**Universidad Nacional Mayor de San Marcos**
**Universidad del Perú. Decana de América**
Facultad de Ciencias Matemáticas
Escuela Profesional de Matemática

# Modelo predictivo basado en el algoritmo CatBoost para la estimación de la probabilidad de default en tarjetas de crédito en una entidad financiera

**TESIS**
Para optar el Título Profesional de Licenciado en Matemática

**AUTOR**

Jesús Adalberto ZELAYA CONTRERAS

**ASESOR**

Dr. Carlos Alberto PEÑA MIRANDA

Lima, Perú

2025




Reconocimiento - No Comercial - Compartir Igual - Sin restricciones adicionales

https://creativecommons.org/licenses/by-nc-sa/4.0/

Usted puede distribuir, remezclar, retocar, y crear a partir del documento original de modo no comercial, siempre y cuando se dé crédito al autor del documento y se licencien las nuevas creaciones bajo las mismas condiciones. No se permite aplicar términos legales o medidas tecnológicas que restrinjan legalmente a otros a hacer cualquier cosa que permita esta licencia.

**Referencia bibliográfica**

---

Zelaya, J. (2025). *Modelo predictivo basado en el algoritmo CatBoost para la estimación de la probabilidad de default en tarjetas de crédito en una entidad financiera*. [Tesis de pregrado, Universidad Nacional Mayor de San Marcos, Facultad de Ciencias Matemáticas, Escuela Profesional de Matemática]. Repositorio institucional Cybertesis UNMSM.

---

## Metadatos complementarios

| | |
|---|---|
| **Datos de autor** | |
| Nombres y apellidos | Jesús Adalberto Zelaya Contreras |
| Tipo de documento de identidad | DNI |
| Número de documento de identidad | 72943195 |
| URL de ORCID | https://orcid.org/0009-0003-4767-6366 |
| **Datos de asesor** | |
| Nombres y apellidos | Carlos Alberto Peña Miranda |
| Tipo de documento de identidad | DNI |
| Número de documento de identidad | 10699143 |
| URL de ORCID | https://orcid.org/0000-0002-4339-4615 |
| **Datos del jurado** | |
| **Presidente del jurado** | |
| Nombres y apellidos | Alex Armando Cruz Huallpara |
| Tipo de documento | DNI |
| Número de documento de identidad | 40949706 |
| **Miembro del jurado 1** | |
| Nombres y apellidos | Frank Duberlee Alvarez Huertas |
| Tipo de documento | DNI |
| Número de documento de identidad | 45173557 |
| **Datos de investigación** | |
| Línea de investigación | Optimización |

| | |
|---|---|
| Grupo de investigación | METNUM |
| Agencia de financiamiento | Ninguno |
| Ubicación geográfica de la investigación | Lima, Perú |
| Año o rango de años en que se realizó la investigación | 2024-2025 |
| URL de disciplinas OCDE | MATEMATICA PURA<br>https://purl.org/pe-repo/ocde/ford#1.01.00 |

UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS
(Universidad del Perú, DECANA DE AMÉRICA)
FACULTAD DE CIENCIAS MATEMATICAS
Ciudad Universitaria - Av. Venezuela S/N cuadra 34
Teléfono: 619-7000, Anexo 1610
Correo Postal: 05-0021, E-mail: eapmat@unmsm.edu.pe
Lima - Perú

## *Escuela Profesional de Matemática*

*ACTA DE SUSTENTACIÓN DE TESIS PARA OPTAR EL TÍTULO PROFESIONAL DE LICENCIADO EN MATEMÁTICA*
*(Modalidad Virtual)*

*En la UNMSM - Ciudad Universitaria - Facultad de Ciencias Matemáticas, siendo las 16:00 horas del jueves 15 de mayo del 2025, se reunieron los docentes designados como Miembros del Jurado Evaluador de Tesis: Dr. Alex Armando Cruz Huallpara (PRESIDENTE), Mg. Frank Duberlee Alvarez Huertas (MIEMBRO) y el Dr. Carlos Alberto Peña Miranda (MIEMBRO ASESOR), para la sustentación de la tesis titulada:* ***"MODELO PREDICTIVO BASADO EN EL ALGORITMO CATBOOST PARA LA ESTIMACIÓN DE LA PROBABILIDAD DE DEFAULT EN TARJETAS DE CRÉDITO EN UNA ENTIDAD FINANCIERA"*** *presentado por el señor Bachiller* ***Jesús Adalberto Zelaya Contreras****, para optar el Título Profesional de Licenciado en Matemática.*

*Luego de la exposición de la tesis, el Presidente invitó al expositor a dar respuesta a las preguntas formuladas.*

*Realizada la evaluación correspondiente por los miembros del Jurado Evaluador, el expositor mereció la aprobación* ***Sobresaliente con mención*** *con un calificativo promedio de* ***diecinueve (19).***

*A continuación, el Presidente del Jurado, Dr. Alex Armando Cruz Huallpara, manifestó que el señor Bachiller Jesús Adalberto Zelaya Contreras, en vista de haber aprobado la sustentación de su tesis, será propuesto para que se le otorgue el Título Profesional de Licenciado en Matemática.*

*Siendo las 17:00 horas se levantó la sesión firmando para constancia la presente Acta en tres (3) copias originales.*

*Dr. Alex Armando Cruz Huallpara*
*PRESIDENTE*

*Mg. Frank Duberlee Alvarez Huertas*
*MIEMBRO*

*Dr. Carlos Alberto Peña Miranda*
*MIEMBRO ASESOR*

# CERTIFICADO DE SIMILITUD

Yo Dr. Carlos Alberto Peña Miranda en mi condición de asesor acreditado con la Resolución Decanal N.° 000446-2025-D-FCM/UNMSM de la tesis académico, cuyo título es "**MODELO PREDICTIVO BASADO EN EL ALGORITMO CATBOOST PARA LA ESTIMACIÓN DE LA PROBABILIDAD DE DEFAULT DE TARJETAS DE CRÉDITO EN UNA ENTIDAD FINANCIERA"** presentado por el bachiller Sr. **Jesús Adalberto Zelaya Contreras**, para optar el título Profesional de Licenciado en Matemática CERTIFICO que se ha cumplido con lo establecido en la Directiva de Originalidad y de Similitud de Trabajos Académicos, de Investigación y Producción Intelectual. Según la revisión, análisis y evaluación mediante el software de similitud textual, el documento evaluado cuenta con el porcentaje de 7% de similitud, nivel **PERMITIDO** para continuar con los trámites correspondientes y **para su publicación en el repositorio institucional**.

Se emite el presente certificado en cumplimiento de lo establecido en las normas vigentes, como uno de los requisitos para la obtención del grado/ título/ especialidad correspondiente.

Lima, 21 de mayo de 2025





Dr. Carlos Alberto PEÑA MIRANDA

DNI: 10699143

# Dedicatoria

A mis padres, pilares fundamentales de mi vida, cuyo amor incondicional y sacrificios han sido la base de todo lo que soy y logro.

Al profesor Pedro, cuya pasión por las matemáticas en aquellas clases de secundaria no solo enseñaba fórmulas, sino que encendió en mí la chispa que me llevaría a elegir esta hermosa carrera.

A la Universidad Nacional Mayor de San Marcos, que forjó mi carácter académico, y a todos aquellos compañeros y mentores que caminaron junto a mí en esta trayectoria: en las victorias que celebramos y en los desafíos que superamos con resiliencia.

A Dios, por ser mi fortaleza en cada paso de este viaje.

*«Él les dijo: Venid y veréis. Entonces fueron y vieron dónde se hospedaba; y se quedaron con Él aquel día»* **(Jn 1:39)**

# Agradecimientos

Expreso mi sincero agradecimiento:

Al Dr. Carlos Alberto Peña Miranda, por su paciencia, valiosa orientación y compromiso durante el desarrollo de esta investigación.

A mis compañeros de trabajo y superiores, cuyo apoyo constante, orientación profesional contribuyen significativamente a mi crecimiento como profesional.

A mis amigos y familiares, por brindarme su apoyo emocional y motivación incondicional durante los períodos de mayor exigencia académica.

Finalmente, a todos aquellos que, de manera directa o indirecta, hicieron posible la culminación de este trabajo.

# Índice general

**Dedicatoria** **i**

**Agradecimientos** **ii**

**Resumen** **v**

**Abstract** **vi**

**Introducción** **1**

**1. Modelos predictivos** **3**

1.1. Definición de modelos predictivos . . . . . 3

1.2. Clasificación de modelos predictivos . . . . . 5

1.3. Ejemplos de modelos predictivos . . . . . 8

1.3.1. Regresión logística . . . . . 8

1.3.2. Árboles de decisión . . . . . 9

1.3.3. Random forest . . . . . 11

1.4. Limitaciones de los modelos clásicos . . . . . 12

**2. Algoritmo CatBoost** **15**

2.1. Introducción al gradient boosting . . . . . 15

2.2. Fundamento matemático de CatBoost . . . . . 22

2.2.1. Función de pérdida logarítmica . . . . . 24

2.2.2. Manejo de variables categóricas . . . . . 25

2.3. Ventajas de CatBoost sobre otros algoritmos . . . . . 29

**3. Planteamiento del problema** **32**

3.1. Descripción del problema . . . . . 32

3.2. Diseño del modelo . . . . . 32

3.2.1. Preprocesamiento de datos . . . . . 33
3.3. Métricas de evaluación . . . . . 61

**4. Resultados de la investigación** **69**
4.1. Análisis de los resultados . . . . . 69
4.2. Importancia de las variables en el desempeño del modelo . . . . . 72

**5. Conclusiones** **75**

**Referencias bibliográficas** **77**

# RESUMEN

## MODELO PREDICTIVO BASADO EN EL ALGORITMO CATBOOST PARA LA ESTIMACIÓN DE LA PROBABILIDAD DE DEFAULT EN TARJETAS DE CRÉDITO EN UNA ENTIDAD FINANCIERA

**JESUS ADALBERTO ZELAYA CONTRERAS**

Asesor: Dr. Carlos Alberto Peña Miranda.

Título Obtenido: Licenciado en Matemática.

---

En la presente investigación se aborda el problema de la predicción de incumplimiento (*default*) en solicitantes no bancarizados de tarjetas de crédito, población que carece de historial crediticio y representa un desafío para las instituciones financieras. El objetivo fue desarrollar un modelo predictivo basado en el algoritmo *CatBoost*. La metodología se estructuró en cuatro etapas: recolección y preparación de datos históricos, implementación del modelo *CatBoost*, evaluación mediante métricas como el coeficiente GINI e interpretación de resultados. Los resultados mostraron un buen desempeño predictivo, con un GINI de 45 % en entrenamiento y 42.8 % en validación, demostrando capacidad para diferenciar entre clientes de alto y bajo riesgo. Se concluye que el modelo propuesto ofrece una herramienta efectiva para la gestión del riesgo crediticio en poblaciones no bancarizadas, con potencial para mejorar las decisiones de otorgamiento de crédito.

**Palabras clave**: Modelo predictivo, CatBoost, riesgo crediticio, aprendizaje automático.

# ABSTRACT


**PREDICTIVE MODEL BASED ON THE CATBOOST ALGORITHM FOR ESTIMATING THE PROBABILITY OF DEFAULT IN CREDIT CARDS AT A FINANCIAL INSTITUTION**

**JESUS ADALBERTO ZELAYA CONTRERAS**

Advisor: Dr. Carlos Alberto Peña Miranda.

Obtained Title: Degree in Mathematics.

---

This research addresses the problem of predicting default in unbanked credit card applicants, a population lacking credit history that poses challenges for financial institutions. The objective was to develop a predictive model based on the *CatBoost* algorithm. The methodology was structured in four stages: collection and preparation of historical data, implementation of the *CatBoost* model, evaluation using metrics such as the GINI coefficient, and interpretation of results. The results showed good predictive performance, with a GINI of 45 % in training and 42.8 % in validation, demonstrating the ability to differentiate between high-risk and low-risk customers. It is concluded that the proposed model provides an effective tool for credit risk management in unbanked populations, with potential to improve credit granting decisions.



**Key words**: Predictive model, CatBoost, credit risk, machine learning.

# Introducción

La predicción de riesgo crediticio es una de las tareas más críticas en el sector financiero, ya que permite a las instituciones evaluar la probabilidad de que un cliente incumpla con sus obligaciones de pago. En un entorno económico cada vez más complejo y competitivo, la capacidad de anticipar el comportamiento de los deudores se ha convertido en un factor clave para la sostenibilidad de las entidades financieras. Los modelos predictivos, basados en técnicas de aprendizaje automático, han emergido como herramientas esenciales para mejorar la precisión en la estimación del riesgo, optimizar la asignación de crédito y minimizar las pérdidas asociadas al incumplimiento. En este contexto, el desarrollo de modelos robustos y eficientes se ha vuelto una prioridad para las instituciones que buscan mantener un equilibrio entre la rentabilidad y el control del riesgo.

El problema central de esta investigación radica en la predicción de la probabilidad de incumplimiento (default) en solicitantes de tarjetas de crédito. Específicamente, se busca estimar la probabilidad de que un cliente caiga en mora después de un determinado número de días de atraso en los meses siguientes al otorgamiento de la tarjeta. Este problema es particularmente desafiante porque la población sobre la cual se busca estimar la probabilidad de incumplimiento (default) es la denominada *no bancarizada*. Se trata de personas que, en el último año, no han solicitado préstamos, adquirido tarjetas de crédito ni realizado movimientos financieros que no sean pasivos. Esta falta de historial crediticio dificulta la predicción, ya que no se dispone de información suficiente. Por ello, es necesario recurrir a otro tipo de características y explorar fuentes de datos alternativas para lograr una estimación adecuada. Además, este proceso contribuye a la inclusión financiera, permitiendo bancarizar a nuevos usuarios. La falta de modelos precisos puede llevar a decisiones crediticias no óptimas, aumentando el riesgo de pérdidas financieras y afectando la estabilidad de las instituciones.

El objetivo general de esta investigación es desarrollar un modelo predictivo basado en el algoritmo CatBoost para estimar la probabilidad de default en solicitantes de tarjetas de crédito, utilizando información del perfil del cliente. Para alcanzar este objetivo, se plantean cuatro objetivos específicos: (1) investigar el fundamento matemático del algoritmo CatBoost, con énfasis en la optimización de funciones de pérdida y el manejo de variables categóricas; (2) implementar y entrenar un modelo CatBoost para predecir la probabilidad de incumplimiento; (3) evaluar el rendimiento del modelo utilizando métricas como el GINI, AUC-ROC y la prueba de Kolmogórov-Smirnov; y (4) analizar la importancia de las variables en el modelo para identificar los factores más relevantes en la predicción del default.

Esta investigación es relevante tanto desde un punto de vista teórico como práctico. Teóricamente, contribuye al campo de las matemáticas aplicadas al explorar el fundamento matemático del algoritmo CatBoost, específicamente su enfoque en la optimización de funciones de pérdida y su manejo de variables categóricas mediante técnicas como el ordered target encoding. En el ámbito práctico, el desarrollo del modelo ofrecerá una nueva herramienta para las instituciones financieras, permitiéndoles mejorar la gestión del riesgo crediticio y reducir las pérdidas asociadas al incumplimiento de pagos. Un modelo predictivo preciso y eficiente facilitará la toma de decisiones en la asignación de crédito, contribuyendo a una gestión más sostenible y efectiva de los recursos financieros.

El presente trabajo de tesis, está organizado en cinco capítulos. El primer capítulo introduce los conceptos previos, como qué es un modelo predictivo, la clasificación de modelos, algunos ejemplos de modelos clásicos y las limitaciones que suelen presentar. La segunda sección profundiza en el algoritmo CatBoost, explicando su fundamento matemático y sus ventajas sobre otros métodos. La tercera sección describe el problema a tratar, el conjunto de datos, las variables que se utilizarán para el entrenamiento del modelo, el preprocesamiento de estas, el proceso de entrenamiento y las métricas de evaluación. La cuarta sección presenta un análisis de los resultados obtenidos, analizando el rendimiento del modelo y la importancia de las variables. Finalmente, la quinta sección presenta las conclusiones y recomendaciones para trabajos futuros. Esta estructura busca guiar al lector a través de un flujo lógico que va desde los conceptos generales hasta los resultados específicos de la investigación.

# Capítulo 1

# Modelos predictivos

En esta sección, se presentan los fundamentos teóricos y las herramientas matemáticas necesarias para comprender el desarrollo de modelos predictivos en el contexto del riesgo crediticio. Estos modelos son esenciales en el análisis de datos, ya que permiten anticipar eventos futuros a partir de patrones históricos, facilitando la toma de decisiones en instituciones financieras y otras áreas.

Particularmente en el ámbito del riesgo crediticio, los modelos de aprendizaje supervisado, como la regresión logística y los árboles de decisión, han demostrado ser efectivos para predecir la probabilidad de incumplimiento de pago. Sin embargo, cada uno de estos modelos tiene sus propias limitaciones, especialmente en el manejo de variables categóricas, la sensibilidad a datos desbalanceados y la capacidad de modelar relaciones no lineales.

El objetivo de esta sección es proporcionar una comprensión clara de estos modelos clásicos, analizando sus ventajas y desventajas, y justificando la necesidad de enfoques más avanzados, como el algoritmo CatBoost. Este último se ha desarrollado para abordar de manera más eficiente los desafíos asociados con conjuntos de datos financieros complejos, mejorando la precisión y estabilidad de las predicciones.

A lo largo de este capítulo, se explorarán los principios matemáticos que sustentan estos modelos, sus aplicaciones prácticas y los criterios clave para evaluar su desempeño en escenarios del mundo real.

## 1.1. Definición de modelos predictivos

Un **modelo predictivo** es una función matemática $f : X \to Y$ que mapea un conjunto de variables de entrada $X$ (también llamadas características o *features*) a un conjunto de salidas $Y$ (también llamadas etiquetas o *labels*).

Formalmente, dado un conjunto de datos $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$, donde $x_i \in X$ es un vector de características y $y_i \in Y$ es la etiqueta correspondiente, podemos considerar un caso en el que el vector de características $x_i$ esté compuesto por tres valores que representan atributos del cliente, como la edad, el promedio de saldo en cuenta bancaria y la cantidad de transacciones en el último mes. Por su parte, la etiqueta $y_i$ puede tomar valores como *aprobado* o *rechazado*.

**Ejemplo 1.1.**

Por ejemplo, en un modelo de aprobación de créditos, $X$ podría incluir información como el historial de pagos de un cliente, sus ingresos mensuales y su nivel de endeudamiento. La salida $Y$ sería la decisión del modelo, como *aprobado* o *rechazado*. En este caso, la función $f$ aprende a predecir si un cliente es apto para recibir un crédito en función de sus características financieras.

El objetivo de un modelo predictivo es aprender una función $f$ que minimice una función de pérdida $L(f(x_i), y_i)$. La función de pérdida $L$ cuantifica la discrepancia entre las predicciones del modelo $f(x_i)$ y las etiquetas reales $y_i$. Esto se debe a que, mientras menor sea la diferencia entre lo predicho y lo real, mejor será el desempeño del modelo. En otras palabras, la función de pérdida nos permite cuantificar qué tan bien está funcionando nuestro modelo entrenado. Matemáticamente, esto se expresa como:

$$f^* = \arg\min_{f \in \mathcal{F}} \sum_{i=1}^{n} L(f(\mathbf{x}_i), y_i),$$

donde $\mathcal{F}$ es el espacio de funciones posibles que el modelo puede aprender.

El operador arg mín se utiliza para indicar que buscamos la función $f^*$ dentro del espacio de funciones $\mathcal{F}$ que minimiza la suma de la función de pérdida $L$ en el conjunto de datos. En otras palabras, no solo nos interesa el valor mínimo de la suma de pérdidas, sino la función $f^*$ que lo alcanza.

**Ejemplo 1.2.**

Por ejemplo, en un problema de regresión lineal, si consideramos una función lineal del tipo:

$$f(x) = w_0 + w_1 x,$$

donde $w_0$ y $w_1$ son parámetros a ajustar, y utilizamos la función de pérdida de error cuadrático medio:

$$L(f(x_i), y_i) = (f(x_i) - y_i)^2,$$

entonces el proceso de entrenamiento del modelo consiste en encontrar los valores óptimos de $w_0$ y $w_1$ que minimicen la suma de los errores cuadráticos:

$$(w_0^*, w_1^*) = \arg\min_{w_0, w_1} \sum_{i=1}^{n} (w_0 + w_1 x_i - y_i)^2.$$

Aquí, arg mín nos proporciona los valores $w_0^*$ y $w_1^*$ que hacen que la función de pérdida sea mínima, es decir, los parámetros que generan la mejor función predictiva dentro del conjunto de funciones lineales posibles.

### Importancia de los modelos predictivos en la toma de decisiones

Los modelos predictivos son fundamentales en la toma de decisiones, ya que permiten anticipar eventos futuros basados en datos históricos. En el contexto del riesgo crediticio, un modelo predictivo puede estimar la probabilidad de que un cliente incumpla con sus obligaciones de pago, lo que permite a las instituciones financieras optimizar la asignación de crédito y minimizar las pérdidas asociadas al incumplimiento.
Según Thomas, Crook y Edelman (2017), en su obra *Credit Scoring and Its Applications*, el uso de modelos predictivos en el análisis de crédito ha demostrado ser una de las herramientas más eficaces para reducir la morosidad y mejorar la rentabilidad de las entidades financieras. La precisión de estos modelos es crucial, ya que decisiones basadas en predicciones erróneas pueden tener consecuencias financieras significativas, afectando tanto a las instituciones como a los clientes.

## 1.2. Clasificación de modelos predictivos

Los modelos predictivos se clasifican en dos grandes categorías: **supervisados** y **no supervisados**. Además, dentro de los modelos supervisados, se distinguen dos tipos principales: **regresión** y **clasificación**.

- **Modelos supervisados**: En estos modelos, el conjunto de datos $\mathcal{D}$ incluye tanto las características $\mathbf{x}_i$ como las etiquetas $y_i$. El objetivo es aprender una función $f$ que generalice bien a nuevos datos.

**Ejemplo 1.3.**

Por ejemplo, en un sistema de clasificación de correos electrónicos, cada correo puede representarse mediante un conjunto de características $\mathbf{x}_i$, como la cantidad de palabras en el asunto, la presencia de ciertas palabras clave y el remitente del mensaje. La etiqueta $y_i$ indicaría si el correo es *spam* o *no spam*. Un modelo supervisado entrenado con datos etiquetados puede aprender a clasificar correctamente nuevos correos basándose en patrones identificados en el conjunto de entrenamiento.

