,35538397	0.981735
MaxTotalVencido36M	0,68029773	0,36059547	3.12e-06
MaxTotalVencido6M	0,65198999	0,30397998	0.988360
MaxTotalVencido9M	0,66006357	0,32012713	3.41e-08
MaxTotalVencidoSinHip12M	0,67241798	0,34483596	0.009291
MaxTotalVencidoSinHip24M	0,67896939	0,35793879	3.61e-09
MaxTotalVencidoSinHip36M	0,682144	0,36428801	0.001538

Cuadro D.1: Tabla de p-valores del segundo proceso

Variable	AUC	GINI	Pvalue
MaxTotalVencidoSinHip3M	0,62512115	0,25024231	0.000408
MaxTotalVencidoSinHip6M	0,6505743	0,30114861	0.011846
MaxTotalVencidoSinHip9M	0,66009982	0,32019963	0.993330
mayorValorVencido	0,68244304	0,36488609	0.995453
peorCalfActual	0,63908503	0,27817006	0.989456
peorCalfHistorica	0,7057998	0,41159961	1.53e-05
PromedioDiasAtrasoTotales12M	0,66969582	0,33939163	5.55e-07
PromedioDiasAtrasoTotales24M	0,67781623	0,35563246	0.419224
PromedioDiasAtrasoTotales36M	0,68477066	0,36954133	0.991521
PromedioDiasAtrasoTotales6M	0,64896512	0,29793024	0.174467
PromedioDiasAtrasoTotales9M	0,6579625	0,31592499	0.992281
PromedioProductosSaldoVencido12M	0,64379491	0,28758982	0.035429
PromedioProductosSaldoVencido24M	0,65247754	0,30495509	0.102398
PromedioProductosSaldoVencido36M	0,66032479	0,32064957	0.789034
PromedioSaldoVencidoConsumo12M	0,66489565	0,3297913	0.712405
PromedioSaldoVencidoConsumo24M	0,67538992	0,35077983	0.117674
PromedioSaldoVencidoConsumo36M	0,6786586	0,3573172	0.001209
PromedioSaldoVencidoConsumo9M	0,65282983	0,30565965	0.993460
PromedioTotalVencido12M	0,67366299	0,34732598	0.983196
PromedioTotalVencido24M	0,68088536	0,36177071	0.999695
PromedioTotalVencido36M	0,68512102	0,37024203	0.993430
PromedioTotalVencido3M	0,62837485	0,2567497	0.061838
PromedioTotalVencido6M	0,65265709	0,30531418	0.995160
PromedioTotalVencido9M	0,66104416	0,32208831	0.001394
PromedioTotalVencidoSinHip12M	0,6719841	0,34396819	0.995195
PromedioTotalVencidoSinHip24M	0,68050072	0,36100144	1.78e-07
PromedioTotalVencidoSinHip36M	0,6851358	0,37027161	0.742618
PromedioTotalVencidoSinHip9M	0,66007036	0,32014072	0.995450
RazonDiasAtrasoTotales6M_36M	0,6432143	0,28642859	0.999873
RazonProductosSaldoVencido9M_36M	0,63284967	0,26569933	0.259313
RazonSaldoVencidoConsumo6M_36M	0,637859	0,27571799	0.015058
RazonSaldoVencidoConsumo9M_36M	0,64749073	0,29498147	0.000198
RazonTotalVencido3M_6M	0,62533992	0,25067984	4.45e-06

