Escuela de Negocios
Tipo de documento: Tesis de maestría

Master in Management + Analytics

Predicción de Churn temprano como herramienta en la industria fintech

Autoría: Vinetz, Natali
Año: 2025

UNIVERSIDAD TORCUATO DI TELLA

MASTER IN MANAGEMENT + ANALYTICS

PREDICCIÓN DE *CHURN* TEMPRANO COMO HERRAMIENTA EN LA INDUSTRIA FINTECH

TESIS

Natali Vinetz

Mayo 2025

Tutor: Yanina Soledad Dip

Resumen

La industria fintech en Argentina ha experimentado un crecimiento acelerado en los últimos años, lo que ha intensificado la competencia. Como resultado, uno de los principales desafíos que enfrentan las empresas del sector es retener a sus usuarios y evitar que migren a la competencia.

Ante este escenario, surge la necesidad de aprovechar la cantidad de datos disponibles para entender por qué los usuarios abandonan los servicios que ofrece la empresa bajo análisis. Para centralizar y profundizar el estudio, se ha elegido un único producto: las tarjetas de débito. Este trabajo se enfoca en analizar el comportamiento de los usuarios de este servicio, con el fin de encontrar el mejor algoritmo para predecir el *churn* y entender los factores clave que impulsan a los usuarios a quedarse o a abandonar.

Más específicamente, el análisis se centrará en predecir si los usuarios dejarán de utilizar el servicio luego de sus primeros 30 días desde la solicitud de su primera tarjeta de débito. Los hallazgos serán propuestos a la empresa para diseñar estrategias de retención enfocadas en los factores más relevantes.

Entre los principales logros de este trabajo, se destacan la identificación de un modelo de machine learning con alta capacidad de discriminación entre clases (1 = *churn*, 0 = no *churn*) y la detección de las variables más influyentes en la retención y el abandono de usuarios.

Abstract

The fintech industry in Argentina has experienced accelerated growth in recent years, which has intensified competition. As a result, one of the main challenges companies in the sector face is retaining their users and preventing them from migrating to competitors.

In this context, the need arises to take advantage of the available data to understand why users abandon the services offered by the company under analysis. To focus and deepen the study, a single product has been selected: debit cards. This project focuses on analyzing the behavior of users of this service, with the aim of finding the best algorithm to predict *churn* and identifying the key factors that drive users to stay or leave.

More specifically, the analysis will focus on predicting whether users will stop using the service after the first 30 days following their request for a debit card. The findings will be presented to the company to design retention strategies focused on the most relevant factors.

Among the main achievements of this work are the identification of a machine learning model with high discriminative capacity between classes (1 = *churn*, 0 = no *churn*) and the detection of the most influential variables in user retention and *churn*.

Índice

Índice de Tablas

Índice de Figuras

1. Introducción

1.1 Contexto

El ecosistema fintech en Argentina continúa mostrando un crecimiento acelerado. Según el informe Radar Argentina 2024 elaborado por Finnovista, el país cuenta con 383 empresas fintech, lo que representa un incremento del 11,7 % respecto a 2023. Este dinamismo se sostiene en una tasa de crecimiento anual compuesta del 15,3% desde el año 2020. Además, sólo un 6,7 % de las startups fundadas en 2023 cerraron sus operaciones, lo que da cuenta de la resiliencia del sector (Finnovista, 2024).

De acuerdo con la Cámara Argentina de Fintech, la industria se organiza en nueve verticales principales: pagos digitales, activos virtuales, finanzas empresariales, préstamos y financiamiento colectivo, tecnología para instituciones financieras, insurtech, activos financieros y mercado de capitales, gestión de finanzas personales, asesoría financiera y entidades financieras disruptivas. En una publicación reciente (septiembre de 2024), la Cámara destacó que las billeteras virtuales ya gestionan el 5 % de los depósitos del sector privado, consolidándose como una herramienta central en la vida financiera de las personas.

