ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE CIENCIAS

MODELOS ESTADÍSTICOS PARA LA ESTIMACIÓN DE LA CAPACIDAD DE PAGO DE PERSONAS NATURALES TARJETAHABIENTES CON INFORMACIÓN EN EL SISTEMA DE REGISTRO DE DATOS CREDITICIOS

TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE INGENIERO MATEMÁTICO

FARHAD KHOSSRO GHADIRI AYALA

farhadkg.fkg@outlook.com

DIRECTOR: MSC.DIEGO PAÚL HUARACA SAGÑAY

diego.huaracas@epn.edu.ec

CODIRECTOR: MSC.MENTHOR OSWALDO URVINA MAYORGA

menthor.urvina@epn.edu.ec

DMQ, AGOSTO 2023

CERTIFICACIONES

Yo, FARHAD KHOSSRO GHADIRI AYALA, declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

Farhad Khossro Ghadiri Ayala

Certifico que el presente trabajo de integración curricular fue desarrollado por Farhad Khossro Ghadiri Ayala, bajo mi supervisión.

MSc.Diego Paúl Huaraca Sagñay
DIRECTOR

MSc.Menthor Oswaldo Urvina Mayorga
CODIRECTOR

DECLARACIÓN DE AUTORÍA

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el(los) producto(s) resultante(s) del mismo, es(son) público(s) y estará(n) a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.

Farhad Khossro Ghadiri Ayala

MSc.Diego Paúl Huaraca Sagñay

MSc.Menthor Oswaldo Urvina Mayorga

AGRADECIMIENTO

Deseo expresar mi profundo agradecimiento a mis padres, Patricia y Farhad, por toda su paciencia, amor, cariño y apoyo que me han brindado durante toda mi vida. A mi querida hermana Nilufar, por todas las vivencias que hemos tenido juntos desde niños. Extiendo mi gratitud a mis tías, Luisa y Lola, cuyo apoyo incondicional ha sido un pilar fundamental para toda nuestra familia a lo largo de los años.

A mi profesor Diego Huaraca, que me ha transmitido valiosos conocimientos y me ha brindado parte de su tiempo para terminar este trabajo. A mis profesores de la facultad, pues ellos me han transmitido los conocimientos, la pasión y el amor que tengo por las matemáticas. A mis compañeros que me acompañaron durante toda la carrera, pues todos me han enseñado cosas valiosas que me servirán para toda la vida personal y profesional.

Finalmente a mis amigos, quienes se han convertido en la familia que he elegido a lo largo de esta travesía. Su apoyo y amistad han sido un bálsamo curativo en los momentos de desafío.

RESUMEN

El presente trabajo de integración curricular tiene por objetivo desarrollar modelos estadísticos paramétricos y no paramétricos para la estimación adecuada de la capacidad de pago de pago de personas naturales bancarizadas con información en el sistema de registros crediticio. Las metodologías utilizadas para las estimaciones de los ingresos se realizaron mediante los modelos: Regresión Lineal Múltiple (RLM), Random Forest (RF), Gradient Boosting Machine (GBM) y Extreme Gradient Boosting (xgBoost). La base de datos de estudio es una muestra representativa y aleatoria. Para estudiar la base de datos, la misma se dividió en tres grupos para mejorar la calidad de las predicciones, considerando un modelo para cada grupo. Para seleccionar las variables más representativas que ingresen a estos modelos, se usaron las técnicas de Kolmogorov-Smirnov (KS) para variables cuantitativas y la técnica de Valor de Información (VI) para variables cualitativas; además, la técnica del remuestreo resultó ser una herramienta formidable para mejorar las predicciones de los modelos en comparación con los modelos iniciales. Los resultados de este estudio fueron favorables de acuerdo con los objetivos planteados. Se mejoró la estimación de los ingresos con la técnica del remuestreo en comparación con el modelo inicial, en particular se mejoró la estimación en las colas de las distribuciones del ingreso real (sujetos con ingresos muy bajos y sujetos con ingresos muy altos).

Palabras clave: **modelos estadísticos, estimación de ingresos, personas naturales bancarizadas, metodologías, colas de distribuciones.**

ABSTRACT

This curricular integration project aims to develop parametric and non-parametric statistical models for the accurate estimation of repayment capacity among banked individuals using information from the credit registry system. Income estimation methodologies were executed using the following models: Multiple Linear Regression (MLR), Random Forest (RF), Gradient Boosting Machine (GBM), and Extreme Gradient Boosting (xgBoost). In the present project, the database consists of a random and representative sample. To study the database, it was divided into three groups to enhance prediction quality, considering a model for each group. To select the most representative variables for inclusion in these models, Kolmogorov-Smirnov (KS) techniques were applied for quantitative variables, and the Value of Information (VI) technique was used for qualitative variables. Additionally, the resampling technique was a potent tool for improving model predictions, compared to the initial models. The outcomes of this study were favorable according to the objectives. Income estimation was enhanced using the resampling technique compared to the initial model, particularly improving estimation in the tails of real income distributions (individuals with very low and very high incomes).

Keywords: **statistical models, income estimation, banked individuals, methodologies, distribution tails.**

Índice general

Índice de figuras

Capítulo 1

Descripción del componente desarrollado

1.1. Descripción del proyecto

La estimación de la capacidad de pago como una medida de liquidez y solvencia que puede presentar una persona natural para hacer frente a sus obligaciones crediticias, constituye un soporte técnico importante para las entidades financieras a la hora de asignar los montos y los cupos en los diferentes tipos de crédito a colocar en el mercado, debido a que una estimación adecuada permitirá mitigar la posible existencia de escenarios de sobreendeudamiento, lo cual ocurre cuando el monto de la deuda es superior al patrimonio del cliente, por otra parte, en el área de cobranzas la estimación de la capacidad de pago permitirá priorizar las acciones aumentando los índices de recupero.

El estimador de la capacidad de pago es un modelo analítico que se construye mediante la utilización de técnicas estadísticas de regresión como son los modelos lineales generalizados (GLM) y con técnicas de Machine Learning (ML), dichas técnicas permiten estimar la capacidad de pago de una persona natural a partir de los conjuntos de información referentes a datos crediticios actuales e históricos, información sociodemográfica e información socioeconómica.

Considerando las fuentes de información, en particular la información crediticia que es la de mayor importancia en este proyecto, se hace nece-

saria la segmentación de la población en tres grupos distintos:

1. Población Tarjetahabiente: Formada por los sujetos que a la fecha de consulta cuentan con un cupo asignado de Tarjeta de Crédito y además cuentan con información crediticia histórica (al menos 3 meses anteriores a la fecha de consulta).
2. Población Bancarizada: Formada por los sujetos que a la fecha de consulta no cuentan con Tarjetas de Crédito pero tienen información crediticia histórica (al menos 3 meses anteriores a la fecha de consulta).
3. Población No Bancarizada: Formada por los sujetos que presentan menos de 3 meses de información crediticia histórica.

En este proyecto nos centraremos en estimar modelos estadísticos que se ajusten lo mejor posible a la capacidad de pago de las personas naturales en la **Población Tarjetahabiente**.

1.2. Objetivo general

Construir modelos analíticos que permitan estimar la capacidad de pago de una persona natural para hacer frente a sus obligaciones crediticias.

1.3. Objetivos específicos

1. Construir indicadores que permitan estimar la liquidez disponible de las personas naturales a fin de determinar los cupos de crédito adecuados, mitigando la probabilidad de existencia de sobreendeudamiento.
2. Obtener modelos analíticos que minimicen la sobre-estimación de ingresos en clientes con capacidades de pago bajas y la sub-estimación de ingresos en clientes con capacidades de pago altas.
3. Evaluar modelos paramétricos y no paramétricos dependiendo de la distribución de los ingresos en las subpoblaciones y del conjunto de

variables crediticias disponibles, con el fin de maximizar la exactitud y minimizar el error.

1.4. Alcance

Para alcanzar nuestro objetivo principal, es necesario adquirir un conocimiento en modelos lineales generalizados y modelos de regresión no paramétricos, así que se comenzará por estudiar estos tipos de modelos. Se consolidará, se analizará y se depurará la información crediticia, sociodemográfica y socioeconómica disponible para el desarrollo del proyecto (información con fecha de corte diciembre de 2021).

Se calcularán al menos 2 medidas de divergencia en función de cada tipo de variable de manera que se pueda generar un ranking entre las variables candidatas a formar parte de cada uno de los modelos.

Posteriormente, se entrenarán al menos 3 diferentes tipos de modelos que permitan estimar la capacidad de pago de una persona natural y se evaluará el poder predictivo y el error de ajuste de cada uno de los modelos candidatos en la población de estudio.

Finalmente, se realizará una ejecución del modelo ganador sobre una base de clientes más actualizada con la finalidad de medir la estabilidad del modelo.

Capítulo 2

Marco Teórico

En este capítulo se explica la base matemática de la metodología utilizada en el proyecto. Primero se explican los algoritmos de modelamiento predictivo: Regresión Lineal Múltiple, Random Forest, Gradient Boosting Machine, xgBoost. Después se explican los test estadísticos, junto con los estadísticos más útiles, que se usan para evaluar la bondad de ajuste del mejor modelo escogido.

2.1. Modelos estadísticos

2.1.1. Modelos de Aprendizaje Supervisado

El aprendizaje supervisado es una categoría de algoritmos de aprendizaje automático donde el modelo se entrena utilizando un conjunto de datos que posee una variable de interés, Y.

Cada muestra del conjunto de entrenamiento consta de variables explicativas y por lo menos una variable dependiente. El objetivo es que el modelo asocie a los individuos de la muestra, hacia los valores observados de la muestra para la variable dependiente y así poder hacer predicciones precisas en individuos que no formaron parte de la muestra para calcular el modelo. Ejemplos de algoritmos:

1. Regresión Lineal Múltiple

2. Regresión Logística
3. Árboles de decisión
4. Random Forest
5. Gradient Boosting Machine
6. xgBoost

2.1.2. Modelos de Aprendizaje No Supervisado

El aprendizaje no supervisado es otra categoría de algoritmos de aprendizaje automático donde el modelo se entrena con un conjunto de datos no etiquetado. En este caso, el modelo intenta encontrar patrones o estructuras ocultas en los datos sin tener información previa sobre las salidas esperadas, Y. El objetivo principal del aprendizaje no supervisado es explorar la estructura inherente en los datos para obtener información valiosa. Ejemplos de algoritmos:

1. K-Means
2. Análisis de Componentes Principales (PCA)

2.1.3. Modelos Paramétricos

Los modelos paramétricos son aquellos que tienen un número fijo de parámetros y hacen suposiciones específicas sobre la distribución de los errores en un modelo poblacional. Una vez que se han estimado los parámetros utilizando el conjunto de entrenamiento, el modelo está completamente definido y puede ser utilizado para hacer predicciones en nuevos datos. Ejemplos:

1. Regresión Lineal Múltiple
2. Regresión logística

2.1.4. Modelos No Paramétricos

Los modelos no paramétricos, no hacen suposiciones específicas sobre la distribución de los datos ni de los errores, y por tanto no tienen parámetros que se relacionen con la distribución de los datos ni de los errores. Esto les permite ser más flexibles y capaces de ajustarse mejor a datos complejos y no lineales. Ejemplos:

1. Árboles de decisión
2. Random Forest
3. Gradient Boosting Machine
4. xgBoost

2.2. Modelo de Regresión Múltiple

Este es el único modelo paramétrico que se utiliza para estimar los resultados del proyecto. El propósito de implementar dicho modelo fue establecer una comparación entre sus resultados y los obtenidos mediante modelos no paramétricos.

Es importante tener en cuenta que al ser un modelo paramétrico es necesario que se cumplan los siete supuestos (hipótesis) del modelo lineal clásico, no solo para los datos de la muestra, sino también con respecto a la distribución de los errores. En esta sección no se explicitan los detalles matemáticos de este modelo, dado que se lo estudia ampliamente a lo largo de la carrera y es fácil de encontrar en cualquier bibliografía.

En general, algunas de las desventajas que tiene el modelo de regresión múltiple son:

1. *Restricciones en la forma funcional:* El modelo de regresión múltiple asume una relación lineal entre las variables independientes y la variable dependiente. Sin embargo, en la realidad, las relaciones son en su mayoría no lineales. Esto puede llevar a un mal ajuste cuando los datos siguen patrones no son lineales.

2. *Sensibilidad a outliers (valores atípicos):* Los modelos de regresión múltiple pueden ser sensibles a valores atípicos en los datos. Los outliers pueden influir significativamente en la estimación de los coeficientes y afectar la calidad del ajuste del modelo.

3. *Multicolinealidad:* Si existe una alta correlación entre algunas de las variables independientes, se puede presentar un problema de multicolinealidad. La multicolinealidad puede hacer que las estimaciones de los coeficientes sean inestables y difíciles de interpretar correctamente.

4. *Sobreajuste (overfitting) o subajuste (underfitting):* Los modelos de regresión múltiple paramétricos pueden tener una alta tendencia a sobreajustar los datos de entrenamiento. Es decir, el modelo puede capturar ruido y características irrelevantes en los datos, lo que lleva a una mala generalización a nuevos datos no utilizados en la base de entrenamiento.

5. *Dificultad para modelar relaciones no lineales:* Los modelos de regresión múltiple asumen una relación lineal entre las variables. Modelar relaciones no lineales requiere transformar manualmente las variables, lo que puede ser complicado y consumir mucho tiempo. Además, puede ser que no se capture información sobre patrones que no se pueden ver fácilmente en los gráficos.

6. *No apto para datos complejos:* En problemas con muchas variables independientes y relaciones no lineales complejas, los modelos de regresión múltiple paramétricos pueden no ser lo suficientemente flexibles como para capturar la información relevante de los datos.

7. *Requiere supuestos sobre los errores:* Los modelos de regresión múltiple paramétricos asumen que los errores tienen una distribución normal con media cero y varianza constante. Si estos supuestos no se cumplen, las inferencias y predicciones del modelo pueden ser incorrectas.

Aunque los modelos de regresión múltiple paramétricos tienen estas desventajas, también tienen ventajas, como su interpretabilidad (que es ampliamente estudiado en la Econometría).

Sin embargo, en situaciones donde los datos son altamente no lineales o cuando se desconoce la verdadera forma funcional de la relación entre las variables, los modelos no paramétricos pueden ser más adecuados.

2.3. Random Forest

El Random Forest es un algoritmo de aprendizaje supervisado no paramétrico, que genera varios árboles de decisión sobre un conjunto de datos de entrenamiento, y luego los resultados obtenidos para cada árbol se combinan para obtener una estimación más cercana al valor verdadero.

Para entender de mejor forma lo que realiza este algoritmo, a continuación se explican algunos conceptos relacionados, ya que fueron de importancia para después utilizar correctamente las librerías en R.

2.3.1. Algoritmo de Árbol de decisión (AD):

Es de mucha importancia conocer el concepto de *Árbol de Decisión*, ya que este modelo es la base para definir los tres métodos no paramétricos que se utilizaron en este proyecto: Random Forest, Gradient Boosting Machine y xgBoost.

Definición 2.1 (Árbol de decisión)**.** *Un árbol de decisión, es un grafo dirigido con estructura de árbol (no necesariamente binario), en el que también usa estadísticos provenientes de la* ***Teoría de información*** *para ir generando, a partir del nodo padre, sus nodos hijos hasta llegar a los nodos hojas.*

En un árbol de decisión, cada nodo interno representa una pregunta o condición sobre una variable explicativa del conjunto de datos. A medida que se sigue el árbol desde el nodo raíz hasta las hojas, las respuestas a estas preguntas guían el camino de la predicción. Cada nodo hoja representa una clase o valor de predicción final.

Existen dos tipos de árboles de decisión: **árboles de clasificación** y **árboles de regresión**. En el proyecto se utilizó la idea de un árbol de regresión ya que la variable dependiente que se desea predecir, es continua y no

discreta.

Por otro lado, el algoritmo general para crear un solo árbol de decisión está dado en el Algoritmo(1).

Algoritmo 1 Generación de *árbol de decisión*

Entrada: Base de datos de aprendizaje, $\mathcal{L}$.
Salida: Árbol de decisión, φ.

1: Crear un árbol de decisión, φ, con nodo raíz t_0
2: Crear una pila vacía S de nodos de la forma $(t, \mathcal{L}_t)$
3: $S.push((t, \mathcal{L}_t))$
4: **while** $S \neq \emptyset$ **do**
5: $\quad t, \mathcal{L}_t = S.pop()$
6: $\quad$ **if** Se cumple el criterio de parada **then**
7: $\quad\quad \overline{y}_t =$constante
8: $\quad$ **else**
9: $\quad\quad$ Hallar la partición de $\mathcal{L}_t$ que maximiza la ganancia de información:

$$s_t^* = \underset{A \in \text{Var-Explicativas}}{\text{máx}} \triangle i(A, t)$$

10: $\quad\quad$ Particionar $\mathcal{L}_t$ en $\bigcup_{v \in \text{niveles(A)}} \mathcal{L}_{t_v}$
11: $\quad\quad$ Crear los v nodos hijo de t
12: $\quad\quad$ $S.push((t, \mathcal{L}_{t_v})), \forall v \in$ niveles(A)
13: **return** φ

2.3.2. Explicación detallada del algoritmo AD

Un árbol de decisión sirve para predecir valores de una variable dependiente. Se lo realiza individuo por individuo (fila a fila) y en cada nodo intermedio se van realizando preguntas sobre las variables de estudio, de esta manera se va particionando la base de entrenamiento de forma que los nodos hojas contengan individuos de la muestra con características similares.

Viéndolo de forma más gráfica, el algoritmo de árbol de decisión está particionando individuos en regiones lo más homogéneas posible. Ver Figura(2.2).

Dentro del algoritmo, se pueden identificar tres acciones importantes que se deben analizar y realizar para el correcto funcionamiento del mismo:

1. *Definir un **criterio de parada** para la generación de nodos hojas del*

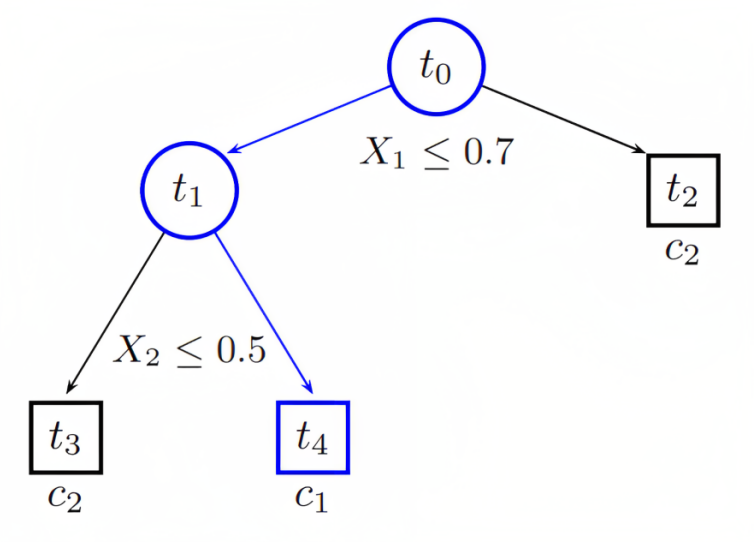


Figura 2.1: *Ejemplo: Arbol de decisión para variable binaria. Obtenido de [8]*

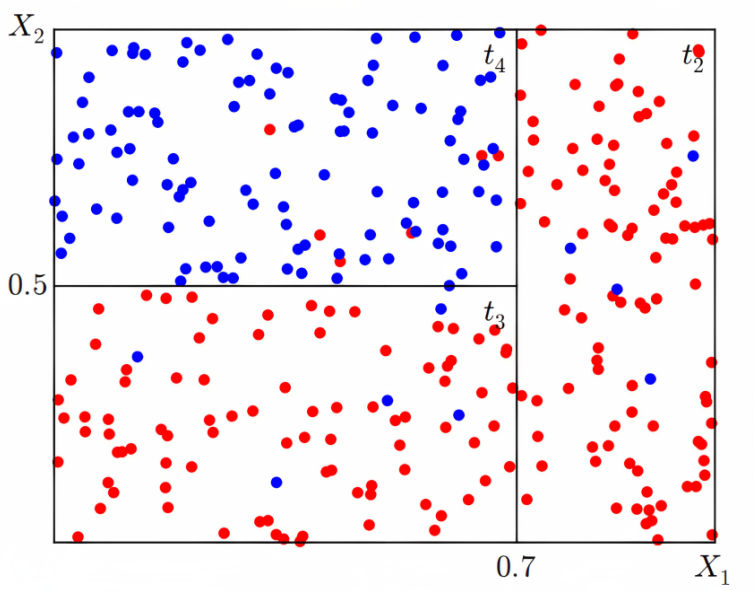


Figura 2.2: *Clasificación de árbol de la* Figura(2.1).

árbol.

El porcentaje de individuos en cada nodo hoja debe ser de tal forma que no caigamos en una sobre-estimación o en una sub-estimación en el modelo final. Este porcentaje es muy importante y va a ser el criterio de parada de los algoritmos, ya que si no se fija un criterio de parada, el algoritmo va a crear nodos hijos hasta que en cada hoja solamente exista 1 individuo.

2. *Calcular una posible* ***estimación de la variable objetivo*** *que se está tratando de modelar.*

   De forma general, se utiliza el promedio muestral (en cada nodo hoja) para estimar el valor de los ingresos reales de las personas. El promedio es muy común por las propiedades estadísticas que posee (converge con probabilidad 1 al parámetro poblacional), y ha sido implementado en las librerías de R que se utilizaron para calcular los modelos de prueba.

3. *Hallar la* ***partición del conjunto de entrenamiento*** ***($\mathcal{L}$)****, que maximice la* ***ganancia de información*** *en cada nodo del árbol, hasta llegar a tener nodos hojas.*

   Como se muestra en la gráfica(2.2), en cada nodo del árbol de decisión se realiza una partición de los individuos, hasta que en los nodos hoja se llega a tener un porcentaje fijo de individuos del conjunto de entrenamiento.

   Para saber de qué forma ir creando cada nodo del árbol, se utiliza un estadístico llamado ganancia de información, dado por la fórmula general:

$$\triangle i(A,t) = i(t) - \sum_{v \in \text{Niveles(A)}} \frac{|S_v|}{|S|} \cdot i(t_v) \tag{2.1}$$

   en donde:

   a) A es una variable explicativa cualquiera,

   b) t es el nodo al que se quiere calcular la ganancia de información,

   c) $|S|$ es la cardinalidad del conjunto de individuos que actualmente se tiene en el nodo t,

   d) $|S_v|$ es la cardinalidad del subconjunto de S que contiene los individuos en el nivel v de la variable A,

   e) Finalmente, $i(t)$ es una medida de impureza de cada nodo, que ayuda a medir la homogeneidad de los niveles de una variable. En este sentido, $i(t_v)$ mide la impureza de cada nodo hijo del nodo actual t (en el supuesto caso que se decida usar la variable A para particionar el nodo t).

En árboles de clasificación es muy común utilizar las funciones de impureza de: *Entropía de Shannon* o también de *Índice de Gini*. Por otro lado, en árboles de regresión se utiliza una función de impureza relacionada con el error cuadrático medio en cada nodo:

$$i(t) = \frac{1}{N_t} \sum_{\mathbf{X}, y \in \mathcal{L}_t} (y - \overline{y}_t)^2 \tag{2.2}$$

en donde:

a) N_t es el número de individuos que hay actualmente en el nodo t,

b) $\mathcal{L}_t$ es el subconjunto, del conjunto de entrenamiento, que se encuentra actualmente en el nodo t,

c) $\mathbf{X}$ es la matriz de información muestral de las variables explicativas, que se obtiene de $\mathcal{L}_t$, de forma similar, y es el vector muestral de la variable dependiente que estamos estudiando y se obtiene de $\mathcal{L}_t$,

d) y es el valor real de la variable objetivo, y $\overline{y}_t$ es el valor estimado de la variable, que se calcula en el nodo t.

Es importante resaltar también que existe una relación entre la función de impureza de Gini (clasificación) y la función de impureza dada antes (regresión):

$$i(t) = \frac{1}{2} \cdot i_{gini}(t) \tag{2.3}$$

por lo que se puede implementar un árbol de clasificación o de regresión usando un mismo criterio para ambos.

Para finalizar el cálculo de la maximización de la ganancia de información, se debe calcular, para cada variable que no ha sido considerada hasta ese momento en la creación del nodo t:

$$s_t^* = \max_{A \in \text{Var-Explicativas}} \triangle i(A, t) \tag{2.4}$$

donde s_t^* es la partición en $\mathcal{L}_t$ que maximiza la ganancia de información, en el nodo actual t.

Si se modifican los estadísticos de los tres pasos anteriores y se los utiliza en el algoritmo indicado (ver algoritmo 1), va a cambiar la forma de generar el árbol de decisión.

2.3.3. Algoritmo de Random Forest:

Los bosques aleatorios (Random Forest) consisten en construir un conjunto de varios árboles de decisión, que se calculan a partir de una modificación en la aleatoriedad del algoritmo (1), presentado antes para árboles de decisión.

Existen varios métodos para calcular el bosque aleatorio, y estos se diferencian en la forma de introducir la perturbación aleatoria que va a influir en el cálculo de cada árbol de decisión por separado.

En el presente proyecto, se utiliza la función *ranger* de la librería de mismo nombre, del lenguaje R. Este algoritmo para calcular el bosque aleatorio, se resume en los siguientes pasos:

1. **Conjunto de datos:** El algoritmo de Bosques Aleatorios parte de un conjunto de datos de entrenamiento que incluye variables independientes, junto con la variable objetivo que se desea predecir.

   Cada fila en el conjunto de datos representa una observación.

2. **Muestreo bootstrap:** Se inicia el proceso generando múltiples conjuntos de datos de entrenamiento mediante el muestreo bootstrap con reemplazo. Esto implica seleccionar al azar filas del conjunto de datos original para formar una nueva muestra, permitiendo que una misma observación aparezca varias veces o no aparezca en absoluto.

   Cada conjunto de datos de entrenamiento generado de esta manera es del mismo tamaño que el conjunto de datos de entrenamiento original.

3. **Construcción de árboles:** Para cada conjunto de datos de entrenamiento, se construye un árbol de decisión.

4. **Votación y predicción:** Una vez que se han construido todos los árboles, para realizar una predicción sobre una nueva observación, se evalúa esa observación por cada árbol del bosque.

En el caso de clasificación, cada árbol emite una predicción para la clase de la observación. Luego, se realiza una votación entre todos los árboles para determinar la clase final más frecuente. En el caso de regresión, cada árbol emite una predicción numérica, y el resultado final es el *promedio* de las predicciones de todos los árboles.

5. **Evaluación y ajuste:** Una vez que se ha construido el bosque y se han realizado las predicciones, se evalúa la precisión del modelo utilizando un conjunto de datos de prueba. Este conjunto de prueba contiene individuos no utilizados en el proceso de entrenamiento y permite evaluar cómo se desempeña el modelo frente a datos no vistos previamente.

   Se pueden ajustar hiperparámetros del algoritmo, como el número de árboles en el bosque o la profundidad máxima de los árboles, mediante técnicas de validación cruzada para obtener un modelo final con un rendimiento óptimo.

Lo mencionado en el paso 5 forma parte del trabajo realizado en el proyecto. Específicamente, después de efectuar las predicciones iniciales, se procedió a ajustar los hiperparámetros de la función *ranger*. El objetivo fue la minimización del error cuadrático medio (MSE), la reducción de la cantidad de individuos en cada nodo hoja y la limitación del número de árboles generados por el algoritmo. Estos ajustes se llevaron a cabo con el propósito de evitar tanto el sobreajuste como el subajuste en las estimaciones posteriores, al mismo tiempo que se buscó mantener la eficiencia computacional.

Cabe señalar que se tomó esta medida considerando que las computadoras empleadas en los experimentos poseían recursos limitados en términos de capacidad de procesamiento.

2.4. Gradient Boosting Machine (GBM)

El algoritmo de *GBM* es el segundo algoritmo de aprendizaje supervisado no paramétrico que se utilizó en el proyecto. A diferencia del algoritmo de Random forest, este genera árboles de decisión de forma secuencial sobre un conjunto de datos de entrenamiento; es decir, cada árbol de decisión

que se genera, depende del anterior árbol que se creó para ser generado (ver más en [5] y [10]).

La metología de Gradient Boosting Machine se resume en el Algoritmo(2). Los parámetros que se observan son:

1. y_i es el valor real de la observación de la variable dependiente para el individuo i,
2. La función de pérdida es: $L(y_i, \gamma) = (y_i - \gamma)^2$,
3. *M* denota el número de árboles que se van a crear, mientras que m es el m-ésimo árbol generado,
4. F_{m-1} es la predicción del paso anterior, empieza con F_0,
5. j representa el nodo hoja, mientras que J_m representa el total de hojas en el árbol m,
6. $\gamma_{j,m}$ representa la estimación del error en la hoja j del árbol m
7. $n_{j,m}$ representa el número de individuos en la hoja j del árbol m,
8. $r_{i,m}$ representa el error en el individuo i del árbol m, del subconjunto de individuos $R_{j,m}$,
9. el parámetro η representa el coeficiente de aprendizaje del algoritmo,
10. y por último se tiene la función indicadora $\mathbb{1}(x \in R_{j,m})$, esta asegura que el individuo x se vincule apropiadamente al único nodo hoja que le corresponde, ya que cada individuo está asociado a un único nodo hoja del árbol de decisión.

2.4.1. Explicación detallada de algoritmo

A continuación se explican los pasos que se realiza en el algoritmo para construir el modelo predictivo. El ejemplo que se presenta fue tomado de [9] :

1. *Inicialización del algoritmo con el promedio muestral:*

Algoritmo 2 Algoritmo de Gradient Boosting Machine

Entrada: Base de datos de aprendizaje, $\mathcal{L}$.
Salida: Estimación de variable dependiente $Y = F_M(x)$.

1: Calcular el valor constante:

$$F_0(x) = \min_{\gamma} \sum_{i=1}^{n} L(y_i, \gamma) = \overline{Y_n} \tag{2.5}$$

2: **for** $m = 1$ hasta M **do**

3: Calcular los residuales, $\forall i = 1, \ldots, n$:

$$r_{i,m} = -\frac{\delta L(y_i, \gamma)}{\delta \gamma}\Big|_{\gamma = F_{m-1}(x)} = y_i - F_{m-1} \tag{2.6}$$

4: Entrenar el árbol de decisión, $r_m \sim X_1, \ldots, X_k$, y hallar las regiones de individuos en cada hoja, $R_{j,m}$, para $j = 1, \ldots, J_m$

5: Calcular, $\forall j = 1, \ldots, J_m$:

$$\gamma_{j,m} = \min_{\gamma} \sum_{x_i \in R_{j,m}} L(y_i, F_{m-1}(x_i) + \gamma) = \frac{1}{n_{j,m}} \sum_{x_i \in R_{j,m}} r_{i,m} \tag{2.7}$$

6: Actualizar las estimaciones del modelo:

$$F_m(x) = F_{m-1}(x) + \eta \cdot \sum_{j=1}^{J_m} \gamma_{j,m} \cdot \mathbb{1}(x \in R_{j,m}) \tag{2.8}$$

7: **return** $F_M(x)$

Para predecir la variable objetivo Y en función de las variables explicativas $X_1, \ldots, X_k$, el algoritmo comienza utilizando el promedio muestral como el primer predictor para dicha variable. Es decir, se calcula el promedio de todos los valores de Y en el conjunto de entrenamiento y se utiliza este valor como la predicción inicial para todas las instancias de los datos de entrenamiento. Ver figura (2.3).

$$F_0 = \overline{Y}_0$$

2. *Generación de nuevos modelos débiles para reducir errores:*

El residuo se calcula restando las predicciones del primer árbol, de

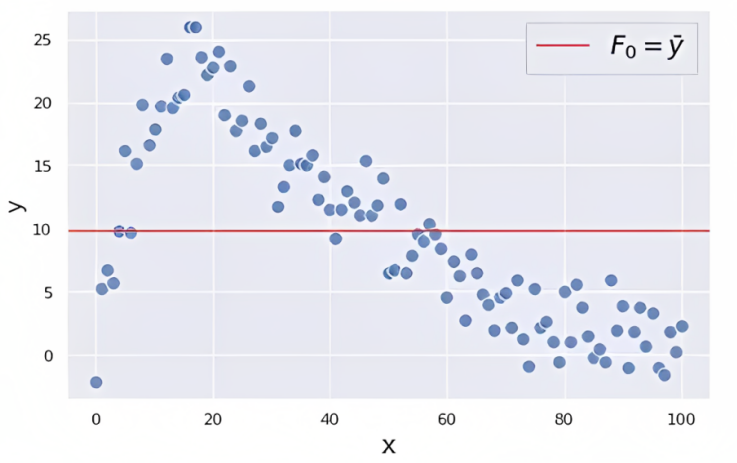


Figura 2.3: *Muestra y promedio muestral.*

los valores reales Y en el conjunto de entrenamiento. Ver figura (2.4).

$$r_0 = Y - \overline{Y}_0$$

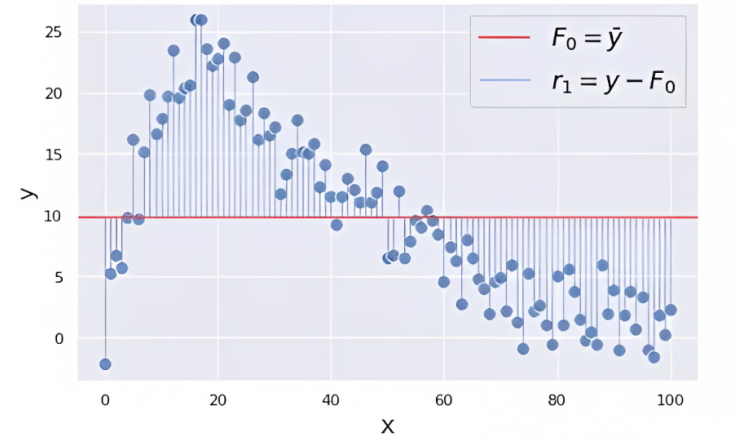


Figura 2.4: *Errores para* F_0.