Dentro de los modelos supervisados, se encuentran:

- **Regresión**: En problemas de regresión, la variable objetivo $y_i$ es continua. El objetivo es predecir un valor numérico, como el precio de una vivienda o el rendimiento de un activo financiero. Un ejemplo clásico es la regresión lineal, donde $f(\mathbf{x}_i) = \mathbf{w}^T\mathbf{x}_i + b$, con $\mathbf{w}$ siendo un vector de pesos y $b$ un término de sesgo.

  **Ejemplo 1.4.**

  Supongamos que queremos predecir el precio de una vivienda en función de su área (en metros cuadrados). Si el modelo aprendido es:

  $$f(x) = 5000 \cdot x + 20000$$

  donde $x$ es el área de la vivienda en metros cuadrados, el coeficiente 5000 representa el precio por metro cuadrado y el término independiente 20000 es un ajuste base. Si una vivienda tiene un área de $x = 100\ \text{m}^2$, el precio estimado sería:

  $$f(100) = 5000 \times 100 + 20000 = 520000$$

- **Clasificación**: En problemas de clasificación, la variable objetivo $y_i$ es discreta. El objetivo es asignar una etiqueta a cada instancia, como predecir si un cliente incurrirá en default (clasificación binaria) o clasificar un correo electrónico como spam o no spam. Un ejemplo común es la regresión logística, donde $f(\mathbf{x}_i) = \sigma(\mathbf{w}^T\mathbf{x}_i + b)$, con $\sigma$ siendo la función sigmoide.

  **Ejemplo 1.5.**

  Supongamos que queremos predecir si un cliente aprobará un crédito ($y = 1$)

o no ($y = 0$) en función de su ingreso mensual ($x$, en miles de dólares). Si el modelo aprendido es:

$$f(x) = \sigma(-3 + 2x),$$

donde $\sigma(z) = \frac{1}{1+e^{-z}}$ es la función sigmoide. Si un cliente tiene un ingreso de $x = 3$ mil dólares, la probabilidad de que se le apruebe el crédito es:

$$f(3) = \frac{1}{1 + e^{-(-3+2\times3)}} = \frac{1}{1 + e^{-3}} \approx 0.95$$

Esto indica que el cliente tiene una alta probabilidad de ser aprobado para el crédito.

- **Modelos no supervisados**: En estos modelos, el conjunto de datos $\mathcal{D}$ solo incluye las características $\mathbf{x}_i$, sin etiquetas $y_i$. El objetivo es descubrir patrones o estructuras subyacentes en los datos.

  Por ejemplo, en la segmentación de clientes de un banco, cada cliente puede describirse con características como el monto promedio de sus transacciones, la frecuencia de uso de la tarjeta de crédito y su historial de depósitos. Sin embargo, no se cuenta con etiquetas que indiquen a qué tipo de cliente pertenece cada uno. Un modelo no supervisado, como el algoritmo de k-means, puede agrupar a los clientes en diferentes segmentos según sus comportamientos financieros, permitiendo a la institución ofrecer estrategias personalizadas para cada grupo.

  Ejemplos comunes incluyen:

  - **Clustering**: Agrupa los datos en clusters basados en similitudes, como el algoritmo K-means.

    **Ejemplo 1.6.**

    Supongamos que un banco quiere segmentar a sus clientes en tres grupos según sus hábitos de gasto. Se tienen los siguientes datos de clientes representados por dos características: ingreso mensual ($1000 a $5000) y gasto promedio en tarjeta de crédito ($200 a $2000).

    Aplicando el algoritmo K-means con $k = 3$, se obtienen tres grupos:

    - Cluster 1: Clientes con ingresos bajos y gastos moderados.
    - Cluster 2: Clientes con ingresos altos y gastos elevados.

- Cluster 3: Clientes con ingresos medios y gastos balanceados.

Con esta información, el banco puede ofrecer estrategias personalizadas para cada grupo, como promociones o límites de crédito adecuados.

- **Reducción de dimensionalidad**: Reduce el número de características mientras preserva la estructura de los datos, como el Análisis de Componentes Principales (PCA).

  **Ejemplo 1.7.**

  Supongamos que tenemos un conjunto de datos con 10 variables financieras para cada cliente, como ingresos, deudas, historial de pagos, uso de tarjetas, etc. Aplicando PCA, descubrimos que solo dos combinaciones lineales de estas variables explican el 90 % de la variabilidad de los datos.

  En lugar de trabajar con las 10 variables originales, podemos representar a cada cliente con solo dos valores (las dos primeras *componentes principales*), reduciendo la complejidad del modelo y facilitando la visualización de los datos en un gráfico bidimensional.

## 1.3. Ejemplos de modelos predictivos

Entre los ejemplos de modelos clásicos utilizados en la predicción de riesgo crediticio se encuentran:

### 1.3.1. Regresión logística

La **regresión logística** es un modelo de clasificación binaria que estima la probabilidad de que una instancia pertenezca a una clase específica. Dado un vector de características $\mathbf{x}_i$, el modelo predice la probabilidad $p_i$ de que la etiqueta $y_i$ sea 1 (default) mediante la función logística:

$$p_i = \sigma(\mathbf{w}^T \mathbf{x}_i + b) = \frac{1}{1 + e^{-(\mathbf{w}^T \mathbf{x}_i + b)}},$$

donde $\mathbf{w}$ es un vector de pesos, $b$ es el término de sesgo, y $\sigma$ es la función sigmoide. La función de pérdida utilizada en la regresión logística es la **pérdida logarítmica** (log loss), definida como:

$$L(y_i, p_i) = -\left[y_i \log(p_i) + (1 - y_i) \log(1 - p_i)\right].$$

Para esta fórmula, los logaritmos hacen referencia al logaritmo neperiano en base $e$. El objetivo es encontrar los parámetros $\mathbf{w}$ y $b$ que minimicen la pérdida logarítmica sobre el conjunto de entrenamiento.

**Ejemplo 1.8.**

Supongamos que un banco quiere predecir si un cliente entrará en *default*, lo que significa que no pudo pagar su deuda o incumplió con el pago de un crédito. Se define $y = 1$ si esto ocurrió, y $y = 0$ si no ocurrió, en función de su ingreso mensual ($x$, en miles de dólares). Si el modelo aprendido es:

$$p_i = \frac{1}{1 + e^{-(-2+3x_i)}}$$

y un cliente tiene un ingreso de $x_i = 2{,}5$ mil dólares, la probabilidad de *default* es:

$$p_i = \frac{1}{1 + e^{-(-2+3(2.5))}} = \frac{1}{1 + e^{-5.5}} \approx 0.995$$

Esto indica que el modelo predice una probabilidad del 99.5 % de que el cliente entre en *default*.

Ahora, si el valor real de $y_i$ es 1 (es decir, el cliente efectivamente entró en default), la pérdida logarítmica correspondiente es:

$$L(1, 0.995) = -[1 \cdot \log(0.995) + (1 - 1) \cdot \log(1 - 0.995)].$$

Dado que el segundo término se anula, queda:

$$L(1, 0.995) = -\log(0.995) \approx 0.005.$$

Como la probabilidad predicha es cercana a 1 y el valor real también es 1, la pérdida es pequeña, lo que indica que el modelo hizo una buena predicción.

### 1.3.2. Árboles de decisión

Un **árbol de decisión** es un modelo no paramétrico que divide recursivamente el espacio de características en regiones más simples, basándose en reglas de decisión. Un modelo es no paramétrico porque no asume una forma matemática fija para los datos (como una ecuación lineal). En su lugar, aprende patrones directamente de los datos sin hacer suposiciones previas sobre cómo se relacionan las variables. El espacio de características es el conjunto de todas las posibles combinaciones de valores que pueden tomar las variables

de entrada. Por ejemplo, si se evalúan clientes para un préstamo en función de su ingreso y su historial de pagos, el espacio de características incluye todas las combinaciones posibles de ingresos e historiales.

El **árbol de decisión** está formado por nodos, donde cada nodo representa una prueba sobre una característica de los datos. Es decir, en cada nodo se formula una pregunta, como: **¿El ingreso es mayor a 3000 dólares?**. También contiene ramas, que representan los posibles resultados de la prueba. Siguiendo el ejemplo anterior, si el nodo pregunta **¿El ingreso es mayor a 3000 dólares?**, habrá dos ramas: una para *Sí* y otra para *No*. Finalmente, el árbol tiene hojas, que representan una etiqueta de clase o una decisión final después de seguir todas las reglas de decisión. En el caso del modelo de préstamos, una hoja puede indicar si el préstamo es *aprobado* o *rechazado*.

Formalmente, un árbol de decisión puede expresarse como una función $f(\mathbf{x}_i)$ que asigna una etiqueta $y_i$ basada en una serie de condiciones:

$$f(\mathbf{x}_i) = \sum_{m=1}^{M} c_m \cdot \mathbb{I}(\mathbf{x}_i \in R_m),$$

donde $R_m$ son regiones disjuntas del espacio de características, $c_m$ es la etiqueta asociada a la región $R_m$.

$\mathbb{I}$ es la **función indicadora**, la cual toma el valor de 1 si se cumple una determinada condición y 0 en caso contrario. Matemáticamente, se define como:

$$\mathbb{I}(x \in R) = \begin{cases} 1, & \text{si } x \in R, \\ 0, & \text{si } x \notin R. \end{cases}$$

En el contexto de los árboles de decisión, esta función se usa para indicar si una instancia $x_i$ pertenece a una región específica $R_m$, lo que determina la asignación de una etiqueta correspondiente.

**Ejemplo 1.9.**

Supongamos que un banco quiere decidir si aprueba ($y = 1$) o rechaza ($y = 0$) un préstamo en función del ingreso mensual ($x_1$, en miles de dólares) y el historial de pagos ($x_2$, donde 1 significa buen historial y 0 significa mal historial). Definimos tres regiones $R_1, R_2$ y $R_3$, con etiquetas $c_1 = 0$ (rechazo), $c_2 = 1$ (aprobado) y $c_3 = 0$ (rechazo), respectivamente.

Las reglas del árbol dividen el espacio de características en las siguientes regiones:

$$R_1 = \{(x_1, x_2) \mid x_1 < 3\}, \quad R_2 = \{(x_1, x_2) \mid x_1 \geq 3, x_2 = 1\}, \quad R_3 = \{(x_1, x_2) \mid x_1 \geq 3, x_2 = 0\}.$$

El árbol de decisión se expresa como:

$$f(\mathbf{x}_i) = c_1 \cdot \mathbb{I}(\mathbf{x}_i \in R_1) + c_2 \cdot \mathbb{I}(\mathbf{x}_i \in R_2) + c_3 \cdot \mathbb{I}(\mathbf{x}_i \in R_3).$$

Ahora, evaluemos un cliente con un ingreso mensual de $x_1 = 4$ mil dólares y un mal historial de pagos ($x_2 = 0$):

$$f(4,0) = 0 \cdot \mathbb{I}((4,0) \in R_1) + 1 \cdot \mathbb{I}((4,0) \in R_2) + 0 \cdot \mathbb{I}((4,0) \in R_3).$$

Dado que $(4,0)$ pertenece a $R_3$, donde la etiqueta es $c_3 = 0$, se obtiene:

$$f(4,0) = 0.$$

Por lo tanto, el préstamo es **rechazado**.

### 1.3.3. Random forest

El **random forest** es un método de ensamblaje que combina múltiples árboles de decisión para mejorar la precisión y reducir el sobreajuste.

Un método de ensamblaje es una técnica que combina las predicciones de varios modelos base (en este caso, árboles de decisión) para obtener un modelo más robusto y preciso. Al fusionar múltiples modelos, se reduce la variabilidad y se mejora la generalización en nuevos datos.

El sobreajuste ocurre cuando un modelo aprende demasiado bien los detalles y ruido del conjunto de entrenamiento, lo que le impide generalizar correctamente a nuevos datos. En el caso de los árboles de decisión, esto sucede cuando el árbol es demasiado profundo y se ajusta excesivamente a las observaciones de entrenamiento. Random forest ayuda a mitigar este problema al promediar múltiples árboles, reduciendo la dependencia de un solo modelo y mejorando la estabilidad de las predicciones.

Cada árbol en el bosque se entrena con una muestra aleatoria del conjunto de datos (muestreo con reemplazo) y un subconjunto aleatorio de características. La predicción final es el promedio (en regresión) o la moda (en clasificación) de las predicciones de todos los árboles. Formalmente, la predicción de un random forest puede expresarse como:

$$f(\mathbf{x}_i) = \frac{1}{T} \sum_{t=1}^{T} f_t(\mathbf{x}_i),$$

donde $T$ es el número de árboles y $f_t$ es la predicción del árbol $t$-ésimo. random forest es robusto frente al sobreajuste y puede manejar grandes conjuntos de datos, pero su complejidad computacional puede ser un desafío.

**Ejemplo 1.10.**

Supongamos que un banco utiliza un modelo de random forest para predecir si un cliente será aprobado para un préstamo ($y = 1$) o no ($y = 0$). Se han entrenado $T = 5$ árboles de decisión, los cuales producen las siguientes predicciones para un cliente con ingreso mensual de \$4 000 y un buen historial de pagos:

$$f_1(\mathbf{x}_i) = 1, \quad f_2(\mathbf{x}_i) = 0, \quad f_3(\mathbf{x}_i) = 1, \quad f_4(\mathbf{x}_i) = 1, \quad f_5(\mathbf{x}_i) = 0.$$

Para clasificación, la predicción final se obtiene mediante la **moda** de las predicciones individuales:

$$f(\mathbf{x}_i) = \text{moda}([1, 0, 1, 1, 0]) = 1.$$

Por lo tanto, el modelo predice que el cliente será aprobado para el préstamo.

Si en lugar de clasificación fuera un problema de regresión, por ejemplo, estimar el monto del préstamo aprobado en miles de dólares, y los cinco árboles produjeran las siguientes predicciones:

$$f_1(\mathbf{x}_i) = 20, \quad f_2(\mathbf{x}_i) = 25, \quad f_3(\mathbf{x}_i) = 22, \quad f_4(\mathbf{x}_i) = 23, \quad f_5(\mathbf{x}_i) = 21.$$

En este caso, la predicción final sería el **promedio** de los valores predichos:

$$f(\mathbf{x}_i) = \frac{1}{5}(20 + 25 + 22 + 23 + 21) = \frac{111}{5} = 22.2$$

Por lo tanto, el modelo estima que el cliente recibiría un préstamo de aproximadamente \$22 200.

## 1.4. Limitaciones de los modelos clásicos

A pesar de su popularidad y amplio uso en la predicción de riesgo crediticio, los modelos clásicos presentan varias limitaciones que pueden afectar su rendimiento en ciertos escenarios. A continuación, se describen las principales limitaciones de los modelos clásicos, como la regresión logística, los árboles de decisión, random forest.

### Limitaciones de la regresión logística

La regresión logística, aunque es un modelo simple y eficiente, tiene varias limitaciones:

- **Linealidad**: La regresión logística asume una relación lineal entre las características y el logaritmo de la probabilidad de default. Esto limita su capacidad para capturar relaciones no lineales entre las variables, lo que puede resultar en un modelo subóptimo cuando las relaciones en los datos son complejas (Hosmer, Lemeshow & Sturdivant, 2013).
- **Manejo de variables categóricas**: Aunque la regresión logística puede manejar variables categóricas mediante codificación one-hot, este enfoque puede aumentar significativamente la dimensionalidad del problema, especialmente cuando hay muchas categorías. Esto puede llevar a un sobreajuste y a un mayor costo computacional (Menard, 2002).
- **Sensibilidad a datos desbalanceados**: En problemas de riesgo crediticio, donde la mayoría de los clientes no incurren en default, la regresión logística puede tener dificultades para predecir correctamente la clase minoritaria (default), ya que tiende a favorecer la clase mayoritaria (King & Zeng, 2001).

### Limitaciones de los árboles de decisión

Los árboles de decisión, aunque son intuitivos y fáciles de interpretar, presentan las siguientes limitaciones:

- **Sobreajuste**: Los árboles de decisión tienden a sobreajustarse a los datos de entrenamiento, especialmente cuando son muy profundos. Esto ocurre porque el modelo puede memorizar los datos en lugar de generalizar patrones subyacentes (Breiman, Friedman, Olshen & Stone, 1984).
- **Inestabilidad**: Pequeños cambios en los datos de entrenamiento pueden resultar en árboles de decisión completamente diferentes. Esto hace que el modelo sea menos robusto frente a variaciones en los datos (Breiman, Friedman, Olshen & Stone, 1984).
- **Manejo de variables categóricas**: Aunque los árboles de decisión pueden manejar variables categóricas de manera nativa, su rendimiento puede verse afectado cuando hay muchas categorías o cuando las categorías tienen una distribución desigual (Breiman, Friedman, Olshen & Stone, 1984).

### Limitaciones de random forest

Aunque random forest es más robusto que un solo árbol de decisión, todavía tiene algunas limitaciones:

- **Complejidad computacional**: random forest requiere entrenar múltiples árboles de decisión, lo que puede ser computacionalmente costoso, especialmente en conjuntos de datos grandes o con muchas características (Breiman, 2001).
- **Interpretabilidad**: Aunque random forest mejora la precisión, pierde la interpretabilidad de un solo árbol de decisión, ya que combina las predicciones de muchos árboles (Breiman, 2001).
- **Manejo de datos desbalanceados**: Al igual que los árboles de decisión, random forest puede tener dificultades para predecir correctamente la clase minoritaria en problemas desbalanceados, a menos que se utilicen técnicas de balanceo (Breiman, 2001).

# Capítulo 2

# Algoritmo CatBoost

El *Gradient Boosting* es una de las técnicas más utilizadas en aprendizaje automático para problemas de clasificación y regresión. Este método construye modelos de manera aditiva, combinando múltiples árboles de decisión para minimizar una función de pérdida de forma iterativa. Sin embargo, a pesar de su efectividad, presenta ciertos desafíos, como el manejo adecuado de variables categóricas y la eficiencia computacional.

**CatBoost** (Categorical Boosting) es un algoritmo desarrollado para abordar estas limitaciones, optimizando el tratamiento de variables categóricas y mejorando la estabilidad del modelo. A diferencia de otros métodos de boosting como XGBoost y LightGBM, CatBoost introduce una novedosa estrategia de *permutación ordenada* para la codificación de variables categóricas, evitando la fuga de información (*data leakage*) y proporcionando estimaciones más precisas.

En este capítulo, exploraremos los fundamentos matemáticos del *Gradient Boosting*, los principios clave detrás de CatBoost, su implementación y sus ventajas con respecto a otros algoritmos de boosting. A través de este análisis, se demostrará cómo CatBoost mejora la eficiencia en el entrenamiento y optimiza el rendimiento en tareas de predicción.

## 2.1. Introducción al gradient boosting

El **Gradient Boosting** es un método de aprendizaje supervisado que construye modelos de manera secuencial, ajustando cada nueva iteración para minimizar una función de pérdida. Este enfoque se basa en la idea de combinar múltiples modelos débiles (generalmente árboles de decisión) para crear un modelo fuerte que mejore la precisión en la predicción. A continuación, se presenta una descripción formal del algoritmo y su fundamento matemático.

### Descripción formal del algoritmo y fundamento matemático

El algoritmo de *Gradient Boosting* optimiza una función de pérdida mediante un enfoque iterativo, en el cual cada nuevo modelo aprende a corregir los errores cometidos por la suma de los modelos anteriores. Formalmente, dado un conjunto de datos de entrenamiento $\{(x_i, y_i)\}_{i=1}^{n}$, donde $x_i$ representa las características y $y_i$ la variable objetivo, el modelo se construye de la siguiente manera:

1. Se inicializa el modelo con un predictor constante:

$$F_0(x) = \arg\min_{c} \sum_{i=1}^{n} L(y_i, c)$$

donde $L(y, F(x))$ es la función de pérdida. Aquí, $c$ representa un valor escalar óptimo que minimiza la función de pérdida en la primera etapa del modelo. La elección de $c$ depende del tipo de problema: en regresión con pérdida cuadrática suele ser la media de los valores objetivo, mientras que en clasificación con pérdida logarítmica suele ser la moda de las clases. Este predictor inicial proporciona un punto de partida antes de que el modelo comience a iterar y mejorar sus predicciones.

2. Para cada iteración $m = 1, \ldots, M$:

- Se calcula el residuo o pseudo-residuo, que representa la dirección del gradiente negativo de la función de pérdida:

$$r_i^{(m)} = -\left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}\right]_{F(x)=F_{m-1}(x)}$$

  donde $F_{m-1}(x)$ es el modelo construido hasta la iteración $m-1$. Se utiliza esta notación porque el objetivo del nuevo modelo es corregir los errores del modelo actual, por lo que el cálculo del gradiente se evalúa en la predicción del modelo en su estado anterior.

  En este paso, el residuo se toma como el gradiente negativo de la función de pérdida porque en optimización, el gradiente indica la dirección de mayor incremento de la función. Para minimizar la pérdida, el modelo debe moverse en la dirección opuesta, es decir, en el gradiente negativo. Esto permite que cada iteración corrija los errores acumulados y mejore la precisión del modelo.

- Se ajusta un nuevo modelo débil $h_m(x)$, generalmente un árbol de decisión, que predice los residuos $r_i^{(m)}$. Este modelo débil tiene la tarea de aproximar los residuos actuales y suele ser un árbol de decisión poco profundo para evitar el sobreajuste.

Su función es detectar patrones en los errores del modelo anterior y corregirlos en la siguiente actualización.

- Se encuentra el coeficiente de actualización $\gamma_m$ resolviendo:

$$\gamma_m = \arg\min_{\gamma} \sum_{i=1}^{n} L(y_i, F_{m-1}(x_i) + \gamma h_m(x_i))$$

donde $\gamma_m$ es un factor de ajuste que controla cuánto debe contribuir el nuevo modelo $h_m(x)$ a la actualización del modelo global. Este coeficiente se encuentra minimizando la función de pérdida, ya que el objetivo es garantizar que la corrección aplicada por $h_m(x)$ reduzca los errores de la mejor manera posible. En otras palabras, $\gamma_m$ es el peso óptimo que equilibra la influencia del nuevo modelo débil en la predicción final.

- Se actualiza el modelo:

$$F_m(x) = F_{m-1}(x) + \gamma_m h_m(x)$$

lo que significa que el modelo global se ajusta agregando el modelo débil $h_m(x)$ multiplicado por su peso óptimo $\gamma_m$. Este proceso iterativo continúa hasta alcanzar el número máximo de iteraciones $M$ o hasta que la mejora en la función de pérdida sea insignificante.