La fintech que brindó los datos para esta tesis se encuentra enfocada en expandir su servicio de billetera virtual, con el objetivo de ofrecer una plataforma desde la cual los usuarios puedan realizar todo tipo de transacciones desde donde estén y sin fricciones. Su visión es convertirse en una billetera integral, que combine soluciones de pagos digitales y gestión de finanzas personales.

En línea con el crecimiento de las billeteras virtuales, el uso de tarjetas prepagas y de débito también ha mostrado una expansión sostenida en el ecosistema fintech argentino. De acuerdo con el informe GP Insight - Segundo semestre 2024, los productos de débito emitidos por fintechs continúan ganando participación en el mercado, impulsados por una mayor digitalización de los pagos, la inclusión financiera de nuevos usuarios y la demanda de soluciones ágiles para el manejo del dinero (Global Processing, 2024).

Este tipo de productos no solo permite a los usuarios realizar pagos presenciales y virtuales, sino que además actúan como punto de entrada para personas que históricamente estuvieron excluidas del sistema bancario tradicional. Así, el crecimiento en la emisión de tarjetas está directamente asociado al auge de las billeteras virtuales, que buscan convertirse en

plataformas integrales desde donde los usuarios puedan recibir ingresos, pagar servicios, comprar en comercios físicos y digitales, e incluso invertir.

La compañía cuyos datos se analizaron en este trabajo, opera precisamente dentro de esta lógica: su objetivo es consolidar su billetera como centro de operaciones financieras cotidianas, incluyendo el uso activo de tarjetas prepaga/débito como complemento fundamental del ecosistema digital. La comprensión del comportamiento de los usuarios frente a este producto —especialmente en los primeros días de uso— resulta clave para mejorar la adopción y mitigar el *churn* (abandono o pérdida de usuarios) temprano. Este foco no es menor si se tiene en cuenta que adquirir nuevos usuarios implica un costo considerable para las empresas, tanto en términos económicos como operativos. Por eso, fortalecer la retención no solo mejora la eficiencia del modelo de negocio, sino que se vuelve una estrategia esencial para sostener el crecimiento en el tiempo.

1.2 Problema

El problema que se aborda en esta tesis es la predicción del *churn* de usuarios en el contexto de una fintech que ofrece tarjetas de débito como uno de sus productos principales. En este caso, el *churn* se define como la conducta de aquellos usuarios que, luego de solicitar la tarjeta y completar un primer mes de uso (M0, días 0 a 30), dejan de realizar transacciones durante el segundo mes (M1, días 31 a 60). Esta inactividad en M1 es una señal temprana de abandono, ya que la mayoría de los usuarios que dejan de operar en esta fase inicial no retoman luego el uso del producto.

El objetivo de la tesis es predecir este comportamiento para identificar, de forma anticipada, a los usuarios con alta probabilidad de abandono. Esto permitirá activar intervenciones personalizadas antes de que ocurra el *churn*, como mejoras en la experiencia, incentivos o campañas de fidelización.

En un entorno de alta competencia y con un costo creciente para adquirir nuevos usuarios, mejorar la retención desde las primeras semanas de uso no solo es deseable, sino estratégico. Prevenir el abandono temprano ayuda a maximizar el valor de cada usuario incorporado y a reforzar la sostenibilidad del modelo de negocio.

1.3 Objetivo

En primer lugar, el objetivo es desarrollar un modelo predictivo de *churn* que permita identificar a los nuevos usuarios del servicio de tarjetas de débito de una fintech que tienen una alta probabilidad de abandonar el servicio.

En segundo lugar, se busca identificar los factores que impulsan este comportamiento de abandono para comprender mejor las causas subyacentes del *churn* temprano e implementar una estrategia acorde que mejore el *churn* ratio.