Con el fin de reducir los errores de predicción causados por la primera estimación (media muestral), se genera un primer árbol de decisión. En este caso, el árbol se construye de manera que la variable dependiente sea el residuo de la primera estimación, r_0, mientras que las variables explicativas iniciales se mantienen, es decir, $r_0 \sim X_1, \ldots, X_k$. Ver figura (2.5).

Las predicciones brindadas por este primer árbol se las denota como γ_1. Luego, la nueva predicción de la variable Y ahora se obtiene

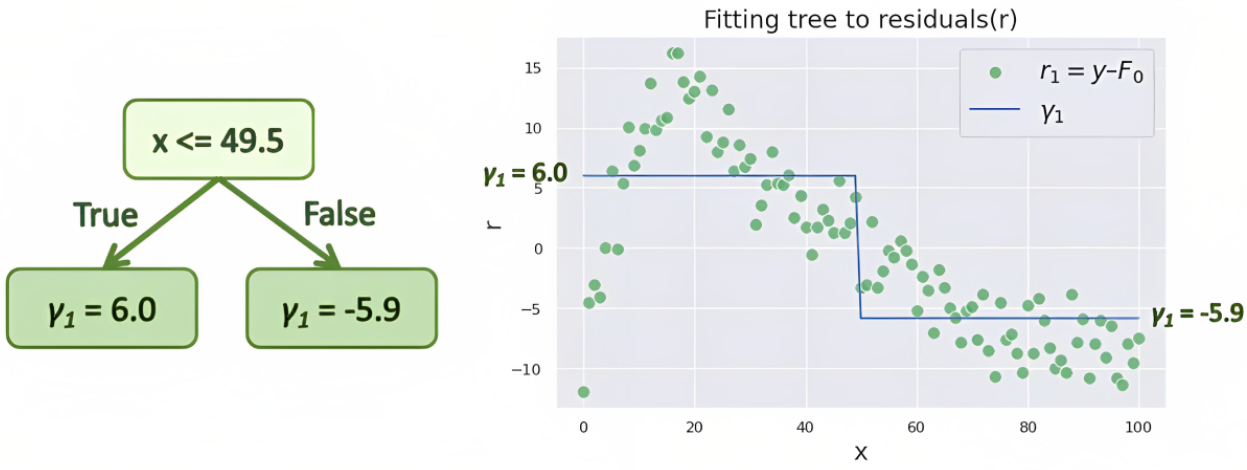


Figura 2.5: *Árbol generado para los errores y gráfico de errores.*

sumando el promedio inicial (calculado en el primer paso) con el valor que se obtiene en la hoja correspondiente del árbol de decisión resultante. Sin embargo, para controlar la contribución de este nuevo modelo al ensamble final, se multiplica por un coeficiente de aprendizaje que está en el rango de 0 a 1. Ver figura (2.6).

$$F_1 = F_0 + \eta \cdot \gamma_1$$

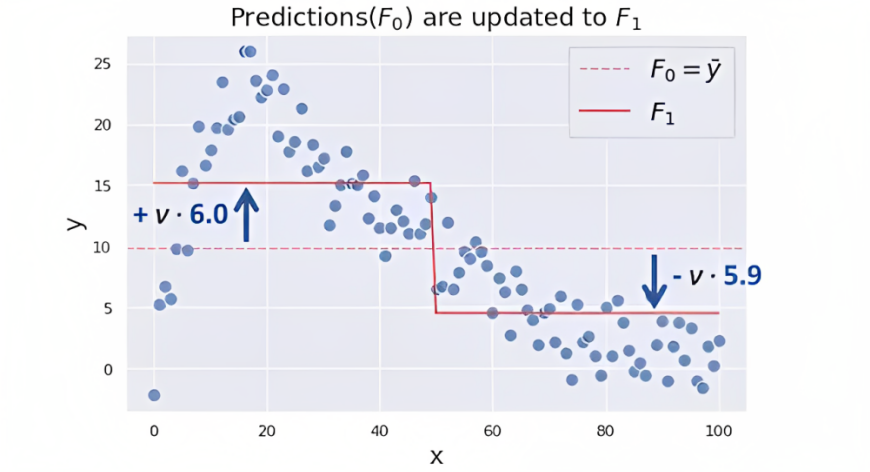


Figura 2.6: *Actualización de estimación de* Y.

Este coeficiente de aprendizaje, η_0, es un hiperparámetro del algoritmo y determina cuánto impacto tiene el nuevo modelo en la predicción final, pero nunca es 1 para evitar sobre-ajuste o sub-ajuste de

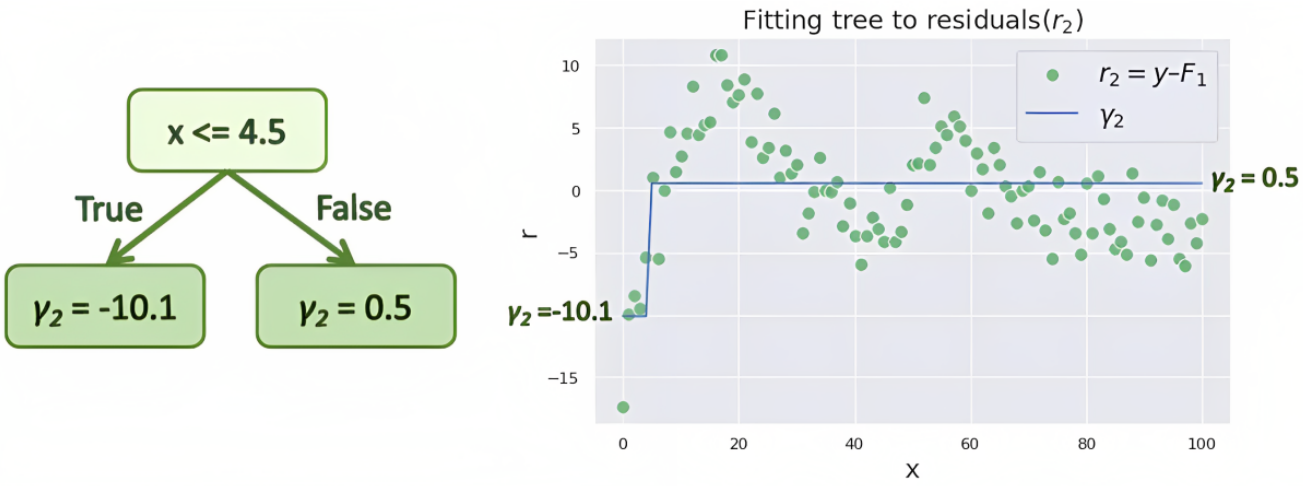


Figura 2.7: *Se calculan nuevamente los errores.*

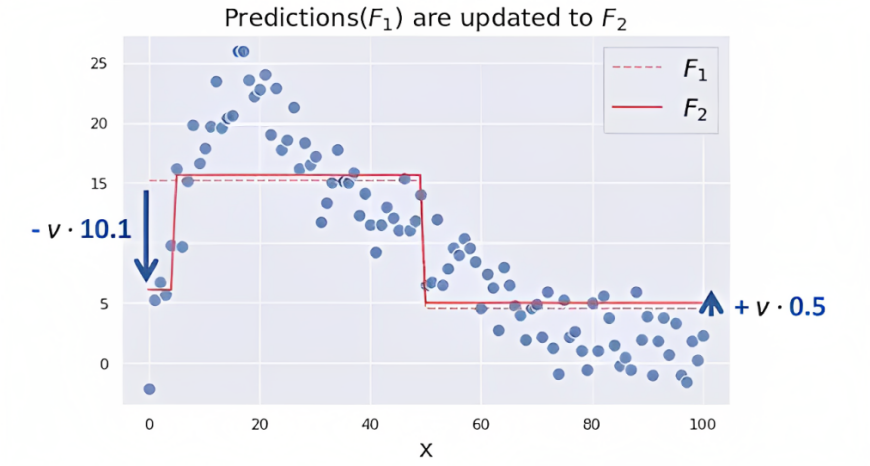


Figura 2.8: *Segunda actualización de* Y.

información que no fué usada para crear el modelo. En el presente proyecto, se tomó $\eta = 0{,}03$ para todas las iteraciones.

3. *Repetición del proceso hasta convergencia*

   El paso 2 se repite varias veces para construir más modelos débiles y mejorar el rendimiento del ensamble. En cada iteración, se construye un nuevo árbol de decisión que se enfoca en estimar los residuos

de los árboles anteriores:

$$\begin{aligned}
&\text{Iteración 1: } r_1 = r_0 - \gamma_1 \implies \text{estimar: } \gamma_2 \implies F_2 = F_1 + \eta \cdot \gamma_2 \\
&\text{Iteración 2: } r_2 = r_1 - \gamma_2 \implies \text{estimar: } \gamma_3 \implies F_3 = F_2 + \eta \cdot \gamma_3 \\
&\text{Iteración 3: } r_3 = r_2 - \gamma_3 \implies \text{estimar: } \gamma_4 \implies F_4 = F_3 + \eta \cdot \gamma_4 \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \vdots
\end{aligned}$$

Por lo tanto, las predicciones del algoritmo se actualizan mediante la suma de los resultados obtenidos en las hojas de los nuevos árboles, creados para los errores en cada iteración.

El proceso de construcción de nuevos árboles de decisión, y la actualización de las predicciones, se repite iterativamente hasta que los errores de predicción, r_k, ya no cambian drásticamente de un árbol a otro, o hasta que se alcance un número predefinido de árboles de decisión en el ensamble.

Cuando se estudia al algoritmo graficando cada iteración sobre un diagrama de puntos, como se realiza desde la Figura(2.3) hasta la Figura(2.8), se observa que la estimación de la variable dependiente se va dando poco a poco, de forma que variables indicadoras se van aproximando más a la tendencia de los puntos.

2.5. Extreme Gradient Boosting (xgBoost)

El algoritmo de *xgBoost* es el tercer algoritmo de aprendizaje supervisado no paramétrico que se utilizó en el proyecto. De forma similar que el algoritmo de *GBM*, este método utiliza árboles de decisión y una técnica de gradiente descendente.

En este caso se minimiza la suma entre la pérdida de entrenamiento y la regularización del árbol generado. Además, para cada árbol que se genera, se trata de crecer el árbol hasta una altura en la que no vaya a existir sub o sobreestimación. Luego, la estimación final es la sumatoria de la predicción inicial y la predicción de cada árbol siguiente.

La diferencia entre GBM y xgBoost se puede ver en la gráfica(2.9). Es

decir que, a veces, el algoritmo de *GBM* genera muchas particiones a la base de datos (ver [3]).

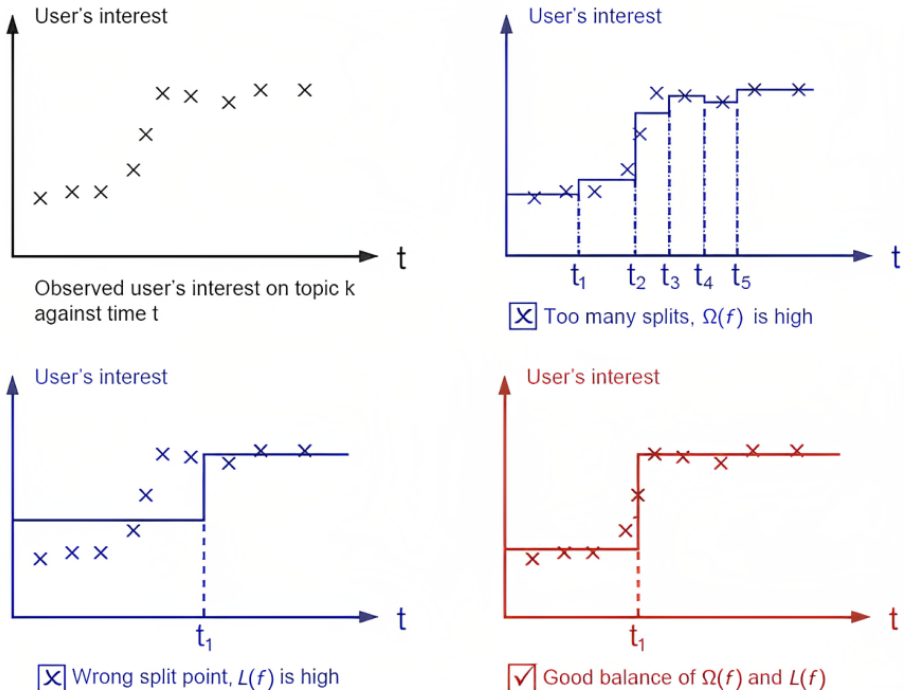


Figura 2.9: *Idea gráfica del algoritmo xgBoost (ver [2])*

Aún así, este algoritmo tiene mejoras en los modelos matemáticos que se calculan en cada iteración en conjunto con una optimización del sistema computacional. La diferencia en la minimización de la función objetivo, es que se agrega un término de regularización que tiene que ver con la creación del árbol:

$$\min_{\gamma} \sum_{i=1}^{n} L(y_i, F_{m-1}(x_i) + \gamma(x_i)) + \Omega(\gamma) \tag{2.9}$$

donde γ representa todos los valores de los nodos hoja del árbol generado, F_{m-1} representa lo mismo pero para el árbol generado en el paso anterior.

El primer término se lo conoce como *pérdida de entrenamiento (training loss en inglés)*, y mide qué tan bien se incorpora el modelo a los datos de entrenamiento, y trabaja de forma similar que en *GBM*.

El término $\Omega(\gamma)$ es la regularización del algoritmo, que está relacionado con el número de regiones(o particiones), $R_{j,m}$, de individuos que se crean en la hoja j del árbol m. Este término de regularización mide la complejidad de los árboles que se van creando en cada iteración. En este

algoritmo se considera:

$$\Omega(\gamma) = \eta \cdot T + \frac{1}{2}\lambda \cdot \sum_{j=1}^{T} \gamma_j^2 \tag{2.10}$$

En donde η es el coeficiente de aprendizaje para cada nodo hoja *(en el caso de GBM se utilizó* $\eta = 0{,}03$*)*, T es el número de nodos hoja en el árbol, λ es un coeficiente de penalización para el tamaño del árbol generado, y γ_j es la estimación de la variable dependiente en cada hoja del árbol.

Si no se controla la partición de individuos, los árboles que se generan en cada iteración crecen hasta su máxima extención, lo que causa que el árbol creado *"pierda generalidad"*, y no realice buenas predicciones con individuos que no pertenecieron a la base de entrenamiento.

Tomando en cuenta el primer término en la función objetivo, y aplicando *Series de Taylor de orden 2, sobre la segunda componente de la pérdida L*, se obtiene:

$$\sum_{i=1}^{n} L(y_i, F_{m-1}(x_i) + \gamma(x_i)) \approx \sum_{i=1}^{n} \left[L(y_i, F_{m-1}(x_i)) + g_i \cdot \gamma(x_i) + \frac{1}{2} h_i \gamma^2(x_i) \right] \tag{2.11}$$

en donde: $g_i = \delta_{F_{m-1}}(F_{m-1}(x_i) - y_i)^2$ y $h_i = \delta^2_{F_{m-1}}(y_i - F_{m-1}(x_i))^2$, son la primera y segunda derivada respectivamente, con respecto a $F_{m-1}(x_i)$ (ver [11]).

Ahora, sumando los dos términos de la función objetivo, y simplificando la notación, se obtiene:

$$\min_{\gamma} \sum_{j=1}^{T} \left[G_j \cdot \gamma_j + \frac{1}{2}(H_j + \lambda)\gamma_j^2 \right] + \eta \cdot T \tag{2.12}$$

en donde se considera $L(y_i, F_{m-1}(x_i)) \approx 0$, pues es lo que se espera de la estimación; además: $G_j = \sum_{i \in I_j} g_i$ y $H_j = \sum_{i \in I_j} h_i$, donde $I_j = \{i \in 1, \ldots, n : x_i$ está en la hoja $j\}$.

Observar que esta última sumatoria es la suma de T funciones cuadráticas respecto a γ_j cada una, por lo tanto al derivar e igualar a cero, se

obtiene el estimador de los valores en cada hoja del árbol:

$$\widehat{\gamma}_j = -\frac{G_j}{H_j + \lambda} \quad \forall j = 1, \ldots, T \tag{2.13}$$

Por lo tanto, el mínimo de esta función objetivo es:

$$-\frac{1}{2} \cdot \sum_{j=1}^{T} \frac{G_j^2}{H_j + \lambda} + \eta \cdot T.$$

Ahora, con los cálculos realizados antes, es posible definir una nueva forma de generar un árbol de decisión, denominada *método "greedy"*. Tomar en cuenta que al mismo tiempo, en la creación de nodos considera que pueden existir valores perdidos (valores NA) en la base de entrenamiento, por lo que también se introduce una técnica de *"sparsity aware split finding"*[1], la cual solo se menciona pero no se la estudia a profundidad ya que no es el objetivo de este proyecto. Ver Algoritmo(3).

Algoritmo 3 Generación *Greedy* para *árbol de decisión* (Y binaria)

Entrada: Base de datos de aprendizaje, $\mathcal{L}$.
Salida: Árbol de decisión, φ.

1: Crear un árbol de decisión, φ, con nodo raíz t_0
2: Particionar t_0 en nodo izquierda y derecha si:

$$Gain = \frac{1}{2} \cdot \left[\frac{G_L^2}{H_L + \lambda} + \frac{G_R^2}{H_R + \lambda} - \frac{(G_L + G_R)^2}{H_L + H_R + \lambda} \right] - \eta \geq 0 \tag{2.14}$$

   En el caso que $Gain < 0$, se deja de particionar el nodo (técnica de podado de árboles).
3: Realizar el paso anterior para cada nodo que se vaya creando pero de forma ordenada, es decir, de izquierda a derecha.
4: **return** φ

Finalmente, el algoritmo para el *Modelo xgBoost* está dado por el Algoritmo(4). Aquí, M denota el número de árboles que vamos a generar.

La metodología del xgBoost es mucho más eficiente computacionalmente que las otras metodologías como Random Forest, Gradient Boosting Machine y Regresión Lineal Múltiple. En este proyecto no se estudia estas

[1]Esta técnica está diseñada para manejar conjuntos de datos donde muchas de las características tienen un gran número de valores nulos o cero, durante el proceso de construcción del árbol de decisión.

Algoritmo 4 Algoritmo de Modelo xgBoost

Entrada: Base de datos de aprendizaje, $\mathcal{L}$.
Salida: Predicción, F_M, de la variable dependiente.

1: Empezar con un predictor "débil"de la variable: $F_0 = \overline{Y}$.
2: **for** $m = 1, \ldots, M$ **do**
3: Calcular las primeras y segundas derivadas:

$$\forall i = 1, \ldots, n: \quad g_i = \delta_{F_{m-1}}(F_{m-1}(x_i) - y_i)^2$$
$$h_i = \delta^2_{F_{m-1}}(y_i - F_{m-1}(x_i))^2$$

4: Entrenar el árbol "greedy"de decisión. Al final se denota al árbol como:

$$E_m(x_i) = \sum_{j=1}^{T} \widehat{\gamma_j} \cdot \mathbb{1}(x \in R_{j,m}),$$

en donde $\{R_{j,m}\}_{j=1}^{T}$ es la mejor partición de individuos en los nodos en el árbol m. Además: $\widehat{\gamma_j} = -\frac{G_j}{H_j+\lambda}$.
5: Actualizar: $F_m = F_{m-1} + E_m$
6: Calcular:

$$F_M = F_0 + \sum_{m=1}^{M} E_m$$

7: **return** F_M

mejoras computacionales pues no se halla dentro del alcance de conocimientos de la carrera.

2.6. Test de Kolmogorov-Smirnov (KS)

El test de Kolmogorov-Smirnov, o también conocido como el test KS, es un test estadístico que permite comparar dos distribuciones diferentes.

Este test se puede utilizar para comparar una distribución muestral con una distribución teórica muestral (test KS de una muestra), o también dos distribuciones muestrales (test KS de dos muestras).

La prueba de hipótesis que se estudia con este estadístico es:

H_0 : las muestras vienen de una población con la misma distribución.

H_a : las muestras vienen de una población con diferente distribución.

De manera general, el estadístico de prueba que se utiliza para realizar un test KS de dos muestras, es (Revisar [1]):

$$D_i = \max_z |F_{X_i}(z) - F_Y(z)|$$

Por otro lado, si la variable X tiene una muestra de n individuos y Y tiene una muestra de m individuos, el estadístico crítico para aceptar o rechazar H_0 es:

$$D_{crit} = c(\alpha) \cdot \sqrt{\frac{n+m}{n \cdot m}},$$

en donde $c(\alpha) = \sqrt{-\ln\left(\frac{\alpha}{2}\right) \cdot 0{,}5}$ y α es el nivel de confianza. Así, se rechaza H_0 si $D_i > D_{crit}$. Revisar más en [13]. **Por ejemplo**, vamos a comparar los siguientes vectores de resultados:

$$X : (4, 7, 9, 12, 21, 23, 23, 24, 28)$$
$$Y : (1, 5, 7, 13, 13, 18, 20, 20, 25)$$

Se calculan las funciones de distribución acumuladas:

$$CDF(X) = \begin{cases} 0 & x < 4 \\ \frac{1}{9} & 4 \leq x < 7 \\ \frac{2}{9} & 7 \leq x < 9 \\ \frac{1}{3} & 9 \leq x < 12 \\ \frac{4}{9} & 12 \leq x < 21 \\ \frac{5}{9} & 21 \leq x < 23 \\ \frac{7}{9} & 23 \leq x < 24 \\ \frac{8}{9} & 24 \leq x < 28 \\ 1 & 28 \leq x \end{cases} \qquad CDF(Y) = \begin{cases} 0 & x < 1 \\ \frac{1}{9} & 1 \leq x < 5 \\ \frac{2}{9} & 5 \leq x < 7 \\ \frac{1}{3} & 7 \leq x < 13 \\ \frac{5}{9} & 13 \leq x < 18 \\ \frac{2}{3} & 18 \leq x < 20 \\ \frac{8}{9} & 20 \leq x < 25 \\ 1 & 25 \leq x \end{cases}$$

Se grafican las distribuciones acumuladas para visualizar:

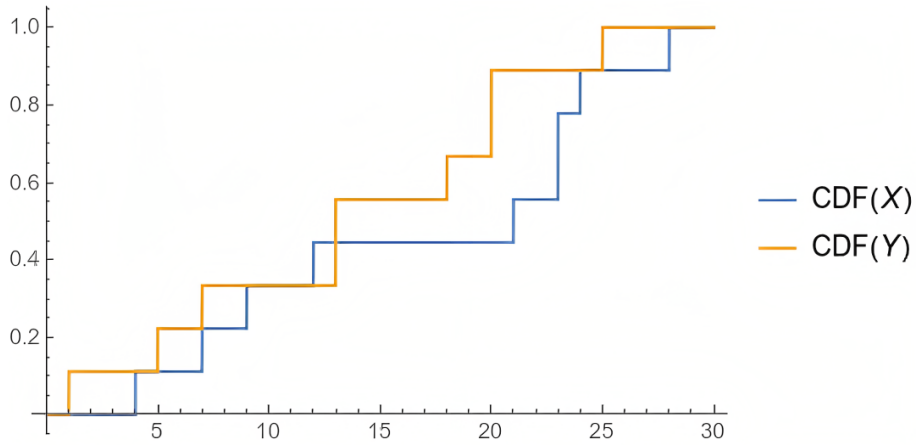


Figura 2.10: *Gráficas de las distribuciones acumuladas de X e Y.*

Ahora, calculamos el estadístico de prueba: $D = \max_x(F_X(x) - F_Y(x))$:

$$D = \begin{cases} 0 & x < 1 \\ \frac{1}{9} & 1 \leq x < 4 \\ 0 & 4 \leq x < 5 \\ \frac{1}{9} & 5 \leq x < 9 \\ 0 & 9 \leq x < 12 \\ \frac{1}{9} & 12 \leq x < 18 \\ \frac{2}{9} & 18 \leq x < 20 \\ \frac{4}{9}* & 20 \leq x < 21 \\ \frac{1}{3} & 21 \leq x < 23 \\ \frac{1}{9} & 23 \leq x < 24 \\ 0 & 24 \leq x < 25 \\ \frac{1}{9} & 25 \leq x < 28 \\ 0 & 28 \leq x \end{cases}$$

Por lo tanto, el estadístico de prueba tiene el valor $\frac{4}{9} = 0.\overline{4}$; mientras que el estadístico crítico, con confianza de $1 - \alpha = 0{,}95$, es $0{,}64021\ldots$. Como el estadístico de prueba es menor que el valor crítico, no se rechaza la hipótesis nula. Ejemplo tomado de [12].

En el proyecto se utilizó el test KS de dos muestras, para comparar las distribuciones muestrales de cada variable explicativa con la variable de-

pendiente que es la **Capacidad de pago**. Solamente para el cálculo del test KS y del test VI (que se explica en la siguiente sección), se discretizó la variable dependiente en una variable categórica con 3 niveles diferentes.

También, para realizar los cálculos del estadístico de prueba en el proyecto, se lo realizó de forma ponderada. Es decir, se calculó el estadístico de prueba de una variable explicativa con respecto a los posibles niveles de la variable dependiente; y este cálculo se lo realiza para cada combinación posible de dos niveles en la variable dependiente.

Después, para combinar todos los resultados, se realiza una suma ponderada:

$$estad_{KS} = \sum_{i,j=1}^{k} D_{i,j} \cdot pond_{i,j}$$

en donde $pond_{i,j} = \left(\frac{frec_{nivel[i]} + frec_{nivel[j]}}{n(k-1)} \right)$, k es el número de niveles distintos de la variable dependiente Y, y n es el número de observaciones de la variable X. También, se tiene que $k \leq n$, pues como máximo existen n valores diferentes para Y.

En este caso, al ser Y una variable numérica, se genera una nueva variable que recoge la información de la variable Y en tres niveles. La nueva variable se trata del *Rango de ingresos*. Además, este cálculo ponderado se lo realiza para cada una de las variables explicativas.

De esta manera, escogiendo aquellas variables que cumplen $estad_{KS} \geq 0{,}20$, se obtienen las variables que más aportan para describir a la variable dependiente. Hay que tomar en cuenta que, mientras más se acerca el valor del estadístico de prueba a 1, mayor **poder predictivo** tendrá la variable comparada.

2.7. Valor de Información (VI)

El Valor de Información (VI) es una medida utilizada en el contexto del análisis de tablas de contingencia para evaluar la asociación entre dos variables categóricas. Es una medida de la ganancia o pérdida de información proporcionada por una variable categórica al predecir otra varia-

ble categórica.

Cuando se trabaja con **tablas de contingencia**, se presentan las frecuencias conjuntas de dos variables categóricas. Estas tablas se organizan en filas y columnas, donde cada celda representa el recuento de observaciones que pertenecen a una combinación particular de categorías de ambas variables. A partir de esta información, se pueden calcular diversas medidas de asociación, y el Valor de Información es una de ellas. Ver Tabla(2.1).

$X \setminus Y$	d_1	$\cdots$	d_k	$\cdots$	d_s	total
c_1	n_{11}	$\cdots$	n_{1k}	$\cdots$	n_{1s}	$n_{1\bullet}$
$\vdots$	$\vdots$		$\vdots$		$\vdots$	$\vdots$
c_h	n_{h1}	$\cdots$	n_{hk}	$\cdots$	n_{hs}	$n_{h\bullet}$
$\vdots$	$\vdots$		$\vdots$		$\vdots$	$\vdots$
c_r	n_{r1}	$\cdots$	n_{rk}	$\cdots$	n_{rs}	$n_{r\bullet}$
Total	$n_{\bullet 1}$	$\cdots$	$n_{\bullet k}$	$\cdots$	$n_{\bullet s}$	n

Tabla 2.1: *Tabla de contingencia de VI.*

El cálculo del Valor de Información implica comparar la distribución conjunta de las dos variables con la distribución que se esperaría si fueran independientes. La fórmula para calcular el Valor de Información es la siguiente:

$$VI = \sum_{i=1}^{r} \sum_{j=1}^{s} p_{ij} \log \left(\frac{p_{ij}}{p_i \cdot p_j} \right)$$

donde:

- p_{ij}: Frecuencia conjunta observada en la celda (i, j) de la tabla de contingencia.
- p_i: Frecuencia marginal de la fila i, es decir, la suma de las frecuencias en la fila i.
- p_j: Frecuencia marginal de la columna j, es decir, la suma de las frecuencias en la columna j.
- r: Número de filas de la tabla de contingencia, es decir, el número de categorías de la variable 1.

- s: Número de columnas de la tabla de contingencia, es decir, el número de categorías de la variable 2.

El Valor de Información es una medida que varía entre 0 y ∞, donde un valor cercano a 0 indica que las dos variables son independientes, mientras que un valor mayor indica una mayor asociación entre ellas. Cuanto mayor sea el Valor de Información, mayor es la ganancia de información y mayor es la asociación entre las variables categóricas. Revisar [6] y [7].

El Valor de Información se utiliza en el análisis de datos, la minería de datos y la inteligencia artificial para medir la relevancia de una variable categórica al predecir otra variable categórica, y es especialmente útil en el contexto de selección de características y análisis de atributos en modelos predictivos.

2.8. Prueba Chi-Cuadrado para tablas de contingencia

Al momento de realizar predicciones sobre la muestra de validación, es necesario medir las frecuencias de las predicciones para saber si nuestro modelo predice de la misma forma a la muestra de entrenamiento y la muestra de validación. En otras palabras, lo que se quiere y se espera es que, la distribución de las predicciones en la muestra de entrenamiento sea muy similar a la distribución de las predicciones con la muestra de validación.

Para poder saber si lo último mencionado se está cumpliendo, se utilizó el estadístico para la prueba de Chi-Cuadrado en una tabla de contingencia generada por las variables *Rango de Ingresos Real* y *Rango de Ingresos Estimados*. Esta tabla de contingencia es calculada para la muestra de entrenamiento y la muestra de validación.

El estadístico de prueba es:

$$\chi^2 = \sum_{i,j=1}^{k} \frac{(t_{ij} - v_{ij})^2}{t_{ij}},$$

en donde k es el número de niveles en los rangos de ingresos, t_{ij} es la

frecuencia de las estimaciones con la muestra test en la posición (i, j) en la primera tabla de contingencia, v_{ij} es la frecuencia de las estimaciones con la muestra de validación en la posición (i, j) en la segunda tabla de contingencia. Se calcularon los rangos de tal forma que haya un número igual de rangos reales y estimados.

El estadístico crítico está dado por una variable aleatoria χ^2 con $(k-1)^2$ grados de libertad y una confianza del 95 %. Además, la hipótesis nula es *"Las distribuciones en modelamiento y test son similares."*; la cual se rechaza si $\chi^2 > \chi^2_{crit}$. Revisar [4].