3. El modelo final se obtiene tras $M$ iteraciones:

$$F_M(x) = F_0(x) + \sum_{m=1}^{M} \gamma_m h_m(x).$$

Este proceso permite que el modelo aprenda de manera secuencial, ajustando cada nueva iteración para minimizar la función de pérdida en la dirección del gradiente negativo, lo que da origen al nombre de *Gradient Boosting*.

**Ejemplo 2.1.**

Supongamos que tenemos un conjunto de datos con tres observaciones y queremos entrenar un modelo de Gradient Boosting para efectos prácticos, utilizando tres iteraciones. Emplearemos la función de pérdida cuadrática media (MSE):

$$L(y, F(x)) = \frac{1}{2}(y - F(x))^2$$

Inicialización:

El modelo inicial $F_0(x)$ se obtiene minimizando la función de pérdida. Para MSE, la mejor constante $c$ es la media de los valores objetivo:

$$F_0(x) = \frac{1}{n}\sum_{i=1}^{n} y_i$$

Supongamos que nuestros datos son:

$$\{(x_1, y_1) = (1, 3), (x_2, y_2) = (2, 5), (x_3, y_3) = (3, 7)\}$$

Entonces, la inicialización del modelo es:

$$F_0(x) = \frac{3 + 5 + 7}{3} = 5$$

**Iteración 1:**

1. Cálculo de los residuos: Usamos la derivada de la función de pérdida respecto a $F(x_i)$. Para una función de pérdida cuadrática:

$$L(y, F(x)) = \frac{1}{2}(y - F(x))^2$$

Calculamos la derivada parcial con respecto a $F(x)$:

$$\frac{\partial L(y, F(x))}{\partial F(x)} = \frac{\partial}{\partial F(x)}\left[\frac{1}{2}(y - F(x))^2\right]$$

Aplicamos la regla de la cadena:

$$\frac{\partial L(y, F(x))}{\partial F(x)} = \frac{1}{2} \cdot 2(y - F(x)) \cdot (-1)$$

$$= -(y - F(x))$$

Así, los residuos en la primera iteración se calculan como:

$$r_i^{(1)} = -\left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}\right] = y_i - F_0(x_i)$$

Calculamos los residuos para cada punto:

$$r_1^{(1)} = 3 - 5 = -2, \quad r_2^{(1)} = 5 - 5 = 0, \quad r_3^{(1)} = 7 - 5 = 2$$

2. Ajuste del modelo débil $h_1(x)$: En lugar de predecir los residuos con su media, utilizamos la mediana por subgrupos para hacerlo más robusto. Dividimos los datos en subgrupos según $x$ y calculamos la mediana de los residuos en cada uno:

- Para el grupo 1, con valores de residuos {-2, 0}, la mediana es:

$$\text{mediana}(\{-2, 0\}) = -1$$

- Para el grupo 2, con el único valor de residuo {2}, la mediana es:

$$\text{mediana}(\{2\}) = 2$$

Así, el modelo débil se define como:

$$h_1(x) = \begin{cases} -1, & \text{si } x \in \{1, 2\} \\ 2, & \text{si } x = 3 \end{cases}$$

3. Cálculo del coeficiente de actualización $\gamma_1$: En regresión cuadrática, se puede demostrar que:

$$\gamma_m = \frac{\sum_{i=1}^{n} r_i^{(m)} h_m(x_i)}{\sum_{i=1}^{n} h_m(x_i)^2}$$

Esto es para encontrar el valor óptimo de $\gamma_m$, debemos minimizar la función de pérdida:

$$\gamma_m = \arg\min_{\gamma} \sum_{i=1}^{n} L(y_i, F_{m-1}(x_i) + \gamma h_m(x_i))$$

Dado que estamos usando la pérdida cuadrática media (MSE), la función de pérdida es:

$$L(y_i, F(x_i)) = \frac{1}{2}(y_i - F(x_i))^2$$

Sustituyendo $F_m(x) = F_{m-1}(x) + \gamma_m h_m(x)$, la función de pérdida se convierte en:

$$L(y_i, F_m(x_i)) = \frac{1}{2}(y_i - (F_{m-1}(x_i) + \gamma_m h_m(x_i)))^2$$

Diferenciamos con respecto a $\gamma_m$:

$$\frac{\partial}{\partial \gamma_m} \sum_{i=1}^{n} \frac{1}{2}(y_i - (F_{m-1}(x_i) + \gamma_m h_m(x_i)))^2 = 0$$

Aplicamos la regla de la cadena:

$$\sum_{i=1}^{n} (y_i - (F_{m-1}(x_i) + \gamma_m h_m(x_i)))(-h_m(x_i)) = 0$$

Reescribimos en términos del residuo $r_i^{(m)} = y_i - F_{m-1}(x_i)$:

$$\sum_{i=1}^{n}(r_i^{(m)} - \gamma_m h_m(x_i))(-h_m(x_i)) = 0$$

Distribuimos:

$$\sum_{i=1}^{n} -r_i^{(m)} h_m(x_i) + \gamma_m \sum_{i=1}^{n} h_m(x_i)^2 = 0$$

Despejamos $\gamma_m$:

$$\gamma_m = \frac{\sum_{i=1}^{n} r_i^{(m)} h_m(x_i)}{\sum_{i=1}^{n} h_m(x_i)^2}$$

Ahora procedemos a realizar el cálculo:

$$\gamma_1 = \frac{r_1^{(1)} h_1(x_1) + r_2^{(1)} h_1(x_2) + r_3^{(1)} h_1(x_3)}{h_1(x_1)^2 + h_1(x_2)^2 + h_1(x_3)^2} = \frac{(-2)(-1) + (0)(-1) + (2)(2)}{(-1)^2 + (-1)^2 + (2)^2}$$

Por lo tanto, tenemos $\gamma_1 = 1$.

4. Actualización del modelo:

$$F_1(x) = F_0(x) + \gamma_1 h_1(x)$$

$$F_1(x) = \begin{cases} 4, & \text{si } x \in \{1, 2\} \\ 7, & \text{si } x = 3 \end{cases}$$

**Iteración 2:**

1. Cálculo de los residuos:

Usamos la fórmula general:

$$r_i^{(m)} = -\left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}\right]_{F(x)=F_{m-1}(x)}$$

Para una función de pérdida cuadrática, como vimos en la iteración 1, tenemos:

$$r_i^{(2)} = y_i - F_1(x_i)$$

Evaluamos para cada punto:

$$r_1^{(2)} = 3 - 4 = -1, \quad r_2^{(2)} = 5 - 4 = 1, \quad r_3^{(2)} = 7 - 7 = 0$$

2. Ajuste del modelo débil $h_2(x)$

Suponemos que el nuevo modelo predice los residuos mediante la mediana de los valores en cada grupo:

- Para $x = 2$, el residuo es $\{1\}$, por lo que la mediana es 1.
- Para $x \in \{1, 3\}$, los residuos son $\{-1, 0\}$, cuya mediana es -0.5.

Entonces, el modelo débil queda definido como:

$$h_2(x) = \begin{cases} \text{-0.5}\,, & \text{si } x \in \{1, 3\} \\ 1, & \text{si } x = 2 \end{cases}$$

3. Cálculo de $\gamma_2$:

$$\gamma_2 = \frac{r_1^{(2)} h_2(x_1) + r_2^{(2)} h_2(x_2) + r_3^{(2)} h_2(x_3)}{h_2(x_1)^2 + h_2(x_2)^2 + h_2(x_3)^2} = \frac{(-1)(\text{-0.5}) + (1)(1) + (0)(\text{-0.5})}{(\text{-0.5})^2 + (1)^2 + (\text{-0.5})^2}$$

Por lo tanto, tenemos $\gamma_2 = 1$.

4. Actualización del modelo:

$$F_2(x) = F_1(x) + \gamma_2 h_2(x)$$

$$F_2(x) = \begin{cases} 3.5\,, & \text{si } x = 1 \\ 5\,, & \text{si } x = 2 \\ 6.5\,, & \text{si } x = 3 \end{cases}$$

**Iteración 3:**

1. Cálculo de los residuos:

Usamos la fórmula general:

$$r_i^{(m)} = -\left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}\right]_{F(x)=F_{m-1}(x)}$$

Para una función de pérdida cuadrática, como vimos en la iteración 1, tenemos:

$$r_i^{(3)} = y_i - F_2(x_i)$$

Evaluamos para cada punto:

$$r_1^{(3)} = 3 - 3.5 = -0.5\ , \quad r_2^{(3)} = 5 - 5 = 0\ , \quad r_3^{(3)} = 7 - 6.5 = 0.5$$

2. Ajuste del modelo débil $h_3(x)$

Suponemos que el nuevo modelo predice los residuos mediante la mediana de los valores en cada grupo:

- Para $x \in \{1, 2\}$, los residuos son {-0.5 , 0}, cuya mediana es -0.25.
- Para $x = 3$, el residuo es $\{0.5\}$, por lo que la mediana es 0.25.

Entonces, el modelo débil queda definido como:

$$h_3(x) = \begin{cases} \text{-0.25 }, & \text{si } x \in \{1, 2\} \\ 0.25 \,, & \text{si } x = 3 \end{cases}$$

3. Cálculo de $\gamma_3$:

$$\gamma_3 = \frac{r_1^{(3)} h_3(x_1) + r_2^{(3)} h_3(x_2) + r_3^{(3)} h_3(x_3)}{h_3(x_1)^2 + h_3(x_2)^2 + h_3(x_3)^2} = \frac{\text{(-0.5)(-0.25)} + (0)\text{(-0.25)} + (0.5)(0.25)}{\text{(-0.25)}^2 + \text{(-0.25)}^2 + (0.25)^2}$$

Por lo tanto, tenemos $\gamma_3 = 1.3$.

4. Actualización del modelo:

$$F_3(x) = F_2(x) + \gamma_3 h_3(x)$$

Por lo tanto el modelo final después de $M$ iteraciones, para nuestro caso particular con $M = 3$:

$$F_3(x) = \begin{cases} 3.175 \,, & \text{si } x = 1 \\ 4.675 \,, & \text{si } x = 2 \\ 6.825 \,, & \text{si } x = 3 \end{cases}$$

Este ejemplo ilustra el proceso iterativo del Gradient Boosting. En la práctica, el modelo débil $h_m(x)$ suele ser un árbol más sofisticado, lo que permite mejorar la predicción en cada iteración. Como resultado, el modelo final es una combinación ponderada de estos modelos débiles, donde la cantidad de iteraciones y la configuración de los hiperparámetros determinan su desempeño final.

Este marco del *Gradient Boosting* es fundamental para nosotros, ya que sobre él se basa el algoritmo *CatBoost*.

## 2.2. Fundamento matemático de CatBoost

El algoritmo *CatBoost* surge como una solución a los desafíos presentes en el manejo de variables categóricas dentro de los métodos de *boosting*. Propuesto por Prokhorenkova

(2019), este método introduce estrategias innovadoras para reducir el sesgo introducido por la asignación de valores a categorías raras y mejorar la eficiencia en el entrenamiento. A diferencia de otros enfoques tradicionales como *XGBoost* o *LightGBM*, *CatBoost* emplea una técnica de permutación ordenada para calcular las estadísticas de las características categóricas, evitando la fuga de información y proporcionando estimaciones más robustas. A continuación, se presenta el fundamento matemático que sustenta este algoritmo.

Dado un conjunto de datos $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$, donde $\mathbf{x}_i$ es un vector de características y $y_i$ es la etiqueta correspondiente, el objetivo del Gradient Boosting es aprender una función $F(\mathbf{x})$ que minimice una función de pérdida $L(y, F(\mathbf{x}))$. El modelo se construye de manera aditiva, donde en cada iteración $m$ se añade un nuevo modelo débil $h_m(\mathbf{x})$ para corregir los errores residuales de la iteración anterior. Formalmente, el modelo en la iteración $m$ se expresa como:

$$F_m(\mathbf{x}) = F_{m-1}(\mathbf{x}) + \nu \cdot h_m(\mathbf{x}),$$

La cual tiene una forma similar a la vista en la sección anterior, donde:

- $F_{m-1}(\mathbf{x})$ es el modelo en la iteración anterior.
- $h_m(\mathbf{x})$ es el modelo débil añadido en la iteración $m$.
- $\nu$ es la tasa de aprendizaje, un hiperparámetro que controla la contribución de cada modelo débil.

En cada iteración, el modelo débil $h_m(\mathbf{x})$ se ajusta para minimizar la función de pérdida $L(y, F_{m-1}(\mathbf{x}) + h_m(\mathbf{x}))$. Esto se realiza utilizando el gradiente de la función de pérdida con respecto a $F_{m-1}(\mathbf{x})$. En particular, el gradiente $g_m(\mathbf{x})$ se calcula como:

$$g_m(\mathbf{x}_i) = \left[\frac{\partial L(y_i, F(\mathbf{x}_i))}{\partial F(\mathbf{x}_i)}\right]_{F(\mathbf{x}_i)=F_{m-1}(\mathbf{x}_i)}.$$

El modelo débil $h_m(\mathbf{x})$ se entrena para aproximar el gradiente negativo $-g_m(\mathbf{x})$, lo que permite corregir los errores residuales de la iteración anterior.

### 2.2.1. Función de pérdida logarítmica

En problemas de clasificación, una función de pérdida comúnmente utilizada es la **pérdida logarítmica** (log loss), definida como:

$$L(y, F(\mathbf{x})) = -\left[y \ln(p) + (1-y) \ln(1-p)\right],$$

donde $p = \sigma(F(\mathbf{x}))$ es la probabilidad predicha de que la etiqueta $y$ sea 1, $\sigma$ es la función sigmoide y el logaritmo aplicado es el logaritmo natural (base $e$). En problemas de regresión, se utiliza comúnmente la **pérdida cuadrática**:

$$L(y, F(\mathbf{x})) = \frac{1}{2}(y - F(\mathbf{x}))^2.$$

El objetivo del Gradient Boosting es minimizar la función de pérdida $L(y, F(\mathbf{x}))$ sobre el conjunto de datos $\mathcal{D}$. Esto se logra mediante la optimización iterativa de los modelos débiles $h_m(\mathbf{x})$, que se ajustan para reducir el gradiente de la función de pérdida en cada iteración.

**Ejemplo 2.2.**

Supongamos que tenemos un conjunto de datos con tres observaciones representado en la tabla 2.1, donde cada observación tiene una etiqueta binaria $y_i$ y una predicción del modelo $F(\mathbf{x}_i)$.

Tabla 2.1: Conjunto de datos para el ejemplo 2.2.

| $y_i$ | $F(\mathbf{x}_i)$ |
|---|---|
| 1 | 1.2 |
| 0 | -0.8 |
| 1 | 0.5 |

Calculamos la probabilidad predicha usando la función sigmoide:

$$p_i = \sigma(F(\mathbf{x}_i)) = \frac{1}{1 + e^{-F(\mathbf{x}_i)}}$$

Aplicamos esta transformación a cada observación:

$$p_1 = \frac{1}{1 + e^{-1.2}} \approx 0.7685$$

$$p_2 = \frac{1}{1 + e^{0.8}} \approx 0.3100$$

$$p_3 = \frac{1}{1 + e^{-0.5}} \approx 0.6225$$

La función de pérdida logarítmica se define como:

$$L(y, F(\mathbf{x})) = -\sum_{i=1}^{n} [y_i \ln(p_i) + (1 - y_i) \ln(1 - p_i)]$$

Sustituyamos los valores para cada observación:

$$L_1 = -[1 \cdot \ln(0.7685) + (1 - 1) \cdot \ln(1 - 0.7685)] \approx -\ln(0.7685) \approx 0.2633$$

$$L_2 = -[0 \cdot \ln(0.3100) + (1 - 0) \cdot \ln(1 - 0.3100)] \approx -\ln(0.6900) \approx 0.3711$$

$$L_3 = -[1 \cdot \ln(0.6225) + (1 - 1) \cdot \ln(1 - 0.6225)] \approx -\ln(0.6225) \approx 0.4740$$

Por lo tanto la pérdida total:

$$L_{\text{total}} = \frac{1}{n} \sum_{i=1}^{n} L_i = \frac{0.2633 + 0.3711 + 0.4740}{3} \approx 0.3695$$

El valor obtenido de log loss es 0.3695 nos indica qué tan bien el modelo está prediciendo las probabilidades correctas para cada clase. Un valor más cercano a 0 sugiere que el modelo está haciendo predicciones más seguras y precisas, mientras que un valor mayor indica que las probabilidades asignadas por el modelo no se alinean bien con las etiquetas reales.

En este caso, el log loss de 0.3695 sugiere que el modelo está logrando una calidad de predicción razonable, pero aún podría mejorarse.

### 2.2.2. Manejo de variables categóricas

Las **variables categóricas** son un tipo de variable que representa datos cualitativos, divididos en categorías discretas. A diferencia de las variables numéricas, estas no tienen un orden intrínseco ni una escala de medida cuantitativa.

En el contexto de modelos estadísticos y machine learning, las variables categóricas suelen clasificarse en:

- **Nominales**: Cuando no existe un orden entre las categorías (ej.: color, género).
- **Ordinales**: Cuando las categorías tienen un orden natural (ej.: nivel educativo: primaria, secundaria, universidad).

Para incluirlas en modelos matemáticos, generalmente se aplican técnicas como:

- **One-Hot Encoding**: Creación de variables binarias (0/1) para cada categoría.
- **Label Encoding**: Asignación de un valor numérico a cada categoría (útil para variables ordinales).

El manejo adecuado de estas variables es crucial, ya que una codificación incorrecta puede afectar significativamente el desempeño del modelo.

### Manejo de variables categóricas en CatBoost

Una de las principales ventajas de CatBoost es su capacidad para manejar eficientemente variables categóricas mediante la técnica de **ordered target encoding**. Esta técnica es especialmente útil para evitar el problema de **data leakage**, que ocurre cuando la información del conjunto de prueba se filtra en el conjunto de entrenamiento, lo que puede llevar a una sobreestimación del rendimiento del modelo, debido a que se ha entrenado con datos ya conocidos.

En el **ordered target encoding**, las variables categóricas se codifican utilizando estadísticas basadas en la variable objetivo, como la media de la variable objetivo para cada categoría. Sin embargo, a diferencia de otras técnicas de codificación, CatBoost utiliza una permutación aleatoria de los datos durante el entrenamiento para evitar el data leakage. Formalmente, para una categoría $c$ en la variable categórica $x$, el valor codificado $\text{Enc}(c)$ se calcula como:

$$\text{Enc}(c) = \frac{\sum_{i=1}^{n} \mathbb{I}(x_i = c) \cdot y_i + \alpha \cdot p}{\sum_{i=1}^{n} \mathbb{I}(x_i = c) + \alpha},$$

donde:

- $\mathbb{I}(x_i = c)$ es la función indicadora que vale 1 si $x_i = c$ y 0 en caso contrario.
- $y_i$ es la variable objetivo.
- $\alpha$ es un parámetro de regularización que controla el efecto de la media global $p$ (la proporción de la clase positiva en el conjunto de datos).

Al utilizar una permutación aleatoria de los datos, CatBoost asegura que la codificación de cada instancia se base únicamente en las instancias anteriores en la permutación, evitando así el data leakage.

**Ejemplo 2.3.**

A continuación, mostramos un ejemplo de uso de **ordered target encoding** para la predicción de compras por distrito en Lima. Consideremos la tabla 2.2 un dataset de 6 clientes con la variable categórica *Distrito* y la variable objetivo $y$ (1=Compra, 0=No compra):

Tabla 2.2: Conjunto de datos para el ejemplo 2.3.

| Cliente | Distrito | $y$ |
|---|---|---|
| 1 | Miraflores | 1 |
| 2 | Barranco | 0 |
| 3 | San Isidro | 1 |
| 4 | Miraflores | 1 |
| 5 | Barranco | 0 |
| 6 | San Isidro | 0 |

Parámetros:

- $\alpha = 1$
- $p = 0.5$
- Permutación aleatoria de clientes: [3, 5, 1, 2, 6, 4]

Además el valor de $c$ para este caso toma 3 únicos valores : Miraflores, Barranco y San Isidro.