La pregunta de investigación: ¿Cuáles son los factores clave que impulsan el *churn* entre los nuevos usuarios de una tarjeta de débito en la fintech, y cómo se puede diseñar una estrategia de retención efectiva que permita una asignación eficaz de recursos financieros y de personal?

Criterio de éxito:

El éxito de la investigación se evaluará en función de dos aspectos principales:

1. Capacidad del modelo predictivo para identificar con precisión los usuarios en riesgo de abandono de los nuevos usuarios.

2. Identificación de factores de *churn*, lo que implica descubrir las variables más influyentes que impulsan el *churn* y su relación con el comportamiento del usuario.

2. Datos

Para construir la base que alimentará el modelo se trabajó con distintos datasets tabulares provistos por la entidad. Estas bases estaban estructuradas a nivel transaccional, mientras que el objetivo era generar una versión agregada por usuario. Por eso, se implementó un proceso de ETL (Extracción, Transformación y Carga) para unificar, depurar y estructurar la información.

El proceso ETL es clave en ingeniería e integración de datos, ya que permite consolidar información desde múltiples fuentes hacia un repositorio centralizado, optimizando su análisis y facilitando la toma de decisiones. En el marco de esta tesis, el proceso comenzó con la extracción de datos desde las siguientes tablas:

1. **Tabla Debit Cards:** contiene todas las transacciones realizadas con tarjetas de débito emitidas entre el 1/1/2024 y el 31/12/2024. Se filtraron las transacciones efectuadas entre el 1/1/2024 y el 28/2/2025. Incluye detalles como el user ID, fecha de emisión de la tarjeta utilizada, comercio, tipo y estado de la transacción, monto, moneda, entre otros.
2. **Tabla Users:** reúne información básica de cada usuario, incluyendo la fecha de alta en la plataforma y fecha de nacimiento.
3. **Tabla Kustomer:** registra los tickets generados en atención al cliente entre el 1/1/2024 y el 28/2/2025. Se filtraron únicamente aquellos relacionados con el producto debit cards. La tabla contiene datos como el user ID, fecha de apertura y cierre del ticket, y tipo de inconveniente reportado.
4. **Tabla Promos:** incluye las promociones y recompensas obtenidas por los usuarios, con transacciones dentro del período 1/1/2024 al 28/2/2025. Se detalla el user ID, fecha, tipo de recompensa e ID de la promoción, entre otros campos.
5. **Tabla programa de Coins:** presenta información transaccional sobre las coins obtenidas y canjeadas por cada usuario en el mismo rango de fechas (1/1/2024 al 28/2/2025).

En la etapa de transformación, se trabajó sobre cada una de las tablas para agrupar la información a nivel de usuario. Esto implicó consolidar las transacciones y calcular métricas relevantes por persona. Además, se generaron variables dummy a partir de las variables categóricas, lo que permitió cuantificar la presencia o frecuencia de distintos tipos de eventos asociados a cada usuario (por ejemplo, tipos de transacción, comercios, cantidad de tickets abiertos, etc.). De esta manera, se fue construyendo una base estructurada, donde cada fila representa un usuario y cada columna, una característica de su comportamiento en los primeros días de uso.

Dado que el objetivo del modelo es predecir *churn* temprano —es decir, identificar usuarios que dejan de usar la tarjeta en el primer mes después de su adopción— se definió cuidadosamente la ventana de análisis para evitar *data leakage* (contaminación o filtración de datos).

En particular, para cada usuario se consideraron únicamente los datos correspondientes a los primeros 30 días desde la emisión de su primera tarjeta de débito. Primero, este rango de fechas se identificó en la tabla de Debit Cards, y luego se aplicó como filtro en las demás tablas (Kustomer, Promos, Coins) para restringir la información a esa misma ventana. Así, la base final solo incluye información disponible en los primeros 30 días de cada usuario.