Capítulo 3

Metodología

3.1. Esquema metodológico y de resultados

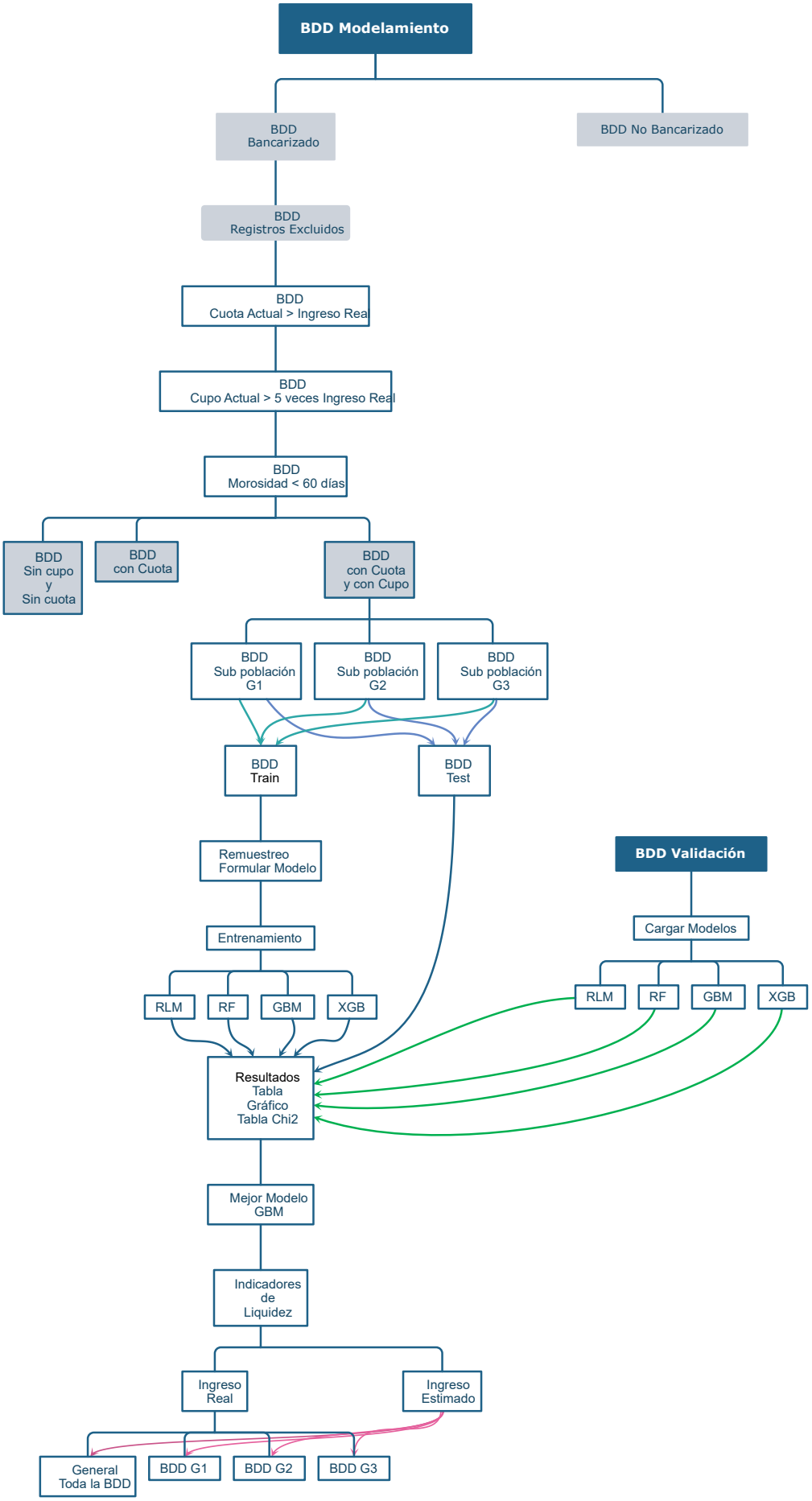
En este capítulo, se proporciona un desglose de las etapas que conformaron el proyecto. Se inició abordando el tratamiento de la base de datos, seguido por la selección de las variables más relevantes para cada modelo considerado. Posteriormente, se llevó a cabo un proceso de remuestreo en la base de datos con el objetivo de garantizar la representatividad de los datos.

Una parte crucial del enfoque adoptado implicó la cuidadosa elección de los hiperparámetros para la ejecución de los modelos. Además, se crearon múltiples modelos, cada uno con sus propias configuraciones e hiperparámetros específicos. Este proceso resultó esencial para acercar más las predicciones a la realidad de los datos.

Luego, se documentó el criterio que sustentó la elección del modelo que arrojó los resultados más prometedores.

Finalmente, se presenta una evaluación de los modelos obtenidos. Estos resultados se compararon con una base consolidada final, lo que permitió verificar la eficacia y solidez de los modelos al emplear información que no había sido considerada en la fase de creación de los mismos.

A continuación se observa un **esquema metodológico** para mejor entendimiento:


BDD Modelamiento
BDD Bancarizado
BDD No Bancarizado
BDD Registros Excluidos
BDD Cuota Actual > Ingreso Rea
BDD Cupo Actual > 5 veces Ingreso Real
BDD Morosidad < 60 días
BDD Sin cupo y Sin cuota
BDD con Cuota
BDD con Cuota y con Cupo
BDD Sub población G1
BDD Sub población G2
BDD Sub población G3
BDD Train
BDD Test
Remuestreo Formular Modelo
Entrenamiento
RLM
RF
GBM
XGB
BDD Validación
Cargar Modelos
RLM
RF
GBM
XGB
Resultados Tabla Gráfico Tabla Chi2
Mejor Modelo GBM
Indicadores de Liquidez
Ingreso Real
Ingreso Estimado
General Toda la BDD
BDD G1
BDD G2
BDD G3

3.2. Exploración y descripción de la Base de Datos

3.2.1. Población de modelamiento

La población que se considera para realizar este proyecto, agrupa a los sujetos del Sistema Crediticio Ecuatoriano con ingreso real reportado a una entidad financiera.

Pto. Obs	Casos	%
dic-21	950.821	100,00

Tabla 3.1: Total de individuos en la base de datos inicial.

La base de datos inicial tiene 1172 variables. Se brindan más detalles de las variables en la sección de Elección de Variables.

3.2.2. Identificación de Bancarizado

En vista que la estimación de la capacidad de pago se realiza con información histórica de Buró, se identifican los sujetos que no cuentan con dicha información.

Se genera la marca NO BANCARIZADO. Los sujetos no bancarizados son aquellos que no tienen historial crediticio, por ende, no sería posible realizar una estimación de la capacidad de pago al no contar con variables históricas.

Se enfatiza que, el primer predictor de ingresos que se utiliza previo a la investigación realizada para el presente proyecto, fue modelado por el Buró, y con este modelo se realizan las primeras comparaciones que incluyen las gráficas de las densidades estimadas.

BANCARIZADO_36M	N	%
BANCARIZADO	927.647	97,56 %
NO BANCARIZADO	23.174	2,44 %
Total	**950.821**	**100,00 %**

Tabla 3.2: Individuos bancarizados *VS* Individuos no bancarizados

En este primer paso se retira de la base a los individuos que no tienen información disponible como para predecir sus ingresos, y se observa un total de $2{,}44\,\%$.

3.2.3. Descripción del ingreso real y estimado actual

Las primeras estimaciones de los ingresos brindadas por el Buró no son tan eficientes, lo que causa mucho sobreajuste y subajuste de los valores estimados de ingreso real. Se puede observar en las gráficas de distribuciones que las estimaciones no se ajustan bien a los valores reales. Ver la figura (3.1).

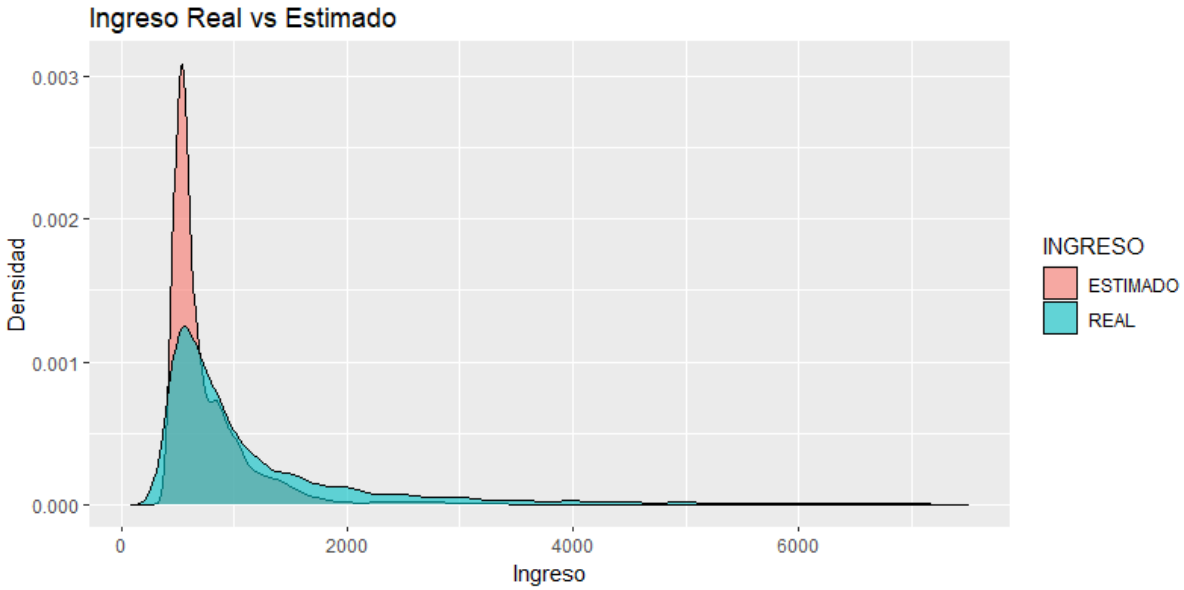


Figura 3.1: *Densidad Real VS Densidad Estimada.*

En este gráfico se puede observar que se subestima demasiado a los individuos con ingresos mayores a los *$2000* y todas las estimaciones se están concentrando alrededor de los *$500*. Esta subestimación proviene del modelo que está utilizando actualmente en el Buró de Crédito y es lo que se desea mejorar en este proyecto.

3.2.4. Registros excluidos

Se aplican tres criterios para excluir individuos de la base de modelamiento. Estos criterios presentan ideas lógicas, y fueron sugeridas por el

tutor del proyecto, en base a su experiencia profesional. Además, es importante tomar en cuenta que, para generar un buen modelo, se necesita información no tan sesgada y que sepamos que pertenece a una población de estudio en específico, ya que si usamos toda la información que se nos brinde es probable que el modelo final no realice buenas predicciones para información que no pertenece a la base muestral de modelamiento.

1. **Se excluyen los registros cuya cuota estimada actual es superior al ingreso.**

   Se excluyen individuos tales que, la suma de las cuotas de amortización de los créditos sea superior al ingreso de la persona, pues debe cubrir gastos básicos para subsistir.

   Se observa que el $23{,}32\,\%$ de la muestra disponible se excluye del modelo estudiado.

CUOTA ESTIMADA SUPERIOR AL INGRESO		
MARCA	**N**	**%**
NO	711.320	76,68 %
SI	216.327	23,32 %
Total	**927.647**	**100,00 %**

Tabla 3.3: Primera exclución de individuos.

2. **Se excluyen los registros cuyo Máximo cupo de TC es superior en 5 veces al ingreso**

   Esta regla aplica para los individuos con ingresos menores a $\$1000$, es decir, los individuos que tengan ingreso superior podrán tener un máximo cupo de TC mayor o menor a los límites establecidos.

   En este caso, se excluye un $2{,}73\,\%$ extra.

CUPO TC SUPERIOR EN 5 VECES AL INGRESO		
MARCA	**N**	**%**
NO	691.920	97,27 %
SI	19.400	2,73 %
Total	**711.320**	**100,00 %**

Tabla 3.4: Segunda exclución de individuos.

3. **Se excluyen los registros que presentan morosidad al punto de observación mayor a 60 días**

   Los individuos que tienen morosidad mayor a 60 días se los excluye, ya que lo más posible es que se les niegue un nuevo crédito o un incremento en el cupo de la tarjeta de crédito al no encontrarse al día en el pago de su haberes.

MOROSIDAD MAYOR A 60 DIAS		
MARCA	**N**	**%**
NO	607.457	87,79 %
SI	**84.463**	**12,21 %**
Total	**691.920**	**100,00 %**

Tabla 3.5: Tercera exclución de individuos.

3.2.5. Poblaciones autónomas para el estudio

Luego de los tres criterios para excluir individuos de la muestra, se procede a realizar una última partición de esta muestra: Individuos que tienen Cuota VS Individuos que tienen Cupo. Esta partición es muy importante, ya que la población *tarjetahabiente* del presente proyecto, resulta ser aquella que *TIENE_CUOTA="SI"* y *TIENE_CUPO="SI"*. Los filtros se resumen en la tabla(3.6).

	TIENE_CUOTA		
TIENE_TC	**NO**	**SI**	**Total**
NO	77.819	298.449	**376.268**
SI	3	231.186	**231.189**
Total	**77.822**	**529.635**	**607.457**

Tabla 3.6: Población individuos tarjetahabiente (color verde).

Aquellos individuos que tienen cuota pero no tienen tarjeta de crédito, y los que no tienen cuota ni tarjeta de crédito, son grupos de individuos que se estudian en otros proyectos.

3.2.6. Especificación de la población de estudio

La población con la que se realiza este estudio, es de *231.186* individuos (tarjetahabientes). Las estimaciones con el modelo anterior siguen

brindando fallas a pesar de haber filtrado la base de muestra, y esto se observa claramente en los percentiles. Para ser más específicos, se están subestimando los valores de los ingresos reales de los individuos. Esto se lo puede observar de forma más clara en las tablas(3.7) y (3.8).

INGRESO REAL						
Quintil	**Mínimo**	**20 %**	**40 %**	**60 %**	**80 %**	**Máximo**
Valor	400	670	963	1450	2566	35000

Tabla 3.7: Quintiles Ingreso Real.

INGRESO ESTIMADO						
Quintil	**Mínimo**	**20 %**	**40 %**	**60 %**	**80 %**	**Máximo**
Valor	233	653	842	961	1242	7721

Tabla 3.8: Quintiles Ingreso estimado Buró.

Estas subestimaciones se confirman una vez que se grafican las distribuciones con los ingresos reales y los ingresos estimados para la sub población de individuos tarjetahabiente, como se lo realizó en la figura(3.2).

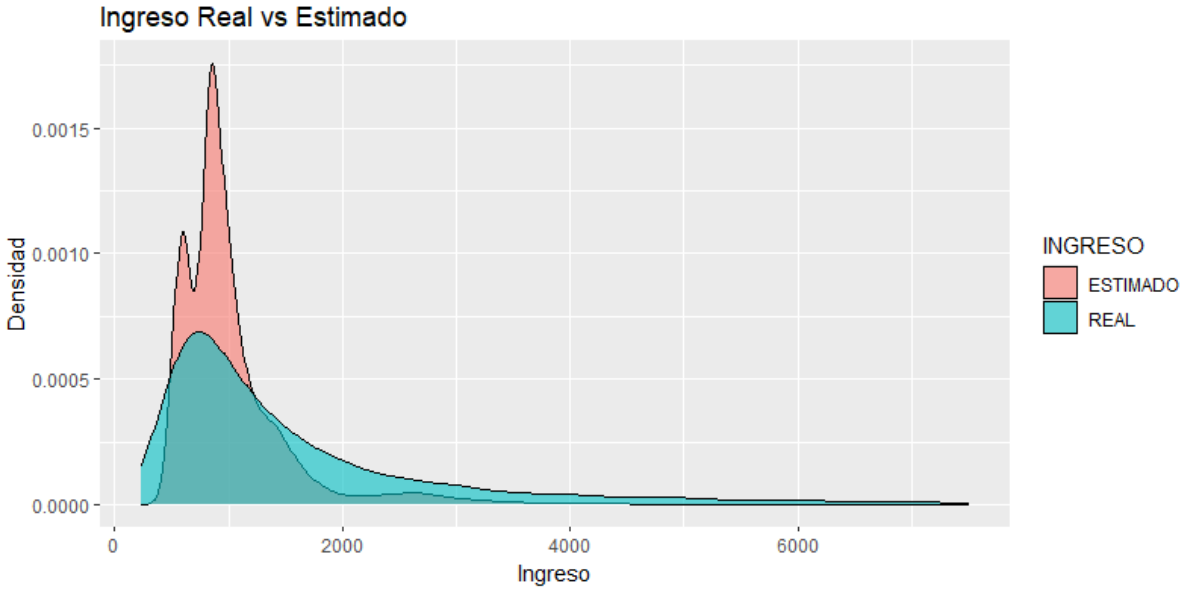


Figura 3.2: *Densidad Real VS Densidad Estimada AVAL.*

Una observación que se puede realizar solo observando la gráfica de comparación de las densidades, es que existen muchos individuos con un ingreso estimado alrededor de la media. Esto implica que las predicciones no son nada buenas en las colas de la distribución (cola izquierda y cola

derecha), por lo que desde un inicio, una técnica muy importante de tomar en cuenta, es realizar un remuestreo adecuado para que se suavice la gráfica de densidad estimada.

De esta manera, un buen criterio para verificar si las estimaciones son buenas (o malas) es, observando que las colas de las densidades reales y muestrales no se alejen mucho unas de otras. El utilizar un solo criterio no es recomendable, es por esto que se utilizan matrices de coincidencias y test estadísticos en conjunto con las gráficas, para escoger el mejor modelo.

3.2.7. Relación entre Ingresos, y Cupo de TC Actual.

Con el objetivo de excluir registros de ingresos de la base de modelamiento que no tengan una relación lógica con los cupos de tarjetas de crédito asignados, se realiza una tabla de valores cruzados, ya que aquí se puede visualizar los individuos que causarían un sesgo demasiado elevado en la base de datos de modelamiento.

Para construir la tabla de valores cruzados considerando una buena representatividad en cada corte, primero se observan los cuartiles para el *Cupo Máximo de TC* y con estos se discretiza esta misma variable. Ver tabla(3.9).

Cupo Máximo				
Quintil	**Mínimo**	**25 %**	**75 %**	**Máximo**
Valor	0	1004	4000	200000

Tabla 3.9: Cuartiles Cupo Máximo Tarjeta de Crédito.

Para discretizar la variable de Ingreso Real, se utilizan rangos de los ingresos reales que son comunes para discriminar a los individuos del sistema crediticio, y se los redondea a un número cerrado. Luego, la tabla de valores cruzados para comparar los ingresos reales y los cupos de tarjeta de crédito, está dada por la tabla(3.10).

Identificación de subpoblaciones G1, G2 y G3

Observar que en la tabla (3.10), se subrayan algunas casillas de color anaranjado. Estas casillas contienen individuos que se excluyen de la

	CUPO TC			
INGRESOS	**<= 1000**	**1000 - 4000**	**>4000**	
<= 450	4.186	2.436	0	
451 - 1000	35.389	57.272	1.223	
1001 - 2500	13.319	42.047	28.014	
2501 - 5000	3.286	11.102	13.850	
>5000	1.486	6.164	11.412	**231.186**
	57.666	**119.021**	**54.499**	

Tabla 3.10: Valores cruzados: Ingreso Real VS Cupo Máximo Tarjeta de Crédito.

generación de los modelos estadísticos debido a las siguientes consideraciones:

1. Los individuos con un ingreso menor al salario básico ($\$450$), no tienen una buena capacidad de pago, pues lo más posible es que son personas que no tienen ingresos fijos o que no trabajan con relación de dependencia.

   Lo más probable para este grupo de personas es que se les niegue la solicitud de tarjeta de crédito, o que se les asigne el cupo mínimo disponible por parte de la entidad financiera.

2. Los individuos con un ingreso entre $\$2500$ o más, y que se les ha asignado un cupo en la tarjeta de crédito menor a los $\$1000$, también se excluyen del estudio. En este caso los ingresos de las personas son muy altos pero los cupos en las tarjetas de crédito son muy bajos.

   Estos individuos presentan mucho sesgo al modelo, ya que lo que se espera con el modelo es asignar un cupo elevado a una persona que tiene un ingreso muy elevado, pero eso no es lo que sucede en esta población. Es probable que para esta porción de la muestra, las personas utilizan su tarjeta para gastos pequeños; también puede ser que a las personas les interese pagar de contado todo lo que compran y así no generar un gasto innecesario por el pago de intereses.

3. Los individuos con un ingreso mayor a los $\$5000$ y que se les ha asignado un cupo de tarjeta entre $\$1000 - \$4000$, también causan

sesgo al modelo, y las causas pueden ser similares a lo que se explica en el punto 2.

Finalmente, luego de realizar los filtros necesarios a la base de datos proporcionada al inicio, y también tomando en cuenta las exclusiones de los individuos que van a causar sesgo dentro del modelo, se llega a tener una base de datos dividida con respecto a los cupos de la tarjeta de crédito.

En otras palabras, no se calcula un solo modelo para toda la población tarjetahabiente, sino que se separa a esta población en tres grupos diferentes, y se genera el mejor modelo posible para cada uno de estos grupos. En resumen, los grupos que se consideran en el presente proyecto están dados en la tabla(3.11).

Sujetos que tienen TC			
CUPO TC	**GRUPO**	**Sujetos**	**%**
Menor a 1000 usd	G1	48.708	22,8%
De 1000 a 4000 usd	G2	110.421	51,7%
Más de 4000 usd	G3	54.499	25,5%
Total		**213.628**	**100,0%**

Tabla 3.11: Base de datos Tarjetahabiente para modelar.

Por otro lado, si se vuelven a graficar las densidades de Ingreso Real y las densidades de Ingreso Estimado por el Buró, pero se lo realiza en cada sub grupo de la población tarjetahabiente; se observan grandes diferencias entre los tres grupos. Ver figuras 3.3 y 3.4.

Se debe tomar en cuenta que son densidades muy diferentes a pesar de que su tendencia es similar. Basta con fijarse en la escala de los gráficos para verificar que son densidades totalmente diferentes. En el eje X, se tiene la misma escala, mientras que en el eje Y la escala varía dependiendo de las frecuencias en el histograma asociado a la variable de ingresos.

El resultado final al que se desea llegar con los modelos descritos para este proyecto, es obtener densidades lo más similares posibles, almenos que se asemejen más las colas de los ingresos estimados, con las colas de los ingresos reales.

Además se debe tomar en cuenta que al finalizar la división de la base

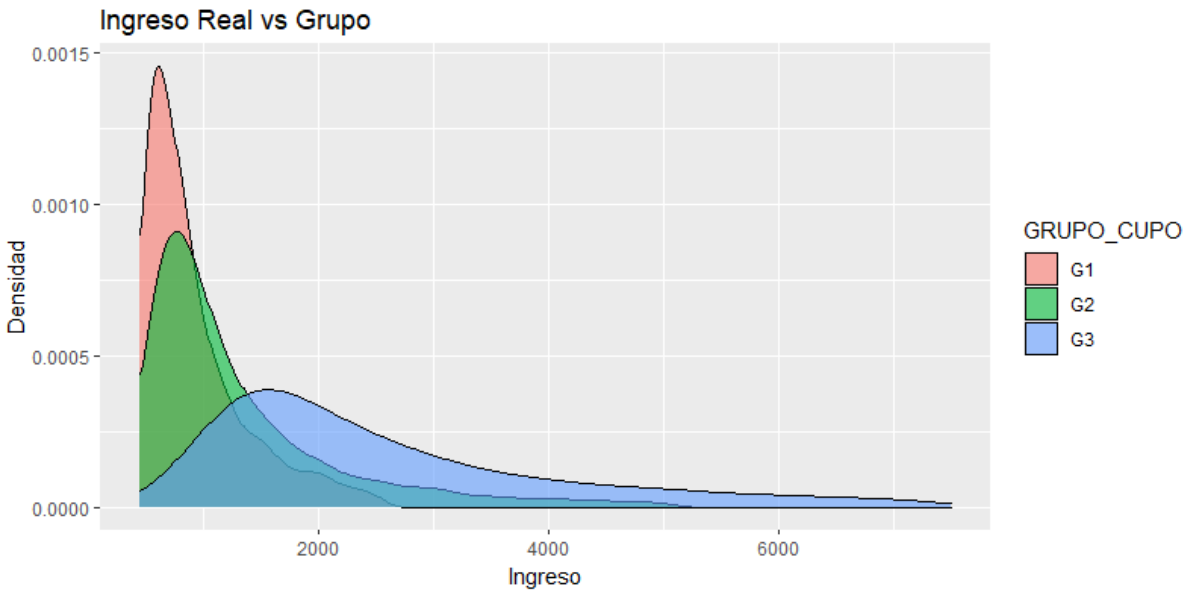


Figura 3.3: *Densidad Real en cada grupo: G1, G2, G3.*

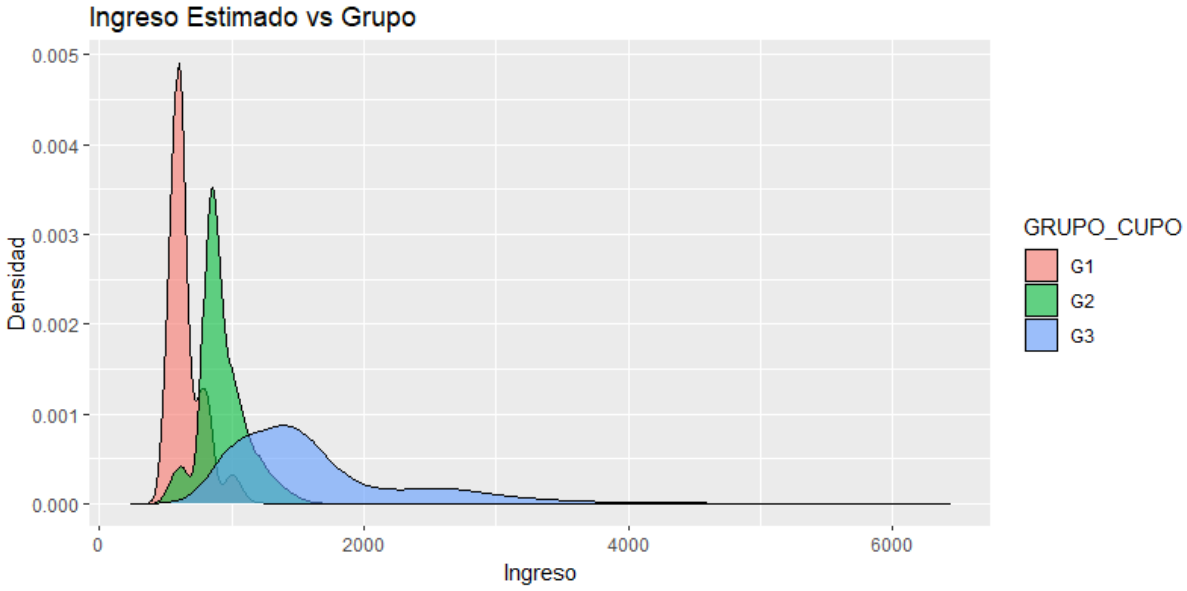


Figura 3.4: *Densidad Estimada en cada grupo: G1, G2, G3.*

original en tres grupos, se van a trabajar con 213,628 **individuos** y por las variables auxiliares de rangos que fueron creadas para la división y conteo de ingreso reales, el número de variables aumenta a 1180 **variables**.

3.3. Elección de variables

Hasta el momento se está trabajando con 1180 variables, las cuales están divididas entre numéricas y categóricas.

Las variables utilizadas para crear los modelos, son variables proporcionadas por Buró y de fácil replicación, y tienen que ver con deudas, cuotas, pagos, días de atrasos, cupos de TC, atrasos de pagos, y demás variables que se utiliza en el sistema financiero crediticio.

Creación de bases de datos de Train y Test

Para la base de datos train y la de test, se dividió la información en partes iguales, al 50% cada una. Esta partición fue pensada para crear un modelo que se adapte incluso a individuos con información muy alejada de la que se utiliza en la base de entrenamiento, y se debe al gran tamaño en la muestra.

Dado al gran tamaño de la muestra, resulta beneficioso construir y evaluar el modelo de manera efectiva mediante esta partición.

Creación y combinación de variables

Se identifican algunas variables adicionales que se generan como la suma de otras variables ya presentes en la base de datos original. Son las variables proporcionadas por Buró.

En este punto, el número total de variables aumenta hasta llegar a 1885. Es relevante destacar que algunas de las variables seleccionadas posteriormente mediante el test KS corresponden a las nuevas variables obtenidas al sumar otras variables preexistentes.

Ingresos Reales vistos como variable categórica

Para tener un resultado final más robusto en los tests KS y VI, se transforma la variable de Ingresos Reales, que es continua, hacia una variable de rangos de ingresos, que es discreta, y se añade esta variable a la base de datos actualizada hasta ese momento.

Recordar que se trabaja la población en tres grupos diferentes de individuos que tienen cupo en la tarjeta de crédito, por lo que los rangos de ingresos reales ahora se los calculan para cada grupo de la población tarjetahabiente.

Ingreso Real (Grupo 1)	
Percentil	**Valor**
30 %	600
70 %	990

Ingreso Real (Grupo 2)	
Percentil	**Valor**
30 %	800
70 %	1404

Ingreso Real (Grupo 3)	
Percentil	**Valor**
30 %	1671
70 %	3750

Los rangos de ingresos reales por grupo se calculan tomando en cuenta los percentiles 30 y 70. Al escoger los percentiles como valores en los rangos de ingreso real, se asegura la representatividad en la muestra que pertenece a cada rango.

Ahora, con esta nueva variable de rango de ingreso real por grupo, se procede a calcular los test KS y VI para escoger las variables que más aporten a describir el rango de ingresos real. Además, como el rango de ingresos real se calcula por grupo, se está escogiendo las mejores variables descriptivas para el mejor modelo, pero en cada grupo.

Selección de variables cuantitativas y cualitativas (Test KS y VI)

Como se mencionó previamente, se ejecutaron los tests de KS y VI para cada grupo dentro de la población, seleccionando conjuntos específicos de variables para cada uno de estos análisis.

Es importante destacar que en el test de KS se eligieron las variables cuyos porcentajes fueran mayores al $19{,}5\,\%$, y el mismo umbral se aplicó

en el test de VI. No se llevó a cabo un análisis de correlación entre las variables, ya que los modelos previstos para su utilización final son de naturaleza no paramétrica y tienden a considerar como equivalentes a variables con correlaciones. Por ende, la correlación entre variables no tiene influencia en el alcance del presente proyecto.

Las tablas con las variables escogidas para cada grupo se pueden observar en las tablas: A.1, A.2 y A.3. Las variables son diferentes de un grupo a otro, además cada grupo tiene un número diferente de variables:

1. El grupo 1 tiene un total de 20 variables numéricas y 1 categórica,
2. El grupo 2 tiene un total de 27 variables numéricas y 1 categórica,
3. El grupo 3 cuenta con 41 variables numéricas y ninguna categórica.

3.4. Representatividad en los nodos hoja

La representatividad en los nodos hoja es uno de los hiperparámetros de interés (para los modelos estadísticos), ya que se busca el porcentaje más adecuado en los nodos hojas de los árboles que se generan en cada iteración, de forma que no exista tanto subajuste ni sobreajuste en las estimaciones finales del modelo.

Para hallar los hiperparámetros más óptimos, se generan *grids de hiperparámetros*. Para hallar la grid, se generan 30 modelos diferentes de *Random Forest* con hiperparámetros diferentes y se comparan los MSE en cada caso.

Al final, se escoge un porcentaje de $3{,}6\,\%$ (como mínimo) para cada nodo hoja de cada árbol, creado para cada árbol de decisión. Este porcentaje se elige tomando en cuenta que, el MSE no varía mucho en cada combinación de parámetros; y también, para los tres grupos, se estarían generando árboles de decisión con 28 nodos hojas finales. Además, como máximo se generan 400 árboles de decisión por motivos de poder computacional.

En las tablas A.4, A.5, A.6, se observan las grids de hiperparámetros de cada grupo, junto con la combinación seleccionada para los modelos.

Tomar en cuenta que la selección de hiperparámetros se la realiza solamente para el modelo de Random Forest, ya que se utilizan los mismos hiperparámetros en los otros dos modelos no paramétricos.

Además, se calcula el grid para el Random forest ya que de los 3 modelos paramétricos es el más sencillo de interpretar. Por otro lado, en Gradient Boosting Machine y xgBoost se utilizan, a parte del porcentaje de 3.6 %, otros criterios de parada dentro de sus algoritmos, para la generación de árboles de decisión.

3.5. Función de balanceo para la muestra

Para la población tarjetahabiente, luego de realizar la partición en las tres poblaciones de estudio, se puede observar que existen individuos con ingresos elevados en cada grupo; pero también se deduce que los individuos de estudio se agrupan, con mucha densidad, alrededor de la media.

Esta forma irregular de la distribución muestral ocasiona las malas predicciones realizadas por Buró en un principio. Para tratar de solucionar este problema, se realizan remuestreos sobre las colas de la distribución de la base train. De esta manera lo que se busca es que, en las predicciones finales, las distribuciones de la estimación de ingresos se asemejen a las distribuciones reales de ingresos.

El remuestreo consiste en escoger individuos de forma aleatoria en la muestra, mezclar las características con otro individuo de la misma base, y finalmente guardar como individuo recién observado a la simulación realizada.