Ahora vamos recorriendo en el orden de la permutación aleatoria:

**Cálculo para Cliente 3 (San Isidro, y=1):** Primera aparición de San Isidro y datos anteriores ninguno.

$$\text{Enc(San Isidro)} = \frac{\alpha \cdot p}{\alpha} = 0.5$$

**Cálculo para Cliente 5 (Barranco, y=0):** Primera aparición de Barranco y datos anteriores cliente 3.

$$\begin{aligned}\text{Enc(Barranco)} &= \frac{\mathbb{I}(x_1 = \text{Barranco}) \cdot y_1 + \alpha \cdot p}{\mathbb{I}(x_1 = \text{Barranco}) + \alpha} \\ &= \frac{0 \cdot 1 + \alpha \cdot p}{0 + \alpha} = 0.5\end{aligned}$$

**Cálculo para Cliente 1 (Miraflores, y=1):** Primera aparición de Miraflores y datos anteriores clientes [3,5].

$$\begin{aligned}\text{Enc(Miraflores)} &= \frac{\mathbb{I}(x_1 = \text{Miraflores}) \cdot y_1 + \mathbb{I}(x_2 = \text{Miraflores}) \cdot y_2 + \alpha \cdot p}{\mathbb{I}(x_1 = \text{Miraflores}) + \mathbb{I}(x_2 = \text{Miraflores}) + \alpha} \\ &= \frac{0 \cdot 1 + 0 \cdot 0 + \alpha \cdot p}{0 + 0 + \alpha} = 0.5\end{aligned}$$

**Cálculo para Cliente 2 (Barranco, y=0):** Barranco ya apareció, se ajusta el encoding y datos anteriores [3,5,1].

$$\begin{aligned}\text{Enc(Barranco)} &= \frac{\mathbb{I}(x_1 = \text{Barranco}) \cdot y_1 + \mathbb{I}(x_2 = \text{Barranco}) \cdot y_2 + \mathbb{I}(x_3 = \text{Barranco}) \cdot y_3 + \alpha \cdot p}{\mathbb{I}(x_1 = \text{Barranco}) + \mathbb{I}(x_2 = \text{Barranco}) + \mathbb{I}(x_3 = \text{Barranco}) + \alpha} \\ &= \frac{0 \cdot 1 + 1 \cdot 0 + 0 \cdot 1 + \alpha \cdot p}{0 + 1 + 0 + \alpha} = 0.25\end{aligned}$$

**Cálculo para Cliente 6 (San Isidro, y=0):** San Isidro ya apareció, se ajusta el encoding y datos anteriores [3,5,1,2].

$$\begin{aligned}\text{Enc(San Isidro)} &= \frac{\sum_{i=1}^{4} \mathbb{I}(x_i = \text{San Isidro}) \cdot y_i + \alpha \cdot p}{\sum_{i=1}^{4} \mathbb{I}(x_i = \text{San Isidro}) + \alpha} \\ &= \frac{(1 \cdot 1 + 0 \cdot 0 + 0 \cdot 1 + 0 \cdot 0) + 1 \cdot 0{,}5}{1 + 0 + 0 + 0 + 1} \\ &= \frac{1 + 0{,}5}{2} \\ &= 0.75\end{aligned}$$

**Cálculo para Cliente 4 (Miraflores, y=1):** Miraflores ya apareció, se ajusta el encoding y datos anteriores [3,5,1,2,6].

$$\begin{aligned}\text{Enc(Miraflores)} &= \frac{\sum_{i=1}^{5} \mathbb{I}(x_i = \text{Miraflores}) \cdot y_i + \alpha \cdot p}{\sum_{i=1}^{5} \mathbb{I}(x_i = \text{Miraflores}) + \alpha} \\ &= \frac{(0 \cdot 1 + 0 \cdot 0 + 1 \cdot 1 + 0 \cdot 0 + 0 \cdot 0) + 1 \cdot 0{,}5}{0 + 0 + 1 + 0 + 0 + 1} \\ &= \frac{1 + 0{,}5}{2} = 0.75\end{aligned}$$

Tabla 2.3: Valor final asignado a cada categoría.

| Distrito | Encoding Final |
|---|---|
| Miraflores | 0.75 |
| Barranco | 0.50 |
| San Isidro | 0.75 |

Interpretación comercial:

- Miraflores y San Isidro muestran alta probabilidad de compra (75 %)
- Barranco se mantiene en la media global (50 %)
- El encoding refleja patrones reales: distritos comerciales vs. residenciales

Al final, el uso sería que, al momento de entrenar el modelo, se ingresen los valores numéricos en lugar de la categoría como lo mostrado en la tabla 2.3. Por ejemplo, en lugar de la categoría *Miraflores*, iría el valor numérico 0.75, y así con las demás categorías.

## 2.3. Ventajas de CatBoost sobre otros algoritmos

El Gradient Boosting ofrece varias ventajas en comparación con otros métodos de aprendizaje supervisado:

- **Flexibilidad**: Puede manejar tanto problemas de clasificación como de regresión, y es compatible con diversas funciones de pérdida (Prokhorenkova, 2019).

  **Ejemplo 2.4.**

  Una plataforma de e-commerce necesita:

  - **Clasificación**: Predecir si un cliente comprará (Sí/No) usando *LogLoss*.

- **Regresión**: Estimar el monto de compra (USD) con *RMSE*.
- **Datos mixtos**: variables numéricas (edad, historial) y categóricas (ciudad, dispositivo).

CatBoost resuelve ambos casos con la misma librería:

- Cambiando el parámetro *loss_function* según el problema.
- Procesando automáticamente categorías sin preprocesamiento manual.

Esto simplifica la implementación y reduce errores en producción.

- **Capacidad para capturar relaciones no lineales**: Al utilizar árboles de decisión como modelos débiles, el Gradient Boosting puede capturar relaciones no lineales y complejas entre las características y la variable objetivo (Dorogush, 2018).

  **Ejemplo 2.5.**
  Por ejemplo, en un estudio de marketing. Un modelo lineal podría predecir que **a mayor inversión en publicidad, mayores ventas** (relación directa). CatBoost, en cambio, identifica que:

  - Las ventas crecen con la publicidad hasta cierto punto, pero decrecen si hay saturación (relación cuadrática).
  - El efecto cambia según el *grupo demográfico*: jóvenes responden a redes sociales, adultos a TV (interacciones no lineales).

  Esto se debe a que sus árboles de decisión dividen los datos en segmentos y combinan sus resultados jerárquicamente.

- **Robustez frente al sobreajuste**: La tasa de aprendizaje $\nu$ y la limitación de la profundidad de los árboles ayudan a reducir el sobreajuste, especialmente en conjuntos de datos pequeños (Prokhorenkova, 2019).

  **Ejemplo 2.6.**
  En un hospital con pocos registros de pacientes (200 casos), se quiere predecir el riesgo de diabetes. Un modelo complejo:

  - Podría memorizar ruido (como valores atípicos en análisis de sangre) en lugar de patrones reales.

- Tendría alto rendimiento en los datos de entrenamiento pero fallaría con nuevos pacientes.

CatBoost controla esto mediante:

- Árboles de profundidad máxima 4 (evita patrones demasiado específicos).
- Ajuste gradual con tasa de aprendizaje baja ($\nu = 0{,}01$), mejorando iterativamente sin sobreajustar.

Así logra generalizar bien incluso con datos limitados.

# Capítulo 3

# Planteamiento del problema

## 3.1. Descripción del problema

El Score de Admisión de Tarjetas de Crédito NoHit (No bancarizado) es un modelo estadístico utilizado por las entidades financieras para evaluar el nivel de riesgo de una persona que solicita una tarjeta de crédito y que no cuenta con historial en el sistema financiero formal. Este modelo clasifica a los solicitantes en dos categorías: bueno o malo, en función de su probabilidad de caer en default, es decir, en incumplimiento de pago de sus obligaciones financieras.

El score de tarjetas de crédito (TC) es una herramienta clave en el proceso de admisión de nuevos clientes, ya que permite a la entidad financiera tomar decisiones sobre la aprobación o el rechazo de una solicitud. En este caso, el modelo está diseñado específicamente para el segmento No Hit, que hace referencia a personas que no tienen historial crediticio registrado en la entidad o en el sistema financiero en general. Dado que estas personas no cuentan con referencias previas sobre su comportamiento de pago, el modelo se apoya en otras variables disponibles para estimar el nivel de riesgo y determinar si el solicitante es apto para recibir una tarjeta de crédito.

## 3.2. Diseño del modelo

La población objetivo del modelo son las personas con Documento Nacional de Identidad (DNI) que solicitan una tarjeta de crédito (TC) en el sistema financiero. Se considera que un solicitante pertenece al segmento No bancarizado (No Hit) si en los últimos 12 meses disponibles en el Registro de Créditos y Cobranzas (RCC) no ha registrado saldo, línea de crédito ni contingente en ninguna entidad financiera. Para la construcción del modelo,

se trabajó con datos de cuatro meses para el entrenamiento y un mes adicional para la validación fuera de tiempo (out-of-time).

El principal objetivo del modelo predictivo No Hit es estimar la probabilidad de default del solicitante, es decir, la probabilidad de que la persona incurra en un atraso en el pago de su tarjeta de crédito por más de 60 días dentro de los 12 meses siguientes a la emisión del crédito. Esta estimación se basa en variables del perfil del solicitante.

Adicionalmente, se han aplicado exclusiones a nivel de la población analizada por consideraciones del negocio. Por ejemplo, se han retirado aquellos registros de personas con antecedentes de transacciones fraudulentas o posibles casos de lavado de activos.

Para mejorar la capacidad predictiva del modelo, se ha utilizado una base de datos amplia que permite capturar diversas características de los clientes. Esta base incluye datos censales (tipo de vivienda, nivel de gastos, etc.), información sociodemográfica, entre otras fuentes relevantes.

En este modelo, la variable objetivo (target) es la clasificación del solicitante en bueno o malo, en función de su comportamiento crediticio posterior a la obtención de la tarjeta. Una persona es considerada mala si presenta un incumplimiento de pago superior a 60 días en una ventana de observación de 12 meses. Además, se han aplicado filtros adicionales sobre la población para mejorar la precisión del modelo. Por ejemplo, se han excluido aquellas personas que ya presentan morosidad al inicio del período de observación, dado que el objetivo es aprender sobre los clientes que se atrasan después del otorgamiento del crédito. También se han aplicado otros criterios de exclusión alineados con la estrategia del negocio.

### 3.2.1. Preprocesamiento de datos

Ahora pasamos al entrenamiento del modelo. Para ello, vamos a preparar las variables que servirán para el aprendizaje del modelo, es decir, las variables independientes, también conocidas como variables de entrada y denotadas con $X$.

Contamos con una base de datos de un total de 54 337 registros para los periodos de entrenamiento, que abarcan de junio de 2021 a septiembre de 2021. Por otro lado, el periodo fuera de tiempo (OOT, por sus siglas en inglés *Out Of Time*), que se utilizará para evaluar el rendimiento del modelo en un periodo no visto durante el entrenamiento, contiene un total de 17 940 registros y corresponde a octubre de 2021.

La descripción de las 24 variables se expresa en la siguiente tabla 3.1. También agregaremos la variable dependiente de salida, denotada como $y$:

Tabla 3.1: Descripción de las variables utilizadas en el modelo.

| Descripción de la variable | Nombre de variable | Uso en el modelo | Tipo de variable |
|---|---|---|---|
| Promedio de saldo tipo pasivo en los últimos 6 meses | prm_sldtippas06m | Entrada | Numérica |
| Promedio de deuda de telecomunicaciones en los últimos 12 meses | prm_ddatelc12m | Entrada | Numérica |
| Nivel educativo | lvl_edu_new | Entrada | Numérica |
| Número promedio de entidades de telecomunicaciones reportantes en los últimos 12 meses | prm_telcorep12m | Entrada | Numérica |
| Número de correos registrados | nro_email | Entrada | Numérica |
| Agrupación por situación laboral y edad de la persona | segmento_pea_new | Entrada | Numérica |
| Cantidad de transacciones realizadas por app o banca por internet en el último mes | ctd_uso_app_bpi_mon | Entrada | Numérica |
| Flag de situación laboral dependiente | flag_dependiente | Entrada | Numérica |
| Transacción promedio en banca por internet en los últimos 12 meses | bpi_trx_prom | Entrada | Numérica |
| Promedio de transacciones monetarias y no monetarias en el canal ATM en los últimos 12 meses | atm_trx_prom | Entrada | Numérica |
| Transacción promedio en agentes de la entidad financiera en los últimos 12 meses | age_trx_prom | Entrada | Numérica |
| Número de hijos en el hogar | nro_hijoshog | Entrada | Numérica |

| Descripción de la variable | Nombre de variable | Uso en el modelo | Tipo de variable |
|---|---|---|---|
| Número de incrementos en saldo tipo pasivo en los últimos 12 meses | nro_incsldtippas12m | Entrada | Numérica |
| Clasificación por código de ubigeo | ubigeo_cat | Entrada | Numérica |
| Proporción de hogares que poseen refrigeradoras o congeladoras entre la cantidad de hogares en la manzana | rat_reften_2017 | Entrada | Numérica |
| Monto de deuda en otras empresas en los últimos 12 meses | mnt_ddaotremp12m | Entrada | Numérica |
| Proporción de viviendas cuyo tipo es casa independiente o departamento en edificio entre la cantidad de viviendas en la manzana | propviv_tipviv_2017 | Entrada | Numérica |
| Transacción promedio en tiendas de la entidad financiera en los últimos 12 meses | tie_trx_prom | Entrada | Numérica |
| Promedio de saldo en cuenta sueldo vista en 12 meses | prm_sldctasld12m | Entrada | Numérica |
| Promedio ponderado de la suma de todos los gastos promedio mensual de la manzana, según sector económico de la A hasta la E, al año 2022 | promgast_xhog22 | Entrada | Numérica |

| Descripción de la variable | Nombre de variable | Uso en el modelo | Tipo de variable |
|---|---|---|---|
| Proporción de viviendas con material de construcción en las paredes de tipo ladrillo o bloque de cemento entre la cantidad de viviendas en la manzana | propviv_matnoble | Entrada | Numérica |
| Agrupación de variables pertenecientes al rubro farmacia | comp1 | Entrada | Numérica |
| Agrupación de variables pertenecientes al rubro Supermercados Peruanos | comp2 | Entrada | Numérica |
| Agrupación de variables pertenecientes al rubro tiendas por departamento | comp3 | Entrada | Numérica |
| Target Bueno Malo | tm60_12 | Salida | Numérica |

Como primer paso, revisamos el porcentaje de datos faltantes en las variables independientes. De las 24 variables, 17 presentan valores faltantes en la siguiente proporción, como se muestra en la siguente figura 3.1:

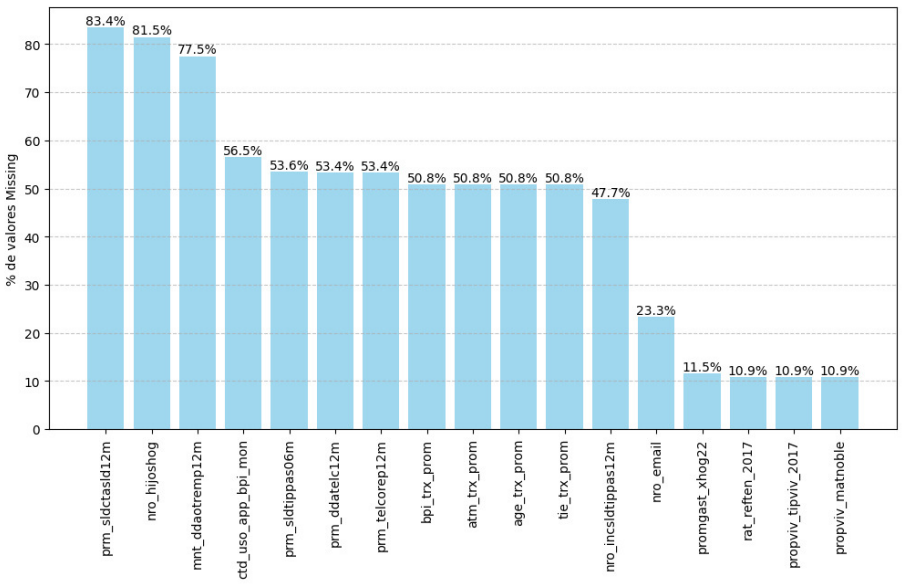


Figura 3.1: Porcentaje de valores faltantes en las variables independientes.

### Tratamiento de datos faltantes

Para tratar estos datos faltantes y según el aprendizaje del modelo, que se detallará en el siguiente apartado sobre el entrenamiento, se ha decidido, por criterio experto, reemplazar los valores faltantes por -99.

Esta decisión se basa en la correlación que tienen las variables con respecto al puntaje final predictivo asignado por el modelo (el score).

En un proceso estándar, este paso suele realizarse después del entrenamiento, ya que requiere evaluar cómo el score del modelo se comporta con respecto a la variable, es decir, si un mayor valor de la variable impacta positiva o negativamente en la predicción. Sin embargo, para efectos de comprensión de la imputación, lo incluimos en esta sección.

- **prm_sldtippas06m :** Esta variable es el promedio de saldo que tiene la persona en sus cuentas de tipo pasivo (por ejemplo, cuentas de ahorros) en los últimos 6 meses. Esta variable presenta una correlación positiva con el *score*, como se puede observar en la figura 3.2. Es decir, a mayor promedio de saldo, el cliente dispone de mejores fondos, lo que se traduce en un mejor perfil para la asignación de una tarjeta de crédito, razón por la cual se le otorga un *score* más alto. En el caso de los valores faltantes (missing), se les asigna el valor de -99, lo que implica en una predicción de un score menor. Esto se debe a que no se tiene información sobre sus fondos, lo que podría indicar un perfil de mayor riesgo.

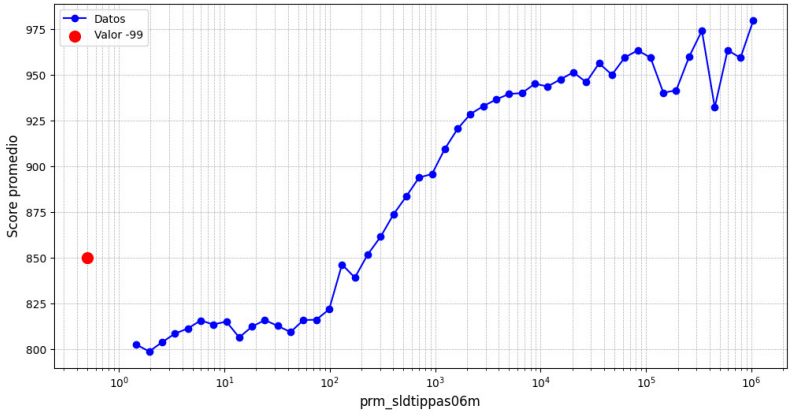


Figura 3.2: Relación entre la variable promedio de saldo en cuenta de tipo pasivo en los últimos 6 meses y el score del modelo.

- **prm_ddatelc12m :** Esta variable es el promedio de deuda que tiene la persona en empresas de telecomunicaciones en los últimos 12 meses (por ejemplo, si tiene deudas

en el pago de su línea telefónica). Esta variable presenta una correlación negativa con el *score* como se muestra en la figura 3.3. Es decir, a mayor deuda, el modelo asigna un *score* menor, ya que se trata de un perfil más riesgoso. En este caso, el dato faltante imputado con -99 indicaría, posiblemente, que la persona no tiene deuda en telecomunicaciones. Por ello, se le asigna un *score* más alto, al considerarse un perfil con menor riesgo.

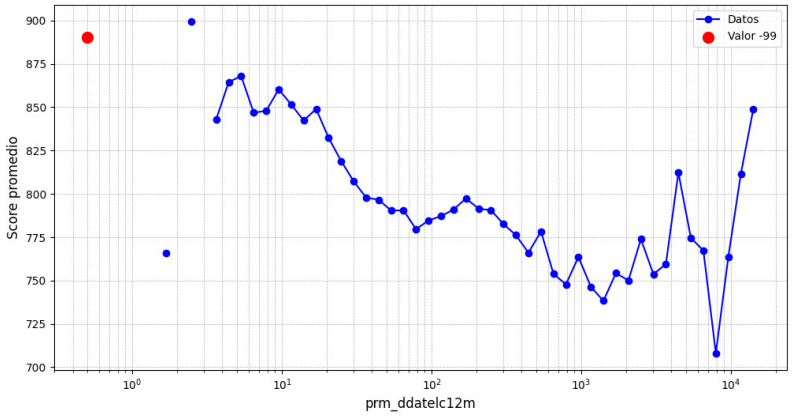


Figura 3.3: Relación entre la variable promedio de deuda en telecomunicaciones en los últimos 12 meses y el score del modelo.

- **prm_telcorep12m :** Esta variable es el número promedio de entidades de telecomunicaciones reportantes en los últimos 12 meses (por ejemplo, si la persona debe a más de una entidad telefónica). Esta variable presenta una correlación negativa con el *score*, como se muestra en la figura 3.4. Es decir, a mayor número de empresas a las que se deba, el modelo asigna un *score* menor, ya que se trata de un perfil más riesgoso. En este caso, el dato faltante imputado con -99 indicaría, posiblemente, que la persona no tiene deudas en empresas de telecomunicaciones. Por ello, se le asigna un *score* más alto, como se muestra en la figura 3.5, al considerarse un perfil con menor riesgo.

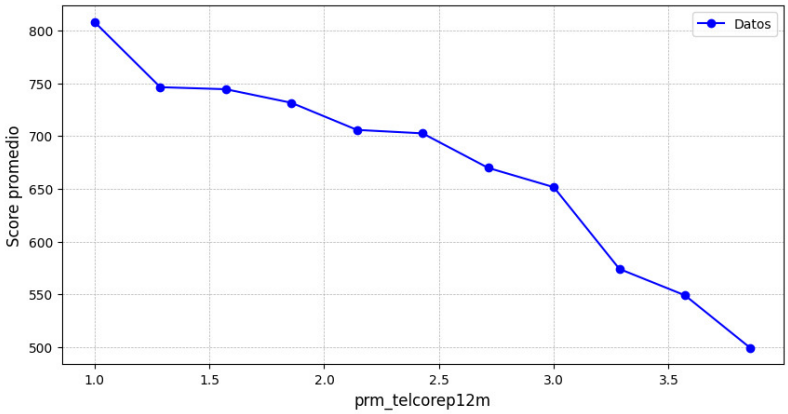


Figura 3.4: Relación entre la variable número promedio de entidades de telecomunicaciones reportantes en los últimos 12 meses y el score del modelo sin considerar el dato faltante.

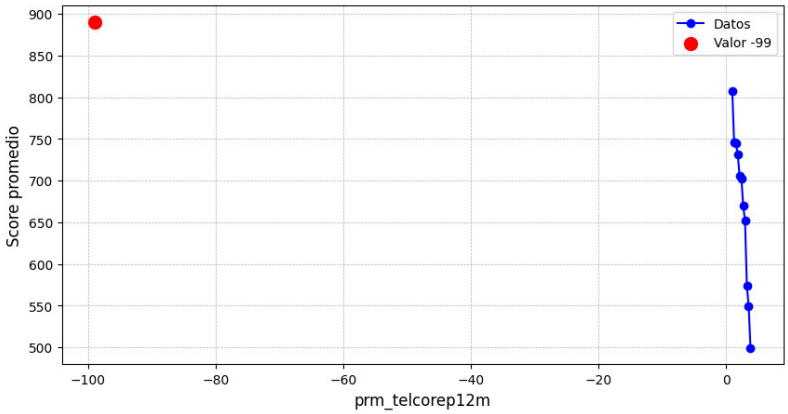


Figura 3.5: Relación entre la variable número promedio de entidades de telecomunicaciones reportantes en los últimos 12 meses y el score del modelo.

- **nro_email :** Esta variable es el número de correos registrados de la persona por la entidad financiera. Esta variable presenta una correlación negativa con el *score*, como se muestra en la figura 3.6. En este caso, el dato faltante imputado con -99 indicaría que la persona no tiene una forma de contacto digital registrada. Por ello, se le asigna un *score* más bajo, como se muestra en la figura 3.7, al considerarse un perfil con mayor riesgo.