A efectos del modelo, se definió que un usuario incurre en *churn* si no realiza ninguna transacción entre los días 31 y 60 desde la creación de su tarjeta. Esta ventana (días 31 a 60) es lo que se denomina M1, y no se incluye como parte del input del modelo, sino que se utiliza únicamente para construir la variable objetivo, también llamada variable target o clase.

La base final, agregada a nivel de usuario, es la que se utiliza en la etapa de carga como input para el modelo de predicción de *churn* temprano. Es importante destacar que los datos provistos por la empresa fueron completamente transformados a los efectos de preservar el anonimato de los usuarios. Por esta razón, no se cuenta con información sensible como nombres, documentos, correos electrónicos o datos de contacto, lo que garantiza la confidencialidad durante todo el proceso de análisis.

2.1 Análisis exploratorio de los datos

Luego de agrupar la información a nivel de usuario, se obtuvo un dataframe final compuesto por **86.915 filas** (una por usuario) y **102 columnas**, es decir, **102 features**. La explicación de cada variable, junto con su tipo de dato, se encuentra detallada en el Apéndice A.

Como primer paso, se revisó el tipo de dato de cada columna. El dataset incluye 46 columnas de tipo float, 52 de tipo int y 4 de tipo object. A continuación, se realizó un chequeo variable por variable para identificar aquellas que requerían transformación de tipo.

Las columnas first_card_creation_date y user_created_date, resolution_total_time contienen información temporal, por lo que se transformaron al formato *datetime*. Por su parte, las columnas user_external_id y semestre, son categóricas y se mantuvieron como tipo *string*, ya que no requerían modificación.

2.1.1 Variable objetivo

La variable objetivo del modelo es user_churn, una variable binaria que toma el valor **1** si el usuario 'abandona' o deja de usar el producto de tarjetas de débito, y **0** en caso contrario.

Se define como *churn temprano* a aquellos usuarios que, luego de los primeros 30 días desde la emisión de su tarjeta (período denominado **M0**), no realizan ninguna transacción entre los días 31 y 60 (**M1**). Esta ausencia de actividad en M1 indica desuso del producto y se utiliza como criterio para etiquetar al usuario como *churner*.

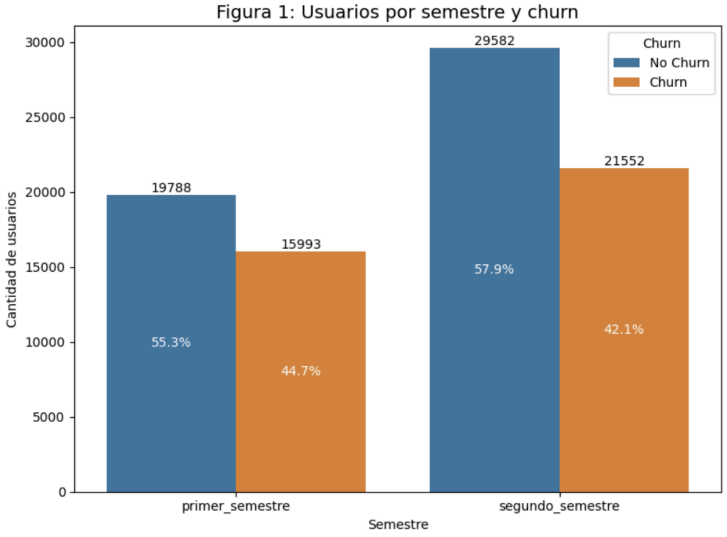
Según se observa en la Figura 1, las clases de la variable user_churn se encuentran balanceadas tanto en el primer como en el segundo semestre. Esta distribución es favorable por dos

motivos. Primero, porque los datos del **primer semestre** se utilizarán para el entrenamiento y prueba del modelo, mientras que los del **segundo semestre** se reservarán para validación. Tener una proporción similar de *churners* y no *churners* en ambos períodos contribuye a una evaluación más robusta y generalizable.