Se debe considerar también que, no necesariamente se desea añadir individuos con el remuestreo, en algunas simulaciones se retiran individuos de las colas y se agregan en la sección central del rango de datos para evitar subajustes o sobreajustes elevados.

La **función de balanceo** opera de una forma muy simple:

1. Se escogen dos percentiles, uno en la cola inferior y otro en la cola superior.

2. Se escogen dos nuevos porcentajes para los percentiles, uno en la cola inferior y otro en la cola superior.

3. En la cola izquierda, si se escoge un porcentaje para percentil mayor al actual, se realiza un remuestro en la cola para añadir individuos en la base hasta que el percentil inicial se convierta en un nuevo percentil, pero con el porcentaje nuevo escogido. En la cola derecha se realiza lo mismo.

4. En el intervalo central, se realiza un muestreo aleatorio para quitar individuos y que se nivelen las colas adecuadamente.

5. La base de datos creada con los remuestreos, se junta con la base de datos train, y se procede a calcular los modelos estadísticos necesarios.

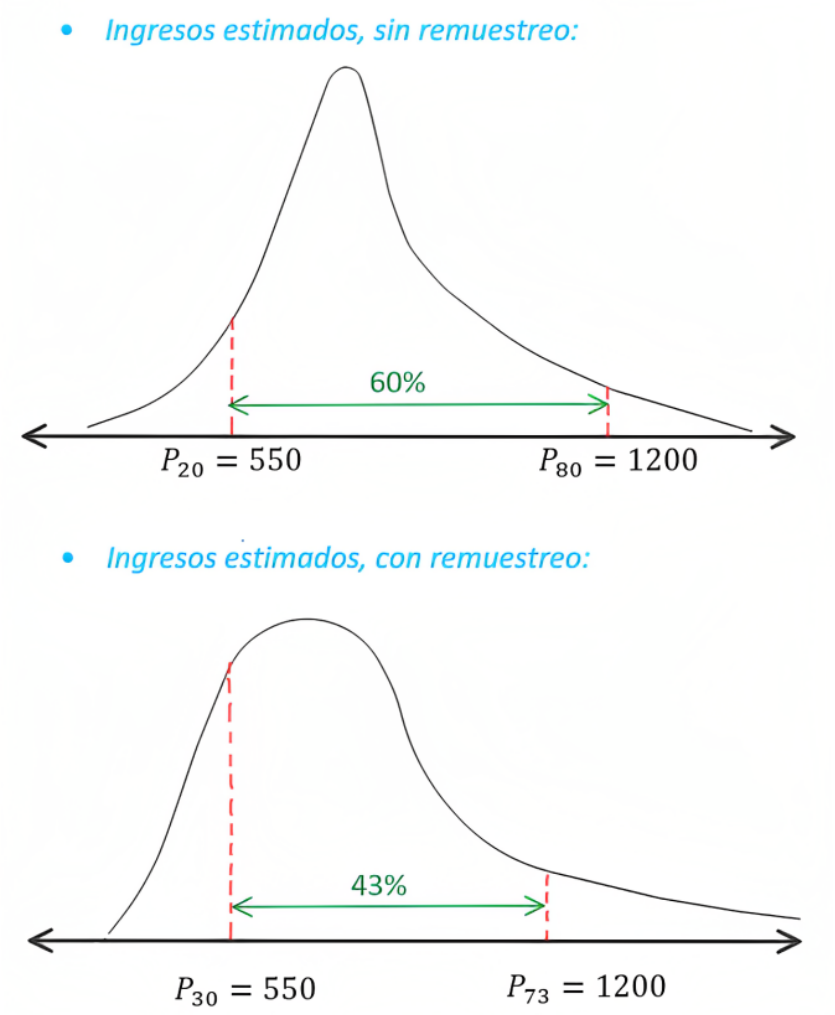


Figura 3.5: *Se puede observar que los percentiles que se escogen incialmente, se modifican en el balanceo de la muestra. Además, los individuos se distribuyen mejor en las colas.*

También, al realizar un balanceo o remuestreo de la base train, el modelo generado es más robusto y predice de mejor manera los ingresos de individuos que no formaron parte de esta base para generar el modelo.

La función de balanceo es muy importante, ya que antes de generar los modelos no paramétricos, se realiza un remuestreo sobre la base de entrenamiento. En cada remuestreo se escogen y modifican los parámetros de la función, y así se van obteniendo varios modelos diferentes para una misma muestra de entrenamiento.

3.6. Modelos para población G1

En la población de individuos tarjetahabientes se realiza una partición del conjunto muestral y se decide trabajar con tres grupos diferentes. Dado que cada grupo tiene características y comportamientos diferentes, si se calcula un solo modelo predictivo para los tres grupos que capture las características de los tres grupos, el modelo va a estar muy sesgado y no van a existir buenas predicciones en ninguno de los tres grupos. Al grupo 1 se lo denominará *G1*, al grupo 2 por *G2* y al grupo 3 por *G3*.

De esta forma, luego de haber realizado el tratamiento adecuado a la base de datos, se proceden a generar los modelos de estimación. Para cada grupo de individuos se calculan cuatro modelos diferentes: 3 modelos no paramétricos (Random Forest, Gradient Boosting Machine y xgBoost), y un modelo paramétrico (Regresión Lineal Múltiple). Por lo tanto, se calculan en total 12 modelos estadísticos diferentes, de los cuales se conservan al final los mejores 3 (1 modelo por cada grupo).

También se ha realizado varias simulaciones para decidir por el modelo predictivo más adecuado para cada grupo. Esta idea quiere decir que se podrían obtener modelos diferentes para grupos diferentes.

Se explicitarán y detallarán los resultados de los modelos para el G1, dado que para los grupos G2 y G3 el procedimiento es similar.

3.6.1. Modelo RF

El modelo Random Forest es el más sencillo de entender con respecto a los tres modelos no paramétricos que se usan en el proyecto. Por este motivo, se explica paso a paso lo que se realiza ya que lo mismo se replica para los otros modelos no paramétricos simulados; y después es lo mismo para los grupos *G2* y *G3*.

Para el *G1*, se utilizan los hiperparámetros escogidos de la tabla A.4, para todas las simulaciones de modelos. Luego, para el *G2* se utilizan los hiperparámetros escogidos de la tabla A.5. Finalmente para el *G3* se utilizan los hiperparámetros escogidos de la tabla A.6.

Antes de empezar a realizar los remuestreos sobre la base train, se calculan los modelos sobre la base inicial. Esto se lo realiza ya que a lo largo de la metodología se utilizan: matriz de coincidencia, métricas MSE-MAE, test Chi-cuadrado para la matriz de coincidencia y también métricas que ayudan a escoger el modelo más adecuado (se enfocan en medir el porcentaje de sub y sobre estimación).

El **primer modelo** que se genera **sin remuestreo** tiene muchas fallas, y es porque los valores de los estadísticos calculados no son para nada deseables. Se confirma la mala estimación observando las matrices de coincidencia: 3.12 y 3.13. Ambas matrices son las mismas, solo que la matriz de porcentajes es en la que se debe prestar atención, pues lo que se requiere para que sea un buen modelo es que la matriz sea lo más cercano a una matriz diagonal (diagonal principal y adyacentes de color verde, y las esquinas en rojo).

Real	Estimado					Total	%
	[450-550]	(550-700]	(700-850]	(850-1200]	(1200-2500]		
[450-550]	0,00	584,00	3302,00	906,00	28,00	4820,00	20 %
(550-700]	0,00	507,00	3659,00	1755,00	63,00	5984,00	25 %
(700-850]	0,00	210,00	2106,00	1687,00	169,00	4172,00	17 %
(850-1200]	0,00	90,00	1793,00	2366,00	727,00	4976,00	20 %
(1200-2500]	0,00	10,00	925,00	2066,00	1452,00	4453,00	18 %

Tabla 3.12: Matriz de coincidencias para Muestra Train

Por otro lado, también se calcula una matriz de coincidencia para la base de test. Solo se presenta la matriz de coincidencias en forma de porcentaje en la tabla 3.14. Es evidente la mala estimación que realiza el modelo,

Real	Estimado					MSE
	[450-550]	(550-700]	(700-850]	(850-1200]	(1200-2500]	
[450-550]	0 %	12 %	69 %	19 %	1 %	53133,45
(550-700]	0 %	8 %	61 %	29 %	1 %	44444,31
(700-850]	0 %	5 %	50 %	40 %	4 %	27597,35
(850-1200]	0 %	2 %	36 %	48 %	15 %	23203,67
(1200-2500]	0 %	0 %	21 %	46 %	33 %	380675,87

Tabla 3.13: Matriz de Porcentajes para Muestra Train

basta ver que las diagonales principales de las matrices de porcentajes de coincidencias no son de color verde; es más, en la primera columna de estas matrices no se ha estimado ningún individuo.

Luego, se procede a realizar el test Chi-cuadrado (al 95 % de confianza, con 16 grados de libertad) para ver qué tan cercana está la distribución de la matriz de coincidencias de la base train, con la matriz de coincidencia de la base test; esta comparación se la observa en la tabla 3.15.

Real	Estimado					MSE
	[450-550]	(550-700]	(700-850]	(850-1200]	(1200-2500]	
[450-550]	0 %	12 %	67 %	21 %	1 %	53133,45
(550-700]	0 %	8 %	60 %	30 %	1 %	44444,31
(700-850]	0 %	5 %	49 %	41 %	4 %	27597,35
(850-1200]	0 %	3 %	36 %	47 %	14 %	23203,67
(1200-2500]	0 %	1 %	22 %	46 %	31 %	380675,87

Tabla 3.14: Matriz de Porcentajes para Muestra de Test

Real	Estimado						
	[450-550]	(550-700]	(700-850]	(850-1200]	(1200-2500]	Calculado	Teórico
[450-550]	0,0	1,9	3,8	8,2	0,0	**205,8**	**26,3**
(550-700]	0,0	1,7	4,3	0,7	0,8	**Decisión**	
(700-850]	0,0	0,7	3,0	0,0	1,3	Se rechaza H0	
(850-1200]	0,0	21,5	0,3	0,8	0,3		
(1200-2500]	0,0	152,1	3,9	0,5	0,2		

Tabla 3.15: Test Chi-cuadrado para matriz de coincidencia.

Cabe recalcar que para calcular el estadístico en el test Chi-cuadrado, se utiliza la matriz de coincidencias y no la de porcentajes, para train y test. Se hace énfasis en que la matriz de porcentajes es muy útil para ver la manera en que el modelo está estimando los valores reales, y por otro lado, la matriz con el cálculo del estadístico Chi-cuadrado nos indica qué tan similares son las distribuciones de individuos estimados con la base train y con la base test.

Además, también se calculan otros estadísticos que sirven para comparar con los resultados de otros modelos, en específico, se calculan: MSE,

MAE, porcentaje de cruces, y porcentajes de sub y sobre estimación. Estos cálculos se los puede observar en la tabla 3.16.

Base Train	Métricas		*Base Test*	Métricas
MSE	139351,11		**MSE**	147337,24
MAE	274,03		**MAE**	281,29
Cruce	26,35 %		**Cruce**	25,77 %
Cruce +/-	70,30 %		**Cruce +/-**	69,46 %
SubEstima	4,20 %		**SubEstima**	4,81 %
SobreEstim	25,50 %		**SobreEstim**	25,73 %

Tabla 3.16: Estadísticos de comparación

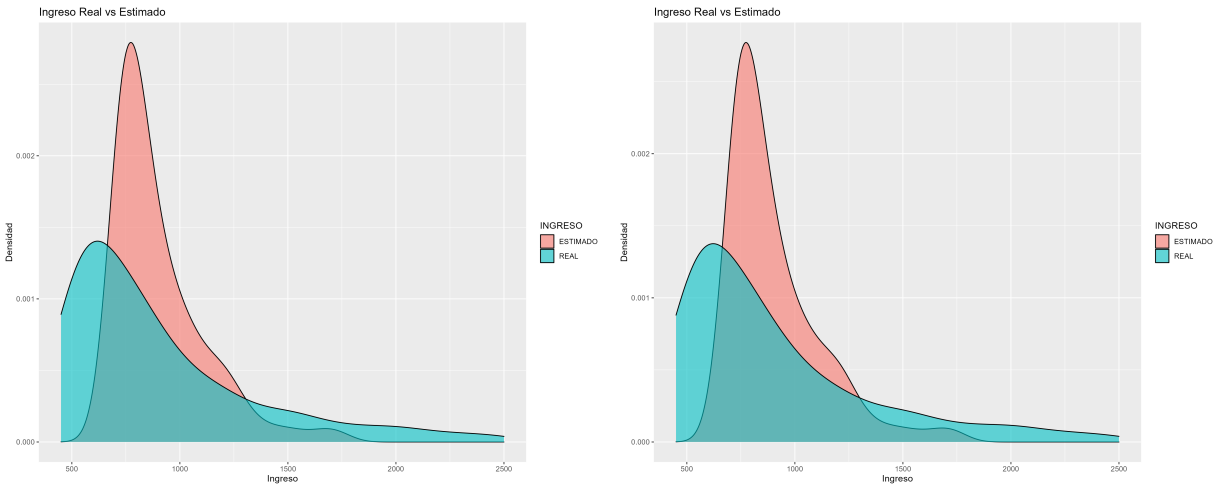
Es importante notar que: *(Cruce +/-) + (SubEstima) + (SobreEstima) = 100 %*. Para que un modelo sea bueno, el *Cruce +/-* tiene que ser mayor al 70 %. Además, según la lógica de que una entidad no desea dar un cupo de tarjeta de crédito de USD$5000 a una persona que gane USD$450 (es decir, sobreestimar los ingresos del individuo), el porcentaje de *SobreEstim* debe ser lo mínimo posible, sin caer ahora en un subajuste.

En la tabla de estadísticos de comparación, se observa una sobreestimación muy elevada, del 25 %. Esta sobreestimación es muy mala en el ejemplo hipotético del párrafo anterior, dado que la entidad financiera no va a dudar en brindar tarjetas créditos a los individuos que soliciten pero que tengan un salario igual o menor al mínimo, lo que representarían pérdidas grandes si el individuo no paga su deuda.

Finalmente, para asegurar que se están realizando buenas (o malas) estimaciones, se procede a ver las gráficas de densidad de los ingresos reales y los ingresos estimados. Estas densidades se observan en la gráfica 3.6.

Para el primer modelo sin remuestreo, no se observan diferencias notables en las densidades de la base train y de la base test. Lo que si se observa claramente es un comportamiento no tan deseable para las densidades de las estimaciones. La altura máxima de las densidades estimadas representa casi el doble que la altura máxima de las densidades reales; esto se traduce a que alrededor de la media de los datos estimados se tiene aproximadamente el doble de individuos que alrededor de la media en los valores reales. Las densidades de las estimaciones tienen un comportamiento como el de la gráfica 3.5.

Ahora, al volver a observar las tablas 3.13, 3.14, 3.15, 3.16, y la gráfica



((a)) Densidad base train ((b)) Densidad base test

Figura 3.6: Densidades reales y estimadas

3.6; la mejor opción que se tiene es **realizar un remuestreo (balanceo)** para la muestra de la base train. Esta decisión es tomada en base a los comentarios negativos que se realizan durante el cálculo de las tablas y los gráficos.

NOTAS :

1. En todos los modelos que se generan para el *G1*, las matrices de coincidencia son cuadradas y los rangos reales y estimados son los mismos (los rangos no son los mismos para *G2* y *G3*, pero son exclusivos en cada grupo, es decir, se calculan 1 sola vez en cada grupo para los ingresos reales y se mantienen para todos los modelos y análisis de ese grupo).

2. Los rangos que se utilizan en las matrices de coincidencia y en el test Chi-cuadrado, representan los percentiles al 20 % de los ingresos reales. Se considera que cada grupo tiene un rango de ingresos real diferente uno del otro, pero en los tres grupos los rangos se los obtiene con los percentiles al 20 %.

   El rango de ingresos reales se escoge para que exista una buena representatividad de los individuos en cada rango, y las comparaciones no sean tan sesgadas.

3. Se debe tomar en cuenta que la división por rangos no es tan precisa, en el sentido que si un individuo en el G1 tiene un ingreso real

de USD$540 y en la estimación con el modelo obtiene un salario de USD$560, en la matriz de coincidencias este individuo va a formar parte del segundo intervalo y ya no del primero.

Esta observación sugiere que se utilicen técnicas de **matemática difusa** para que no pasen este tipo de eventos. En este proyecto no se utiliza la técnica mencionada, pues sale del alcance del proyecto planteado.

4. Por último, para cada modelo simulado, se realizan exactamente los mismos cálculos y test estadísticos para llegar a la conclusión de que la muestra necesita otro remuestreo. Esto se realiza hasta obtener un modelo que tenga una matriz de coincidencias diagonal, un Chi-cuadrado calculado pequeño, y tenga un bajo porcentaje de sobreestimación.

Hiperparámetros para remuestreo

Después de decidir que sí se va a realizar un remuestreo de la base train para que los resultados que se estiman sean más cercanos a la información real, es el momento adecuado para decidir por los hiperparámetros de la función de balanceo.

Para este objetivo, se debe observar nuevamente la gráfica de densidades estimadas para la base train sin remuestreo. Ver gráfica 3.7.

Se observa que aproximadamente para un ingreso de USD$700, la densidad de la estimación crece mucho en comparación con la densidad real, por lo tanto se deben elegir valores cercanos al 700 para el percentil de la izquierda. Por otro lado, aproximadamente en el ingreso de USD$1300, se observa que las estimaciones empiezan a decrecer rápidamente, dejando al gráfico de densidad estimada muy por debajo del gráfico de densidad real; por lo tanto se deben elegir valores cercanos al 1300 para el percentil de la derecha.

Con esta observación en mente, se deben escoger los hiperparámetros hasta llegar a obtener un modelo que se ajuste lo mejor posible a los datos reales, tomando en cuenta los estadísticos de comparación. En el proyecto se realizan **varias simulaciones de remuestreo** para cada mo-

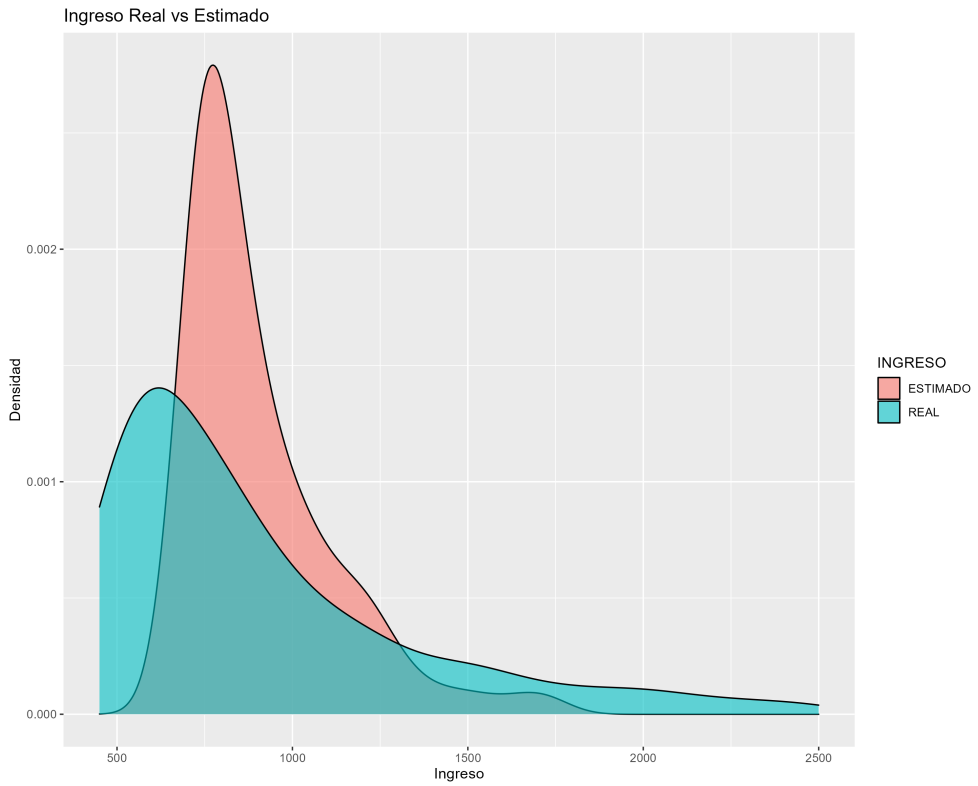


Figura 3.7: *Densidad real y estimada de la base train sin remuestreo.*

delo de predicción, y de estos experimentos se conserva aquel con mejores valores en los estadísticos de comparación. Además de los estadísticos de comparación, las densidades reales y estimadas también dejan de ser tan diferentes, lo que indica que los resultados de la estimación con el remuestreo tienen una distribución más cercana a la original.

No se debe olvidar el uso de las matrices de porcentajes de coincidencias, pues solo con la distribución empírica ó buenos estadísticos de comparación, no se aseguran las buenas predicciones de los datos.

Se debe tomar en cuenta que, la elección de los hiperparámetros **es muy sensible a los cambios**, es decir, pequeños cambios en la elección de estos hiperparámetros representan un gran cambio en los test estadísticos y estimación final.

Después de un exhaustivo proceso de simulaciones con distintos hiperparámetros, se presentan los hiperparámetros que se escogen para que se genere el mejor modelo. Ver la tabla 3.17.

G1_corte1 = 499 // percentil 6%
G1_corte2 = 1200 // percentil 80%
G1_porc1 = 0.3
G1_porc2 = 0.1

Tabla 3.17: Hiperparámetros considerados para el remuestreo

Remuestreo

Se procede a utilizar los hiperparámetros considerados para el remuestreo. Cuando se balancea la muestra, automáticamente los modelos que se generan van a recoger otras características que son más influyentes en el G1.

Se puede observar que la densidad de los ingresos reales se modifica un poco al momento de realizar el remuestreo. Ver gráfica 3.8.

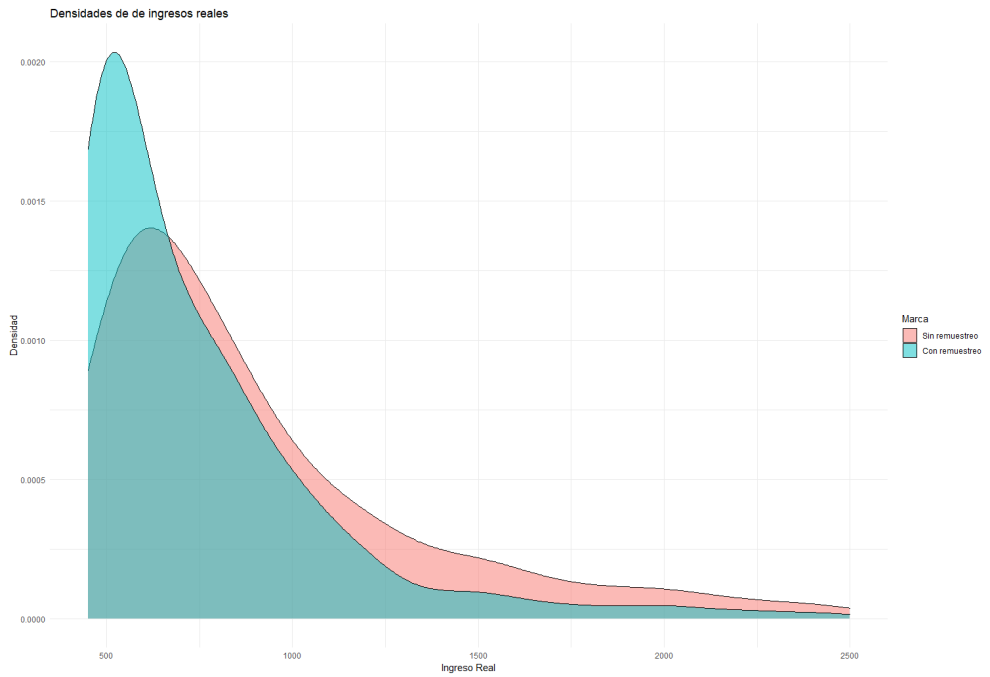


Figura 3.8: *Densidad real de la base train, con y sin remuestreo*

También, está claro en la gráfica lo que se esperaba al remuestrear con los parámetros escogidos: la cola de la izquierda ahora tiene más peso, mientras que la cola de la derecha ahora tiene menor peso. Aún así, se puede observar en el gráfico que no existe una gran diferencia en la tendencia de ambas densidades.

Por lo tanto, queda confirmado que los parámetros escogidos para el re-

muestreo no subestiman ni sobreestiman en exceso al ingreso real.

Representatividad en los nodos hojas en cada árbol

Al momento que se realiza el remuestreo de la base train, cambian de valor todos los percentiles, menos el mínimo y el máximo. Por esta razón, el valor mínimo de individuos en los nodos hoja que se calcula ya no es el mismo que para la base train sin remuestro. Ver tabla A.4.

Se considera que el porcentaje mínimo de individuos en los nodos hoja se mantiene en la base train remuestreada, es decir, se conserva el resultado de *3.6%* y lo que se realiza con este porcentaje es calcular nuevamente el percentil adecuado de la distribución de datos de la base train con resmuestreo.

En este caso se puede calcular el percentil *3.6%*, pero también es posible multiplicar el número de individuos de la base train remuestreada por el *3.6%* y obtener el mismo resultado. El valor calculado representa ya no el porcentaje mínimo, sino el valor mínimo de individuos en los nodos hoja, y este último es el hiperparámetro buscado.

Creación del modelo la base train remuestreada

En el algoritmo de Random Forest se necesita crear varios árboles de decisión para poder comparar los resultados y realizar un promedio de los mismos, así obtener la estimación de los ingresos reales.

Tomando en cuenta que en cada árbol generado se escoge de manera aleatoria (con distribución uniforme discreta) las variables originales de estudio, el último hiperparámetro para el algoritmo es el número de variables que se deben escoger en cada árbol generado, *mtry*.

Se escoge el hiperparámetro $mtry = \frac{\#\text{de variables en el grupo}}{3}$, por lo tanto, para el G1 se tendría $mtry = 7$. Esta asignación del número de variables que se escogen en cada árbol, se realiza para que disminuya la probabilidad de que se generen dos árboles iguales, ya que se tendrían: $21C7 = \frac{21!}{(21-7)! \cdot 7!} = 116\,280$ combinaciones de variables posibles.

El valor de las combinaciones indica que, cada combinación posible de

variables para generar el árbol, tiene una probabilidad de 1 entre 116 280 de que sea escogida. Esto es muy bueno para propósitos de estimación, ya que se disminuyen posibles sesgos (almenos al generar los árboles).

Con este último hiperparámetro, ya se han completado los requisitos en los hiperparámetros para poder generar el modelo de Random Forest para el G1.

Se presenta la línea de código genérica, que construye un modelo de Random Forest en el lenguaje R:

```
RF_G1<-ranger("formula", "base.datos", "num.arboles", "mtry", "num.nodo
    .hoja","importance" = 'impurity', "write.forest" = TRUE, "seed")
```

En este código se observa el hiperparámetro *importance* y lo que realiza son los cálculos con el coeficiente de Gini. Recordar que existe una relación entre la impureza para árboles de clasificación y la impureza de árboles de regresión, dada en la ecuación 2.3.

El hiperparámetro *write.forest* guarda en la memoria al modelo generado, para que posteriormente se pueda guardar un archivo con formato *.rds* y que el modelo ya no se genere cada ves que se utiliza, sino que solo se cargue el modelo para ser usado, como si fuese un archivo *.csv*.

El hiperparámetro *seed* es la semilla que se considera dentro del algoritmo para escoger de forma uniforme las variables para usar en cada árbol de decisión, y los individuos de la base train remuestrada, para remuestrearlos nuevamente y aumentar la muestra.

Por último, el resto de los hiperparámetros de la función *ranger*, son aquellos que se detallan y explican desde la elección de las variables para el G1.

Predicción del grupo G1 sobre toda la base

Después de generar el modelo de Random Forest para la base train con remuestreo, es el momento de evaluar el modelo realizando las predicciones de todos los individuos en el G1, no solamente train sino también los individuos de test.

Para evaluar el modelo generado, también se calcula el tiempo de ejecución del modelo para predecir toda la base de G1. Se presentan los valores

en la tabla 3.18.

Tiempo de ejecución		
user	system	elapsed
13.5	0.54	13.5

Tabla 3.18: Tiempos de ejecución de algoritmo RF para G1

Se debe tomar en cuenta que, los tiempos de ejecución varían mucho de un ordenador a otro. Sin embargo, se puede considerar que el modelo más óptimo para aplicar en la parte industrial o financiera, es aquel que menos tiempo le toma en calcular las estimaciones.

En los cálculos estadísticos posteriores se filtra la base G1 de valores estimados para los ingresos nuevamente, con los individuos considerados en la base train y aquellos de la base test. Esto se debe realizar para poder graficar las densidades reales y empíricas, además para construir la matriz de coincidencia y calcular los estadísticos de comparación.

Una ves que se realizan las predicciones con el modelo Random Forest para la base train remuestreada, se procede a realizar las matrices de coincidencia que se explicaron detalladamente con el modelo Random Forest de la base train sin remuestreo.

Matriz de coincidencias

Se presenta la matriz de porcentajes de coincidencias del modelo Random Forest con la base train remuestreada y también para la base test en la tabla 3.19 y 3.20 respectivamente.

Real	Estimado					MSE
	[450-550]	(550-700]	(700-850]	(850-1200]	(1200-2500]	
[450-550]	0%	72%	23%	5%	0%	37113.05
(550-700]	0%	58%	30%	12%	0%	19147.38
(700-850]	0%	44%	32%	24%	0%	20116.64
(850-1200]	0%	29%	30%	34%	7%	73864.93
(1200-2500]	0%	16%	24%	40%	20%	685494.47

Tabla 3.19: Matriz de porcentaje de coincidencias para la base train

Al comparar los porcentajes de estimación con la tabla de porcentajes para la base train sin remuestrear (3.13), es claro que las estimaciones son superiores.

Real	Estimado					MSE
	[450-550]	(550-700]	(700-850]	(850-1200]	(1200-2500]	
[450-550]	0 %	68 %	27 %	5 %	0 %	40325.16
(550-700]	0 %	57 %	31 %	12 %	0 %	19722.83
(700-850]	0 %	44 %	32 %	24 %	0 %	20466.71
(850-1200]	0 %	30 %	29 %	34 %	6 %	75884.02
(1200-2500]	0 %	17 %	24 %	40 %	19 %	698838.50

Tabla 3.20: Matriz de porcentaje de coincidencias para la base test

A pesar de que las matrices de coincidencias creadas con la base train remuestreada no son de color verde en sus diagonales principales, se nota que los porcentajes se han movido hacia la izquierda, y se asemeja más a la matriz diagonal deseada. Además, los estadísticos MSE por rangos, también presentan mejoras muy notables.

En este mismo sentido, se procede a realizar el test Chi-cuadrado, y se puede observar también una mejora muy significativa en el estadístico calculado para el test, ver tabla 3.21. Recordar que este test solo dice si la matriz de coincidencia de la base train es similar a la matriz de coincidencia de la base test.

Además, se rechaza que las distribuciones de las estimaciones de la base train y test sean similares, pero se debe tomar en cuenta que siempre suceden los eventos a considerar con la *matemática difusa* que provocan cambios en el estadístico calculado.

Real	Estimado						
	[450-550]	(550-700]	(700-850]	(850-1200]	(1200-2500]	Calculado	Teórico
[450-550]	0.0	12.4	19.9	0.0	0.0	**60.6**	**26.3**
(550-700]	0.0	4.1	0.2	0.2	0.0	**Decisión**	
(700-850]	0.0	0.9	0.4	0.0	1.7	Se rechaza H0	
(850-1200]	0.0	4.4	2.4	0.0	0.5		
(1200-2500]	0.0	13.0	0.1	0.1	0.3		

Tabla 3.21: Matriz de test Chi-cuadrado

Métricas

En la matriz de porcentajes de coincidencias ya se presentó el estadístico MSE calculado por rangos, y se observa claramente disminución del valor de este estadístico.

De forma similar, también existen mejoras en las métricas MSE-total, MAE, cruces, subestimaciones y sobreestimaciones. Se puede observar

estas mejoras en la tabla 3.22. La tabla de estadísticos de comparación del modelo Random Forest sin remuestrar la base train, es 3.16.