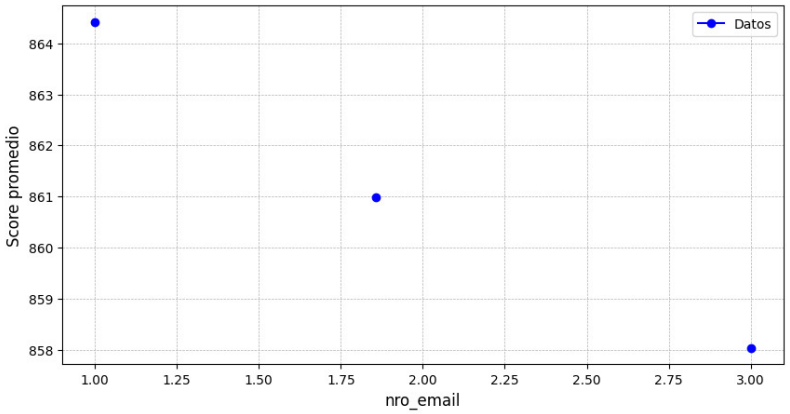


Figura 3.6: Relación entre la variable número de correos registrados de la persona por la entidad financiera y el score del modelo sin considerar el dato faltante.

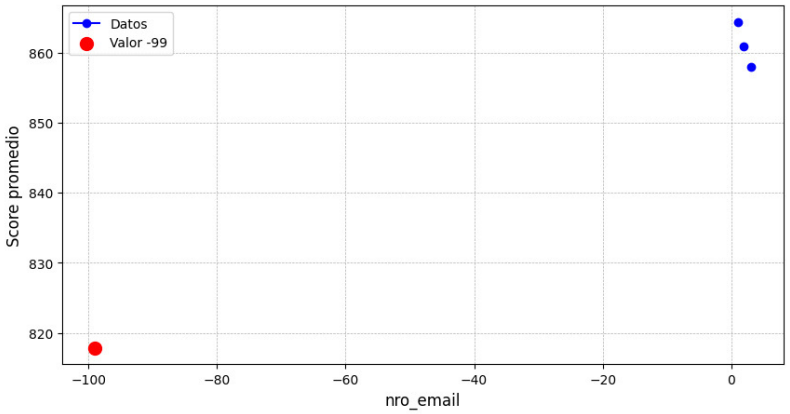


Figura 3.7: Relación entre la variable número de correos registrados de la persona por la entidad financiera y el score del modelo.

- **ctd_uso_app_bpi_mon :** Esta variable es la cantidad de transacciones realizadas por app o banca por internet en el último mes. Esta variable presenta una correlación positiva con el *score*, como se muestra en la figura 3.8. Es decir, a mayor número de transacciones realizadas, el modelo asigna un *score* mayor, ya que la persona también muestra un mejor comportamiento digital. En este caso, el dato faltante imputado con -99 indicaría, posiblemente, que la persona no tiene un comportamiento digital. Por ello, se le asigna un *score* más bajo, como se muestra en la figura 3.9.

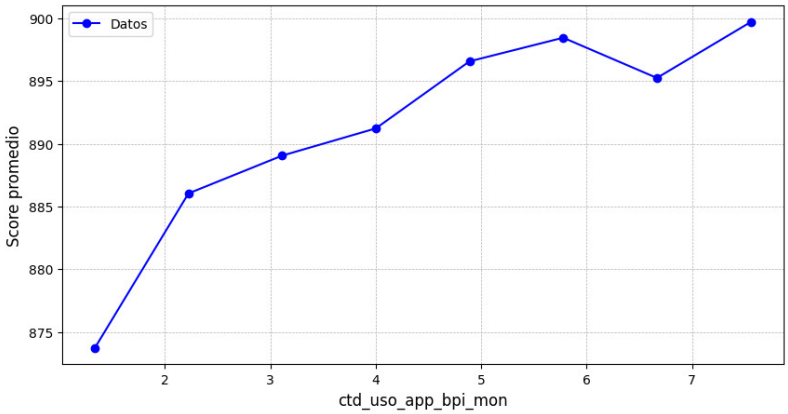


Figura 3.8: Relación entre la variable cantidad de transacciones realizadas por app o banca por internet en el último mes y el score del modelo sin considerar el dato faltante.

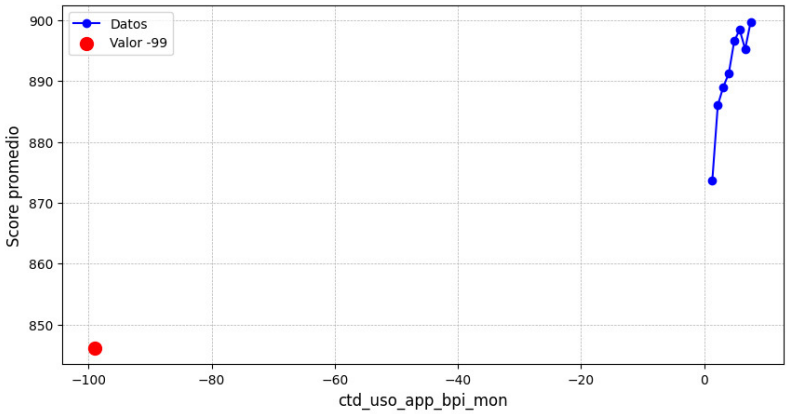


Figura 3.9: Relación entre la variable cantidad de transacciones realizadas por app o banca por internet en el último mes y el score del modelo.

- **bpi_trx_prom :** Esta variable es la transacción promedio en la banca por internet en los últimos 12 meses. Esta variable presenta una correlación positiva con el *score*, como se muestra en la figura 3.10. Es decir, a mayor número de transacciones realizadas, el modelo asigna un *score* mayor, ya que la persona también muestra un frecuente comportamiento digital. En este caso, el dato faltante imputado con -99 indicaría, posiblemente, que la persona no tiene un comportamiento digital. Por ello, se le asigna un *score* más bajo, como se muestra en la figura 3.11.

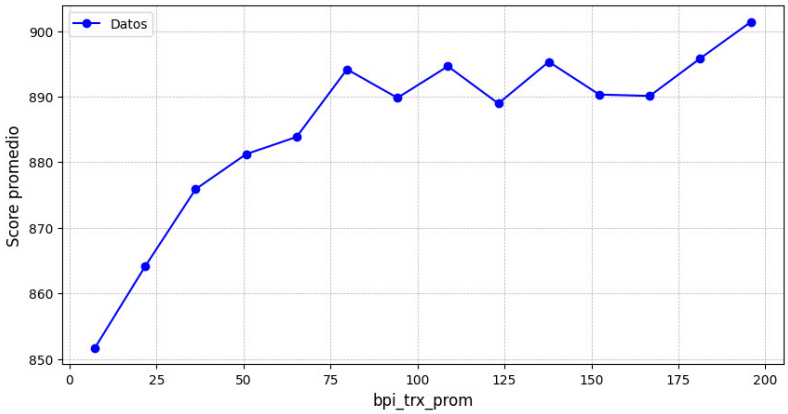


Figura 3.10: Relación entre la variable transacción promedio en la banca por internet en los últimos 12 meses y el score del modelo sin considerar el dato faltante.

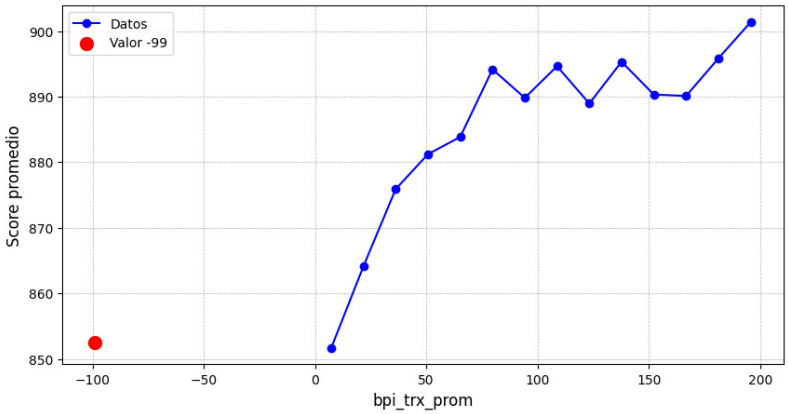


Figura 3.11: Relación entre la variable transacción promedio en la banca por internet en los últimos 12 meses y el score del modelo.

- **atm_trx_prom :** Esta variable es el promedio de transacciones monetarias y no monetarias en el canal ATM (cajero automático) en los últimos 12 meses. Esta variable presenta una correlación negativa con el *score*, como se muestra en la figura 3.12. Es decir, a mayor número de transacciones realizadas por el canal ATM, el modelo asigna un *score* menor, ya que la persona muestra un menor comportamiento digital, prefiriendo el canal presencial. En este caso, el dato faltante imputado con -99 indicaría, posiblemente, que la persona no tiene un comportamiento financiero activo, ya que ni siquiera utiliza el canal ATM. Por ello, se le asigna un *score* más bajo, como se muestra en la figura 3.13.

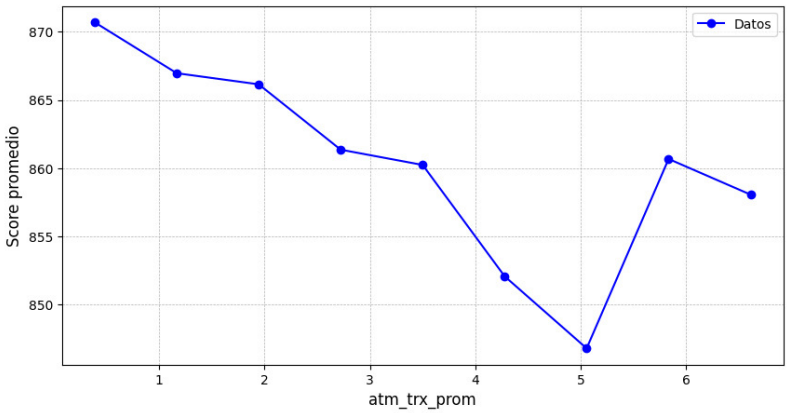


Figura 3.12: Relación entre la variable promedio de transacciones monetarias y no monetarias en el canal ATM en los últimos 12 meses y el score del modelo sin considerar el dato faltante.

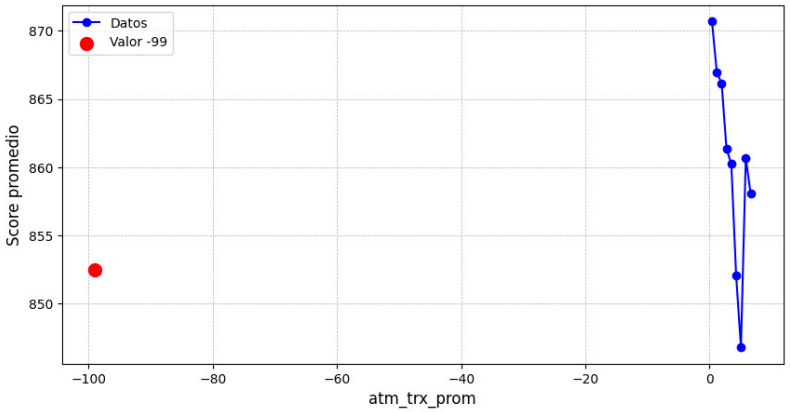


Figura 3.13: Relación entre la variable promedio de transacciones monetarias y no monetarias en el canal ATM en los últimos 12 meses y el score del modelo.

- **age_trx_prom :** Esta variable es la transacción promedio en agentes de la entidad financiera en los últimos 12 meses. Esta variable presenta una correlación negativa con el *score*, como se muestra en la figura 3.14. Es decir, a mayor número de transacciones realizadas en agentes, el modelo asigna un *score* menor, ya que la persona muestra un menor comportamiento digital, prefiriendo el canal presencial. En este caso, el dato faltante imputado con -99 indicaría, posiblemente, que la persona no tiene un comportamiento financiero activo, ya que ni siquiera utiliza el canal de los agentes. Por ello, se le asigna un *score* más bajo, como se muestra en la figura 3.15.

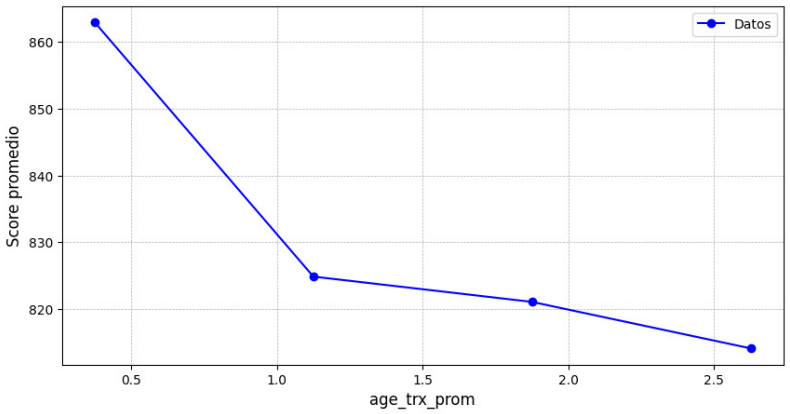


Figura 3.14: Relación entre la variable transacción promedio en agentes de la entidad financiera en los últimos 12 meses y el score del modelo sin considerar el dato faltante.

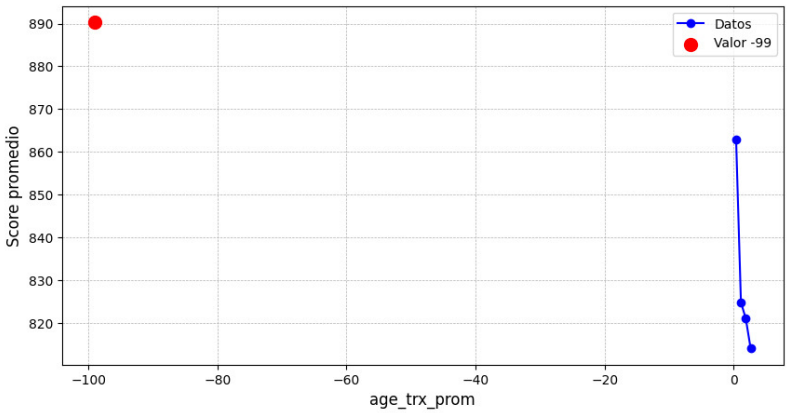


Figura 3.15: Relación entre la variable transacción promedio en agentes de la entidad financiera en los últimos 12 meses y el score del modelo.

- **nro_hijoshog :** Esta variable es el número de hijos en el hogar. Esta variable presenta una correlación negativa con el *score*, como se muestra en la figura 3.16. En este caso, el dato faltante imputado con -99 indicaría que la persona no tiene hijos. Por ello, se le asigna un *score* más alto, como se muestra en la figura 3.17, ya que el modelo entrenado premia a los jóvenes con un perfil digital, como se observa en el comportamiento de las variables anteriores.

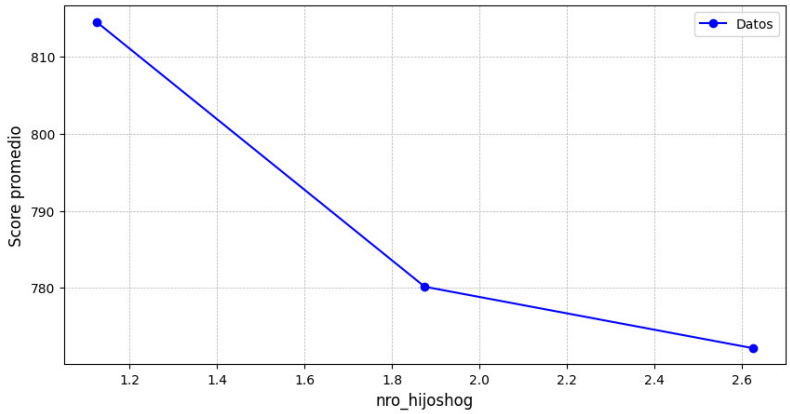


Figura 3.16: Relación entre la variable número de hijos en el hogar y el score del modelo sin considerar el dato faltante.

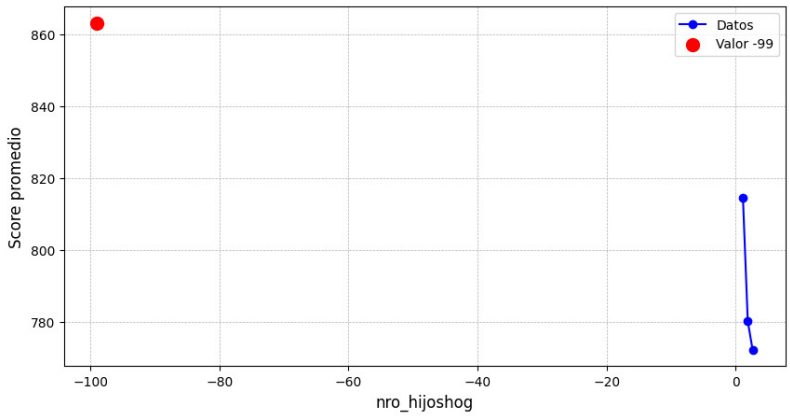


Figura 3.17: Relación entre la variable número de hijos en el hogar y el score del modelo.

- **nro_incsldtippas12m :** Esta variable es el número de incrementos en la cuenta de saldo tipo pasivo en los últimos 12 meses. Esta variable presenta una correlación positiva con el *score*, como se muestra en la figura 3.18. En este caso, el dato faltante imputado con -99 indicaría que la persona no tiene registrada una cuenta de tipo pasivo. Por ello, se le asigna un *score* más bajo, como se muestra en la figura 3.19.

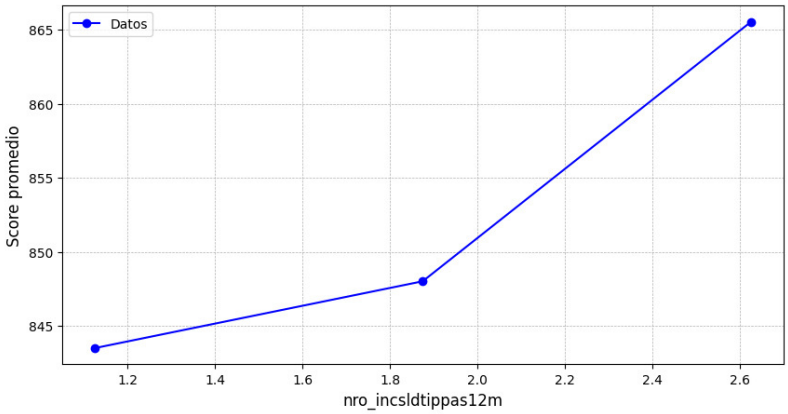


Figura 3.18: Relación entre la variable número de incrementos en la cuenta de saldo tipo pasivo en los últimos 12 meses y el score del modelo sin considerar el dato faltante.

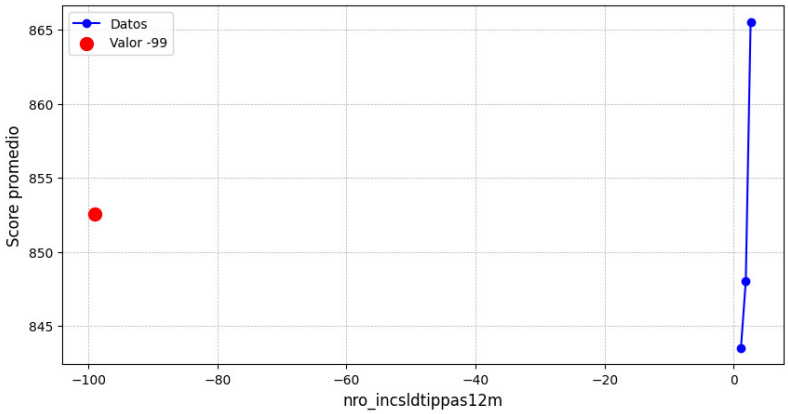


Figura 3.19: Relación entre la variable número de incrementos en la cuenta de saldo tipo pasivo en los últimos 12 meses y el score del modelo.

- **rat_reften_2017 :** Esta variable es la proporción de hogares que poseen refrigeradoras o congeladoras entre la cantidad de hogares en la manzana. Esta variable presenta una correlación positiva con el *score*, como se muestra en la figura 3.20. En este caso, el dato faltante imputado con -99 indicaría que, en la manzana de la persona, probablemente no hay hogares con refrigeradora o congeladora, lo que podría representar un menor nivel socioeconómico. Por ello, se le asigna un *score* más bajo, como se muestra en la figura 3.21.

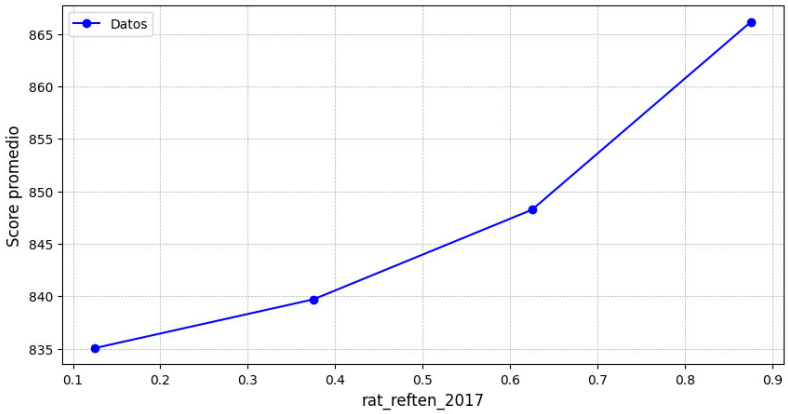


Figura 3.20: Relación entre la variable proporción de hogares que poseen refrigeradoras o congeladoras entre la cantidad de hogares en la manzana y el score del modelo sin considerar el dato faltante.

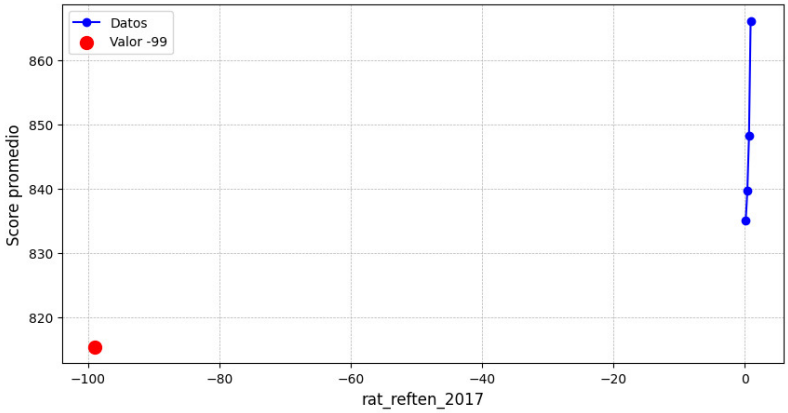


Figura 3.21: Relación entre la variable proporción de hogares que poseen refrigeradoras o congeladoras entre la cantidad de hogares en la manzana y el score del modelo.