Segundo, porque un dataset balanceado facilita el entrenamiento de los modelos, ya que evita que estos se vean sesgados hacia la clase mayoritaria, un problema común en tareas de clasificación binaria. Cuando las clases están desbalanceadas, muchos algoritmos tienden a favorecer la clase más frecuente, afectando la capacidad de detectar correctamente los casos positivos (en este caso, *churners*) (He & Ma, 2017).

Por último, llama la atención la magnitud del *churn*: un 44,7% de los usuarios deja de usar la tarjeta en el primer semestre, y un 42,1% en el segundo. Estos niveles elevados de abandono subrayan la urgencia de contar con un modelo predictivo capaz de anticipar este comportamiento y permitir el diseño de estrategias de retención más efectivas.

Figura 1: Usuarios por semestre y churn

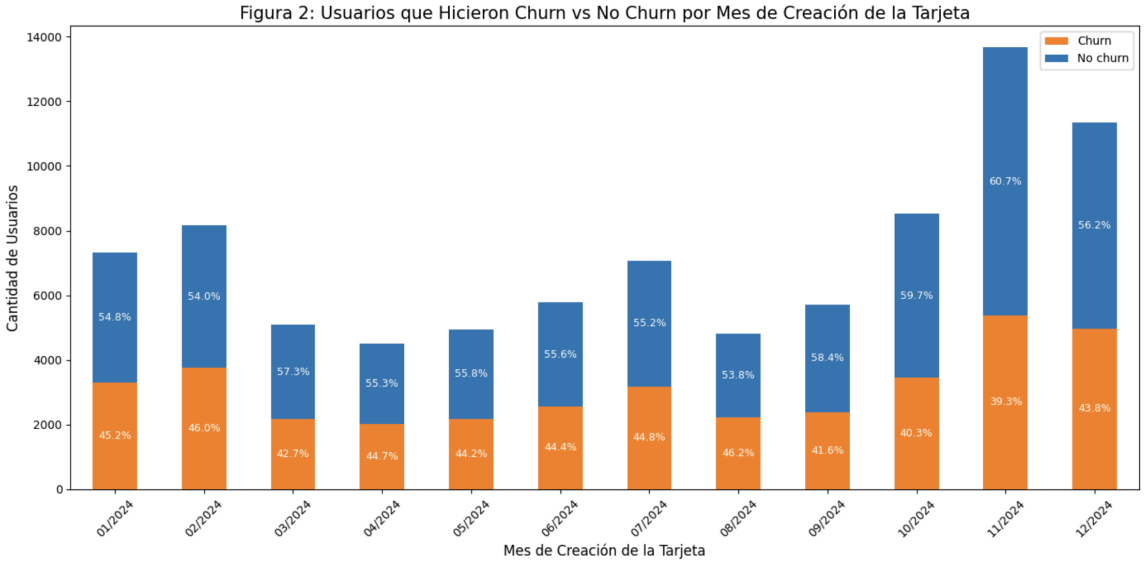


La Figura 2 muestra la cantidad de usuarios que realizaron *churn* y los que no, desglosados por mes de creación de su primera tarjeta. Este desglose permite observar la evolución del comportamiento de los usuarios a lo largo del tiempo y detectar posibles patrones estacionales o cambios en la retención.

Se mantiene una proporción relativamente estable entre *churners* y no *churners* en cada mes, lo cual reafirma que las clases están balanceadas a lo largo del período analizado. Esta consistencia es valiosa, ya que contribuye a la robustez del modelo al entrenar con una muestra representativa de distintos momentos del año.