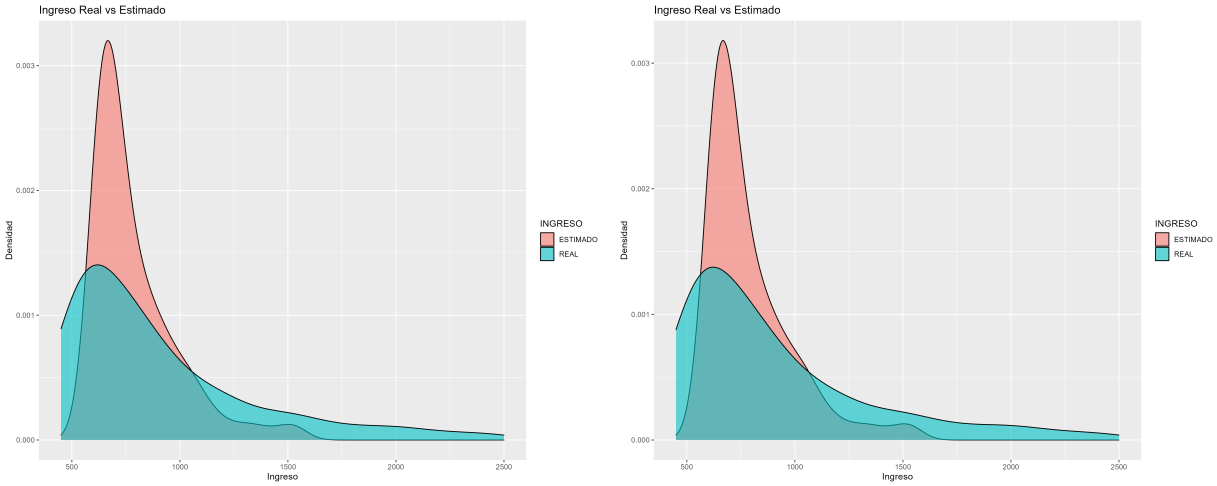
Entrenamiento	Métricas
MSE	155601.24
MAE	259.87
Cruce	30 %
Cruce +/-	78 %
SubEstima	13 %
SobreEstim	9 %

Validación	Métricas
MSE	163054.52
MAE	267.45
Cruce	30 %
Cruce +/-	77 %
SubEstima	14 %
SobreEstim	9 %

Tabla 3.22: Estadísticos de comparación con base train remuestreada

Se observa que el estadístico MSE aumentó de valor lo cual no es tan bueno teóricamente, pero al mismo tiempo el resto de estadísticos han mejorado bastante, en especial el porcentaje de sobreestimación, que del 25 % inicial ha disminuido hasta el 9 %, pero también se ha aumentado el porcentaje de cruce en las diagonales principales, lo que indica estimaciones más cercanas para el ingreso real.

Densidades reales y estimadas



((a)) Densidad base train

((b)) Densidad base test

Figura 3.9: Densidades reales y estimadas

Luego de verificar que las matrices de coincidencia son más cercanas al resultado deseado, es posible realizar las gráficas de las densidades como comparación final. Este paso se realiza al final, pues es posible que las frecuencias de los valores estimados sean muy similares a las

frecuencias reales, pero las estimaciones difieran exageradamente con los valores reales. Ver figura 3.9.

Al observar detenidamente, y comparar las densidades estimadas (remuestreo) con las densidades estimadas (sin remuestreo, figura 3.6), se puede confirmar que se ha disminuido el porcentaje de sobreajuste, pero ahora se está subajustando un poco más. Este subajuste no es tan grave, pues la entidad financiera no perdería tanto dinero, contrario a cuando se tiene un sobreajuste.

Por último, se nota que existe una relación directa entre las matrices de coincidencia y las densidades estimadas con remuestreo: las gráficas de las densidades se desplazaron hacia la izquierda, mientras que en las matrices de coincidencia los porcentajes también se desplazaron hacia la izquierda.

3.6.2. Modelo Gradient Boosting Machine (GBM)

Para el modelo de GBM, se procede de igual forma que en el modelo de Random Forest. Por este motivo, se presentan solamente los resultados con los mejores parámetros.

Hiperparámetros para remuestreo

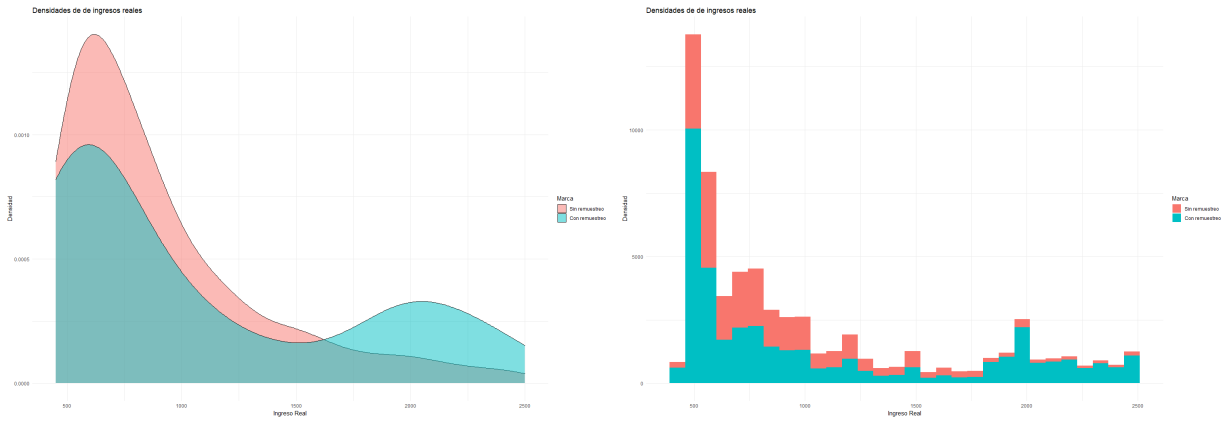
Los mejores hiperparámetros hallados para el modelo GBM sobre el G1, se presentan en la tabla 3.23.

G1_corte1 = 549 // 20%
G1_corte2 = 1815.0836 // 94%
G1_porc1 = 0.49
G1_porc2 = 0.40

Tabla 3.23: Mejores hiperparámetros para GBM sobre G1

Remuestreo

Al aplicar la función de remuestreo con los mejores hiperparámetros, se puede observar que las densidades de ingresos reales varían, pero en los histogramas no se observa una diferencia tan elevada. Ver figura 3.10



((a)) Densidades Reales ((b)) Histogramas reales

Figura 3.10: Comparación de base train real con la remuestreada

Es claro que en el remuestreo, las colas de la derecha e izquierda en el histograma han aumentado, pero esto ayuda a predecir de mejor manera

los valores en las colas. Además, ya se van a observar las matrices de coincidencia y estadísticos de comparación no están alejados de lo que se espera.

Representatividad en los nodos hojas en cada árbol

Para la generación del modelo GBM, se utiliza el mismo porcentaje encontrado para el modelo RF, es decir, el 3.6 % de representatividad en los nodos hoja.

Creación del modelo para la base train remuestreada

Para la creación del modelo, se presenta el código genérico para GBM en el lenguaje R:

```
rgbm_g1 <- gbm("formula", "datos", "n.trees", "n.min.nodo", "shrinkage"
    , "distribution")
```

A diferencia de la función para el RF, se tienen los parámetros *shrinkage* y *distribution*. El hiperparámetro shrinkage representa el coeficiente de aprendizaje del algoritmo, mientras que el parámetro distribution representa la distribución de los errores que el modelo intenta modelar.

Predicción del grupo G1 sobre toda la base

El tiempo de ejecución del modelo GBM sobre toda la base se presenta en la tabla 3.24. De forma similar que en el modelo Random Forest, luego de predecir sobre toda la base se realizan filtros para el G1 y sus respectivas bases train y test.

Tiempo de ejecución		
user	system	elapsed
9,18	0,01	9,18

Tabla 3.24: Tiempos de ejecución de algoritmo GBM para G1

Matriz de coincidencias

Las mejores estimaciones del modelo se presentan en las tablas 3.25 y 3.26.

Real	Estimado					MSE
	[450-550]	(550-700]	(700-850]	(850-1200]	(1200-2500]	
[450-550]	20 %	58 %	12 %	8 %	2 %	56682,85
(550-700]	15 %	52 %	15 %	13 %	5 %	55292,79
(700-850]	11 %	42 %	16 %	20 %	12 %	87910,36
(850-1200]	5 %	33 %	14 %	22 %	26 %	177439,03
(1200-2500]	3 %	21 %	11 %	21 %	44 %	581830,25

Tabla 3.25: Matriz de proporción de coincidencias de la base train

Real	Estimado					MSE
	[450-550]	(550-700]	(700-850]	(850-1200]	(1200-2500]	
[450-550]	20 %	57 %	13 %	7 %	2 %	56920,47
(550-700]	14 %	52 %	16 %	13 %	4 %	54811,59
(700-850]	11 %	42 %	16 %	19 %	12 %	91912,99
(850-1200]	6 %	33 %	14 %	22 %	25 %	178005,80
(1200-2500]	3 %	22 %	10 %	21 %	44 %	581084,86

Tabla 3.26: Matriz de proporción de coincidencias de la base test

Por otro lado, la matriz de cálculo para el test Chi-cuadrado está dado por la tabla 3.27.

Real	Estimado					Calculado	Teórico
	[450-550]	(550-700]	(700-850]	(850-1200]	(1200-2500]	**31.8**	**26.3**
[450-550]	0.0	2.2	2.4	0.2	0.0	**Decisión**	
(550-700]	2.4	1.1	0.9	0.0	3.6	Se rechaza H0	
(700-850]	1.1	0.1	0.5	2.1	0.4		
(850-1200]	8.6	0.6	0.0	0.0	0.1		
(1200-2500]	0.8	2.1	0.1	0.9	1.7		

Tabla 3.27: Cálculo del Chi-cuadrado para comparar distribución train y test

Se rechaza la hipótesis nula, sin embargo el estadístico calculado es suficientemente pequeño.

Métricas

Los estadísticos de comparación se los observa en la tabla 3.28.

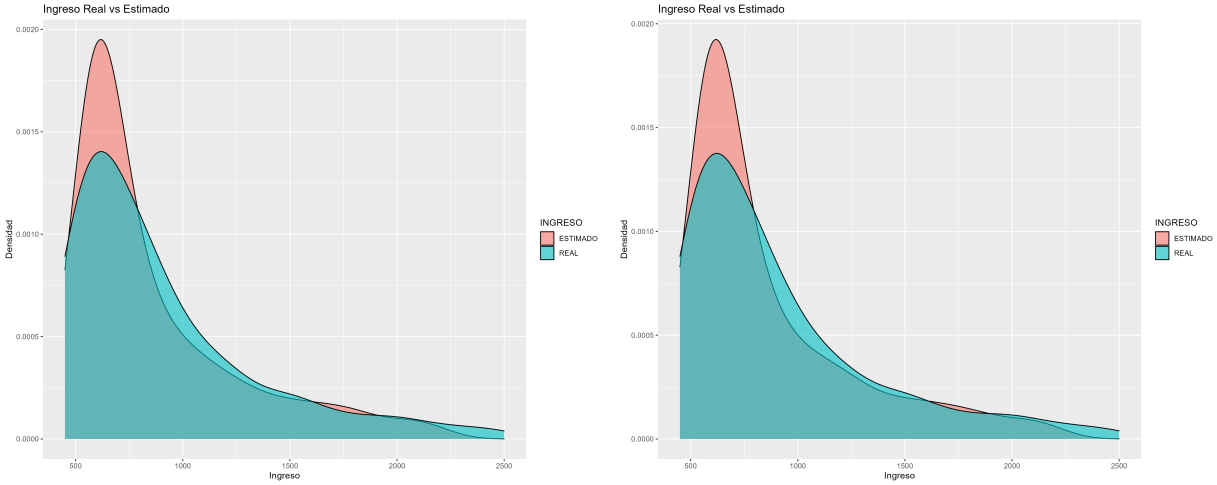
Entrenamiento	Métricas
MSE	182121.38
MAE	292.50
Cruce	32 %
Cruce +/-	73 %
SubEstima	16 %
SobreEstim	11 %

Validación	Métricas
MSE	185701.38
MAE	296.22
Cruce	32 %
Cruce +/-	73 %
SubEstima	16 %
SobreEstim	11 %

Tabla 3.28: Estadísticos de comparación para el modelo GBM

Densidades reales y estimadas

Por último se presenta las gráficas de las densidades reales y estimadas con el modelo de GBM en la figura 3.12. Se observa una mejora muy buena en la estimación de la densidad, pues en la cola derecha no existe sobreestimación de individuos, aunque aún existe un poco de subestimación sobre la cola izquierda.



((a)) Densidad base train ((b)) Densidad base test

Figura 3.11: Densidades reales y estimadas

3.6.3. Modelo XGBOOST

Para el modelo de xgBoost, se procede de igual forma que en el modelo de Random Forest. Por este motivo, se presentan solamente los resultados con los mejores parámetros.

Hiperparámetros para remuestreo

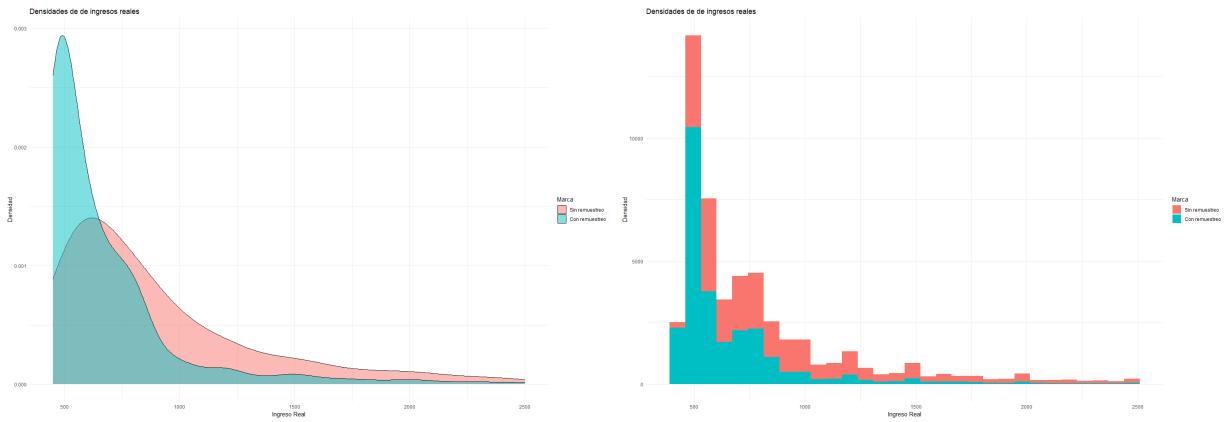
Los mejores hiperparámetros hallados para el modelo xgBoost sobre el G1, se presentan en la tabla

G1_corte1 = 481 // 4 %
G1_corte2 = 850 // 61 %
G1_porc1 = 0.40
G1_porc2 = 0.15

Tabla 3.29: Mejores hiperparámetros para modelo xgBoost

Remuestreo

La comparación de la densidad de ingresos reales con las bases sin remuestrear y remuestreada, está dada en la figura 3.12.



((a)) Densidades Reales ((b)) Histogramas reales

Figura 3.12: Comparación de base train real con la remuestreada

Representatividad en los nodos hojas en cada árbol

Para la generación del modelo XGB, se utiliza el mismo porcentaje encontrado para el modelo RF, es decir, el 3.6 % de representatividad en los nodos hoja.

Creación del modelo para la base train remuestreada

Se presenta la función utilizada para el algoritmo de xgBoost en el lenguaje R. Por la forma en que se programó la función, es necesario separar la base en dos pedazos, la matriz con los datos X y el vector de la variable dependiente Y.

```
my_xgb_g1 <- h2o.xgboost(x = x_g1_em,
                         y = y_em,
                         model_id = "XGB",
                         training_frame = rmod_g1_em,
                         ntrees = 300,
                         learn_rate = 0.03,
                         max_depth = 7,
                         min_rows = g1_min_nodo,
                         nfolds = 5,
                         fold_assignment = "Auto",
                         keep_cross_validation_predictions = TRUE,
                         seed = 12345,
                         stopping_rounds = 500,
                         stopping_metric = "RMSE",
                         stopping_tolerance = 0)
```

Para entender mejor la función, se explica cada hiperparámetro desconocido dado en la función:

1. **x:** Este parámetro especifica las columnas del conjunto de datos que se utilizaron como características (variables independientes) para entrenar el modelo. Se proporciona una lista de nombres de las variables que corresponden a las columnas de la base.

2. **y:** Se refiere al nombre de la columna del conjunto de datos que se utilizará como variable objetivo (variable dependiente) durante el entrenamiento del modelo XGBoost.

3. **model_id:** Se utiliza para asignar un identificador único al modelo que se está entrenando con la función h2o.xgboost. Esto permite asignar un nombre significativo al modelo para facilitar su identificación en análisis posteriores. Si no se proporciona un valor, H2O generará automáticamente un nombre para el modelo.

4. **training_frame:** Este parámetro especifica el conjunto de datos de entrenamiento que se utilizará para entrenar el modelo XGBoost. Debes proporcionar un objeto de tipo H2OFrame que contenga tus datos de entrenamiento.

5. **min_rows:** Determina el número mínimo de individuos requeridos en un nodo para que se realice una partición durante la construcción del árbol en el proceso de entrenamiento. Este valor ayuda a controlar el tamaño mínimo de las hojas del árbol y puede contribuir a evitar el sobreajuste.

6. **nfolds:** El hiperparámetro "nfoldsrepresenta el número de rangos utilizados en el proceso de matriz de crear una matriz de coincidencia de forma automática.

7. **fold_assignment:** Determina cómo se asignan los individuos a los rangos durante la validación cruzada. El valor .ªUTO.ªsigna los rangos automáticamente por H2O.

8. **keep_cross_validation_predictions:** Cuando se establece en TRUE, este parámetro conserva las predicciones realizadas durante la validación cruzada. Esto puede resultar útil para análisis posteriores de las predicciones de validación cruzada.

9. **stopping_rounds:** Representa el número de rondas de entrenamiento en las que no se observa mejora en la métrica de parada antes de detener el proceso de entrenamiento. Ayuda a evitar un entrenamiento excesivo y ahorra tiempo computacional.

10. **stopping_tolerance:** Un valor más bajo de "stopping_tolerance"significa que se requiere una mejora más pequeña para que el entrenamiento continúe, mientras que un valor más alto exige una mejora más sustancial para seguir entrenando el modelo. Este parámetro ayuda

a controlar cuánta paciencia debe tener el algoritmo antes de detenerse en caso de que las mejoras sean pequeñas y no valga la pena seguir entrenando.

Predicción del grupo G1 sobre toda la base

El tiempo de ejecución del modelo xgBoost sobre toda la base se presenta en la tabla 3.30. De forma similar que en el modelo Random Forest, luego de predecir sobre toda la base se realizan filtros para el G1 y sus respectivas bases train y test.

Tiempo de ejecución		
user	system	elapsed
9,5	0,03	9,8

Tabla 3.30: Tiempos de ejecución de algoritmo xgBoost para G1

Matriz de coincidencias

Las mejores estimaciones del modelo xgBoost se presentan en las tablas 3.31 y 3.32.

Real	Estimado					MSE
	[450-550]	(550-700]	(700-850]	(850-1200]	(1200-2500]	
[450-550]	6 %	82 %	10 %	2 %	0 %	20273,19109
(550-700]	3 %	73 %	17 %	7 %	0 %	13864,64652
(700-850]	2 %	60 %	21 %	15 %	2 %	32803,33824
(850-1200]	1 %	45 %	22 %	21 %	11 %	107623,7711
(1200-2500]	1 %	28 %	19 %	28 %	24 %	743416,1121

Tabla 3.31: Matriz de proporción de coincidencias de la base train

Real	Estimado					MSE
	[450-550]	(550-700]	(700-850]	(850-1200]	(1200-2500]	
[450-550]	4 %	82 %	11 %	3 %	0 %	22728,71006
(550-700]	3 %	72 %	18 %	7 %	0 %	13872,91761
(700-850]	2 %	59 %	21 %	15 %	3 %	34102,65441
(850-1200]	1 %	45 %	22 %	22 %	10 %	108256,6019
(1200-2500]	1 %	28 %	19 %	28 %	24 %	751330,8935

Tabla 3.32: Matriz de proporción de coincidencias de la base test

Por otro lado, la matriz de cálculo para el test Chi-cuadrado está dado por la tabla 3.33.

Real	Estimado						
	[450-550]	(550-700]	(700-850]	(850-1200]	(1200-2500]	Calculado	Teórico
[450-550]	21,8	0,1	0,8	3,3	0,0	**55,1**	**26,3**
(550-700]	0,1	3,2	0,5	0,9	1,3	**Decisión**	
(700-850]	7,6	1,8	0,2	0,2	1,9	Se rechaza H0	
(850-1200]	3,9	0,2	0,0	1,1	1,9		
(1200-2500]	1,0	2,3	0,5	0,3	0,2		

Tabla 3.33: Cálculo del Chi-cuadrado para comparar distribución train y test

Métricas

Los estadísticos de comparación se observan en la tabla 3.34.

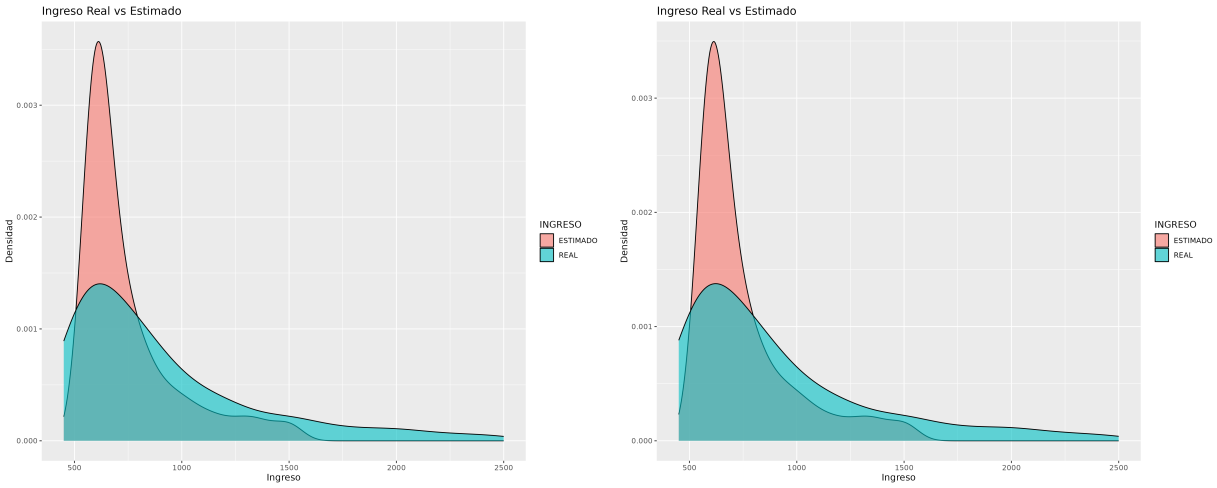
Entrenamiento	Métricas
MSE	170600,54
MAE	266,906887
Cruce	31 %
Cruce +/-	77 %
SubEstima	18 %
SobreEstim	5 %

Validación	Métricas
MSE	177010,883
MAE	273,990049
Cruce	31 %
Cruce +/-	76 %
SubEstima	19 %
SobreEstim	5 %

Tabla 3.34: Estadísticos de comparación

Densidades reales y estimadas

Por último se presenta las gráficas de las densidades reales y estimadas con el modelo xgBoost en la figura 3.13.



((a)) Densidad base train

((b)) Densidad base test

Figura 3.13: Densidades reales y estimadas

3.6.4. Modelo de Regresión Lineal Múltiple (RLM)

El motivo para generar un modelo de regresión lineal múltiple en este proyecto, es para comparar con los otros modelos, ya que es bien conocido que se necesitan muchas hipótesis sobre los datos disponibles y también se necesitan calcular parámetros de distribuciones conocidas para que al final el modelo se ajuste bien a los individuos que no formaron parte de la muestra de entrenamiento.

Además, los modelos principales no requieren hipótesis sobre los datos iniciales, incluyendo todas las hipótesis de la regresión lineal múltiple, es por este motivo que en el modelo de RLM no se realiza ningún remuestreo sobre la base.

Sin realizar las verificaciones de las hipótesis para la regresión múltiple, al final se puede observar en las gráficas de densidades muestrales, que las predicciones de los ingresos no se ajustan tan bien a los ingresos reales.

Creación del modelo para la base train

Para el modelo de RLM, se utiliza la función *lm(·)* que se incluye dentro de las funciones básicas de R. Por lo tanto no se explican sus hiperparámetros.

Predicción del grupo G1 sobre toda la base

El tiempo de ejecución del modelo RLM sobre toda la base se presenta en la tabla 3.35. De forma similar que en el modelo Random Forest, luego de predecir sobre toda la base se realizan filtros para el G1 y sus respectivas bases train y test.

Tiempo de ejecución:		
user	**system**	**elapsed**
0,03	0,01	0,03

Tabla 3.35: Tiempos de ejecución de algoritmo RLM para G1

Matriz de coincidencias

Las mejores estimaciones del modelo se presentan en las tablas 3.36 y 3.37.

Real	Estimado					MSE
	[450-550]	(550-700]	(700-850]	(850-1200]	(1200-2500]	
[450-550]	2%	11%	29%	57%	1%	140568,65
(550-700]	1%	9%	27%	62%	1%	80036,64
(700-850]	1%	6%	25%	67%	1%	29783,02
(850-1200]	0%	3%	17%	77%	2%	31640,72
(1200-2500]	0%	1%	9%	85%	4%	637487,06

Tabla 3.36: Matriz de proporción de coincidencias de la base train

Real	Estimado					MSE
	[450-550]	(550-700]	(700-850]	(850-1200]	(1200-2500]	
[450-550]	1%	11%	29%	58%	1%	144044,12
(550-700]	1%	8%	29%	61%	1%	75824,13
(700-850]	1%	7%	26%	66%	1%	26944,86
(850-1200]	0%	4%	18%	77%	2%	34180,83
(1200-2500]	0%	1%	10%	85%	4%	641140,67

Tabla 3.37: Matriz de proporción de coincidencias de la base test

Por otro lado, la matriz de cálculo para el test Chi-cuadrado está dado por la tabla

Real	Estimado						
	[450-550]	(550-700]	(700-850]	(850-1200]	(1200-2500]	Calculado	Teórico
[450-550]	9,8	0,2	0,5	0,1	0,0	**42,3**	**26,3**
(550-700]	6,9	6,9	3,3	4,7	2,6	**Decisión**	
(700-850]	0,8	0,4	0,0	1,6	0,0	Se rechaza H0	
(850-1200]	0,3	1,0	0,1	0,1	0,3		
(1200-2500]	0,0	0,2	0,7	1,9	0,1		

Tabla 3.38: Cálculo del Chi-cuadrado para comparar distribución train y test

Métricas

Los estadísticos de comparación se los observa en la tabla 3.39.

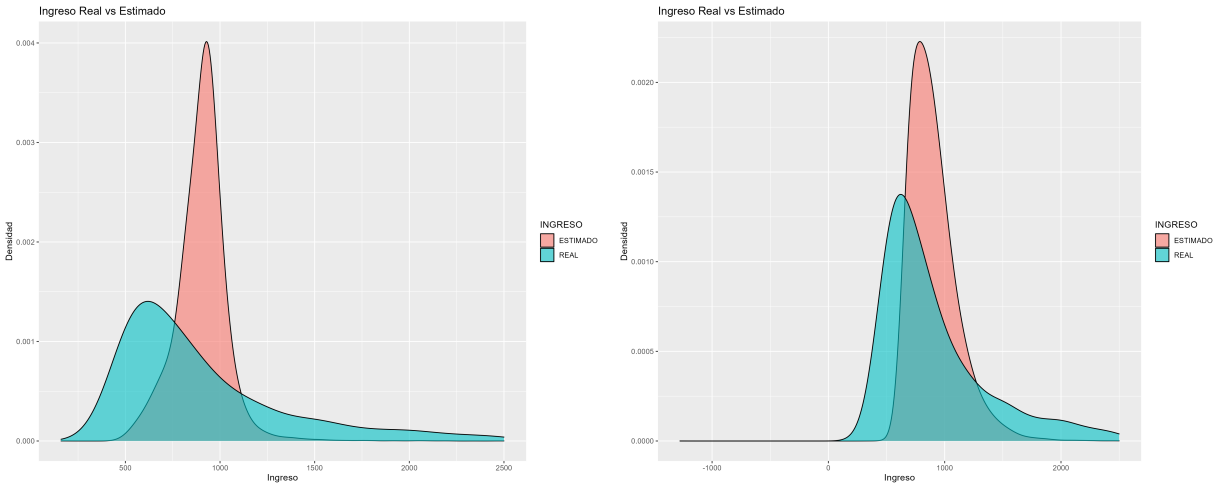
Entrenamiento	Métricas
MSE	175247,243
MAE	311,652831
Cruce	23%
Cruce +/-	64%
SubEstima	3%
SobreEstim	33%

Validación	Métricas
MSE	178626,62
MAE	314,97
Cruce	23%
Cruce +/-	65%
SubEstima	3%
SobreEstim	32%

Tabla 3.39: Estadísticos de comparación

Densidades reales y estimadas

Por último se presenta las gráficas de las densidades reales y estimadas con el modelo de RLM en la figura 3.14.



((a)) Densidad base train ((b)) Densidad base test

Figura 3.14: Densidades reales y estimadas

3.6.5. Elección del mejor modelo entre RLM, RF, GBM y XGB

Luego de generar los cuatro modelos estadísticos, el mejor modelo es el GBM. Esto es claro, no solo por sus matrices de coincidencia y sus métricas, sino también por las gráficas de densidad finales.

Con el modelo de GBM se va a utilizar una base de validación, es decir que ningún individuo de esta base se utilizó para generar los modelos; así, se podrá revisar la matriz de coincidencia de la base de validación, sus métricas y sus densidades.

Lo que se explica, se lo realiza en la sección de *Evaluación del mejor modelo con BDD de Validación para grupo G1*.

También, se calculan algunos indicadores de liquidez para el modelo de GBM, en la sección de *Indicadores de Liquidez*.

3.7. Modelos para población G2

Como se explica en la sección de *Modelos para población G1*, se aplica la misma metodología para el G2. De forma similar, se empiezan presentando los resultados para el modelo de Random Forest, Gradient Boosting Machine, xgBoost y finalmente Regresión Lineal Múltiple.

En el G2 se omite la explicación del modelo generado por la base train sin remuestreo. Esto se debe a que los pasos y las conclusiones sin similares.

Se hace énfasis que la densidad de los ingresos reales es muy diferente a la densidad del modelo utilizado por el Buró. Esto se lo puede observar en la gráfica 3.15.

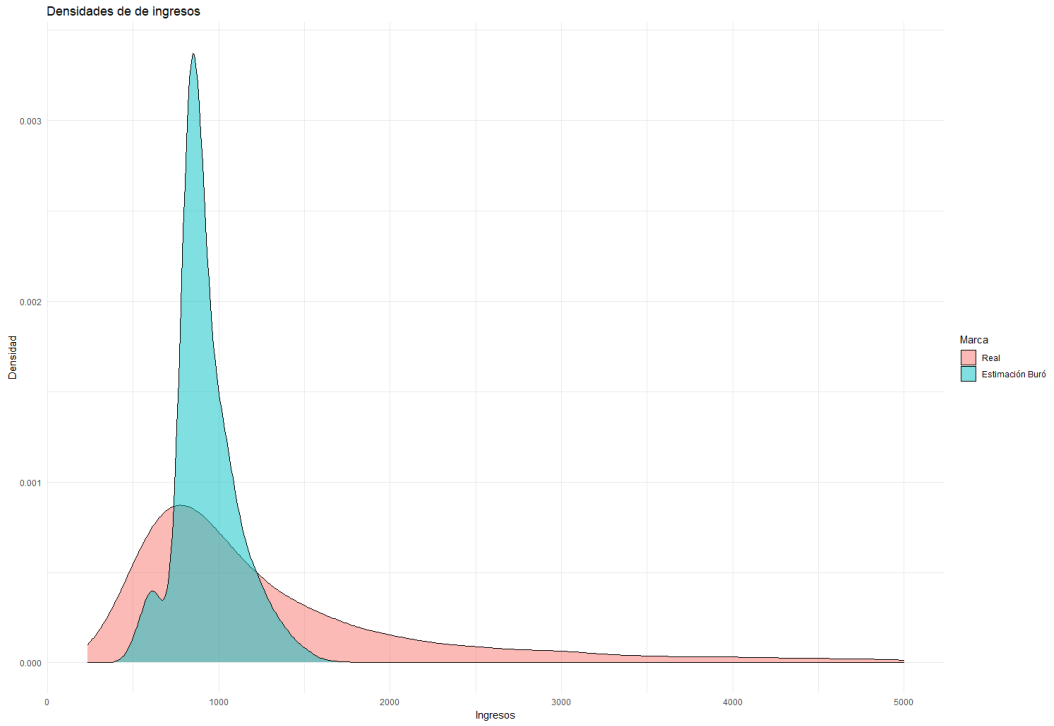


Figura 3.15: Densidad real y estimada por el Buró para G2

3.7.1. Modelo Random Forest (RF)

Se presentan los mejores resultados con los mejores parámetros hallados para el G2.