- **mnt_ddaotremp12m :** Esta variable es el monto de deuda en otras empresas en los últimos 12 meses. Esta variable presenta una correlación negativa con el *score*, como se muestra en la figura 3.22. En este caso, el dato faltante imputado con -99 indicaría que la persona no tiene deuda registrada. Por ello, se le asigna un *score* más alto, como se muestra en la figura 3.22.

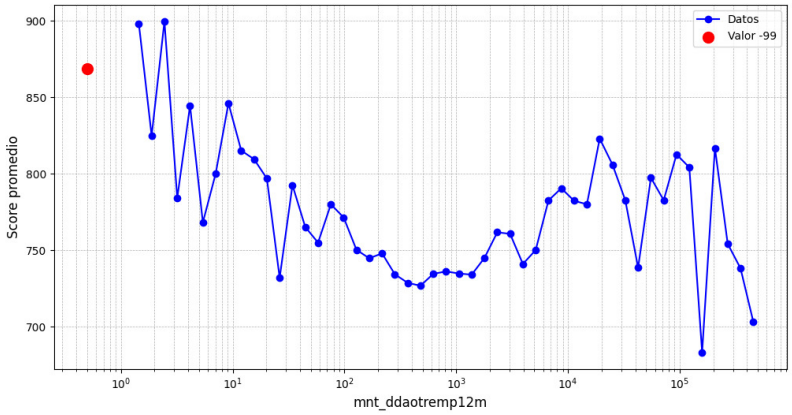


Figura 3.22: Relación entre la variable monto de deuda en otras empresas en los últimos 12 meses y el score del modelo.

- **propviv_tipviv_2017 :** Esta variable representa la proporción de viviendas cuyo tipo es casa independiente o departamento en edificio entre la cantidad de viviendas en la manzana. Esta variable presenta una correlación positiva con el *score*, como se muestra en la figura 3.23. En este caso, el dato faltante imputado con -99 indicaría que, en la manzana de la persona, probablemente no hay viviendas independientes o departamentos en edificio, lo que podría representar un nivel socioeconómico bajo. Por ello, se le asigna un *score* más bajo, como se muestra en la figura 3.24.

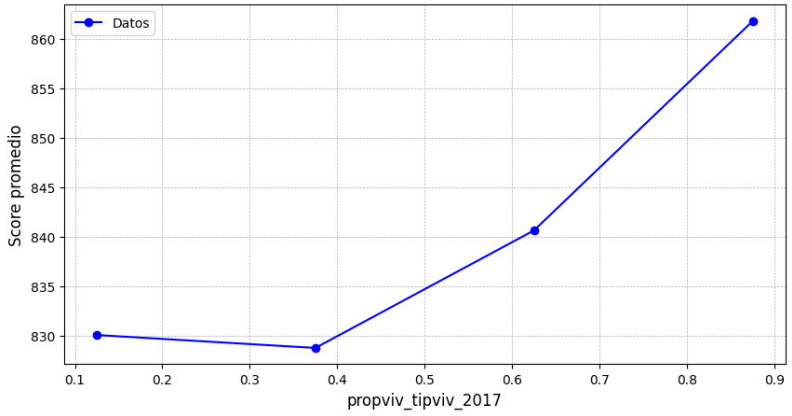


Figura 3.23: Relación entre la variable proporción de viviendas cuyo tipo es casa independiente o departamento en edificio entre la cantidad de viviendas en la manzana y el score del modelo sin considerar el dato faltante.

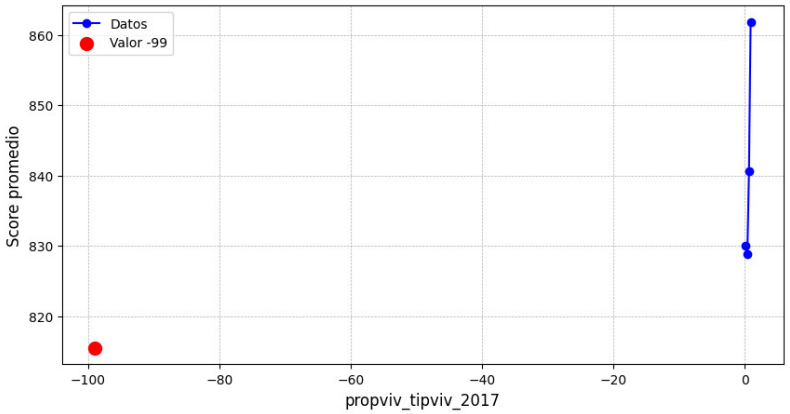


Figura 3.24: Relación entre la variable proporción de viviendas cuyo tipo es casa independiente o departamento en edificio entre la cantidad de viviendas en la manzana y el score del modelo.

- **tie_trx_prom :** Esta variable es la transacción promedio en tiendas de la entidad financiera en los últimos 12 meses. Esta variable presenta una correlación positiva con el *score*, como se muestra en la figura 3.25. En este caso, el dato faltante imputado con -99 indicaría que la persona no tiene actividad financiera frecuente, por lo que podría ser un perfil riesgoso. Por ello, se le asigna un *score* más bajo, como se muestra en la figura 3.26.

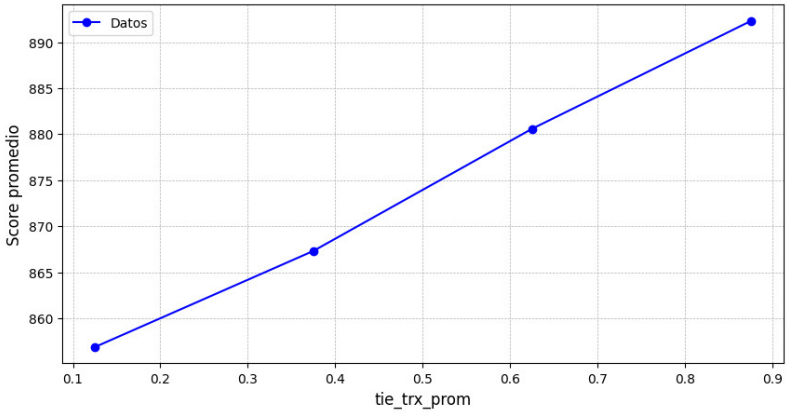


Figura 3.25: Relación entre la variable transacción promedio en tiendas de la entidad financiera en los últimos 12 meses y el score del modelo sin considerar el dato faltante.

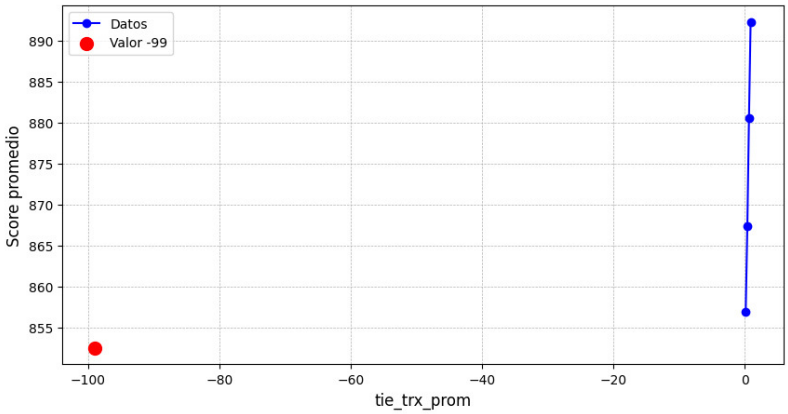


Figura 3.26: Relación entre la variable transacción promedio en tiendas de la entidad financiera en los últimos 12 meses y el score del modelo.

- **prm_sldctasld12m :** Esta variable es el promedio de saldo en cuenta sueldo de los últimos 12 meses. Esta variable presenta una correlación positiva con el *score*, como se muestra en la figura 3.27. En este caso, el dato faltante imputado con -99 indicaría que la persona no tiene cuenta sueldo, es decir, no tiene comportamiento financiero. Por ello, se le asigna un *score* más bajo, como se muestra en la figura 3.27.

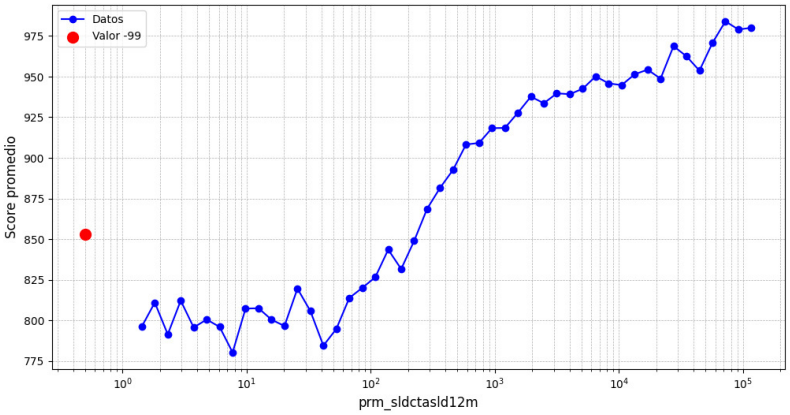


Figura 3.27: Relación entre la variable promedio de saldo en cuenta sueldo de los últimos 12 meses y el score del modelo.

- **promgast_xhog22 :** Esta variable es el promedio ponderado de la suma de todos los gastos promedio mensual de la manzana, según sector económico de la A hasta la E, al año 2022. Esta variable presenta una correlación positiva con el *score*, como se muestra en la figura 3.28. En este caso, el dato faltante imputado con -99 indicaría que la persona no tiene registrados sus gastos promedio. Por ello, se le asigna un *score* más bajo, como se muestra en la figura 3.29.

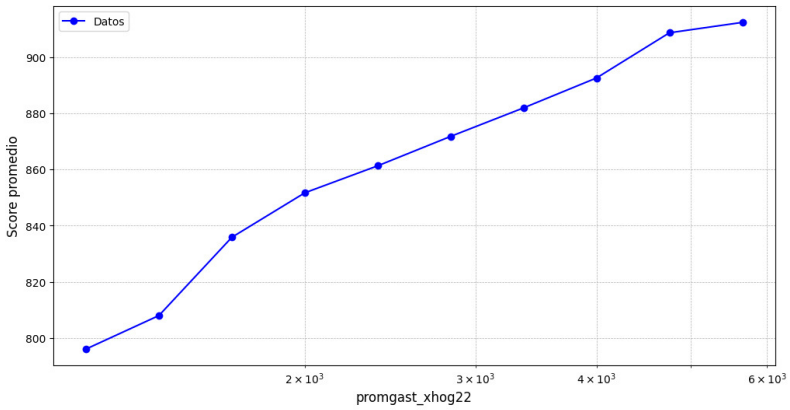


Figura 3.28: Relación entre la variable promedio ponderado de la suma de todos los gastos promedio mensual de la manzana, según sector económico de la A hasta la E, al año 2022 y el score del modelo sin considerar el dato faltante.

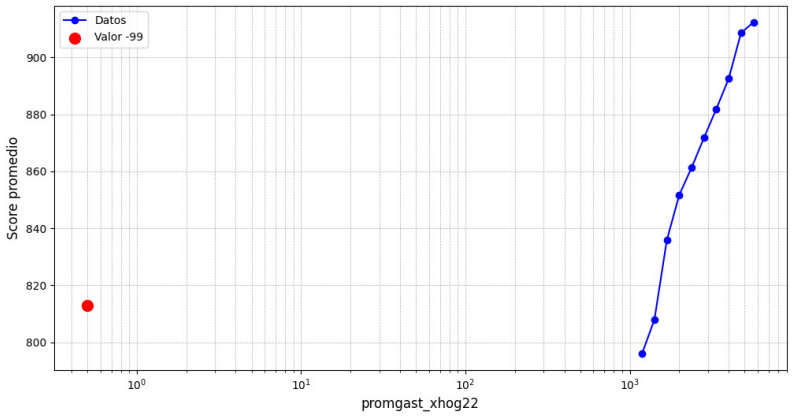


Figura 3.29: Relación entre la variable promedio ponderado de la suma de todos los gastos promedio mensual de la manzana, según sector económico de la A hasta la E, al año 2022 y el score del modelo.

- **propviv_matnoble :** Esta variable representa la proporción de viviendas con material de construcción en las paredes de tipo ladrillo o bloque de cemento entre la cantidad de viviendas en la manzana. Esta variable presenta una correlación positiva con el *score*, como se muestra en la figura 3.30. En este caso, el dato faltante imputado con -99 indicaría que, en la manzana de la persona, no hay construcciones de material noble, lo que indicaría un nivel socioeconómico bajo. Por ello, se le asigna un *score* más bajo, como se muestra en la figura 3.31.

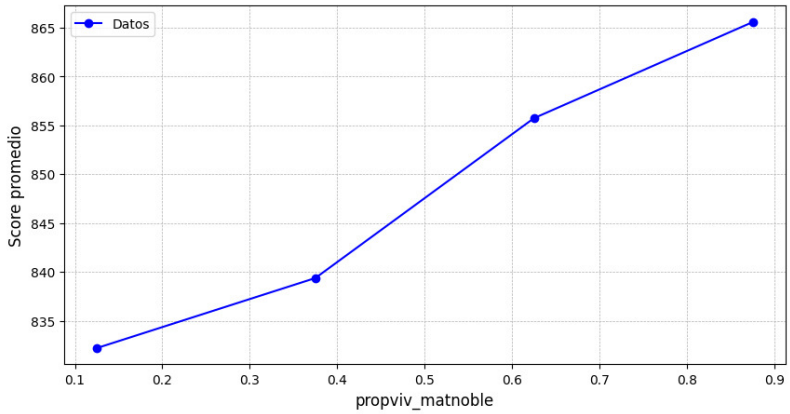


Figura 3.30: Relación entre la variable proporción de viviendas con material de construcción en las paredes de tipo ladrillo o bloque de cemento entre la cantidad de viviendas en la manzana y el score del modelo sin considerar el dato faltante.

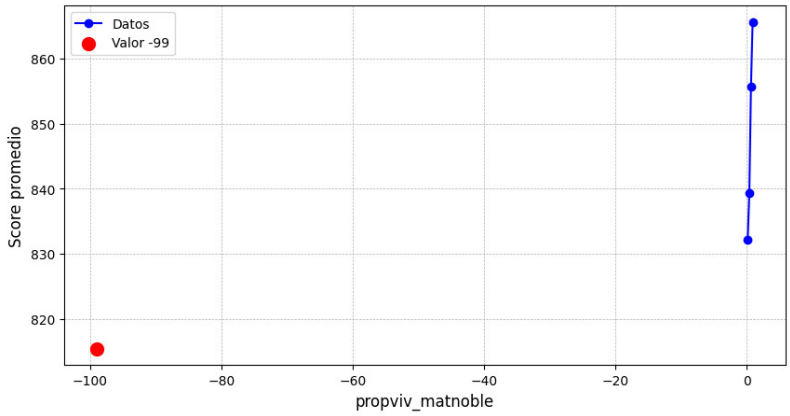


Figura 3.31: Relación entre la variable proporción de viviendas con material de construcción en las paredes de tipo ladrillo o bloque de cemento entre la cantidad de viviendas en la manzana y el score del modelo.

Además de las 24 variables totales, 3 de ellas, las variables *lvl_edu_new*, *segmento_pea_new*, *ubigeo_cat* son de tipo categórico. Cada una de estas variables tiene las siguientes categorías, cuyo significado se detalla a continuación. Han sido ordenadas de manera que presentan una correlación significativa con el valor predicho, como se muestra a continuación:

- **lvl_edu_new :** Esta variable representa el nivel educativo de la persona. La categoría 1 representa haber culminado los estudios de secundaria completa, la categoría 2 representa haber llegado hasta el nivel técnico o universitario, y la categoría 3

representa tener títulos de posgrado. Para nuestra población de entrenamiento representado en la tabla 3.2, la mayor parte se concentra en la categoría 1 con un 49.4 %, seguido de la categoría 2 con 43.5 % y, finalmente, la categoría 3 con 7.1 %. También notamos que sigue una correlación positiva con el *score*, como se muestra en la figura 3.32, esto debido a que un mayor nivel de estudios también se relaciona con un mayor poder adquisitivo, por lo que representa un mejor perfil.

Tabla 3.2: Distribución de la variable nivel educativo.

| lvl_edu_new | cantidad | proporción |
|---|---|---|
| 1 | 26 819 | 49.4 % |
| 2 | 23 642 | 43.5 % |
| 3 | 3 876 | 7.1 % |

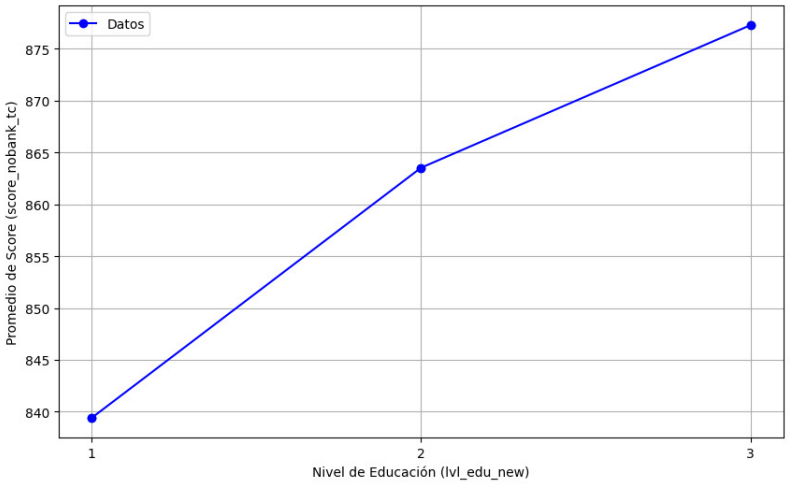


Figura 3.32: Relación entre la variable nivel educativo de la persona y el score del modelo.

- **segmento_pea_new :** Esta variable representa una agrupación por situación laboral y edad de la persona, su distribución en los periodos de entrenamiento esta representado en la tabla 3.3. La categoría 1 agrupa a jóvenes o personas en etapas tempranas de su carrera laboral, con menor estabilidad laboral o ingresos más bajos. La categoría 2 agrupa a personas con mayor estabilidad laboral y experiencia, pero que aún no alcanzan un nivel de ingresos o antigüedad significativo. Por último, la categoría 3 corresponde a personas con mayor experiencia laboral, ingresos altos y estabilidad financiera. Es por ello que sigue una correlación positiva con el *score*, como se muestra en la figura 3.33, debido a que las personas en la categoría 1, al encontrarse

en una etapa temprana de su carrera profesional, podrían representar un perfil más riesgoso. En contraste, aquellas en la categoría 3, con empleo más estable y mayor experiencia, tienen un mayor nivel adquisitivo.

Tabla 3.3: Distribución de la variable agrupación por situación laboral y edad de la persona.

| segmento_pea_new | cantidad | proporción |
|---|---|---|
| 1 | 16 622 | 30.6 % |
| 2 | 20 227 | 37.2 % |
| 3 | 17 488 | 32.2 % |

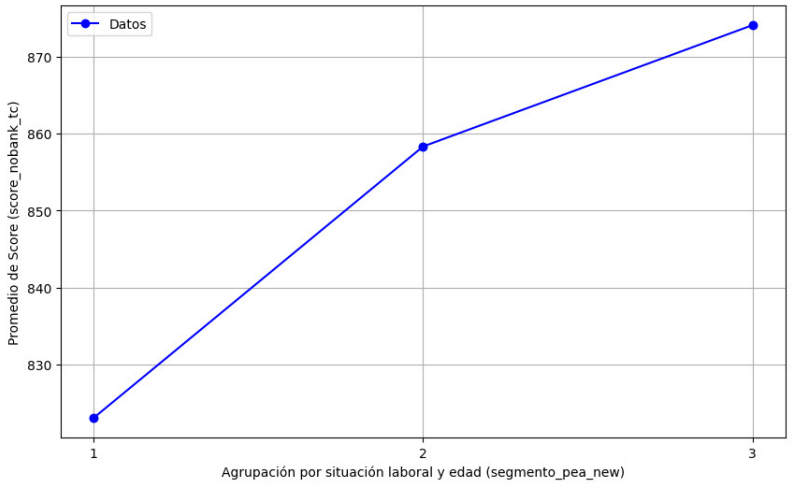


Figura 3.33: Relación entre la variable agrupación por situación laboral y edad de la persona y el score del modelo.

- **ubigeo_cat :** Esta variable representa la clasificación por código de ubigeo, el cual indica la ubicación geográfica del lugar en el que nació la persona, su distribución en los periodos de entrenamiento esta representado en la tabla 3.4. El valor 1 representa zonas rurales o de menor desarrollo económico. El valor 2 representa zonas semi-rurales o en proceso de desarrollo. El valor 3 representa ciudades pequeñas o intermedias. El valor 4 representa ciudades medianas con economías en crecimiento. El valor 5 representa ciudades grandes con alto desarrollo económico. Por último, el valor 6 representa zonas urbanas de mayor desarrollo y acceso a recursos. Es por ello que sigue una correlación positiva con el *score*, como se muestra en la figura 3.34, porque las zonas con mayor desarrollo económico (valores más altos) tienden a tener mayor acceso a servicios financieros, mejores oportunidades laborales y mayores

ingresos. Por el contrario, las zonas con menor desarrollo (valores más bajos) pueden presentar menor acceso a servicios financieros, economías menos diversificadas y menores ingresos.

Tabla 3.4: Distribución de la variable clasificación por código de ubigeo.

| ubigeo_cat | cantidad | proporción |
|---|---|---|
| 1 | 4 603 | 8.5 % |
| 2 | 3 822 | 7.0 % |
| 3 | 8 250 | 15.2 % |
| 4 | 12 367 | 22.8 % |
| 5 | 18 719 | 34.4 % |
| 6 | 6 576 | 12.1 % |

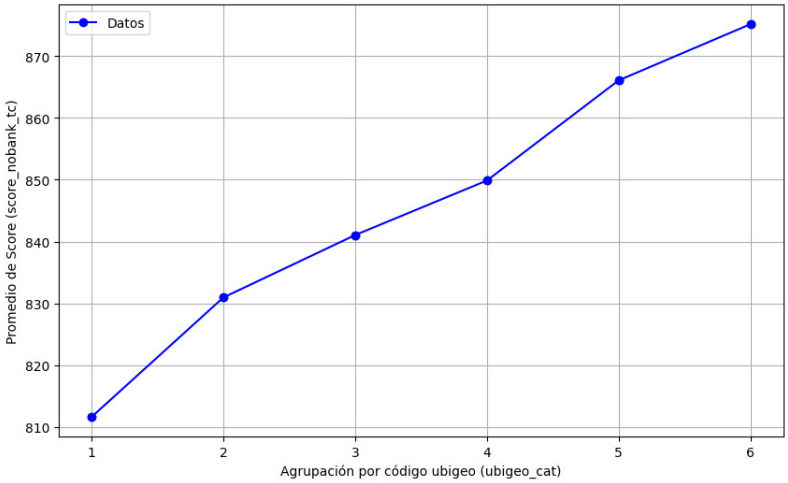


Figura 3.34: Relación entre la variable clasificación por código de ubigeo y el score del modelo.