Cuadro D.2: Continuación tabla de p-valores del segundo proceso

Variable	AUC	GINI	Pvalue
RazonTotalVencido6M_12M	0,64863754	0,29727509	0.501435
RazonTotalVencido6M_36M	0,64734882	0,29469765	0.048214
RazonTotalVencido9M_36M	0,65442716	0,30885433	0.994967
RazonTotalVencidoSinHip6M_36M	0,6452667	0,29053339	0.140830
RazonTotalVencidoSinHip9M_36M	0,65483557	0,30967114	0.994609
SumaDiasAtrasoTotales12M	0,66969582	0,33939163	0.093793
SumaDiasAtrasoTotales24M	0,67781623	0,35563246	0.994533
SumaDiasAtrasoTotales36M	0,68477066	0,36954133	0.995541
SumaDiasAtrasoTotales3M	0,62481368	0,24962735	0.995554
SumaDiasAtrasoTotales6M	0,64896512	0,29793024	0.841867
SumaDiasAtrasoTotales9M	0,6579625	0,31592499	0.984228
SumaProductosSaldoVencido12M	0,64379491	0,28758982	0.006995
SumaProductosSaldoVencido24M	0,65247754	0,30495509	0.992017
SumaProductosSaldoVencido36M	0,66032479	0,32064957	0.995842
SumaSaldoVencidoConsumo12M	0,66615321	0,33230642	0.995029
SumaSaldoVencidoConsumo24M	0,67588752	0,35177505	0.992182
SumaSaldoVencidoConsumo36M	0,67966475	0,35932949	0.976526
SumaSaldoVencidoConsumo6M	0,64370124	0,28740248	0.993307
SumaSaldoVencidoConsumo9M	0,65355725	0,30711451	0.988979
SumaTotalVencido12M	0,67419294	0,34838589	0.992002
SumaTotalVencido24M	0,6808226	0,36164521	0.985927
SumaTotalVencido36M	0,68573955	0,3714791	0.029759
SumaTotalVencido3M	0,62837478	0,25674956	0.999679
SumaTotalVencido6M	0,65265708	0,30531415	0.989180
SumaTotalVencido9M	0,6610435	0,32208701	0.987611
SumaTotalVencidorSinHip9M	0,66079519	0,32159038	0.472024
SumaTotalVencidoSinHip12M	0,67323661	0,34647322	0.141180
SumaTotalVencidoSinHip24M	0,6809983	0,3619966	1.08e-05
SumaTotalVencidoSinHip36M	0,68613882	0,37227764	0.018809
SumaTotalVencidoSinHip3M	0,6253722	0,25074439	0.378999
usoCupoPorcentaje	0,64228598	0,28457196	0.986642
valorPeorCalf	0,66194608	0,32389216	1.15e-07

Cuadro D.3: Continuación tabla de p-valores del segundo proceso

Apéndice E

Tabla de p-valores del tercer proceso

Variable	AUC	GINI	Pvalue
peorCalfActualV1	0,56471965	0,1294393	0.00000118
MinTotalSaldoActualTC9M_Conyuge	0,76814766	0,53629531	0.0000122
RazonTotalDeudaSinTC3M_9M	0,8030793	0,6061586	0.0000188
usoCupoPorcentaje	0,82114299	0,64228598	0.0000885
peorCalfActualO3	0,56436615	0,1287323	0.000665
AcrescimoTotalDeudaSinTC6M	0,74433023	0,48866046	0.000742
peorCalfHistoricaO3	0,55841513	0,11683027	0.001186
RazonCantidadOperacionesSinTC3M_6M	0,80047816	0,60095631	0.001531
PromedioTotalDeudaRotativos3M	0,76207836	0,52415671	0.002158
cantPagosConse12M	0,77791128	0,55582257	0.002747
RazonTotalVencido3M_9M	0,8122643	0,62452859	0.003214
peorCalfActualY1	0,55490765	0,1098153	0.004006
RazonCantidadProductos6M_12M	0,78633536	0,57267072	0.004128
RazonComTotalVencido9M_36M	0,78231849	0,56463699	0.004591
MaxTotalCarteraCastigadaSinHip36M	0,74642568	0,49285136	0.005233
PromedioNentidadesSinHip9M	0,7692208	0,53844159	0.011882
MaxCuotaEstimada12M	0,76962927	0,53925855	0.012132
bancoPeorCalfBELCORP	0,54991965	0,09983929	0.012541
RazonCantidadProductos9M_36M	0,79518919	0,59037839	0.013311
SumaProductosAlDia24M	0,76450014	0,52900028	0.019833
RazonTotalDeudaSinTC6M_36M	0,80656181	0,61312362	0.019839