Hiperparámetros para remuestreo

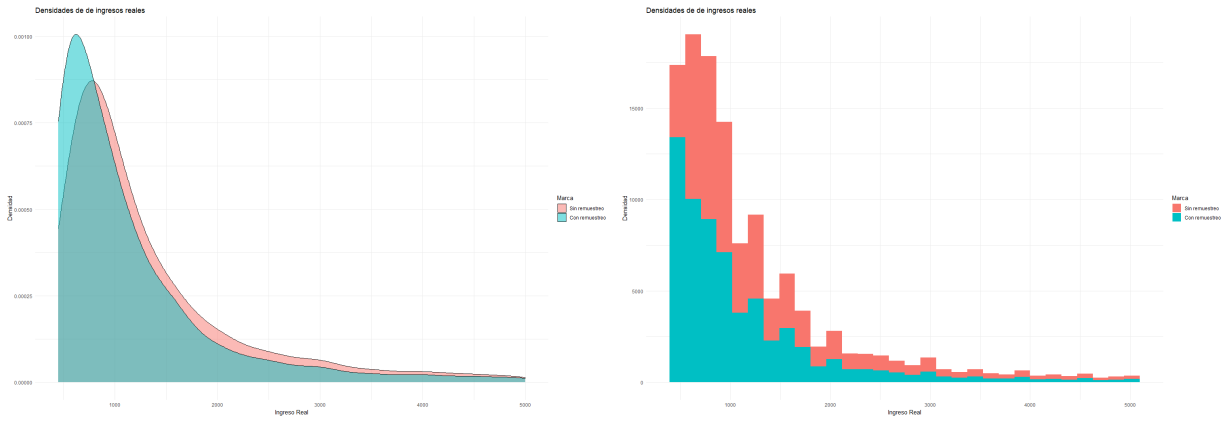
Los mejores hiperparámetros hallados para el modelo RF sobre el G2, se presentan en la tabla 3.40.

G2_corte1 = 553 // 8 %
G2_corte2 = 1800 // 80 %
G2_porc1 = 0.27
G2_porc2 = 0.16

Tabla 3.40: Mejores hiperparámetros para RF sobre G2

Remuestreo

La comparación de la densidad de ingresos reales con las bases sin remuestrear y remuestrada, se da en la figura 3.16.



((a)) Densidades Reales ((b)) Histogramas reales

Figura 3.16: Comparación de base train real con la remuestreada

Representatividad en los nodos hojas en cada árbol

Se utiliza el porcentaje del 3.6 % de representatividad en los nodos hoja de los árboles en cada iteración. Revisar la tabla A.5.

Creación del modelo para la base train remuestreada

Se genera el modelo de RF con la función *ranger* en el lenguaje R. Los detalles de los hiperparámetros ya fueron detallados en el modelo RF para G1.

Predicción del grupo G2 sobre toda la base

El tiempo de ejecución del modelo RF sobre toda la base se presenta en la tabla 3.41.

Tiempo de ejecución		
user	system	elapsed
14.63	0.62	14.63

Tabla 3.41: Tiempos de ejecución de algoritmo RF para G2

Matriz de coincidencias

Las mejores estimaciones del modelo RF se presentan en las tablas 3.42 y 3.43.

Real	Estimado					MSE
	[450-700]	(700-900]	(900-1200]	(1200-1800]	(1800-5000]	
[450-700]	27%	44%	24%	6%	0%	99885.39
(700-900]	10%	27%	40%	23%	0%	106267.02
(900-1200]	5%	18%	34%	40%	4%	115730.02
(1200-1800]	2%	13%	25%	41%	18%	226165.12
(1800-5000]	1%	6%	15%	35%	44%	1949338.07

Tabla 3.42: Matriz de proporción de coincidencias de la base train

Real	Estimado					MSE
	[450-700]	(700-900]	(900-1200]	(1200-1800]	(1800-5000]	
[450-700]	25%	44%	25%	6%	0%	107411.44
(700-900]	10%	28%	39%	23%	1%	109950.68
(900-1200]	4%	18%	33%	40%	4%	120653.79
(1200-1800]	3%	12%	26%	42%	18%	222576.57
(1800-5000]	1%	6%	15%	35%	43%	1926928.81

Tabla 3.43: Matriz de proporción de coincidencias de la base test

Por otro lado, la matriz de cálculo para el test Chi-cuadrado está dado por la tabla 3.44

Real	Estimado						
	[450-700]	(700-900]	(900-1200]	(1200-1800]	(1800-5000]	Calculado	Teórico
[450-700]	9.0	0.4	5.8	3.5	2.4	**67.9**	**26.3**
(700-900]	3.4	0.6	0.5	0.4	8.8	**Decisión**	
(900-1200]	0.4	1.3	0.9	0.3	0.7	Se rechaza H0	
(1200-1800]	9.4	1.9	0.3	0.7	1.9		
(1800-5000]	10.7	0.4	1.5	2.0	0.7		

Tabla 3.44: Cálculo del Chi-cuadrado para comparar distribución train y test

Métricas

Los estadísticos de comparación se observan en la tabla 3.45.

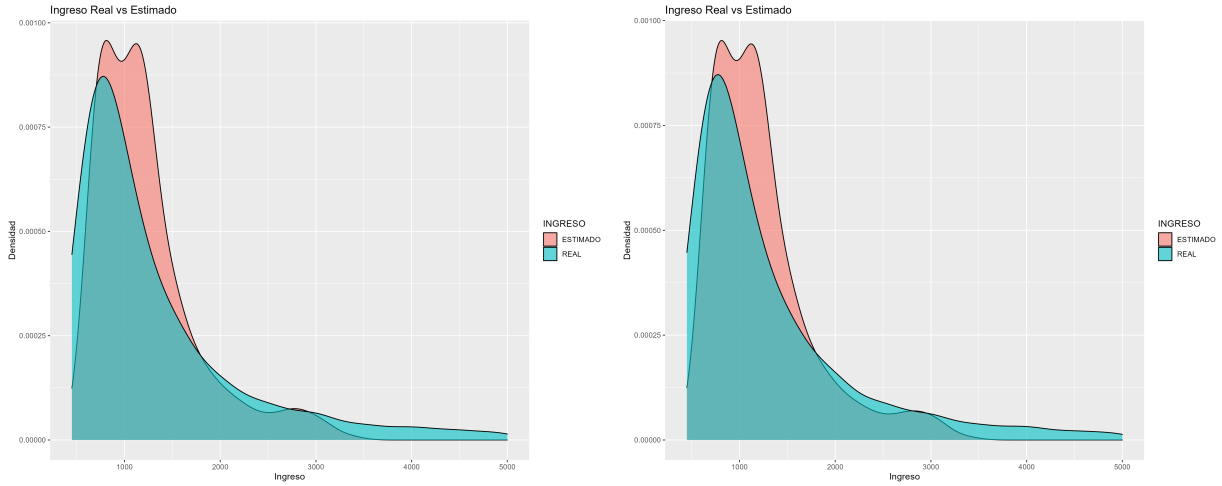
Entrenamiento	Métricas
MSE	478760.21
MAE	450.30
Cruce	34 %
Cruce +/-	80 %
SubEstima	8 %
SobreEstim	12 %

Validación	Métricas
MSE	478866.22
MAE	451.19
Cruce	34 %
Cruce +/-	80 %
SubEstima	8 %
SobreEstim	13 %

Tabla 3.45: Estadísticos de comparación

Densidades reales y estimadas

Por último se presenta las gráficas de las densidades reales y estimadas con el modelo RF en la figura 3.17.



((a)) Densidad base train ((b)) Densidad base test

Figura 3.17: Densidades reales y estimadas

3.7.2. Modelo GBM

Se presentan los mejores resultados con los mejores parámetros hallados para el G2.

Parámetros para remuestreo

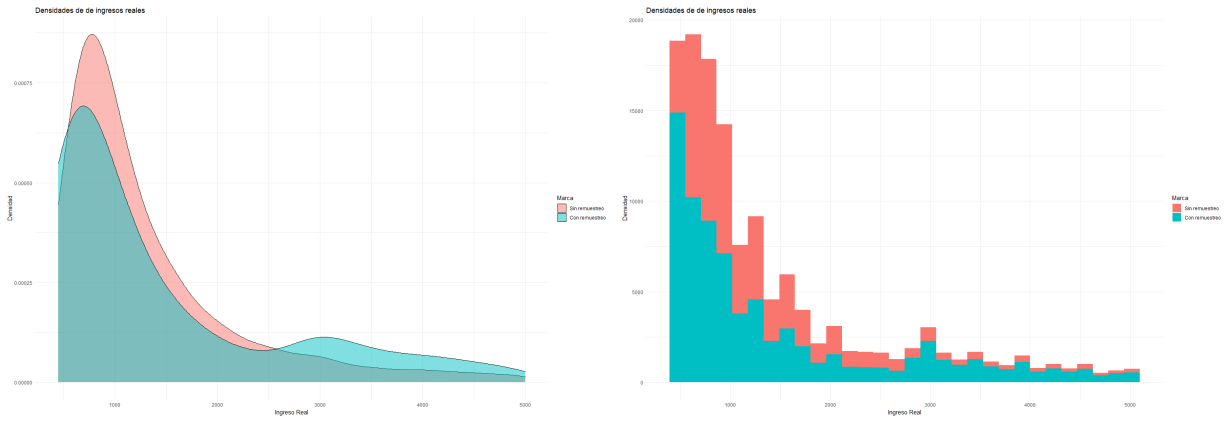
Los mejores hiperparámetros hallados para el modelo GBM sobre el G2, se presentan en la tabla 3.46.

G2_corte1 = 553 // 8 %
G2_corte2 = 2800 // 92 %
G2_porc1 = 0.3
G2_porc2 = 0.25

Tabla 3.46: Mejores hiperparámetros para GBM sobre G2

Remuestreo

La comparación de la densidad de ingresos reales con las bases sin remuestrear y remuestreada, se da en la figura 3.18.



((a)) Densidades Reales ((b)) Histogramas reales

Figura 3.18: Comparación de base train real con la remuestreada

Representatividad en los nodos hojas en cada árbol

Se utiliza el porcentaje del 3.6 % de representatividad en los nodos hoja de los árboles en cada iteración. Revisar la tabla A.6.

Creación del modelo para la base train remuestreada

Se genera el modelo de GBM con la función *gbm* en el lenguaje R. Los detalles de los hiperparámetros ya fueron detallados en el modelo GBM para G1.

Predicción del grupo G2 sobre toda la base

El tiempo de ejecución del modelo GBM sobre toda la base se presenta en la tabla 3.47.

Tiempo de ejecución		
user	system	elapsed
14,75	0	14,75

Tabla 3.47: Tiempos de ejecución de algoritmo GBM para G2

Matriz de coincidencias

Las mejores estimaciones del modelo GBM se presentan en las tablas 3.48 y 3.49.

Real	Estimado					MSE
	[450-700]	(700-900]	(900-1200]	(1200-1800]	(1800-5000]	
[450-700]	63 %	24 %	10 %	3 %	0 %	59280,09
(700-900]	32 %	29 %	24 %	13 %	1 %	93554,52
(900-1200]	20 %	22 %	26 %	25 %	7 %	168868,68
(1200-1800]	12 %	17 %	21 %	27 %	22 %	413762,58
(1800-5000]	7 %	10 %	16 %	25 %	43 %	2241945,66

Tabla 3.48: Matriz de proporción de coincidencias de la base train

Real	Estimado					MSE
	[450-700]	(700-900]	(900-1200]	(1200-1800]	(1800-5000]	
[450-700]	62 %	24 %	10 %	4 %	0 %	64195,54
(700-900]	33 %	29 %	23 %	13 %	2 %	97650,86
(900-1200]	19 %	22 %	25 %	26 %	7 %	172183,95
(1200-1800]	13 %	16 %	22 %	28 %	21 %	402972,20
(1800-5000]	6 %	10 %	16 %	25 %	43 %	2209602,45

Tabla 3.49: Matriz de proporción de coincidencias de la base test

Por otro lado, la matriz de cálculo para el test Chi-cuadrado está dado por la tabla 3.50.

Real	Estimado						
	[450-700]	(700-900]	(900-1200]	(1200-1800]	(1800-5000]	Calculado	Teórico
[450-700]	0,1	0,2	0,0	10,6	1,9	**63,4**	**26,3**
(700-900]	6,5	0,1	0,9	0,1	8,4	**Decisión**	
(900-1200]	3,4	0,2	3,4	2,7	3,6	Se rechaza H0	
(1200-1800]	0,1	3,4	2,7	3,4	3,7		
(1800-5000]	1,8	5,4	0,6	0,3	0,0		

Tabla 3.50: Cálculo del Chi-cuadrado para comparar distribución train y test

Métricas

Los estadísticos de comparación se observan en la tabla 3.51.

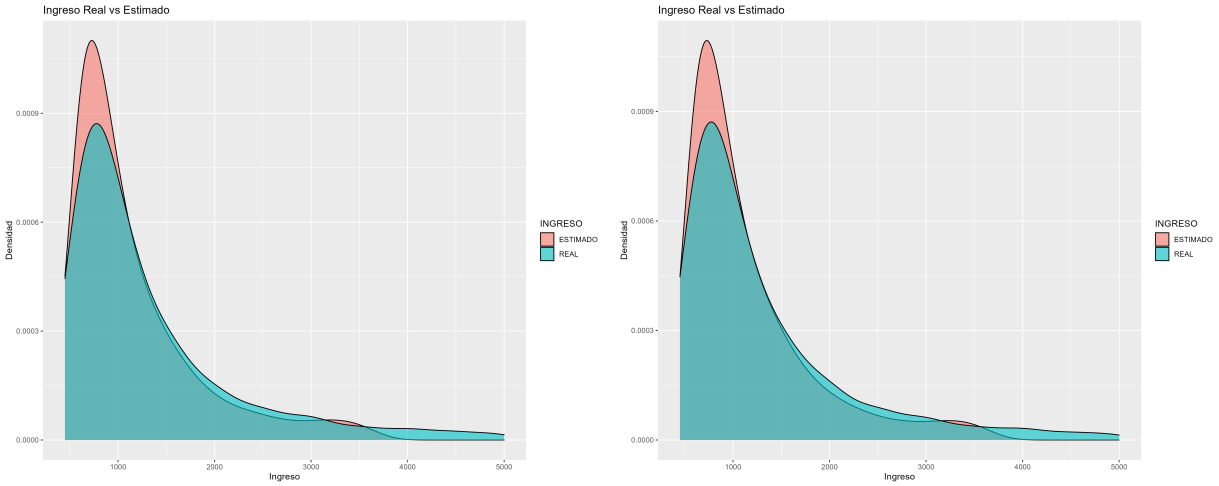
Entrenamiento	Métricas
MSE	566554,06
MAE	475,26
Cruce	39 %
Cruce +/-	77 %
SubEstima	15 %
SobreEstim	7 %

Validación	Métricas
MSE	562697,76
MAE	474,11
Cruce	38 %
Cruce +/-	77 %
SubEstima	15 %
SobreEstim	8 %

Tabla 3.51: Estadísticos de comparación

Densidades reales y estimadas

Por último se presenta las gráficas de las densidades reales y estimadas con el modelo GBM en la figura 3.19.



((a)) Densidad base train

((b)) Densidad base test

Figura 3.19: Densidades reales y estimadas

3.7.3. Modelo XGB

Se presentan los mejores resultados con los mejores parámetros hallados para el G2.

Parámetros para remuestreo

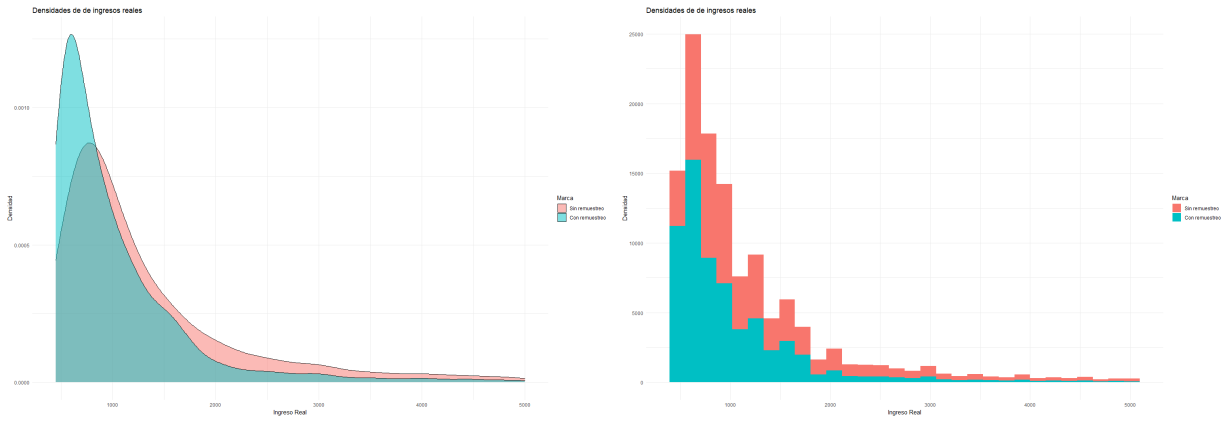
Los mejores hiperparámetros hallados para el modelo xgBoost sobre el G2, se presentan en la tabla

G2_corte1 = 601 // 14 %
G2_corte2 = 1801 // 81 %
G2_porc1 = 0.40
G2_porc2 = 0.10

Tabla 3.52: Mejores hiperparámetros para xgBoost sobre G2

Remuestreo

La comparación de la densidad de ingresos reales con las bases sin remuestrear y remuestreada, se da en la figura 3.20.



((a)) Densidades Reales ((b)) Histogramas reales

Figura 3.20: Comparación de base train real con la remuestreada

Representatividad en los nodos hojas en cada árbol

Se utiliza el porcentaje del 3.6 % de representatividad en los nodos hoja de los árboles en cada iteración.

Creación del modelo para la base train remuestreada

Se genera el modelo de xgBoost con la función *h2o.xgboost* en el lenguaje R. Los detalles de los hiperparámetros ya fueron detallados en el modelo xgBoost para G1.

Predicción del grupo G2 sobre toda la base

El tiempo de ejecución del modelo xgBoost sobre toda la base se presenta en la tabla 3.53.

Tiempo de ejecución		
user	system	elapsed
9	0	16

Tabla 3.53: Tiempos de ejecución de algoritmo xgBoost para G2

Matriz de coincidencias

Las mejores estimaciones del modelo xgBoost se presentan en las tablas 3.54 y 3.55.

Real	Estimado					MSE
	[450-700]	(700-900]	(900-1200]	(1200-1800]	(1800-5000]	
[450-700]	37%	42%	18%	3%	0%	69526,94664
(700-900]	15%	30%	38%	16%	0%	79415,96174
(900-1200]	7%	21%	36%	33%	2%	93279,40504
(1200-1800]	3%	14%	28%	41%	14%	230341,1773
(1800-5000]	1%	7%	17%	38%	36%	2227026,409

Tabla 3.54: Matriz de proporción de coincidencias de la base train

Real	Estimado					MSE
	[450-700]	(700-900]	(900-1200]	(1200-1800]	(1800-5000]	
[450-700]	36%	42%	18%	4%	0%	75417,50005
(700-900]	16%	30%	37%	17%	0%	82665,81735
(900-1200]	7%	20%	35%	35%	2%	96469,36469
(1200-1800]	4%	14%	28%	41%	13%	232183,691
(1800-5000]	1%	7%	17%	38%	35%	2190043,237

Tabla 3.55: Matriz de proporción de coincidencias de la base test

Por otro lado, la matriz de cálculo para el test Chi-cuadrado está dado por la tabla 3.56.

Real	Estimado						
	[450-700]	(700-900]	(900-1200]	(1200-1800]	(1800-5000]	Calculado	Teórico
[450-700]	5,7	0,6	3,0	17,4	0,0	**78,9**	**26,3**
(700-900]	3,3	1,2	3,8	4,8	1,0	**Decisión**	
(900-1200]	0,5	3,4	4,7	5,8	0,4	Se rechaza H0	
(1200-1800]	13,2	0,3	0,8	0,0	3,4		
(1800-5000]	0,7	0,1	2,8	1,0	1,0		

Tabla 3.56: Tiempos de ejecución de algoritmo xgBoost para G2

Métricas

Los estadísticos de comparación se observan en la tabla 3.57.

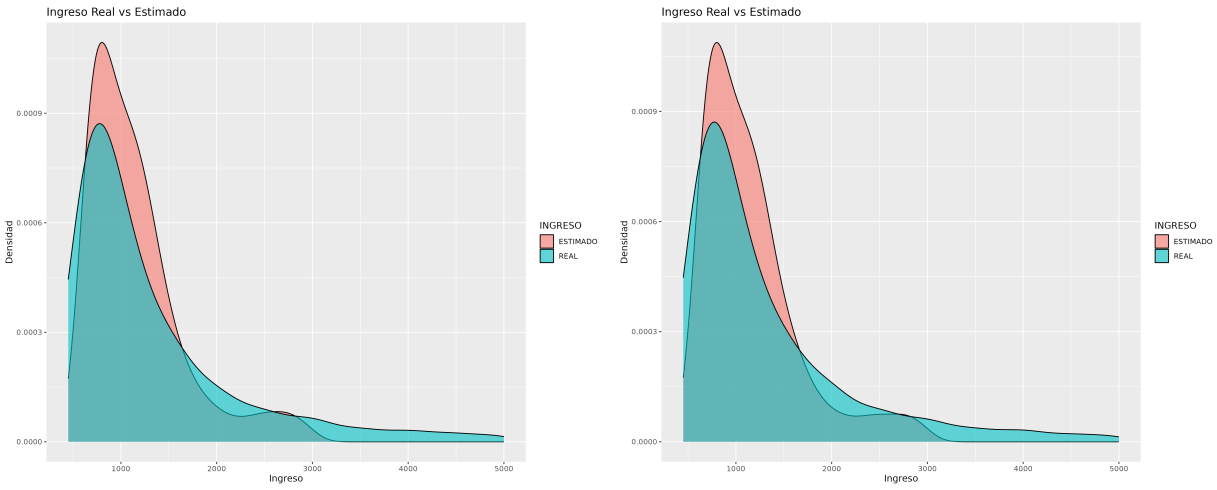
Entrenamiento	Métricas
MSE	515464,7
MAE	444,9901
Cruce	36 %
Cruce +/-	82 %
SubEstima	9 %
SobreEstim	9 %

Validación:	Métricas
MSE	513224,66
MAE	446,17713
Cruce	35 %
Cruce +/-	81 %
SubEstima	10 %
SobreEstim	9 %

Tabla 3.57: Estadísticos de comparación

Densidades reales y estimadas

Por último se presenta las gráficas de las densidades reales y estimadas con el modelo xgBoost en la figura 3.21.



((a)) Densidad base train

((b)) Densidad base test

Figura 3.21: Densidades reales y estimadas

3.7.4. Modelo RLM

Tal y como se procede con el G1, también se lo hace para el G2.

Creación del modelo para la base train

Para el modelo de RLM, se utiliza la función $lm(\cdot)$ que se incluye dentro de las funciones básicas de R. Por lo tanto no se explican sus hiperparámetros.

Predicción del grupo G2 sobre toda la base

El tiempo de ejecución del modelo RLM sobre toda la base se presenta en la tabla 3.58

Tiempo de ejecución:		
user	**system**	**elapsed**
0,03	0,01	0,03

Tabla 3.58: Tiempos de ejecución de algoritmo RLM para G2

Matriz de coincidencias

Las mejores estimaciones del modelo se presentan en las tablas 3.59 y 3.60.

Real	Estimado					MSE
	[450-700]	(700-900]	(900-1200]	(1200-1800]	(1800-5000]	
[450-700]	4 %	17 %	49 %	29 %	1 %	75417,50005
(700-900]	2 %	10 %	39 %	46 %	2 %	82665,81735
(900-1200]	1 %	6 %	30 %	58 %	5 %	96469,36469
(1200-1800]	1 %	4 %	22 %	62 %	10 %	232183,691
(1800-5000]	0 %	2 %	14 %	56 %	27 %	2190043,237

Tabla 3.59: Matriz de proporción de coincidencias de la base train

Real	Estimado					MSE
	[450-700]	(700-900]	(900-1200]	(1200-1800]	(1800-5000]	
[450-700]	5 %	16 %	49 %	30 %	1 %	75417,50005
(700-900]	2 %	10 %	40 %	46 %	2 %	82665,81735
(900-1200]	1 %	6 %	29 %	58 %	5 %	96469,36469
(1200-1800]	1 %	4 %	22 %	62 %	11 %	232183,691
(1800-5000]	0 %	2 %	14 %	58 %	26 %	2190043,237

Tabla 3.60: Matriz de proporción de coincidencias de la base test

Por otro lado, la matriz de cálculo para el test Chi-cuadrado está dado por la tabla 3.61.

Real	Estimado						
	[450-700]	(700-900]	(900-1200]	(1200-1800]	(1800-5000]	Calculado	Teórico
[450-700]	1,5	0,2	0,6	2,6	4,8	**52,0**	**26,3**
(700-900]	0,0	0,0	0,6	0,0	5,3	**Decisión**	
(900-1200]	1,6	1,2	3,1	0,0	5,3	Se rechaza H0	
(1200-1800]	11,5	0,0	1,0	0,0	0,6		
(1800-5000]	2,1	0,2	0,0	6,3	3,6		

Tabla 3.61: Cálculo del Chi-cuadrado para comparar distribución train y test

Métricas

Los estadísticos de comparación se los observa en la tabla 3.62

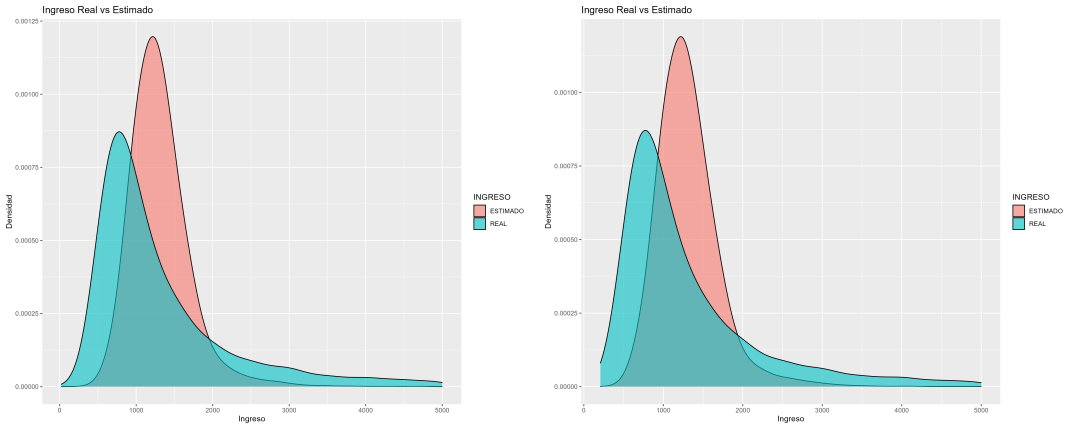
Entrenamiento	Métricas
MSE	622939,11
MAE	552,26
Cruce	25 %
Cruce +/-	67 %
SubEstima	4 %
SobreEstim	29 %

Validación	Métricas
MSE	625948,57
MAE	552,44
Cruce	25 %
Cruce +/-	67 %
SubEstima	4 %
SobreEstim	29 %

Tabla 3.62: Estadísticos de comparación

Densidades reales y estimadas

Por último se presenta las gráficas de las densidades reales y estimadas con el modelo de RLM en la figura 3.22.



((a)) Densidad base train ((b)) Densidad base test

Figura 3.22: Densidades reales y estimadas

3.7.5. Elección del mejor modelo entre RLM, RF, GBM y XGB

Luego de generar los cuatro modelos estadísticos, el mejor modelo es el xgBoost. Se decide que este es el mejor debido a que las matrices de coincidencia, los estadísticos de comparación y las densidades estimadas, son las mejores entre todos los modelos.

Se aclara que, por motivos de poder computacional, se decide utilizar al modelo GBM para realizar las estimaciones sobre la base de validación. Esto se decide porque a pesar que el modelo xgBoost es el mejor, el GBM también tiene muy buenos resultados y además su algoritmo no es tan pesado computacionalmente hablando.

3.8. Modelos para población G3

Como se explica en la sección de Modelos para población G1, se aplica la misma metodología para el G3. De forma similar, se empiezan presentando los resultados para el modelo de Random Forest, Gradient Boosting Machine, xgBoost y finalmente Regresión Lineal Múltiple.

En el G3 también se omite la explicación del modelo generado por la base train sin remuestreo. Para verificar la mala estimación del modelo del Buró, se presentan las densidades para el G3. Ver figura 3.23.

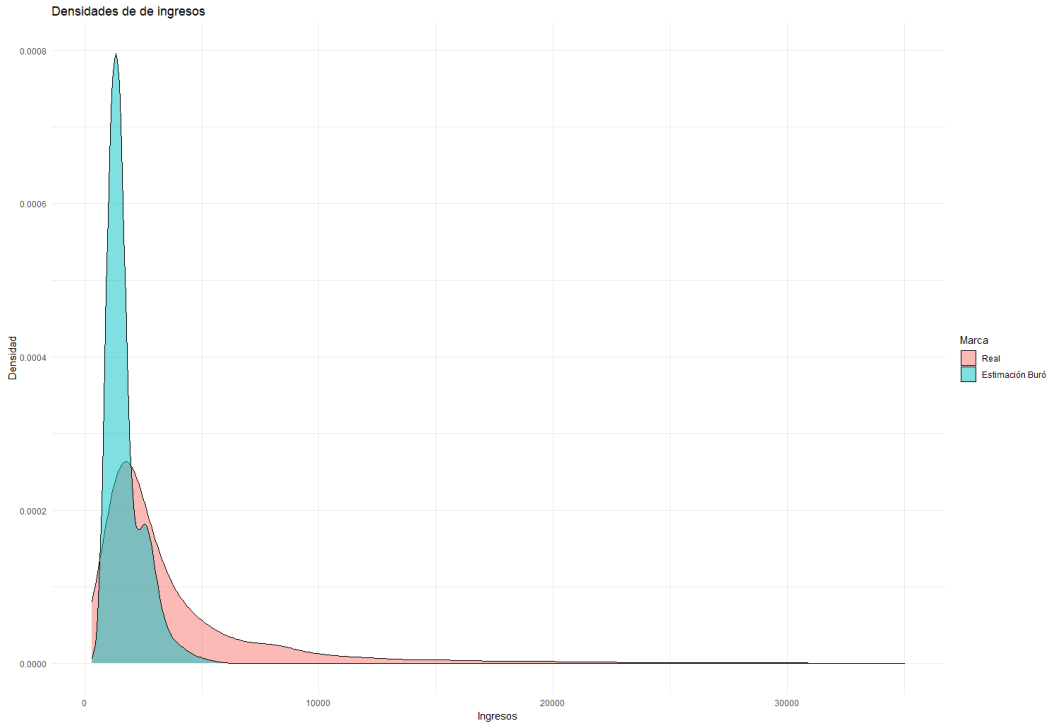


Figura 3.23: Densidad real y estimada por el Buró para G3

Se observa que en el G3, la predicción realizada por el Buró subestima mucho a los ingresos reales. De los tres grupos, esta es la peor estimación realizada, pues el máximo de las estimaciones en este grupo es de USD$7721, pero el máximo de los valores reales es de USD$35 000.

3.8.1. Modelo Random Forest (RF)

Se presentan los mejores resultados con los mejores parámetros hallados para el G3.

Hiperparámetros para remuestreo

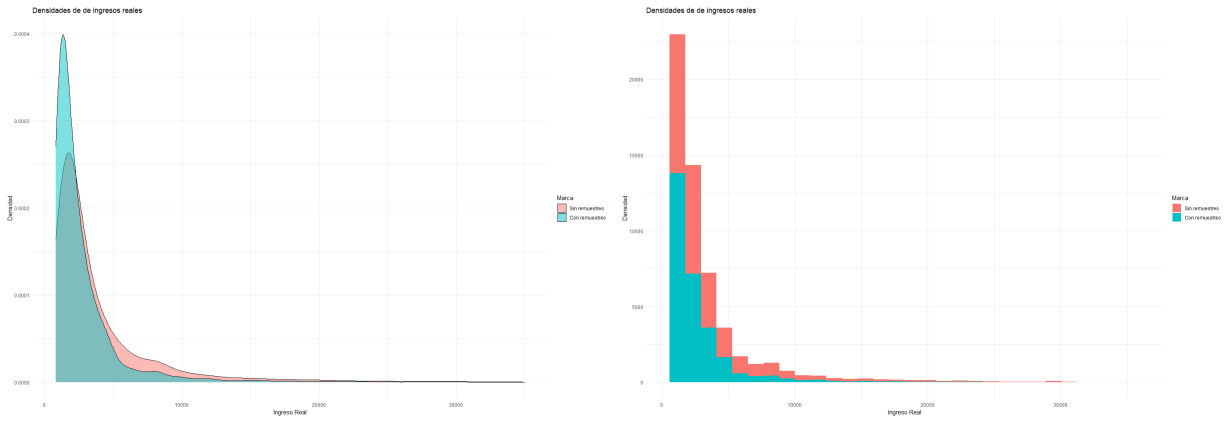
Los mejores hiperparámetros hallados para el modelo RF sobre el G3, se presentan en la tabla 3.63.