De las 24 variables, también 3 de ellas, las siguientes *comp1*, *comp2*, *comp3* son conocidas como componentes disociadas. Han pasado por un proceso de transformación para que se pueda utilizar información de otras compañías sin comprometer directamente los datos, por un tema de privacidad. Cada una de estas variables es producto de una combinación lineal de distintas variables, multiplicadas por un peso respectivo, lo que da como resultado final cada una de las componentes disociadas.

A continuación, mostramos la tabla 3.5 descriptivo de las variables considerando la imputación de valores faltantes. Esta tabla presenta el conteo total de registros de cada variable

con *count*, el valor promedio con *mean*, la desviación estándar con *std*, el valor mínimo con *min*, el primer cuartil con *25 %*, el segundo cuartil con *50 %*, el tercer cuartil con *75 %*, y por último el valor máximo con *max*.

Tabla 3.5: Descripción estadística de las variables utilizadas en el modelo.

| variable | count | mean | std | min | 25 % | 50 % | 75 % | max |
|---|---|---|---|---|---|---|---|---|
| prm_sldtippas06m | 54 337 | 1 412.1 | 14 441.3 | -99.0 | -99.0 | -99.0 | 115.8 | 1 184 547.4 |
| prm_ddatelc12m | 54 337 | 44.9 | 362.6 | -99.0 | -99.0 | -99.0 | 32.5 | 15 451.5 |
| lvl_edu_new | 54 337 | 1.6 | 0.6 | 1.0 | 1.0 | 2.0 | 2.0 | 3.0 |
| prm_telcorep12m | 54 337 | -52.5 | 49.8 | -99.0 | -99.0 | -99.0 | 1.0 | 4.7 |
| nro_email | 54 337 | -22.0 | 42.5 | -99.0 | -99.0 | 1.0 | 1.0 | 3.0 |
| segmento_pea_new | 54 337 | 2.0 | 0.8 | 1.0 | 1.0 | 2.0 | 3.0 | 3.0 |
| ctd_uso_app_bpi_mon | 54 337 | -53.4 | 52.8 | -99.0 | -99.0 | -99.0 | 0.0 | 675.0 |
| flag_dependiente | 54 337 | 0.6 | 0.5 | 0.0 | 0.0 | 1.0 | 1.0 | 1.0 |
| bpi_trx_prom | 54 337 | -27.8 | 93.5 | -99.0 | -99.0 | -99.0 | 6.8 | 2 434.2 |
| atm_trx_prom | 54 337 | -49.7 | 50.1 | -99.0 | -99.0 | -99.0 | 0.2 | 50.6 |
| age_trx_prom | 54 337 | -50.1 | 49.7 | -99.0 | -99.0 | -99.0 | 0.0 | 56.2 |
| nro_hijoshog | 54 337 | -80.4 | 38.9 | -99.0 | -99.0 | -99.0 | -99.0 | 5.0 |
| nro_incsldtippas12m | 54 337 | -45.9 | 50.8 | -99.0 | -99.0 | 0.0 | 2.0 | 11.0 |
| ubigeo_cat | 54 337 | 4.0 | 1.4 | 1.0 | 3.0 | 4.0 | 5.0 | 6.0 |
| rat_reften_2017 | 54 337 | -10.1 | 31.1 | -99.0 | 0.6 | 0.8 | 0.9 | 1.0 |
| mnt_ddaotremp12m | 54 337 | 90.3 | 3 690.0 | -99.0 | -99.0 | -99.0 | -99.0 | 514 961.5 |
| propviv_tipviv_2017 | 54 337 | -10.0 | 31.1 | -99.0 | 0.8 | 1.0 | 1.0 | 2.0 |
| tie_trx_prom | 54 337 | -50.2 | 49.6 | -99.0 | -99.0 | -99.0 | 0.0 | 14.7 |
| prm_sldctasld12m | 54 337 | 70.2 | 1 686.4 | -99.0 | -99.0 | -99.0 | -99.0 | 166 222.0 |
| promgast_xhog22 | 54 337 | 2 045.4 | 1 034.3 | -99.0 | 1 825.9 | 2 046.3 | 2 417.9 | 6 146.0 |
| propviv_matnoble | 54 337 | -10.1 | 31.1 | -99.0 | 0.5 | 0.8 | 0.9 | 1.0 |
| comp1 | 54 337 | -0.2 | 1.5 | -29.2 | -1.1 | -0.6 | 1.1 | 2.3 |
| comp2 | 54 337 | -1.3 | 1.7 | -44.1 | -2.7 | -2.0 | 0.9 | 0.9 |
| comp3 | 54 337 | -0.9 | 1.1 | -33.3 | -1.6 | -0.5 | -0.5 | 1.0 |

Ya con el conjunto dataset de 24 variables definido, procedemos a realizar el entrenamiento del modelo. Para ello, debemos dividir la muestra inicial en dos partes: una muestra para desarrollo (también conocida como de entrenamiento) y otra para validación. La primera se empleará para ajustar el modelo, mientras que la segunda servirá para verificar la precisión de los resultados obtenidos.

El objetivo de esta división es crear modelos que puedan aplicarse de manera efectiva a casos nuevos, distintos de los utilizados para su desarrollo. Si se usara toda la muestra de

análisis para ajustar los modelos, estos podrían diferenciar adecuadamente entre clientes buenos y malos, pero no se garantizaría que los resultados sean similares al aplicar dichos modelos en situaciones nuevas.

La base de desarrollo abarca los períodos de junio 2021 hasta septiembre 2021 (4 stocks). La distribución de la variable *target* (las categorías *bueno* y *malo*) en toda la muestra de desarrollo está representada en la tabla 3.6, y la distribución mensual en la tabla 3.7. Mientras tanto, la base fuera de tiempo comprende el mes de octubre 2021 (1 stock), cuya distribución de la variable *target* (bueno y malo) está representada en la tabla 3.8, y la tasa de malos en el único período se muestra en la tabla 3.9.

Tabla 3.6: Distribución de la target en la población de desarrollo.

| Target | Cantidad | Proporción |
|---|---|---|
| Bueno | 46 442 | 85.5 % |
| Malo | 7 895 | 14.5 % |
| Total | 54 337 | 100 % |

Tabla 3.7: Distribución de la target mensualmente en la población de desarrollo.

| Mes | Bueno | Malo | Total | Tasa Malos |
|---|---|---|---|---|
| Junio 2021 | 13 868 | 2 310 | 16 178 | 14.28 % |
| Julio 2021 | 10 286 | 1 698 | 11 984 | 14.17 % |
| Agosto 2021 | 10 433 | 1 896 | 12 329 | 15.38 % |
| Septiembre 2021 | 11 855 | 1 991 | 13 846 | 14.38 % |

Tabla 3.8: Distribución de la target en la población de validación.

| Target | Cantidad | Proporción |
|---|---|---|
| Bueno | 14 931 | 83.2 % |
| Malo | 3 009 | 16.8 % |
| Total | 17 940 | 100 % |

Tabla 3.9: Distribución de la target mensualmente en la población de validación.

| Mes | Bueno | Malo | Total | Tasa Malos |
|---|---|---|---|---|
| Octubre 2021 | 14 931 | 3 009 | 17 940 | 16.77 % |

Ahora pasamos al entrenamiento del modelo, utilizando el algoritmo *CatBoost*. Para ello, empleamos la librería *catboost*, específicamente la función llamada *CatBoostClassifier*. A continuación, se especifican en detalle los distintos hiperparámetros utilizados para ajustar el entrenamiento del modelo.

- **Número de iteraciones (iterations):** Cantidad máxima de árboles de decisión que se construirán durante el entrenamiento del modelo. Un mayor número de iteraciones puede mejorar la precisión, pero también aumentar el tiempo de entrenamiento.
- **Profundidad (depth):** Número máximo de niveles que puede tener cada árbol de decisión. Una mayor profundidad permite que el modelo capture relaciones más complejas, pero también puede aumentar el riesgo de sobreajuste.
- **Ratio de aprendizaje (learning_rate):** Factor que controla la magnitud de los ajustes en los pesos del modelo después de cada actualización. Valores pequeños hacen que el aprendizaje sea más lento pero estable, mientras que valores grandes pueden acelerar el entrenamiento pero aumentar el riesgo de sobreajuste.
- **Factor de regulación (l2_leaf_reg):** Parámetro que controla la penalización L2 en las hojas del árbol para evitar que el modelo se ajuste demasiado a los datos de entrenamiento. Ayuda a mejorar la generalización.
- **Coeficiente de regularización del tamaño del modelo (model_size_reg):** Influye en el tamaño del modelo cuando se trabajan con variables categóricas. Puede ayudar a reducir el consumo de memoria y mejorar la eficiencia del modelo.
- **Método del subespacio aleatorio (rsm):** Proporción de características del conjunto de datos que se utilizarán en cada división del árbol. Un valor más bajo hace que cada árbol dependa de menos variables, lo que introduce aleatoriedad y puede mejorar la capacidad de generalización del modelo.
- **Iteraciones de estimaciones de hojas (leaf_estimation_iterations):** Número de pasos utilizados para calcular los valores de las hojas en cada árbol. En lugar de hacer una sola actualización, *CatBoost* puede usar múltiples pasos basados en los métodos de gradiente o Newton para mejorar la precisión de los valores de las hojas.
- **langevin:** Activa el método *Stochastic Gradient Langevin Boosting*, que introduce ruido en el proceso de optimización para mejorar la generalización del modelo y reducir el sobreajuste.

- **bagging_temperature:** Controla la aleatoriedad en la selección de muestras para entrenar cada árbol del modelo. Un valor bajo (cercano a 0) hace que la selección de muestras sea más uniforme, lo que da como resultado un modelo más estable pero con menor diversidad entre los árboles. Un valor alto aumenta la aleatoriedad en la selección de datos, generando árboles más diversos, lo que puede mejorar la capacidad del modelo para capturar patrones complejos, pero también podría aumentar la varianza.

- **Semilla (random_seed):** Número que establece un punto de partida fijo para la generación de valores aleatorios durante el entrenamiento. Esto permite obtener resultados reproducibles al entrenar el modelo varias veces.

- **Número máximo de hojas (max_leaves):** Cantidad máxima de hojas que puede tener un árbol. Un número mayor permite modelos más complejos, pero también puede aumentar el riesgo de sobreajuste.

- **Mínimo de datos en una hoja (min_data_in_leaf):** Número mínimo de muestras de entrenamiento que debe contener cada hoja del árbol. Un valor más alto ayuda a evitar que el modelo aprenda patrones demasiado específicos del conjunto de datos de entrenamiento.

- **Función de perdida (loss_function):** Métrica utilizada para evaluar el desempeño del modelo durante el entrenamiento. La función elegida define el tipo de problema que se está resolviendo, como clasificación o regresión.

- **verbose:** Controla la cantidad de información que se muestra en la consola durante el entrenamiento. Valores más altos generan mensajes más detallados sobre el progreso del modelo.

Considerando estos hiperparámetros, se entrenó el modelo con los siguientes valores que se muestran en la tabla 3.10:

Tabla 3.10: Valores númericos de los hiperparámetros del modelo.

| Hiperparámetro | Valor |
|---|---|
| iterations | 230 |
| depth | 5 |
| learning_rate | 0.08 |
| l2_leaf_reg | 15 |
| model_size_reg | 13 |
| rsm | 0.5 |
| leaf_estimation_iterations | 10 |
| langevin | True |
| bagging_temperature | 0.5 |
| random_seed | 1234 |
| max_leaves | 32 |
| min_data_in_leaf | 270 |
| loss_function | Logloss |
| verbose | True |

En la siguiente sección, pasaremos a mostrar los resultados y las métricas de evaluación utilizadas.

## 3.3. Métricas de evaluación

Como estamos desarrollando un modelo de riesgo crediticio, es tomado como estándar el uso del coeficiente GINI para medir su eficiencia. Para ello, pasaremos a definir la curva ROC, ya que será esencial para entender también el concepto de AUC y, por último, el indicador KS (Hand & Till, 2001).

### Curva ROC

La curva ROC (Receiver Operating Characteristic) es una herramienta utilizada para evaluar el desempeño de un modelo de clasificación binaria. Se construye trazando una gráfica donde:

- **El eje horizontal (X)** representa la tasa de falsos positivos (FPR), que es la proporción de elementos de la clase negativa que fueron incorrectamente clasificados como positivos.
- **El eje vertical (Y)** representa la tasa de verdaderos positivos (TPR), que es la proporción de elementos de la clase positiva que fueron correctamente clasificados.

Para construir esta curva, se utilizan las probabilidades estimadas de que un objeto pertenezca a la clase positiva. A partir de estas probabilidades, se calculan las funciones de distribución acumuladas $F(\hat{p})$ y $G(\hat{p})$, que describen cómo se distribuyen la proporción acumulada de ejemplos negativos y ejemplos positivos respectivamente.

- **Función de distribución acumulada $F(\hat{p})$:** Representa la proporción acumulada de ejemplos negativos (clase 0 ) que tienen una probabilidad menor o igual a $\hat{p}$. Es decir, nos dice qué porcentaje de los ejemplos negativos tienen valores de probabilidad por debajo de un cierto umbral.

  $$F(\hat{p}) = P(\hat{p}(x) \leq p \mid Y = 0)$$

  Interpretación: Si $F(0.6) = 0.7$, significa que el 70 % de los ejemplos negativos tienen una probabilidad predicha menor o igual a 0.6.

- **Función de distribución acumulada $G(\hat{p})$:** Representa la proporción acumulada de ejemplos positivos (clase 1 ) que tienen una probabilidad menor o igual a $\hat{p}$. Es decir, nos dice qué porcentaje de los ejemplos positivos tienen valores de probabilidad por

debajo de un cierto umbral.

$$G(\hat{p}) = P(\hat{p}(x) \leq p \mid Y = 1)$$

Interpretación: Si $G(0.6) = 0.4$, significa que el 40 % de los ejemplos positivos tienen una probabilidad predicha menor o igual a 0.6.

**Ejemplo 3.1.**

Supongamos que tenemos un modelo de clasificación binaria que predice la probabilidad de que un paciente tenga una enfermedad el cual sera denotado con 1 en caso de tenerla, y con 0 en caso no tenerla. Tenemos los siguientes datos mostrados en la tabla 3.11:

Tabla 3.11: Conjunto de datos para el ejemplo 3.1.

| Paciente | Clase real | Probabilidad predicha $\hat{p}$ |
|---|---|---|
| 1 | 1 (Positivo) | 0.95 |
| 2 | 0 (Negativo) | 0.85 |
| 3 | 1 (Positivo) | 0.75 |
| 4 | 0 (Negativo) | 0.60 |

Tenemos dos ejemplos negativos (clase 0: Pacientes 2 y 4) y dos ejemplos positivos (clase 1: Pacientes 1 y 3).

Ahora calculamos $F(\hat{p})$ (para clase negativa)

- Para $\hat{p} = 0.60 \rightarrow$ Solo el paciente 4 cumple $\hat{p} \leq 0.60$, así que

$$F(0.60) = \frac{1}{2} = 0.5$$

- Para $\hat{p} = 0.75 \rightarrow$ Sigue solo el paciente 4, $\hat{p} \leq 0.75$, así que

$$F(0.75) = \frac{1}{2} = 0.5$$

- Para $\hat{p} = 0.85 \rightarrow$ Ahora los pacientes 2 y 4 cumplen $\hat{p} \leq 0.85$, así que

$$F(0.85) = \frac{2}{2} = 1$$

- Para $\hat{p} = 0.95 \rightarrow$ Sigue siendo 1

Ahora calculamos $G(\hat{p})$ (para clase positiva)

- Para $\hat{p} = 0.60 \rightarrow$ No cumple ningun paciente la condición $\hat{p} \leq 0.60$, así que

$$G(0.60) = \frac{0}{2} = 0$$

- Para $\hat{p} = 0.75 \rightarrow$ Solo el paciente 3 cumple $\hat{p} \leq 0.75$, así que

$$G(0.75) = \frac{1}{2} = 0.5$$

- Para $\hat{p} = 0.85 \rightarrow$ Sigue solo el paciente 3, $\hat{p} \leq 0.85$, así que

$$G(0.85) = \frac{1}{2} = 0.5$$

- Para $\hat{p} = 0.95 \rightarrow$ Ahora los pacientes 1 y 3 cumplen $\hat{p} \leq 0.95$

$$G(0.95) = \frac{2}{2} = 1$$

En resumen tenemos los siguientes puntos representados en la tabla 3.12 para graficar la curva ROC.

Tabla 3.12: Conjunto de datos para graficar la curva ROC.

| $\hat{p}$ | $F(\hat{p})$ | $G(\hat{p})$ |
|---|---|---|
| 0.60 | 0.5 | 0.0 |
| 0.75 | 0.5 | 0.5 |
| 0.85 | 1.0 | 0.5 |
| 0.95 | 1.0 | 1.0 |

Procedemos a graficar la curva ROC con los datos de la tabla 3.12, donde:

- El eje $X$ representa los valores de $F(\hat{p})$ (tasa de falsos positivos)
- El eje $Y$ representa los valores de $G(\hat{p})$ (tasa de verdaderos positivos)

El resultado se muestra en la figura 3.35, donde:

- La línea azul representa la curva ROC
- La línea punteada representa la línea de referencia (modelo aleatorio sin capacidad discriminativa)

Interpretación de la figura 3.35:

- Si la curva ROC está por debajo de la línea de referencia, el modelo es peor que el azar
- Si está por encima, el modelo tiene poder predictivo
- En nuestro caso, al estar por encima de esta línea, confirmamos que el modelo tiene capacidad predictiva

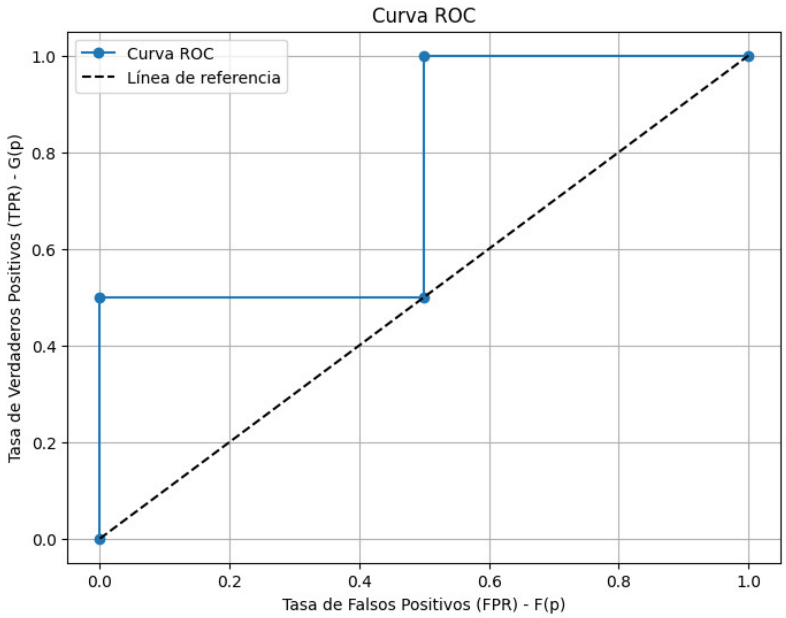


Figura 3.35: Curva ROC. Se muestra el desempeño del modelo en términos de la tasa de verdaderos positivos (TPR) frente a la tasa de falsos positivos (FPR), con la línea de referencia representando un clasificador aleatorio.

### Área bajo la curva ROC (AUC - Area Under the Curve)

El AUC mide el área bajo la curva ROC y representa la probabilidad de que un ejemplo positivo tenga una puntuación mayor que un ejemplo negativo al azar.

- AUC = 1 → Modelo perfecto (separa completamente las clases).
- AUC = 0.5 → Modelo aleatorio (no tiene capacidad de discriminación).
- AUC <0.5 → Modelo peor que el azar (predice al revés).

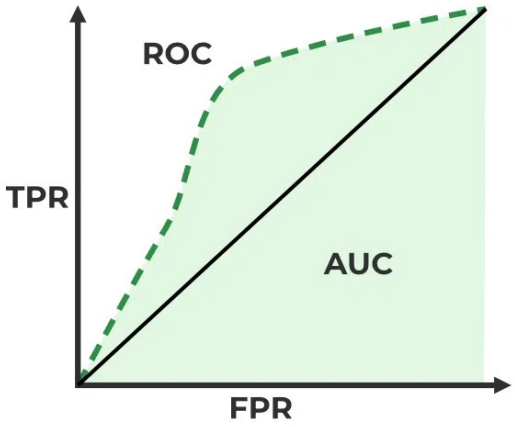


Figura 3.36: Gráfica AUC-ROC, como se observa en la figura, las líneas punteadas representan la curva ROC y el área sombreada debajo de esta corresponde al valor AUC. Adaptado de *AUC-ROC Curve*, por GeeksforGeeks.

**Ejemplo 3.2.**

Imagina que tienes un modelo que predice si una persona tiene una enfermedad (1 = Sí, 0 = No), y las puntuaciones representan la probabilidad de estar enfermo.

Supongamos que el modelo predice estos valores, representados en la siguiente tabla 3.13:

Tabla 3.13: Conjunto de datos para el ejemplo 3.2.

| Persona | Enfermo (1 = Sí, 0 = No) | Probabilidad Predicha |
|---|---|---|
| A | 1 (Sí) | 0.90 |
| B | 0 (No) | 0.40 |
| C | 1 (Sí) | 0.85 |
| D | 0 (No) | 0.60 |

Si seleccionamos al azar una persona enferma y una sana, queremos que la probabilidad predicha de la persona enferma sea mayor que la de la persona sana.

- Si elegimos A (0.90, enfermo) y B (0.40, sano) → El positivo tiene mayor puntuación.
- Si elegimos A (0.90, enfermo) y D (0.60, sano) → El positivo tiene mayor puntuación.
- Si elegimos C (0.85, enfermo) y D (0.60, sano) → El positivo tiene mayor puntuación.