Cuadro E.1: Tabla de p-valores del tercer proceso

Variable	AUC	GINI	Pvalue
RazonCantProdRef6M_36M	0,75906386	0,51812772	0.019931
RazonComTotalPorVencer6M_36M	0,79161111	0,58322223	0.020943
PromedioComTotalVencido36M	0,7892113	0,57842259	0.020958
valorPeorCalf	0,83097304	0,66194608	0.021238
peorCalfActualO2	0,54494615	0,0898923	0.023791
RazonSaldoVencidoTarjetaDeCredito9M_36M	0,78468354	0,56936708	0.026147
PromedioProductosSaldoVencido36M	0,83016239	0,66032479	0.027182
SumaCantidadProductos24M	0,75657038	0,51314076	0.029027
tiempoInhabilitacion	0,75552623	0,51105247	0.031229
peorCalfActualV2	0,54086765	0,0817353	0.034261
PromedioNumeroMesesCentralDeRiesgos36M	0,77843614	0,55687228	0.035372
MinTotalDeudaRotativos3M	0,76205847	0,52411695	0.036128
RazonTotalDeudaSinTC6M_36M__1	0,75002163	0,50004327	0.037017
RazonProductosSaldoVencido6M_12M	0,81193331	0,62386661	0.037819
MaxProductosAlDia3M	0,76723999	0,53447998	0.038423
PromedioTotalVencido36M	0,84256051	0,68512102	0.039602
SumaTotalVencido36M	0,84286977	0,68573955	0.040985
RazonNumAcreedoresSICOM3M_6M	0,79943045	0,5988609	0.041779
SumaProductosSaldoVencido12M	0,82189745	0,64379491	0.046959
RazonSaldoTC3M_6M	0,76426689	0,52853379	0.048893
MaxSaldoAlDiaTarjetaDeCredito24M	0,76138475	0,52276949	0.049087
MaxTotalSaldoActualTC3M	0,75274527	0,50549053	0.049281
peorCalfHistoricaY2	0,5399162	0,0798324	0.049841
PromedioDiasAtrasoTotales36M	0,84238533	0,68477066	0.053754
RazonSaldoAlDiaTarjetaDeCredito9M_36M	0,77708601	0,55417201	0.055276
MinSaldoAlDiaTarjetaDeCredito3M	0,74875144	0,49750288	0.061632
RazonProductosAlDia9M_36M	0,78320932	0,56641864	0.070341
MaxSaldoTC12M	0,75704484	0,51408968	0.075853
RazonSaldoAlDiaTarjetaDeCredito3M_6M	0,76487635	0,5297527	0.080907
SumaTotalDeudaSinTC24M	0,73336245	0,46672491	0.087496
MinComSaldo12M	0,76024655	0,52049309	0.0875
RazonComSaldo3M_6M	0,79957359	0,59914719	0.103318
MaxSaldoVencidoMicrocredito6M	0,75513627	0,51027253	0.110653