G3_corte1 = 1164 // 8 %
G3_corte2 = 5000 // 79 %
G3_porc1 = 0.25
G3_porc2 = 0.11

Tabla 3.63: Mejores hiperparámetros para RF sobre G3

Remuestreo

La comparación de la densidad de ingresos reales con las bases sin remuestrear y remuestreada, se presenta en la figura 3.24.



((a)) Densidades Reales ((b)) Histogramas reales

Figura 3.24: Comparación de base train real con la remuestreada

Representatividad en los nodos hojas en cada árbol

Se utiliza el porcentaje del 3.6 % de representatividad en los nodos hoja de los árboles en cada iteración.

Creación del modelo para la base train remuestreada

Se genera el modelo de RF con la función *ranger* en el lenguaje R. Los detalles de los hiperparámetros ya fueron detallados en el modelo RF para G1.

Predicción del grupo G3 sobre toda la base

El tiempo de ejecución del modelo RF sobre toda la base se presenta en la tabla 3.64.

Tiempo de ejecución		
user	**system**	**elapsed**
17,59	0,49	17,59

Tabla 3.64: Tiempos de ejecución de algoritmo RF para G3

Matriz de coincidencias

Las mejores estimaciones del modelo RF se presentan en las tablas 3.65 y 3.66.

Real	Estimado					MSE
	[800-1450]	(1450-2000]	(2000-2950]	(2950-5200]	(5200-35000]	
[800-1450]	16%	52%	28%	3%	0%	791403,44
(1450-2000]	5%	35%	45%	14%	1%	899755,89
(2000-2950]	2%	21%	39%	34%	3%	1238679,60
(2950-5200]	1%	10%	28%	41%	21%	3346626,39
(5200-35000]	0%	3%	14%	30%	53%	56940913,80

Tabla 3.65: Matriz de proporción de coincidencias de la base train

Real	Estimado					MSE
	[800-1450]	(1450-2000]	(2000-2950]	(2950-5200]	(5200-35000]	
[800-1450]	15%	52%	29%	4%	0%	767512,58
(1450-2000]	5%	36%	44%	15%	1%	1020409,07
(2000-2950]	2%	21%	38%	35%	3%	1334022,53
(2950-5200]	1%	11%	28%	40%	21%	3474387,46
(5200-35000]	0%	4%	13%	29%	53%	59443144,8

Tabla 3.66: Matriz de proporción de coincidencias de la base test

Por otro lado, la matriz de cálculo para el test Chi-cuadrado está dado por la tabla 3.67.

Real	Estimado						
	[800-1450]	(1450-2000]	(2000-2950]	(2950-5200]	(5200-35000]	Calculado	Teórico
[800-1450]	10,6	1,2	0,8	2,2	3,0	**80,3**	**26,3**
(1450-2000]	2,4	1,5	0,8	5,4	6,4	**Decisión**	
(2000-2950]	5,3	1,3	0,1	4,1	1,0	Se rechaza H0	
(2950-5200]	6,9	3,0	0,8	4,5	0,5		
(5200-35000]	0,8	16,5	0,0	0,1	1,4		

Tabla 3.67: Cálculo del Chi-cuadrado para comparar distribución train y test

Métricas

Los estadísticos de comparación se observan en la tabla 3.68

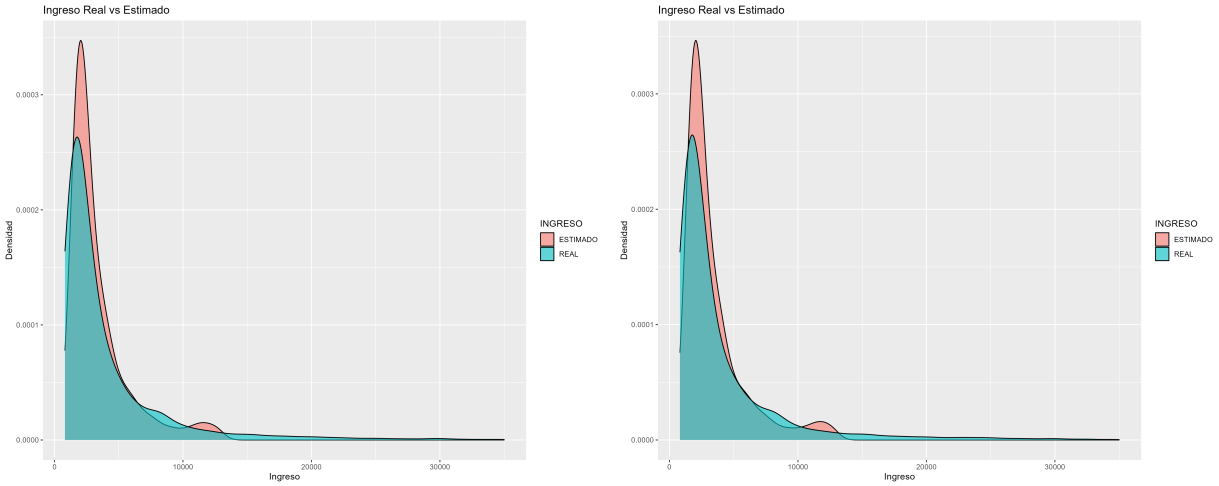
Entrenamiento	Métricas
MSE	12569333
MAE	1800,5
Cruce	37 %
Cruce +/-	84 %
SubEstima	6 %
SobreEstim	10 %

Validación	Métricas
MSE	13299416
MAE	1849,8
Cruce	36 %
Cruce +/-	83 %
SubEstima	6 %
SobreEstim	11 %

Tabla 3.68: Estadísticos de comparación

Densidades reales y estimadas

Por último se presenta las gráficas de las densidades reales y estimadas con el modelo RF en la figura 3.25.



((a)) Densidad base train ((b)) Densidad base test

Figura 3.25: Densidades reales y estimadas

3.8.2. Modelo GBM

Se presentan los mejores resultados con los mejores parámetros hallados para el G3.

Hiperparámetros para remuestreo

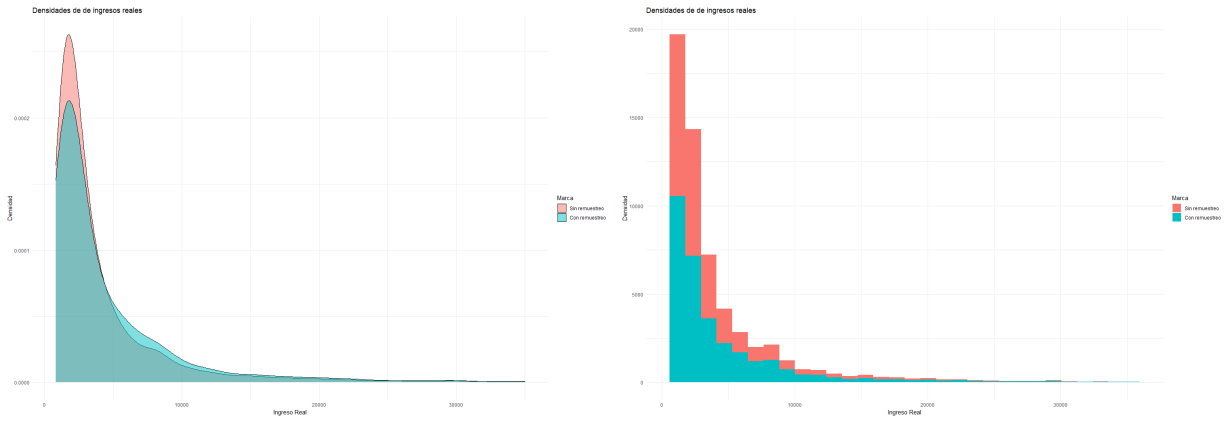
Los mejores hiperparámetros hallados para el modelo GBM sobre el G3, se presentan en la tabla 3.69.

G3_corte1 = 1376 // 18 %
G3_corte2 = 5000 // 79 %
G3_porc1 = 0.23
G3_porc2 = 0.32

Tabla 3.69: Mejores hiperparámetros para GBM sobre G3

Remuestreo

La comparación de la densidad de ingresos reales con las bases sin remuestrear y remuestreada, se presenta en la figura 3.26.



((a)) Densidades Reales ((b)) Histogramas reales

Figura 3.26: Comparación de base train real con la remuestreada

Representatividad en los nodos hojas en cada árbol

Se utiliza el porcentaje del 3.6 % de representatividad en los nodos hoja de los árboles en cada iteración.

Creación del modelo para la base train remuestreada

Se genera el modelo de GBM con la función *gbm* en el lenguaje R. Los detalles de los hiperparámetros ya fueron detallados en el modelo GBM para G1.

Predicción del grupo G3 sobre toda la base

El tiempo de ejecución del modelo GBM sobre toda la base se presenta en la tabla 3.70.

Tiempo de ejecución		
user	**system**	**elapsed**
0,92	0	0,92

Tabla 3.70: Tiempos de ejecución de algoritmo GBM para G3

Matriz de coincidencias

Las mejores estimaciones del modelo GBM se presentan en las tablas 3.71 y 3.72.

Real	Estimado					MSE
	[800-1450]	(1450-2000]	(2000-2950]	(2950-5200]	(5200-35000]	
[800-1450]	35%	50%	12%	2%	0%	520180,46
(1450-2000]	17%	48%	25%	9%	1%	617358,38
(2000-2950]	9%	34%	28%	26%	3%	1226420,79
(2950-5200]	3%	21%	21%	34%	20%	3557364,62
(5200-35000]	1%	8%	13%	25%	52%	62784332,20

Tabla 3.71: Matriz de proporción de coincidencias de la base train

Real	Estimado					MSE
	[800-1450]	(1450-2000]	(2000-2950]	(2950-5200]	(5200-35000]	
[800-1450]	35%	50%	13%	2%	0%	465305,91
(1450-2000]	17%	48%	25%	9%	1%	669950,735
(2000-2950]	8%	36%	27%	26%	3%	1287000,12
(2950-5200]	4%	21%	22%	33%	20%	3626052,71
(5200-35000]	1%	8%	13%	26%	52%	64407703,1

Tabla 3.72: Matriz de proporción de coincidencias de la base test

Por otro lado, la matriz de cálculo para el test Chi-cuadrado está dado por la tabla 3.73.

Real	Estimado						
	[800-1450]	(1450-2000]	(2000-2950]	(2950-5200]	(5200-35000]	Calculado	Teórico
[800-1450]	1,6	1,3	0,2	0,0	2,1	**35,3**	**26,3**
(1450-2000]	0,0	0,1	0,2	0,8	2,9	**Decisión**	
(2000-2950]	0,0	8,8	1,1	1,5	1,6	Se rechaza H0	
(2950-5200]	3,3	1,2	0,5	4,2	0,5		
(5200-35000]	0,4	0,0	0,0	1,7	1,4		

Tabla 3.73: Cálculo del Chi-cuadrado para comparar distribución train y test

Métricas

Los estadísticos de comparación se observan en la tabla 3.74.

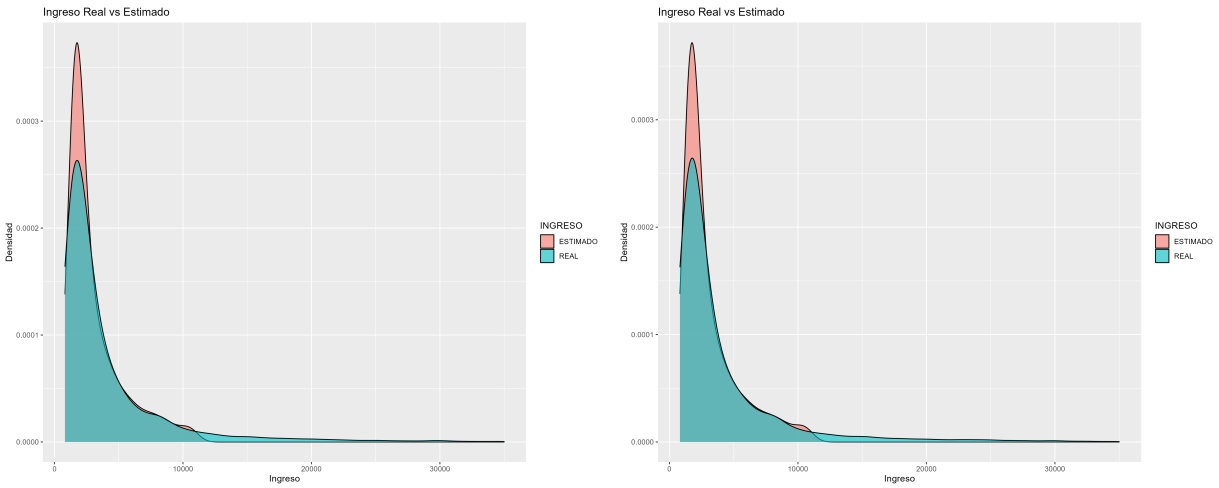
Entrenamiento	Métricas
MSE	13654040
MAE	1800,9
Cruce	40 %
Cruce +/-	83 %
SubEstima	11 %
SobreEstim	6 %

Validación	Métricas
MSE	14185411
MAE	1827,3
Cruce	39 %
Cruce +/-	83 %
SubEstima	11 %
SobreEstim	6 %

Tabla 3.74: Estadísticos de comparación

Densidades reales y estimadas

Por último se presenta las gráficas de las densidades reales y estimadas con el modelo GBM en la figura 3.27.



((a)) Densidad base train ((b)) Densidad base test

Figura 3.27: Densidades reales y estimadas

3.8.3. Modelo XGB

Se presentan los mejores resultados con los mejores parámetros hallados para el G3.

Hiperparámetros para remuestreo

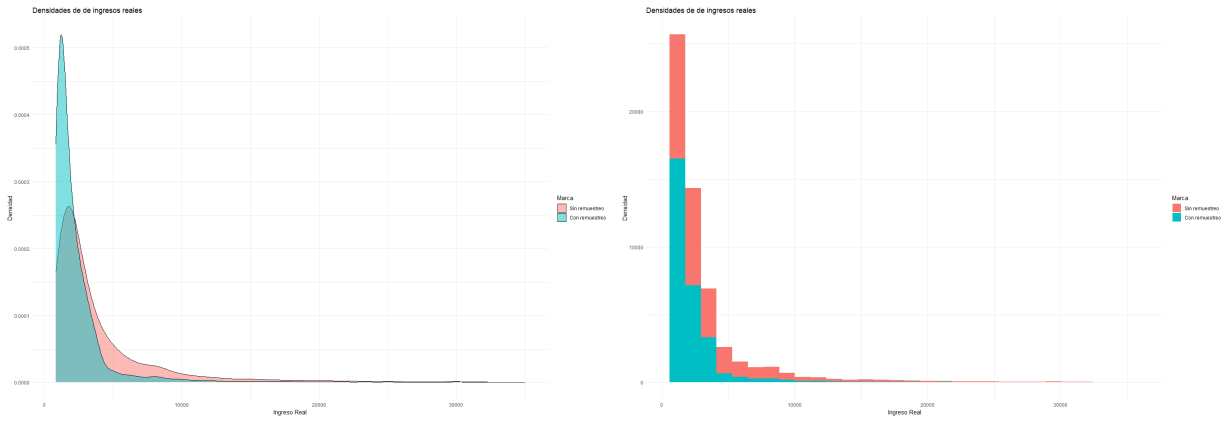
Los mejores hiperparámetros hallados para el modelo xgBoost sobre el G3, se presentan en la tabla 3.75.

G3_corte1 = 1164 // 8 %
G3_corte2 = 4000 // 72 %
G3_porc1 = 0.35
G3_porc2 = 0.10

Tabla 3.75: Mejores hiperparámetros para xgBoost sobre G3

Remuestreo

La comparación de la densidad de ingresos reales con las bases sin remuestrear y remuestreada, se da en la figura



((a)) Densidades Reales ((b)) Histogramas reales

Figura 3.28: Comparación de base train real con la remuestreada

Representatividad en los nodos hojas en cada árbol

Se utiliza el porcentaje del 3.6 % de representatividad en los nodos hoja de los árboles en cada iteración.

Creación del modelo para la base train remuestreada

Se genera el modelo de xgBoost con la función *h2o.xgboost* en el lenguaje R. Los detalles de los hiperparámetros ya fueron detallados en el modelo xgBoost para G1.

Predicción del grupo G3 sobre toda la base

El tiempo de ejecución del modelo xgBoost sobre toda la base se presenta en la tabla 3.76.

Tiempo de ejecución		
user	**system**	**elapsed**
0,92	0	0,92

Tabla 3.76: Tiempos de ejecución de algoritmo xgBoost para G3

Matriz de coincidencias

Las mejores estimaciones del modelo xgBoost se presentan en las tablas 3.77 y 3.78.

Real	Estimado					MSE
	[800-1450]	(1450-2000]	(2000-2950]	(2950-5200]	(5200-35000]	
[800-1450]	39%	43%	16%	2%	0%	487755,2837
(1450-2000]	16%	40%	34%	9%	1%	679467,484
(2000-2950]	7%	27%	37%	26%	3%	1168443,881
(2950-5200]	3%	14%	28%	34%	21%	4645634,89
(5200-35000]	1%	5%	14%	27%	53%	57161828,02

Tabla 3.77: Matriz de proporción de coincidencias de la base train

Real	Estimado					MSE
	[800-1450]	(1450-2000]	(2000-2950]	(2950-5200]	(5200-35000]	
[800-1450]	36%	45%	17%	3%	0%	491835,9737
(1450-2000]	16%	40%	34%	10%	1%	761928,7977
(2000-2950]	8%	26%	37%	26%	3%	1226956,253
(2950-5200]	4%	14%	27%	34%	21%	4584173,406
(5200-35000]	1%	5%	14%	26%	53%	59463522,21

Tabla 3.78: Matriz de proporción de coincidencias de la base test

Por otro lado, la matriz de cálculo para el test Chi-cuadrado está dado por la tabla 3.79.

Real	Estimado						
	[800-1450]	(1450-2000]	(2000-2950]	(2950-5200]	(5200-35000]	Calculado	Teórico
[800-1450]	17,4	1,0	0,7	3,2	6,7	**82,7**	**26,3**
(1450-2000]	0,0	0,1	0,2	8,5	0,1	**Decisión**	
(2000-2950]	4,2	0,5	2,0	0,0	2,1	Se rechaza H0	
(2950-5200]	23,4	0,5	2,8	0,8	1,2		
(5200-35000]	3,4	0,7	0,7	0,5	2,3		

Tabla 3.79: Cálculo de Chi-cuadrado para algoritmo xgBoost para G3

Métricas

Los estadísticos de comparación se observan en la tabla 3.80.

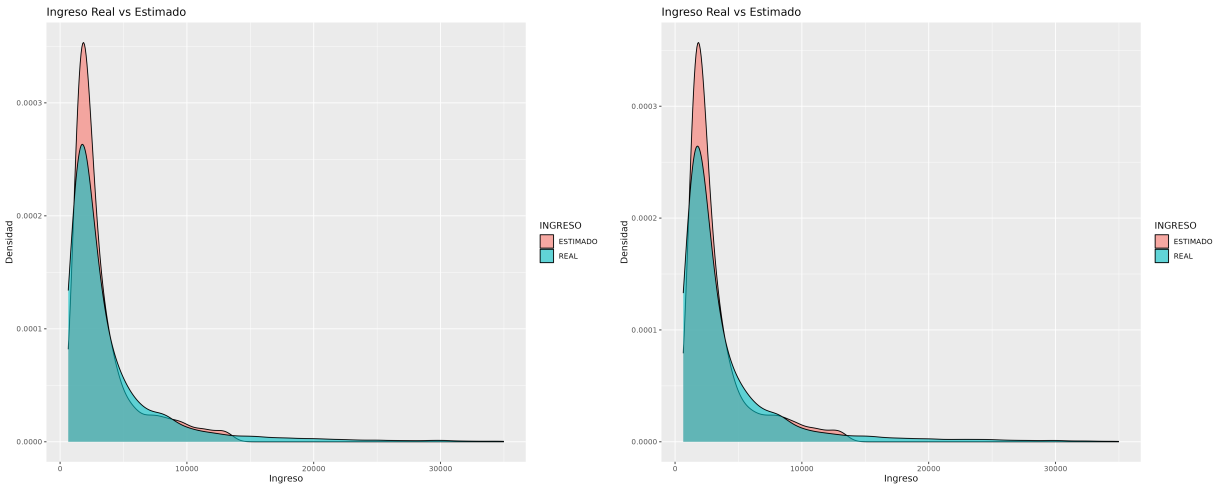
Entrenamiento	Métricas
MSE	12749768
MAE	1780,1952
Cruce	41 %
Cruce +/-	85 %
SubEstima	9 %
SobreEstim	6 %

Validación	Métricas
MSE	13388820
MAE	1820,1063
Cruce	40 %
Cruce +/-	84 %
SubEstima	9 %
SobreEstim	7 %

Tabla 3.80: Estadísticos de comparación

Densidades reales y estimadas

Por último se presenta las gráficas de las densidades reales y estimadas con el modelo xgBoost en la figura



((a)) Densidad base train ((b)) Densidad base test

Figura 3.29: Densidades reales y estimadas

3.8.4. Modelo RLM

Para el G3, también se procede de forma similar que en el G1.

Creación del modelo para la base train

Para el modelo de RLM, se utiliza la función $lm(\cdot)$ que se incluye dentro de las funciones básicas de R. Por lo tanto no se explican sus hiperparámetros.

Predicción del grupo G3 sobre toda la base

El tiempo de ejecución del modelo RLM sobre toda la base se presenta en la tabla 3.81.

Tiempo de ejecución		
user	**system**	**elapsed**
0,2	0,03	0,2

Tabla 3.81: Tiempos de ejecución de algoritmo RLM para G3

Matriz de coincidencias

Las mejores estimaciones del modelo se presentan en las tablas 3.82 y 3.83.

Real	Estimado					MSE
	[800-1450]	(1450-2000]	(2000-2950]	(2950-5200]	(5200-35000]	
[800-1450]	3%	24%	54%	19%	0%	2040274,55
(1450-2000]	2%	15%	41%	41%	2%	2144668,82
(2000-2950]	1%	9%	29%	53%	8%	2511233,35
(2950-5200]	1%	5%	18%	47%	28%	3879461,98
(5200-35000]	0%	2%	8%	31%	58%	53388192,40

Tabla 3.82: Matriz de proporción de coincidencias de la base train

Real	Estimado					MSE
	[800-1450]	(1450-2000]	(2000-2950]	(2950-5200]	(5200-35000]	
[800-1450]	3%	24%	54%	19%	0%	1919878,74
(1450-2000]	2%	14%	41%	41%	2%	2208131,15
(2000-2950]	1%	9%	29%	53%	8%	2606628,55
(2950-5200]	1%	5%	19%	47%	28%	4076611,23
(5200-35000]	0%	2%	8%	31%	59%	55394007,9

Tabla 3.83: Matriz de proporción de coincidencias de la base test

Por otro lado, la matriz de cálculo para el test Chi-cuadrado está dado por la tabla 3.84.

Real	Estimado						
	[800-1450]	(1450-2000]	(2000-2950]	(2950-5200]	(5200-35000]	Calculado	Teórico
[800-1450]	0,0	0,1	1,0	1,0	6,4	**43,6**	**26,3**
(1450-2000]	2,1	0,2	1,1	0,0	0,2	**Decisión**	
(2000-2950]	8,9	0,1	3,2	1,6	0,6	Se rechaza H0	
(2950-5200]	1,5	2,7	1,0	2,2	0,9		
(5200-35000]	2,4	3,0	0,1	0,0	3,3		

Tabla 3.84: Cálculo del Chi-2 para comparar distribución train y test

Métricas

Los estadísticos de comparación se los observa en la tabla 3.85.

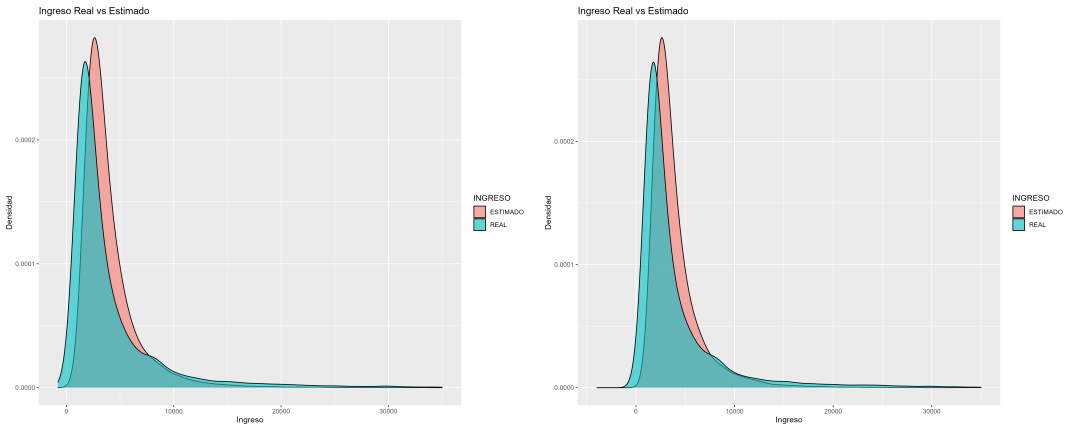
Entrenamiento	Métricas
MSE	12723772
MAE	2053,0
Cruce	30 %
Cruce +/-	71 %
SubEstima	3 %
SobreEstim	26 %

Validación	Métricas
MSE	13323882
MAE	2071,1
Cruce	30 %
Cruce +/-	71 %
SubEstima	3 %
SobreEstim	25 %

Tabla 3.85: Estadísticos de comparación

Densidades reales y estimadas

Por último se presenta las gráficas de las densidades reales y estimadas con el modelo de RLM en la figura 3.30.



((a)) Densidad base train ((b)) Densidad base test

Figura 3.30: Densidades reales y estimadas

3.8.5. Elección del mejor modelo entre RLM, RF, GBM y XGB

Luego de generar los cuatro modelos estadísticos, el mejor modelo es el xgBoost. Se decide que este es el mejor debido a que las matrices de coincidencia, los estadísticos de comparación y las densidades estimadas, son las mejores entre todos los modelos.

Al igual que en el G2, por motivos de poder computacional, se decide utilizar al modelo GBM para realizar las estimaciones sobre la base de validación del G3. Además, el GBM también tiene muy buenos resultados y además su algoritmo no es tan pesado computacionalmente hablando.

Capítulo 4

Discusión de resultados

4.1. Resultados del proceso de creación de los modelos estadísticos

Al dar inicio a los procedimientos de remuestreo, surgen interrogantes acerca de la pertinencia de aplicar estas técnicas a los tres grupos, ya que se podría observar una ligera variación en el comportamiento de las densidades reales al contrastar las bases, antes y después del remuestreo.

Estas incertidumbres dejan de existir cuando se generan los modelos con la base remuestreada. Al parecer, después de remuestrear, los nuevos individuos generados (o excluidos), influyen en que el modelo capte mayor información sobre la población en estudio, y por este motivo se obtienen buenas matrices de coincidencia, estadísticos calculados, y densidades estimadas.

Después de realizar cada simulación para la creación de los modelos en los tres grupos, es claro que el mejor modelo para realizar estas estimaciones es el *xgBoost*. A pesar de que no se tuvo disponible un computador con buenas características, y por tanto no se pudieron realizar tantos remuestreos como se lo hizo con los otros dos modelos no paramétricos, los resultados obtenidos con xgBoost son muy buenos.

Por otro lado, es claro que el modelo de *RLM* a pesar de ser un modelo clásico y muy utilizado en todas las industrias, no brinda resultados deseables para los valores de los ingresos de las personas naturales. El motivo al que se le puede atribuir la mala estimación, es que las variables de estudio no tienen una relación lineal respecto a la variable dependiente, lo que ocasiona que el modelo capte mal las características de la población de estudio.

Además, si bien el modelo de RF es muy robusto y utiliza técnicas para escoger variables e individuos y así evadir el subajuste y sobreajuste, no brinda resultados tan buenos. Sin embargo, es claro que los resultados de este modelo son mejores que el de RLM, pues no se necesita probar ninguna hipótesis sobre las posibles distribuciones de los errores o sobre las correlaciones que puedan existir entre las variables.

Se decide utilizar al modelo de *GBM*. No solo por todas las propiedades de los estadísticos calculados, de las matrices de coincidencia, y de sus estimaciones de los ingresos reales; sino también por su eficiencia en el computador utilizado para realizar el proyecto. El modelo de xgBoost, requiere de realizar varios procesos de forma paralela, como la creación de todos los árboles de decisión al mismo tiempo, lo cual disminuye la eficiencia del computador.

También, las estimaciones realizadas en este proyecto son mejores que las estimaciones realizadas por el Buró, que utilizó la misma base de datos para las estimaciones.

Al tener unas mejores predicciones, se procede a utilizar una **base de validación para evaluar el modelo de GBM** en las tres poblaciones.

Finalmente, se presentan los **indicadores de liquidez** calculados para los tres grupos; para los valores reales, y los valores estimados con el modelo de GBM.

4.2. Evaluación del mejor modelo con BDD de Validación para grupo G1

Se presentan los resultados obtenidos, luego de utilizar la base de validación para evaluar el modelo de GBM, generado para el G1.

Matriz de proporción de coincidencias de la base de validación y estadísticos calculados

Real	Estimado					MSE
	[450-550]	(550-700]	(700-850]	(850-1200]	(1200-2500]	
[450-550]	5%	51%	21%	16%	7%	135110,7194
(550-700]	6%	47%	19%	20%	8%	90703,53141
(700-850]	5%	35%	24%	23%	12%	85686,26519
(850-1200]	4%	26%	19%	26%	24%	164924,448
(1200-2500]	3%	17%	8%	20%	53%	563498,0204

Tabla 4.1: Matriz de proporción de coincidencias de la base

Entrenamiento	Métricas
MSE	318089,716
MAE	414,537159
Cruce	38%
Cruce +/-	71%
SubEstima	20%
SobreEstim	8%

Tabla 4.2: Estadísticos de comparación

Gráficas de densidades

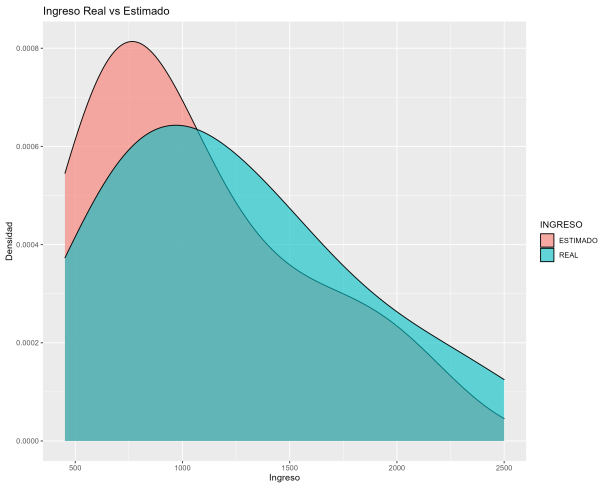


Figura 4.1: Densidad real y estimada de base de validación

4.3. Evaluación del mejor modelo con BDD de Validación para grupo G2

Se presentan los resultados obtenidos, luego de utilizar la base de validación para evaluar el modelo de GBM, generado para el G2.