Esto significa que el modelo tiene buena capacidad para diferenciar entre enfermos y sanos.

### Coeficiente GINI

El Coeficiente de Gini mide la desigualdad en la distribución de las predicciones de un modelo. En el contexto de la clasificación, se define en términos del área bajo la curva ROC (AUC) con la siguiente fórmula:

$$Gini = 2 \times AUC - 1$$

- Si el coeficiente de Gini es cercano a 1, significa que el modelo separa muy bien las clases.
- Si el coeficiente de Gini es cercano a 0, significa que el modelo no tiene capacidad de discriminación.
- Si el coeficiente de Gini es negativo, significa que el modelo está clasificando de manera opuesta a lo esperado (peor que un modelo aleatorio).

### Coeficiente de Kolmogorov-Smirnov (KS)

El Coeficiente de Kolmogorov-Smirnov (KS) mide la capacidad de un modelo de clasificación para distinguir entre las clases positiva y negativa. Se basa en la diferencia máxima entre las funciones de distribución acumulada (FDP) de ambas clases.
Matemáticamente, el coeficiente KS se define como:

$$KS = \max_{\hat{p}} |G(\hat{p}) - F(\hat{p})|$$

- $G(\hat{p}) = P(\hat{p}(X) \leq p \mid Y = 1) \rightarrow$ FDP de las puntuaciones de los positivos.
- $F(\hat{p}) = P(\hat{p}(X) \leq p \mid Y = 0) \rightarrow$ FDP de las puntuaciones de los negativos.
- Un KS cercano a 1 indica una excelente separación entre clases.
- Un KS cercano a 0 indica que el modelo no está separando bien las clases.

Si miramos la curva ROC, el KS se puede interpretar como la máxima distancia vertical entre la curva ROC y la línea diagonal o línea de referencia (random guess).

**Ejemplo 3.3.**

Usando la misma información del ejemplo 3.1 representado en la tabla 3.12, agregando la diferencia de la función de distribución acumulado tenemos lo siguiente en la tabla 3.14:

Tabla 3.14: Conjunto de datos para el ejemplo 3.3.

| $\hat{p}$ | $F(\hat{p})$ | $G(\hat{p})$ | $\lvert G(\hat{p}) - F(\hat{p})\rvert$ |
|---|---|---|---|
| 0.60 | 0.5 | 0.0 | 0.5 |
| 0.75 | 0.5 | 0.5 | 0.0 |
| 0.85 | 1.0 | 0.5 | 0.5 |
| 0.95 | 1.0 | 1.0 | 0.0 |

Como podemos notar, la máxima diferencia es la siguiente:

$$KS = \max_{\hat{p}} |G(\hat{p}) - F(\hat{p})| = 0.5$$

KS = 0.5 significa que, en el peor caso, la diferencia entre las distribuciones acumuladas de positivos y negativos es 50 %.

### Resultados del modelo

Ahora pasamos con los resultados de nuestro modelo. Primero, mostramos la gráfica de la curva ROC de nuestro modelo entrenado, representado en la figura 3.37.

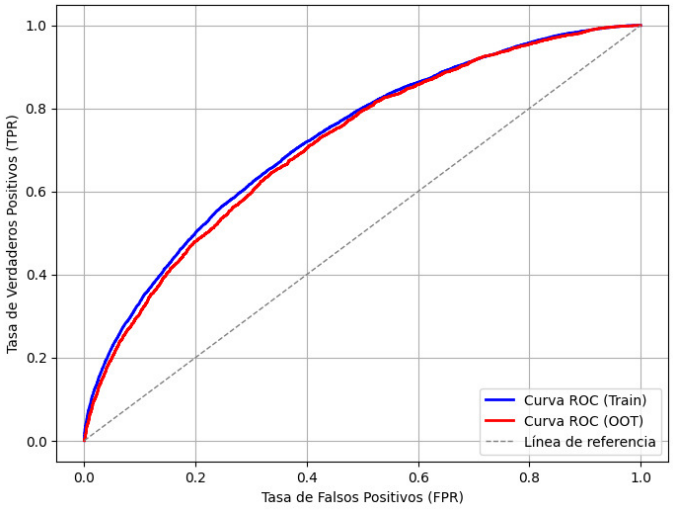


Figura 3.37: Curva ROC de nuestro modelo, la línea azul representa la muestra de desarrollo y la roja, la de validación. Ambas están por encima de la línea de referencia, lo que indica que el modelo tiene capacidad discriminatoria, mejor que el azar. Además, la proximidad de ambas curvas sugiere que el modelo es consistente y no está sobreajustado.

A continuación, mostramos los indicadores GINI, AUC y KS sobre la muestra de entrenamiento, también llamada desarrollo, y sobre la muestra de validación, también llamada OOT, representados en la tabla 3.15. Estos indicadores fueron obtenidos haciendo uso de la librería scipy.stats, específicamente la función ks_2samp para el indicador KS, y la librería sklearn.metrics usando la función roc_auc_score tanto para GINI como AUC.

Tabla 3.15: Métricas de desempeño del modelo en las muestras de desarrollo y validación. Se presentan los valores de GINI, AUC y KS, los cuales muestran una leve disminución en validación, indicando una buena generalización del modelo.

| Base | GINI | AUC | KS |
|---|---|---|---|
| Desarrollo (Train) | 45.14 % | 72.57 % | 32.28 % |
| Validación (OOT) | 42.84 % | 71.42 % | 30.84 % |

# Capítulo 4

# Resultados de la investigación

En este capítulo se presentan y analizan los resultados obtenidos a partir del modelo desarrollado, evaluando su desempeño en distintos periodos y su capacidad de predicción. Para ello, se emplearán métricas clave como el coeficiente GINI, que permite medir la discriminación del modelo, y la tabla de eficiencia, que muestra su capacidad para clasificar correctamente a los individuos según su score.
Además, se analizará la importancia de las variables utilizadas en la predicción mediante valores SHAP, los cuales permiten interpretar la contribución de cada variable en las decisiones del modelo. Esto nos ayudará a comprender qué factores influyen más en las predicciones y cómo estos impactan en el rendimiento general del modelo.

## 4.1. Análisis de los resultados

En este apartado, primero presentaremos el comportamiento del modelo en los períodos de entrenamiento (junio 2021 a septiembre 2021) y validación (octubre 2021), lo cual mostramos en la tabla 4.1.

Tabla 4.1: GINI por períodos.

| Período | GINI |
|---|---|
| Junio 2021 | 45.0 % |
| Julio 2021 | 42.2 % |
| Agosto 2021 | 44.6 % |
| Septiembre 2021 | 48.3 % |
| Octubre 2021 | 42.8 % |

Lo que podemos notar es que el comportamiento es estable para los periodos de entrenamiento alrededor de un 45 %, sin embargo, para el periodo último de entrenamiento de septiembre 2021, se eleva el GINI. Esto se debe a cambios estructurales en las variables más influyentes del modelo.

En particular, la variable *prm_sldtippas06m*, con un peso SHAP relativo del 13.57 % (como veremos más adelante en la tabla 4.3 de importancia de variables) y correlación positiva con el score, experimentó un aumento significativo en su cuartil 3 (Q3, 75 %) en septiembre como se muestra en la figura 4.1. Esto indica que hubo un mayor número de clientes con valores altos en esta variable, lo que está asociado con una mayor probabilidad de ser clasificado como *malo*.

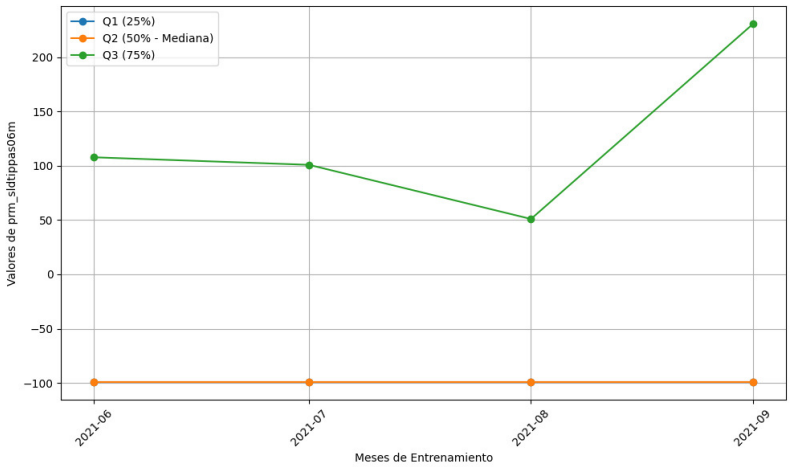


Figura 4.1: Evolución de los cuartiles (Q1, Q2, Q3) de la variable prm_sldtippas06m a lo largo de los meses de entrenamiento.

Por otro lado, la variable *prm_ddatelc12m*, que ocupa el cuarto lugar en importancia con un peso del 5.99 % (como veremos más adelante en la tabla 4.3 de importancia de variables) y tiene una correlación negativa con el score, mostró una caída en su cuartil 3 en septiembre como se muestra en la figura 4.2. Esto sugiere que más clientes presentaron valores bajos en esta variable, lo que está relacionado con una mayor probabilidad de ser *malo*.

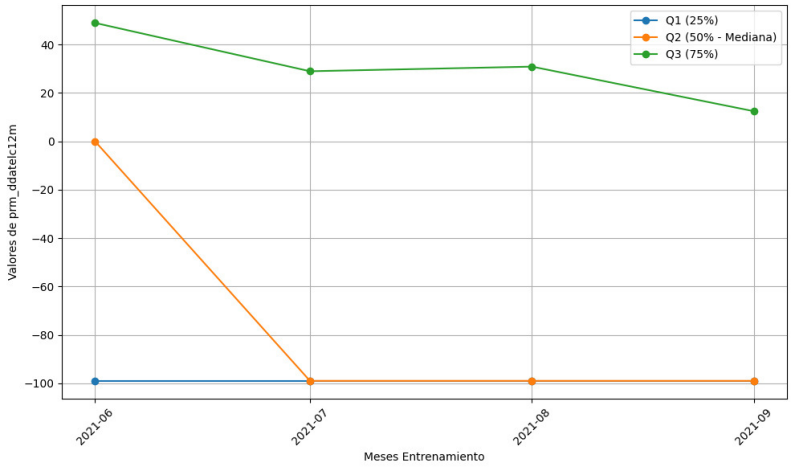


Figura 4.2: Evolución de los cuartiles (Q1, Q2, Q3) de la variable prm_ddatelc12m a lo largo de los meses de entrenamiento.

La combinación de estos factores permitió que el modelo diferenciara mejor entre clientes buenos y malos en septiembre, lo que se reflejó en un incremento del GINI.
Y, como es de esperarse, para el período de validación, la eficiencia del modelo suele disminuir, ya que predice sobre un conjunto de datos no visto previamente. En este caso, la tasa de acierto cae de 48.3 % a 42.8 %, reflejando una diferencia de 5.5 puntos porcentuales.

### Tabla de eficiencia

También agregamos, a continuación, la tabla de eficiencia en la tabla 4.2. Esta muestra el poder de discriminación del modelo en el conjunto de datos de validación, organizando los puntajes en quintiles y comparándolos con la proporción de la tasa de malos. A continuación, presentamos la tabla:

Tabla 4.2: Tabla de eficiencia del modelo, mostrando la distribución de malos según rangos de score. Se observa una menor tasa de malos a medida que el score aumenta.

| Rango score | Promedio | Mínimo | Máximo | Total | Cantidad de malos | Tasa de malos |
|---|---|---|---|---|---|---|
| (0.0, 784] | 700.74 | 181 | 784 | 3 624 | 1,257 | 34.69 % |
| (784, 848] | 819.83 | 785 | 848 | 3 580 | 716 | 20.00 % |
| (848, 890] | 870.44 | 849 | 890 | 3 610 | 531 | 14.71 % |
| (890, 928] | 909.48 | 891 | 928 | 3 589 | 342 | 9.53 % |
| (928, 1000] | 952.50 | 929 | 995 | 3 537 | 163 | 4.61 % |

Lo que nos permite notar esta tabla de eficiencia es la capacidad de discriminación del modelo entre buenos y malos utilizando el score predictivo. Como podemos observar en la tabla 4.2, en los rangos de score más bajos, por ejemplo, entre 0 y 784, se registra una mayor tasa de malos reales de 34.69 %. Esto significa que el modelo asigna puntajes más bajos a quienes verdaderamente presentan un comportamiento financiero negativo, tal como se espera.

Por otro lado, en el rango con score más alto, en este caso de 928 a 1000, la tasa de malos es menor de 4.61 %. Esto indica que, para registros con puntajes elevados, el modelo predice que la mayoría tendrá un buen comportamiento financiero, como era de esperarse.

A continuación, podemos observar esta correlación negativa entre puntaje y tasa de malos, evidenciando que la tasa de malos aumenta a medida que el score disminuye, como se muestra en la figura 4.3. En resumen, nuestro modelo tiene un buen poder predictivo, ya que asigna un score bajo a quienes presentan un comportamiento moroso y un score alto a quienes no.

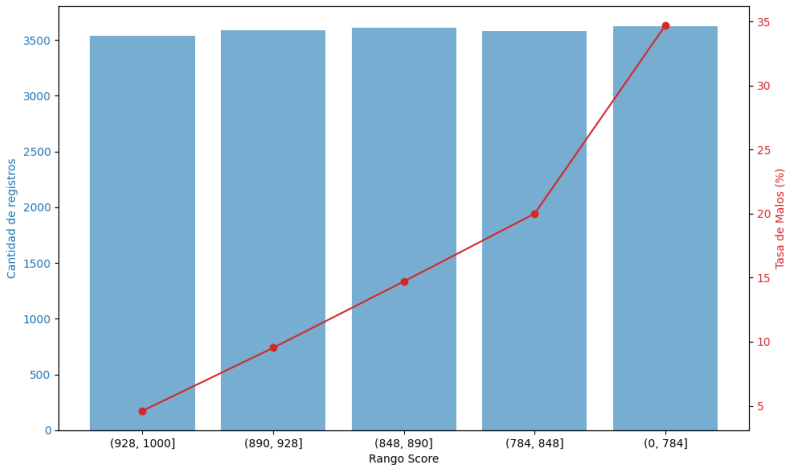


Figura 4.3: Distribución de registros por rango de score (barras) y tasa de malos (línea roja). Se observa que a menor score, la tasa de malos aumenta, validando la capacidad discriminatoria del modelo.

## 4.2. Importancia de las variables en el desempeño del modelo

En este apartado, mostramos cuáles de las 24 variables utilizadas en el modelo son las más influyentes en la predicción. Para ello, nos basamos en el concepto de los valores SHAP, que nos permiten medir la contribución de cada variable en la toma de decisiones del modelo. Además, presentamos una tabla que organiza las variables en un ranking de importancia,

es decir, en orden de relevancia según su impacto en las predicciones. Esto nos permitirá identificar cuáles son los factores clave que el modelo considera al clasificar a los solicitantes.

### Importancia de Variables por SHAP value mean

SHAP (SHapley Additive exPlanations) es un método basado en la teoría de valores de Shapley, que se usa para explicar el impacto de cada variable en la predicción de un modelo. Su objetivo es distribuir de manera justa la contribución de cada variable en la decisión final del modelo, similar a cómo en un juego cooperativo se reparte el valor generado entre los jugadores según su participación.

El concepto subyacente a la importancia de las características SHAP es sencillo: aquellas características con valores de Shapley de magnitud elevada son más relevantes. Para obtener la importancia a nivel global, calculamos el promedio de los valores absolutos de Shapley de cada característica en el conjunto de datos (Lundberg & Lee, 2017).

La importancia de las características SHAP es una alternativa a la importancia de las características de permutación, pero funcionan de manera diferente.

La importancia de las características de permutación mide cuánto empeora el desempeño del modelo cuando una variable se altera al azar. Es decir, si quitamos o desordenamos una variable importante, el modelo hará peores predicciones, lo que indica que dicha variable era relevante.

Por otro lado, SHAP mide la contribución de cada variable en cada predicción individual y luego promedia estas contribuciones. En otras palabras, en lugar de ver cuánto empeora el modelo al quitar una variable, SHAP nos dice cuánto aporta cada variable en cada caso.

**Ejemplo 4.1.**

Imaginemos que tenemos un modelo que predice si una persona aprobará un préstamo. Supongamos que una de las variables es el *historial de pagos*. Si al eliminar o alterar esta variable el modelo empieza a equivocarse más, significa que era una variable importante (esto sería la importancia por permutación). En cambio, SHAP analizaría cuánto peso tiene esta variable en cada decisión del modelo y sacaría un promedio, lo que nos diría qué tan relevante es en general.

A continuación, mostramos la tabla 4.3 de importancia de variables.

Tabla 4.3: Ranking de variables según su importancia en el modelo

| Orden importancia | Variable | Porcentaje de importancia |
|---|---|---|
| 1 | prm_sldtippas06m | 13.57 % |
| 2 | prm_telcorep12m | 7.46 % |
| 3 | lvl_edu_new | 6.00 % |
| 4 | prm_ddatelc12m | 5.99 % |
| 5 | comp1 | 5.53 % |
| 6 | segmento_pea_new | 5.44 % |
| 7 | nro_email | 4.82 % |
| 8 | comp3 | 4.72 % |
| 9 | mnt_ddaotremp12m | 4.62 % |
| 10 | atm_trx_prom | 4.59 % |
| 11 | tie_trx_prom | 4.57 % |
| 12 | comp2 | 4.49 % |
| 13 | age_trx_prom | 3.91 % |
| 14 | ctd_uso_app_bpi_mon | 3.24 % |
| 15 | nro_hijoshog | 2.96 % |
| 16 | nro_incsldtippas12m | 2.82 % |
| 17 | ubigeo_cat | 2.71 % |
| 18 | propviv_tipviv_2017 | 2.61 % |
| 19 | bpi_trx_prom | 2.29 % |
| 20 | promgast_xhog22 | 2.16 % |
| 21 | flag_dependiente | 1.76 % |
| 22 | rat_reften_2017 | 1.34 % |
| 23 | propviv_matnoble | 1.24 % |
| 24 | prm_sldctasld12m | 1.16 % |

# Capítulo 5

# Conclusiones

En esta investigación, se desarrolló y evaluó un modelo predictivo de riesgo crediticio utilizando técnicas de aprendizaje automático, con un enfoque en la capacidad de discriminación y estabilidad del modelo a lo largo del tiempo.
Los principales hallazgos de este estudio se pueden resumir en los siguientes puntos:

- Se desarrolló un modelo basado en CatBoost, véase la subsección **??**, el cual mostró un desempeño competitivo en términos de las métricas estándar utilizadas en modelos de clasificación, véase la tabla 3.15, como el coeficiente de GINI, el área bajo la curva ROC (AUC) y el coeficiente de Kolmogorov-Smirnov (KS).

- Los resultados indicaron que el modelo es capaz de diferenciar adecuadamente entre clientes buenos y malos, véase la tabla 4.2, asignando mayores probabilidades de incumplimiento a aquellos con mayor riesgo financiero.

- La estabilidad del modelo se analizó a lo largo de distintos periodos, véase la sección 4.1, observando que, aunque en general el coeficiente GINI se mantiene estable, hubo fluctuaciones en ciertos meses, explicadas por cambios estructurales en algunas variables clave, véanse las figuras 4.1 y 4.2.

- Se identificaron las variables más influyentes en la predicción del riesgo crediticio utilizando valores SHAP, véase la sección 4.2, lo que permitió interpretar de manera más transparente el impacto de cada factor en la clasificación de los solicitantes.

- Se evidenció que la tasa de malos se comporta de manera inversamente proporcional al score asignado por el modelo, véase la figura 4.3, confirmando su capacidad predictiva.

- Se observó una ligera disminución en la capacidad predictiva del modelo cuando se aplicó a un conjunto de datos de validación fuera del tiempo de entrenamiento, véase la tabla 3.15, lo que resalta la necesidad de un monitoreo continuo y recalibración del modelo en función de nuevas condiciones del mercado.

Finalmente, los resultados de esta investigación demuestran que el uso de modelos avanzados de aprendizaje automático puede mejorar significativamente la capacidad de predicción del riesgo crediticio. No obstante, se recomienda continuar explorando estrategias de robustecimiento del modelo, considerando factores externos que puedan afectar la estabilidad de las variables y la interpretación de los resultados obtenidos.

### Recomendaciones para trabajos futuros

Para futuras investigaciones, se recomienda explorar el uso de enfoques alternativos, como modelos basados en redes neuronales profundas o enfoques híbridos que combinen técnicas de aprendizaje supervisado y no supervisado. Adicionalmente, el uso de modelos de aprendizaje por refuerzo podría ser una estrategia interesante para optimizar la asignación de crédito en función de políticas de negocio dinámicas. También sería valioso evaluar metodologías explicativas adicionales, como modelos de interpretabilidad basados en teoría de juegos o técnicas de sensibilidad de modelos, con el fin de mejorar la transparencia en la toma de decisiones y la confianza en los resultados generados por el modelo.

# Referencias bibliográficas


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

Breiman, L., Friedman, J., Olshen, R., y Stone, C. J. (1984). *Classification and regression trees* (1st ed.). New York: Chapman and Hall/CRC. doi: 10.1201/9781315139470

Dorogush, A. V., Ershov, V., y Gulin, A. (2018). Catboost: gradient boosting with categorical features support. *arXiv preprint arXiv:1810.11363*. `https://arxiv.org/abs/1810.11363`. doi: 10.48550/arXiv.1810.11363

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, *29*(5), 1189–1232. `https://doi.org/10.1214/aos/1013203451`. doi: 10.1214/aos/1013203451

GeeksforGeeks. (2025). *Auc-roc curve.* `https://www.geeksforgeeks.org/auc-roc-curve/`.

Hosmer, D. W. J., Lemeshow, S., y Sturdivant, R. X. (2013). *Applied logistic regression.* Wiley. `https://doi.org/10.1002/9781118548387`. doi: 10.1002/9781118548387

King, G., y Zeng, L. (2001). Logistic regression in rare events data. *Political Analysis*, *9*(2), 137–163. doi: 10.1093/oxfordjournals.pan.a004868

Menard, S. (2002). *Applied logistic regression analysis.* Thousand Oaks: Sage Publications.

Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., y Gulin, A. (2019). Catboost: Unbiased boosting with categorical features. *arXiv preprint*, *v5*. `https://doi.org/10.48550/arXiv.1706.09516`. doi: 10.48550/arXiv.1706.09516