Cuadro E.2: Continuación tabla de p-valores del tercer proceso

Variable	AUC	GINI	Pvalue
MaxTotalVencido36M	0,84014887	0,68029773	0.114086
cupoMaxTarjetas	0,79423006	0,58846013	0.122023
PromedioSerTotalPorVencer6M	0,75002031	0,50004061	0.125437
RazonCuotaEstimada6M_12M	0,78931174	0,57862349	0.13134
RazonTotalSaldoActualTC6M_12M	0,76549497	0,53098994	0.134341
SumaVFAC_12	0,79002321	0,58004642	0.151776
peorCalfActualR	0,53991635	0,0798327	0.153135
RazonTotalVencidoSinHip9M_36M	0,82741779	0,65483557	0.159066
RazonDiasAtrasoTotales3M_6M	0,81134923	0,62269847	0.165916
SumaDiasAtrasoTotales3M	0,81240684	0,62481368	0.176116
RazonSaldoAlDiaMicrocredito9M_36M	0,76260353	0,52520705	0.186673
RazonComSaldo9M_36M	0,79759925	0,5951985	0.197783
MaxCantidadProductos3M	0,78317861	0,56635722	0.231641
DecrescimoSaldoAlDiaConsumo9M	0,76028897	0,52057794	0.240255
RazonTotalPorVencerSinHip6M_36M	0,78620152	0,57240304	0.250805
MaxDiasAtrasoTotales36M	0,84090997	0,68181994	0.260914
PromedioTotalSaldoActualTC12M_Conyuge	0,76498577	0,52997155	0.261488
MaxSaldoVencidoTarjetaDeCredito12M	0,79074528	0,58149057	0.261998
MinSaldoAlDia9M	0,75311581	0,50623162	0.285024
MaxSaldoVencidoConsumo24M	0,83695235	0,6739047	0.291068
MaxTotalVencidoSinHip24M	0,8394847	0,67896939	0.306603
RazonTotalSaldoActualTC6M_12M_Conyuge	0,75336614	0,50673229	0.312502
DecrescimoComSaldo3M	0,77025065	0,5405013	0.315056
RazonTotalSaldoActualTC9M_36M	0,75927376	0,51854753	0.353615
SumaMicTotalVencido6M	0,7520124	0,5040248	0.372047
PromedioTCAbiertas24M	0,7676093	0,5352186	0.372401
RazonDiasAtrasoTotales9M_36M	0,82651086	0,65302171	0.404006
antiguedad	0,7953219	0,59064381	0.405925
bancoPeorCalfPACIFICO	0,53496712	0,06993423	0.407668
RazonMicTotalPorVencer3M_9M	0,7515079	0,5030158	0.41691
fechaPeorCalf	0,77338952	0,54677904	0.442945
mayorValorVencido	0,84122152	0,68244304	0.445598
RazonTotalVencido6M_36M	0,82367441	0,64734882	0.471003

Cuadro E.3: Continuación tabla de p-valores del tercer proceso

Variable	AUC	GINI	Pvalue
DecrescimoTotalDeudaSinHip9M	0,75806629	0,51613258	0.472989
AcrescimoComSaldo6M	0,76563993	0,53127986	0.479791
MaxSaldoVencidoTarjetaDeCredito36M	0,79436079	0,58872159	0.487402
RazonTotalDeudaRotativos9M_36M	0,76883629	0,53767258	0.48829
Rentacomprobada	0,78921321	0,57842642	0.496445
RazonTotalDeuda6M_36M	0,79546051	0,59092102	0.506928
peorCalfActualT	0,52711615	0,0542323	0.509253
RazonSaldoAlDia6M_36M	0,79650067	0,59300133	0.521373
RazonTotalCarteraCastigada3M_6M	0,75110777	0,50221554	0.526786
MaxComSaldo3M	0,80090964	0,60181927	0.533274
MinSerSaldo3M	0,74970325	0,49940651	0.544046
RazonTotalVencido9M_36M	0,82721358	0,65442716	0.547256
SumaCantProdRef9M	0,75898591	0,51797182	0.560132
RazonNumeroDeAcreedoresSBS3M_6M	0,76807007	0,53614014	0.608769
RazonSaldoAlDiaConsumo6M_36M	0,7868054	0,5736108	0.614606
PromedioTotalSaldoActualTC12M	0,74486161	0,48972323	0.63119
SumaTotalVencidoSinHip12M	0,83661831	0,67323661	0.796758
RazonTotalVencidoSinHip3M_6M	0,81128603	0,62257206	0.796985
SumaCantProdRef12M	0,76007629	0,52015258	0.799143
MaxTotalVencidoSinHip36M	0,841072	0,682144	0.810046

Cuadro E.4: Continuación tabla de p-valores del tercer proceso