Matriz de proporción de coincidencias de la base de validación y estadísticos calculados

Real	Estimado					MSE
	[450-700]	(700-900]	(900-1200]	(1200-1800]	(1800-5000]	
[450-700]	28%	38%	26%	8%	0%	121000,774
(700-900]	15%	39%	32%	14%	1%	73086,699
(900-1200]	9%	27%	37%	24%	3%	91795,0568
(1200-1800]	5%	14%	30%	33%	18%	260134,255
(1800-5000]	3%	7%	16%	27%	47%	2090382,86

Tabla 4.3: Matriz de proporción de coincidencias de la base

Entrenamiento	Métricas
MSE	837777,798
MAE	609,402768
Cruce	39%
Cruce +/-	79%
SubEstima	16%
SobreEstim	5%

Tabla 4.4: Estadísticos de comparación

Gráficas de densidades

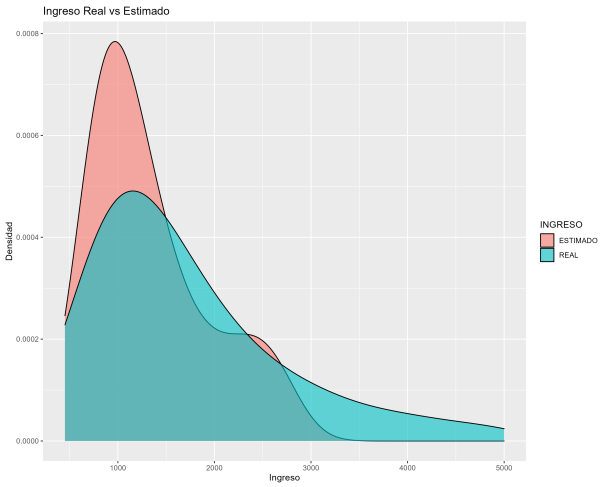


Figura 4.2: Densidad real y estimada de base de validación

4.4. Evaluación del mejor modelo con BDD de Validación para grupo G3

Se presentan los resultados obtenidos, luego de utilizar la base de validación para evaluar el modelo de GBM, generado para el G3.

Matriz de proporción de coincidencias de la base de validación y estadísticos calculados

Real	Estimado					MSE
	[800-1450]	(1450-2000]	(2000-2950]	(2950-5200]	(5200-35000]	
[800-1450]	35%	51%	12%	2%	0%	427577,622
(1450-2000]	17%	47%	26%	8%	0%	521262,371
(2000-2950]	8%	30%	29%	30%	3%	1372026,78
(2950-5200]	3%	16%	20%	34%	27%	3720317,61
(5200-35000]	1%	7%	10%	21%	61%	53476546,3

Tabla 4.5: Matriz de proporción de coincidencias de la base

Entrenamiento	Métricas
MSE	12551674,9
MAE	1764,19535
Cruce	42%
Cruce +/-	86%
SubEstima	10%
SobreEstim	5%

Tabla 4.6: Estadísticos de comparación

Gráficas de densidades

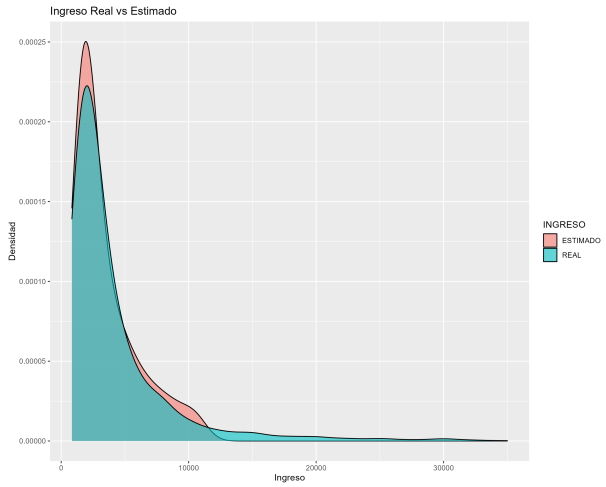


Figura 4.3: Densidad real y estimada de base de validación

4.5. Indicadores de liquidez calculados para la base de validación para grupo G1

Edad\Real	[450-550]	(550-700]	(700-850]	(850-1200]	(1200-2500]
[18,30)	2,05%	2,74%	3,03%	5,82%	7,09%
[30,40)	3,03%	4,11%	4,50%	10,57%	17,32%
[40,55)	2,05%	3,38%	2,64%	6,21%	13,60%
[55,65)	0,73%	0,68%	0,98%	1,61%	3,82%
[65,100)	0,24%	0,49%	0,34%	1,32%	1,61%

Tabla 4.7: Indicador de liquidez variable Edad - Ingreso Real

Edad\Estimado	[450-550]	(550-700]	(700-850]	(850-1200]	(1200-2500]
[18,30)	1,61%	7,44%	3,77%	3,52%	4,40%
[30,40)	1,52%	9,98%	5,82%	9,15%	13,06%
[40,55)	0,78%	7,58%	3,18%	6,07%	10,27%
[55,65)	0,15%	1,86%	1,27%	1,47%	3,08%
[65,100)	0,05%	0,78%	0,68%	1,17%	1,32%

Tabla 4.8: Indicador de liquidez variable Edad - Ingreso Estimado

E Civil \Real	[450-550]	(550-700]	(700-850]	(850-1200]	(1200-2500]
SOLTERO	2,97%	17,17%	8,71%	10,59%	12,77%
CASADO	0,89%	7,97%	4,30%	8,71%	16,13%
DIVORCIADO	0,10%	2,18%	1,48%	1,93%	2,77%
VIUDO	0,00%	0,20%	0,15%	0,10%	0,49%
UNIÓN LIBRE	0,00%	0,10%	0,05%	0,20%	0,05%

Tabla 4.9: Indicador de liquidez variable Estado Civil - Ingreso Real

E Civil \Estimado	[450-550]	(550-700]	(700-850]	(850-1200]	(1200-2500]
SOLTERO	4,90%	6,63%	7,13%	14,45%	19,10%
CASADO	2,47%	4,06%	3,51%	8,56%	19,40%
DIVORCIADO	0,59%	0,69%	0,89%	2,03%	4,26%
VIUDO	0,15%	0,05%	0,05%	0,25%	0,45%
UNIÓN LIBRE	0,05%	0,05%	0,00%	0,05%	0,25%

Tabla 4.10: Indicador de liquidez variable Estado Civil - Ingreso Estimado

Region\Real	[450-550]	(550-700]	(700-850]	(850-1200]	(1200-2500]
Amazonia	1,08%	1,42%	1,81%	3,77%	4,55%
Costa	0,59%	1,32%	1,76%	6,36%	14,77%
Sierra	6,46%	8,66%	7,93%	15,41%	24,12%

Tabla 4.11: Indicador de liquidez variable Región - Ingreso Real

Region\Estimado	[450-550]	(550-700]	(700-850]	(850-1200]	(1200-2500]
Amazonia	0,10%	2,74%	2,50%	2,74%	4,55%
Costa	1,96%	7,73%	2,40%	5,48%	7,24%
Sierra	2,05%	17,17%	9,83%	13,16%	20,35%

Tabla 4.12: Indicador de liquidez variable Región - Ingreso Estimado

4.6. Indicadores de liquidez calculados para la base de validación para grupo G2

Edad\Real	[450-700]	(700-900]	(900-1200]	(1200-1800]	(1800-5000]
[18,30)	0,98%	1,24%	1,80%	2,40%	2,38%
[30,40)	2,72%	4,31%	8,56%	10,69%	12,75%
[40,55)	3,27%	3,89%	5,43%	8,82%	13,78%
[55,65)	1,42%	1,30%	1,76%	2,70%	4,19%
[65,100)	0,52%	0,66%	1,04%	1,38%	2,01%

Tabla 4.13: Indicador de liquidez variable Edad - Ingreso Real

Edad\Estimado	[450-700]	(700-900]	(900-1200]	(1200-1800]	(1800-5000]
[18,30)	1,36%	2,03%	2,24%	1,67%	1,50%
[30,40)	3,20%	7,31%	10,49%	9,80%	8,23%
[40,55)	2,59%	6,46%	8,87%	8,91%	8,35%
[55,65)	0,69%	2,32%	2,84%	3,03%	2,51%
[65,100)	0,35%	1,09%	1,87%	1,34%	0,95%

Tabla 4.14: Indicador de liquidez variable Edad - Ingreso Estimado

E Civil \Real	[450-700]	(700-900]	(900-1200]	(1200-1800]	(1800-5000]
SOLTERO	3,98%	5,46%	8,52%	10,13%	10,71%
CASADO	3,87%	4,44%	8,02%	13,14%	20,57%
DIVORCIADO	0,92%	1,29%	1,73%	2,28%	3,10%
VIUDO	0,08%	0,20%	0,22%	0,37%	0,31%
UNIÓN LIBRE	0,07%	0,03%	0,11%	0,10%	0,33%

Tabla 4.15: Indicador de liquidez variable Estado Civil - Ingreso Real

E Civil \Estimado	[450-700]	(700-900]	(900-1200]	(1200-1800]	(1800-5000]
SOLTERO	4,30%	8,49%	10,68%	8,17%	7,16%
CASADO	2,92%	8,41%	12,64%	14,12%	11,95%
DIVORCIADO	0,80%	1,94%	2,49%	2,07%	2,03%
VIUDO	0,08%	0,24%	0,30%	0,31%	0,24%
UNIÓN LIBRE	0,05%	0,14%	0,13%	0,14%	0,21%

Tabla 4.16: Indicador de liquidez variable Estado Civil - Ingreso Estimado

Region\Real	[450-700]	(700-900]	(900-1200]	(1200-1800]	(1800-5000]
Amazonia	0,75%	0,98%	1,88%	2,30%	2,85%
Costa	0,57%	1,11%	3,39%	6,39%	9,54%
Sierra	7,59%	9,30%	13,31%	17,32%	22,72%

Tabla 4.17: Indicador de liquidez variable Región - Ingreso Real

Region\Estimado	[450-700]	(700-900]	(900-1200]	(1200-1800]	(1800-5000]
Amazonia	0,62%	1,75%	2,38%	2,16%	1,84%
Costa	2,19%	3,43%	5,16%	5,10%	5,12%
Sierra	5,38%	14,03%	18,77%	17,49%	14,57%

Tabla 4.18: Indicador de liquidez variable Región - Ingreso Estimado

4.7. Indicadores de liquidez calculados para la base de validación para grupo G3

Edad\Real	[800-1450]	(1450-2000]	(2000-2950]	(2950-5200]	(5200-35000]
[18,30)	0,81%	0,69%	0,38%	0,57%	0,43%
[30,40)	5,34%	6,12%	6,23%	6,01%	4,05%
[40,55)	5,49%	7,91%	8,42%	10,74%	10,16%
[55,65)	2,43%	3,19%	3,27%	3,83%	4,60%
[65,100)	1,74%	1,94%	1,83%	2,04%	1,78%

Tabla 4.19: Indicador de liquidez variable Edad - Ingreso Real

Edad\Estimado	[800-1450]	(1450-2000]	(2000-2950]	(2950-5200]	(5200-35000]
[18,30)	0,69%	0,98%	0,50%	0,40%	0,31%
[30,40)	4,32%	8,45%	5,63%	5,49%	3,86%
[40,55)	4,10%	11,52%	8,42%	8,70%	9,98%
[55,65)	1,35%	4,70%	3,34%	3,89%	4,04%
[65,100)	0,98%	3,07%	1,70%	1,86%	1,72%

Tabla 4.20: Indicador de liquidez variable Edad - Ingreso Estimado

E Civil \Real	[800-1450]	(1450-2000]	(2000-2950]	(2950-5200]	(5200-35000]
SOLTERO	5,59%	5,20%	4,87%	4,98%	3,67%
CASADO	8,04%	11,96%	12,84%	15,38%	14,78%
DIVORCIADO	1,81%	2,27%	2,05%	2,33%	2,14%
VIUDO	0,32%	0,33%	0,25%	0,33%	0,35%
UNIÓN LIBRE	0,07%	0,12%	0,12%	0,13%	0,08%

Tabla 4.21: Indicador de liquidez variable Estado Civil - Ingreso Real

E Civil \Estimado	[800-1450]	(1450-2000]	(2000-2950]	(2950-5200]	(5200-35000]
SOLTERO	3,97%	8,10%	4,76%	4,23%	3,26%
CASADO	6,11%	16,95%	12,14%	13,59%	14,20%
DIVORCIADO	1,12%	3,09%	2,20%	2,11%	2,08%
VIUDO	0,19%	0,45%	0,33%	0,34%	0,28%
UNIÓN LIBRE	0,04%	0,11%	0,16%	0,10%	0,11%

Tabla 4.22: Indicador de liquidez variable Estado Civil - Ingreso Estimado

Region\Real	[800-1450]	(1450-2000]	(2000-2950]	(2950-5200]	(5200-35000]
Amazonia	1,43%	1,74%	1,69%	1,37%	1,19%
Costa	2,21%	4,27%	4,49%	6,40%	5,34%
Sierra	12,16%	13,85%	13,96%	15,43%	14,49%

Tabla 4.23: Indicador de liquidez variable Región - Ingreso Real

Region\Estimado	[800-1450]	(1450-2000]	(2000-2950]	(2950-5200]	(5200-35000]
Amazonia	0,99%	2,04%	1,67%	1,55%	1,16%
Costa	2,68%	6,09%	3,87%	4,91%	5,15%
Sierra	7,76%	20,59%	14,05%	13,87%	13,60%

Tabla 4.24: Indicador de liquidez variable Región - Ingreso Estimado

4.8. Conclusiones y recomendaciones

4.8.1. Conclusiones

1. A lo largo de todas las simulaciones realizadas, se llega a la conclusión de que los hiperparámetros no se pueden ajustar exactamente a lo que se requiere.

   Se pudo observar que, si se tiene una matriz de coincidencias diagonal, es posible que el estadístico Chi-cuadrado sea grande; y si el Chi-cuadrado es pequeño, la matriz de coincidencias no es diagonal.

2. Es posible que los porcentajes de subestimación y sobreestimación sean pequeños, pero esto ocasione que el Chi-cuadrado sea grande.

3. Cambios pequeños en los hiperparámetros de las funciones de remuestreo, ocasionan grandes cambios en los modelos y en las diferentes estimaciones realizadas. Es por esto que se concluye que no se puede encontrar de manera exacta el modelo que mejor aproxime a la muestra disponible para generar los modelos.

4. La combinación de herramientas estadísticas utilizadas en este proyecto parece ser robusta para obtener buenas predicciones en bases de datos complejas. Al fijarse en las matrices de coincidencia, test Chi-cuadrado y estadísticos de comparación, se tiene la seguridad de que se están realizando buenas predicciones; pero al estudiar también las densidades muestrales de los valores reales y de las estimaciones, se puede asegurar que el modelo está realizando una buena predicción.

5. El modelo xgBoost no se encuentra disponible en el sistema operativo Windows, lo que resultó en la necesidad de recurrir a una computadora con sistema operativo Linux para su utilización. Debido a esta circunstancia, no fue posible llevar a cabo la optimización de los hiperparámetros de remuestreo de la misma manera que se hizo en el caso del modelo GBM. La razón es que la obtención de los valores óptimos para los hiperparámetros del GBM demandó numerosas horas de simulaciones, un proceso que no pudo ser llevado a

cabo utilizando el modelo xgBoost.

6. Después de revisar todos los resultados, se concluye que el G1 es el que peor comportamiento tiene con los modelos. Es probable que la muestra de esta población no sea lo suficientemente significativa, o falte de escoger alguna variable importante para esta población.

7. Por los resultados de la RLM en el G3, se puede concluir que en el G3 existe una mejor recolección de las características de la población en la base de datos. Esto se puede concluir por el hecho de que las matrices de coincidencia si tienen la forma de diagonales principales.

   Además, se observa esto en la distribución de las predicciones vs las reales; y también se puede concluir esto porque no se verificó ninguna hipótesis para la RLM y aún así el modelo para el G3 realizó estimaciones no tan sobreajustadas ni subajustadas como las brindadas por el Buró.

8. En el G1, a priori se conoce que predominan las variables del Sistema Comercial, pues al tratarse de personas con ingresos relativamente bajos, no pueden optar por créditos en bancos, pero no hay ninguna de esas variables escogidas para esta población.

   Por este motivo, el comportamiento del G1 no es tan bueno como esperamos. Encambio en G2 y G3 las variables que más describen a los individuos son las de entidades bancarias.

9. En general, los modelos generados son muy buenos para predecir a la población de individuos tarjetahabientes en el sistema de datos crediticio, pero siempre se debe tener en cuenta que puede existir sesgo por información que se introdujo mal en la base de datos de modelamiento. Puede haber sesgo en cola izquierda, cuando se colocan valores grandes para un individuo individuo con bajo ingreso real. Y también, puede haber sesgo en cola derecha cuando los individuos con ingresos altos tienen información registrada de ingreso real muy pequeño.

10. Se logró obtener mejores resultados para las estimaciones de los

ingresos reales con los modelos estadísticos utilizados en este proyecto, que el modelo usado previamente por el Buró.

Es posible mejorar estas estimaciones con una mayor potencia computacional.

11. Los modelos elegidos se ajustan muy bien a la base de datos de validación. Debido a esto, las densidades reales y estimadas de la variable de ingresos reales, se acercan bastante.

    Además, por la buena estimación de los modelos calculados, se obtienen indicadores de liquidez muy cercanos unos con otros, en la base de modelamiento y en la base de validación. Esto se observa al comparar uno a uno cada entrada de las matrices, en la misma posición.

    Por ejemplo, el 10.16% de individuos en el G3 tiene entre 40 y 55 años de edad y tiene un ingreso real entre USD$5200 y USD$35000 en la base de modelamiento, y el porcentaje en la base de validación es de 9.98%. Así mismo se puede ir comparando para cada variable y cada rango de ingresos.

12. Gracias a que los indicadores de liquidez no varían demasiado al ser calculados con los ingresos reales y los ingresos estimados, estos podrían brindar información útil para generar algún tipo de oferta. Además, debido a que las estimaciones realizadas son muy buenas, se tiene una buena confianza para tomar este tipo de decisiones.

4.8.2. Recomendaciones

1. Se recomienda tener claro la metodología que se va a utilizar para realizar una predicción. Es importante saber las herramientas que se desean utilizar, y también trabajar ordenadamente, de forma que no se cometan errores en el proceso de minería en la base de datos, y luego en la formulación del modelo estadístico de interés, para su posterior evaluación con una base de datos que no se utilizó en la creación del modelo.

2. Cuando se trabaja con variables continuas y el rango de estas variables es grande, se recomienda dividir a la población en grupos

significativos y luego hallar modelos específicos en cada población. Esto se realiza para evitar el sesgo en los ajustes, y en el presente proyecto, se crearon tres grupos: G1, G2, y G3.

3. Se recomienda balancear o remuestrear una base train de tal forma que se logre captar más información importante de la población.

   Para realizar los remuestreos de las bases de datos, se recomienda como paso previo, fijarse en los cuartiles y en las gráficas de densidades reales. En el primer remuestreo escoger parámetros de forma intuitiva y estimar el modelo. A partir del segundo remuestreo, lo mejor es fijarse en los resultados del modelo calculado con la primera base remuestreada.

   Al fijarse en la densidad estimada del modelo generado con la base del primer remuestreo, se pueden actualizar los percentiles para el segundo remuestreo, cambiándolo hacia un percentil cercano al valor en el que la densidad estimada con el primer remuestreo tenga un cambio muy grande (de 500-1000 unidades de USD$). Esto se realiza observando los percentiles.

   El proceso descrito se lo realiza hasta que se escojan parámetros que hagan que la base remuestreada sobreajuste o subajuste demasiado a los ingresos reales.

4. Se recomienda el uso de modelos no paramétricos cuando la variable dependiente sea dificil de estudiar, como por ejemplo las estimaciones de ingresos estudiadas en este proyecto.

   Es conocido que puede existir un sesgo en la introducción de la información de los clientes en la base de datos. Este sesgo operativo no puede ser medido, y tampoco puede ser controlado. Pero con los modelos robustos presentados en este proyecto, se puede realizar un mejor estudio de estas variables difíciles de estimar.

5. Se recomienda utilizar distintas herramientas estadísticas para analizar y describir los modelos que se requieren generar, así se pueden realizar diversas comparaciones y se pueden llegar a encontrar patrones para los datos o las mismas predicciones.

Si bien las distribuciones de las variables son importantes, las variables dependientes necesitan ser explicadas por otras variables independientes, por lo que fijarse solo en la distribución no es lo mejor para realizar predicciones.

6. Se recomienda estudiar los algoritmos y las teorías matemáticas utilizadas para la formulación de los modelos de predicción. Esto permite tener un mejor entendimiento de la herramienta de predicción que se va a estudiar, por tanto los cálculos de los hiperparámetros que mejor optimicen al modelo para el propósito que se le quiera brindar.

7. En caso que se tenga una red de computadoras para mejorar el rendimiento en el tiempo de ejecución de los modelos, es recomendable usar h2o para la formulación y predicción. En particular, siendo de mejor eficiencia y rendimiento la metodología de xgboost.

Referencias bibliográficas

[1] Vance W. Berger and YanYan Zhou. Kolmogorov–smirnov test: Overview. *Encyclopedia of Statistics in Behavioral Science, © John Wiley Sons, Ltd*, 2005. Tomado de: https://onlinelibrary.wiley.com/doi/epdf/10.1002/9781118445112.stat06558.

[2] Tianqi Chen. Introduction to boosted trees, 2014. Tomado de: https://web.njit.edu/~usman/courses/cs675_fall16/BoostedTree.pdf.

[3] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. *Arxiv*, 2016. Tomado de: https://arxiv.org/abs/1603.02754.

[4] Santiago de la Fuente Fernández. Aplicaciones de la chi-cuadrado: Tablas de contingencia, homogeneidad, dependencia e independencia. *Universidad Autónoma de Madrid*, 2016. Tomado de: https://www.fuenterrebollo.com/Aeronautica2016/contingencia.pdf.

[5] Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. *JSTOR*, 1999. Tomado de: https://www.jstor.org/stable/2699986.

[6] Elshaddai Harris. Understanding weight of evidence and information value, 2022. Tomado de: https://www.linkedin.com/pulse/understanding-weight-evidence-information-value-elshaddai-harris/.

[7] Kruthika Kulkarni. Understand weight of evidence and information value, 2021. Tomado de: https://www.analyticsvidhya.com/blog/2021/06/understand-weight-of-evidence-and-information-value/.

[8] Gilles Louppe. *Understanding Random Forests: From Theory to Practice*. PhD thesis, University of Liège, 2014. Tomado de: https://arxiv.org/abs/1407.7502.

[9] Tomonori Masui. All you need to know about gradient boosting algorithm, 2022. Tomado de: https://towardsdatascience.com/all-you-need-to-know-about-gradient-boosting-algorithm-part-1-regression-2520a34a502.

[10] Alexey Natekin and Alois Knoll. Gradient boosting machines, a tutorial. *Frontiers*, 2013. Tomado de: https://www.researchgate.net/publication/259653472_Gradient_Boosting_Machines_A_Tutorial.

[11] Ajit Samudrala. Unveiling mathematics behind xgboost, 2018. Tomado de: https://medium.com/@samudralaajit/unveiling-mathematics-behind-xgboost-c7f1b8201e2a.

[12] Eric Towers. Two-sample kolmogorov-smirnov test, 2020. Tomado de: https://math.stackexchange.com/questions/3577453/two-sample-kolmogorov-smirnov-test.

[13] Wikipedia. Kolmogorov–smirnov test, 2023. Tomado de: https://en.wikipedia.org/wiki/Kolmogorov%E2%80%93Smirnov_test.

Capítulo A

Anexos

A.1. Código, modelos, muestra de validación

Los mejores modelos hallados para cada grupo, el código en R para validar los modelos, y la muestra de validación; se pueden hallar en el siguiente enlace:

https://gitlab.com/FarhadKGA/tic_farhad/-/tree/main

A.2. Selección de variables para los modelos y Grid de Hiperparámetros para representatividad en nodos hoja

Variables escogidas Grupo 1			
CUANTITATIVAS		CUALITATIVAS	
Variable	KS	Variable	VI
ANTIGUEDAD_OP_SC	20.1 %	GRUPO_CUOTA	21.3 %
CUOTA_EST_OP	25.2 %		
cuota052	23.6 %		
cuotaCoo055	19.5 %		
cuotaD053	26.2 %		
cuotaEstimadaD24M416	25.0 %		
cuotaTCS374	26.2 %		
cuotaTotOp059	22.6 %		
DEUDA_TOTAL_SBS_SC_24M	25.7 %		
DEUDA_TOTAL_SC_OP_24M	22.9 %		
maxMontoOp096	24.5 %		
MaxMontoOpD24M417	26.8 %		
PROM_DEUDA_TOTAL_SC_OP_36M	23.4 %		
PROM_XVEN_SC_OP_36M	23.1 %		
salOpDia008	20.3 %		
salProm36M303	23.2 %		
salPromD36M319	26.0 %		
salTotOp040	20.4 %		
SalTotOpD383	23.4 %		
DEUDA_TOTAL_SCE_24M	26.0 %		

Tabla A.1: Variables escogidas para el grupo 1.

Variables escogidas Grupo 2			
CUANTITATIVAS		CUALITATIVAS	
Variable	KS	Variable	VI
cuotaD053	36.5 %	GRUPO_CUOTA	31.9 %
cuotaTCS374	36.5 %		
cuota052	34.4 %		
MaxMontoOpD24M417	33.7 %		
maxMontoOp096	32.3 %		
CUOTA_EST_OP	32.2 %		
DEUDA_TOTAL_SCE_24M	32.0 %		
DEUDA_TOTAL_SBS_SC_24M	32.0 %		
cuotaEstimadaD24M416	31.4 %		
salPromD36M319	31.1 %		
cuotaTotOp059	30.1 %		
salProm36M303	29.1 %		
PROM_DEUDA_TOTAL_SC_OP_36M	28.1 %		
PROM_XVEN_SC_OP_36M	27.9 %		
SalTotOpD383	27.8 %		
salPromD6M316	27.3 %		
disponibleEst094	26.2 %		
DEUDA_TOTAL_SC_OP_24M	26.2 %		
cuotaCoo055	25.3 %		
salProm6M095	25.2 %		
salTotOp040	25.1 %		
salOpDia008	25.1 %		
gastoPersonal093	21.6 %		
salTotOpBCoo036	21.0 %		
salOpDiaCoo004	21.0 %		
ANTIGUEDAD_OP_SC	19.9 %		
maxCupoTC089	19.8 %		

Tabla A.2: Variables escogidas para el grupo 2.

Variables escogidas Grupo 3			
CUANTITATIVAS		CUALITATIVAS	
Variable	KS	Variable	VI
cuotaD053	44.8%		
cuotaTCS374	44.8%		
cuota052	43.8%		
cuotaEstimadaD24M416	41.1%		
DEUDA_TOTAL_SCE_24M	37.5%		
DEUDA_TOTAL_SBS_SC_24M	37.5%		
MaxMontoOpD24M417	34.5%		
CUOTA_EST_OP	33.8%		
gastoPersonal093	33.6%		
maxMontoOp096	33.5%		
salPromD36M319	32.1%		
cuotaTotOp059	31.8%		
salProm36M303	30.5%		
cuotaBan054	30.2%		
disponibleEst094	29.7%		
SalTotOpD383	29.1%		
salPromD6M316	27.9%		
cuotaTotTC060	27.5%		
salProm6M095	26.6%		
salTotOp040	26.5%		
salOpDia008	26.5%		
PROM_DEUDA_TOTAL_SC_OP_36M	24.9%		
cuotaCoo055	24.9%		
PROM_XVEN_SC_OP_36M	24.7%		
TotalDeudaDBan406	24.2%		
CUOTA_EST_OP_TC	23.8%		
DEUDA_TOTAL_SC_OP_24M	22.8%		
TOT_CUPO	22.1%		
cupoTC086	22.1%		
SalCalOVDDBan410	22.1%		
maxCupoTC089	21.7%		
PROM_DEUDA_TOTAL_SBS_TC_36M	21.4%		
PROM_XVEN_SBS_TC_36M	21.2%		
PROM_DEUDA_TOTAL_SBS_OP_36M	20.4%		
salPromTC36M304	20.3%		
PROM_XVEN_SBS_OP_36M	20.2%		
DEUDA_TOTAL_OP_M	20.1%		
salTotTC041	19.9%		
DEUDA_TOTAL_SBS_TC_24M	19.8%		
DEUDA_TOTAL_TC	19.8%		
salTCDia030	19.6%		

Tabla A.3: Variables escogidas para el grupo 3.

num_trees	mtry	min_node	error	Porcentaje	# nodos hoja
400	8	945	325.879497	3.20%	31
300	8	945	325.8929977	3.20%	31
400	7	945	325.9703966	3.20%	31
300	7	945	326.0765568	3.20%	31
400	6	945	326.3740408	3.20%	31
300	6	945	326.3884617	3.20%	31
400	8	1004	326.5478474	3.40%	29
300	8	1004	326.5638765	3.40%	29
400	7	1004	326.6817227	3.40%	29
300	7	1004	326.7649815	3.40%	29
400	6	1004	327.0104837	3.40%	29
300	6	1004	327.0394388	3.40%	29
400	8	1063	327.0619485	3.60%	28
300	8	1063	327.0866645	3.60%	28
400	7	1063	327.2341869	3.60%	28
300	7	1063	327.3198524	3.60%	28
300	8	1122	327.5928637	3.80%	26
400	6	1063	327.5967545	3.60%	28
400	8	1122	327.60078	3.80%	26
300	6	1063	327.608593	3.60%	28
400	7	1122	327.7773946	3.80%	26
300	7	1122	327.847701	3.80%	26
400	8	1181	328.0661071	4.00%	25
300	8	1181	328.0748525	4.00%	25
400	6	1122	328.1049417	3.80%	26
300	6	1122	328.1414022	3.80%	26
400	7	1181	328.2775679	4.00%	25
300	7	1181	328.3519957	4.00%	25
400	6	1181	328.5600298	4.00%	25
300	6	1181	328.5826527	4.00%	25

Tabla A.4: **Grid de Hiperparámetros - Grupo 1**. La combinación de hiperparámetros escogida, es la que está subrayada en amarillo.

num_trees	mtry	min_node	error	Porcentaje	# nodos hoja
300	11	2021	553.5378419	3.20%	31
400	11	2021	553.5386207	3.20%	31
400	10	2021	554.3916545	3.20%	31
300	10	2021	554.4092615	3.20%	31
300	9	2021	554.9019713	3.20%	31
400	9	2021	554.9211004	3.20%	31
300	11	2147	555.012329	3.40%	29
400	11	2147	555.0502108	3.40%	29
300	10	2147	555.7344071	3.40%	29
400	10	2147	555.7532207	3.40%	29
300	11	2273	555.8992655	3.60%	28
400	11	2273	555.9106109	3.60%	28
300	9	2147	556.0970668	3.40%	29
400	9	2147	556.173036	3.40%	29
400	11	2399	556.6939739	3.80%	26
400	10	2273	556.7010713	3.60%	28
300	11	2399	556.7351685	3.80%	26
300	10	2273	556.742675	3.60%	28
300	9	2273	557.1180002	3.60%	28
400	9	2273	557.2510398	3.60%	28
400	10	2399	557.2831821	3.80%	26
300	10	2399	557.298701	3.80%	26
400	11	2526	557.5963488	4.00%	25
300	11	2526	557.6296596	4.00%	25
300	9	2399	557.8332119	3.80%	26
400	9	2399	557.974858	3.80%	26
300	10	2526	558.3658112	4.00%	25
400	10	2526	558.3850107	4.00%	25
300	9	2526	558.6591901	4.00%	25
400	9	2526	558.7473986	4.00%	25

Tabla A.5: **Grid de Hiperparámetros - Grupo 2**. La combinación de hiperparámetros escogida, es la que está subrayada en amarillo.

num_trees	mtry	min_node	error	Porcentaje	# nodos hoja
300	15	1039	2894.569215	3.20%	31
400	15	1039	2895.365471	3.20%	31
400	13	1039	2897.487987	3.20%	31
400	14	1039	2897.787401	3.20%	31
300	13	1039	2898.441389	3.20%	31
300	15	1104	2899.128242	3.40%	29
400	15	1104	2899.786417	3.40%	29
300	14	1039	2899.966098	3.20%	31
400	13	1104	2902.971896	3.40%	29
400	14	1104	2903.301455	3.40%	29
300	13	1104	2904.090539	3.40%	29
300	14	1104	2905.338266	3.40%	29
300	15	1169	2905.537713	3.60%	28
400	15	1169	2906.082904	3.60%	28
400	14	1169	2908.111339	3.60%	28
400	13	1169	2909.068563	3.60%	28
300	14	1169	2909.872657	3.60%	28
300	15	1234	2910.355178	3.80%	26
400	15	1234	2910.588306	3.80%	26
300	13	1169	2910.593118	3.60%	28
400	14	1234	2912.442224	3.80%	26
400	13	1234	2914.150735	3.80%	26
300	14	1234	2915.05526	3.80%	26
300	15	1299	2915.354128	4.00%	25
400	15	1299	2915.387253	4.00%	25
300	13	1234	2915.777125	3.80%	26
400	14	1299	2918.290141	4.00%	25
400	13	1299	2919.37112	4.00%	25
300	13	1299	2920.241223	4.00%	25
300	14	1299	2920.696348	4.00%	25

Tabla A.6: **Grid de Hiperparámetros - Grupo 3**. La combinación de hiperparámetros escogida, es la que está subrayada en amarillo.