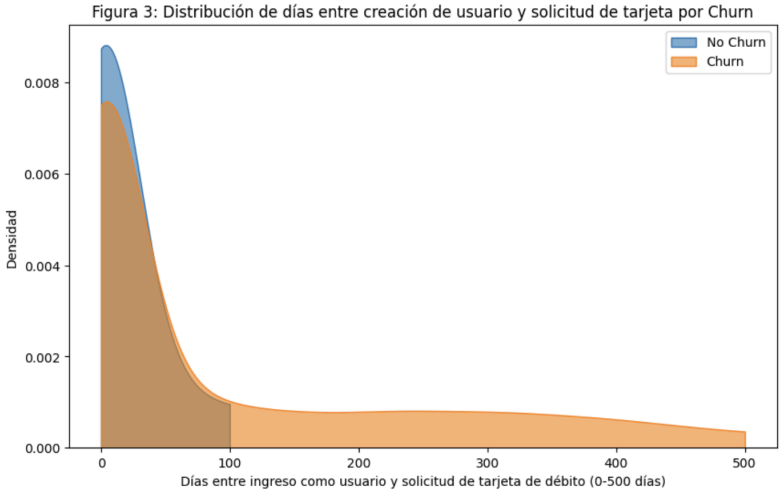
También se destaca el fuerte crecimiento en la cantidad de usuarios a partir de octubre, lo que puede reflejar un aumento en la adopción del producto o campañas comerciales específicas. Sin embargo, este crecimiento viene acompañado de un volumen igualmente alto de abandonos, lo que refuerza la necesidad de actuar tempranamente para retener a estos nuevos usuarios.

Figura 2: Usuarios que Hicieron Churn vs No Churn por Mes de Creación de la Tarjeta



El siguiente histograma comparativo entre clases (Figura 3), muestra el tiempo (en días) entre el registro del usuario y la solicitud de su primera tarjeta, separado por clase (*churn* vs. no *churn*).

Figura 3: Distribución de días entre creación de usuario y solicitud de tarjeta por Churn



Se observa una alta concentración de usuarios que solicitan la tarjeta dentro de los primeros 30 días posteriores al registro, tanto en los que luego abandonan como en los que permanecen activos. Este patrón sugiere una adopción temprana del producto, lo cual puede interpretarse como una señal de buen *engagement* inicial.

Sin embargo, la distribución de los usuarios que realizaron *churn* muestra una mayor dispersión, con una porción relevante que demora más tiempo en solicitar la tarjeta. Este comportamiento podría reflejar dudas iniciales o un menor interés en el producto, factores que aumentan el riesgo de abandono posterior. Además, se destaca una cola larga hacia la derecha en la distribución de *churners*, que representa usuarios con tiempos de adopción considerablemente más largos. Este subgrupo podría corresponder a personas que ingresaron atraídas por otros productos o servicios de la plataforma y que no encontraron suficiente valor en la tarjeta de débito como para adoptarla de forma activa.

Estas observaciones permiten identificar señales tempranas que podrían ser útiles para la predicción del *churn*. En particular, el tiempo entre el registro y la primera solicitud de tarjeta aparece como una posible variable clave: una adopción rápida se asocia con menor probabilidad de abandono, mientras que demoras prolongadas podrían indicar perfiles con mayor riesgo, que podrían beneficiarse de estrategias de retención más personalizadas desde el inicio.

Analizando con mayor profundidad los perfiles de usuario, la entidad los clasificó en cuatro categorías según su comportamiento dentro de la plataforma durante los primeros 30 días: Cross Border, Gamer, Local Shopper y Mixed Players. Esta segmentación, propia de la entidad, permite explorar cómo distintos patrones de uso se relacionan con el *churn*.

Los **Gamers** son usuarios que realizan exclusivamente transacciones dentro del ámbito de la fintech relacionadas con apuestas deportivas y juegos de azar. En contraste, los Cross Border son usuarios que suelen ser nómadas digitales, freelancers o viajeros frecuentes que necesitan operar con libertad a través de distintos países. Para este tipo de perfil, la posibilidad de gestionar, mantener y transferir múltiples monedas no es solo una comodidad, sino una necesidad. Se trata de usuarios que han realizado al menos una transacción genérica (como un depósito o retiro) y que además poseen saldos en cuentas multimoneda o han efectuado transferencias internacionales.

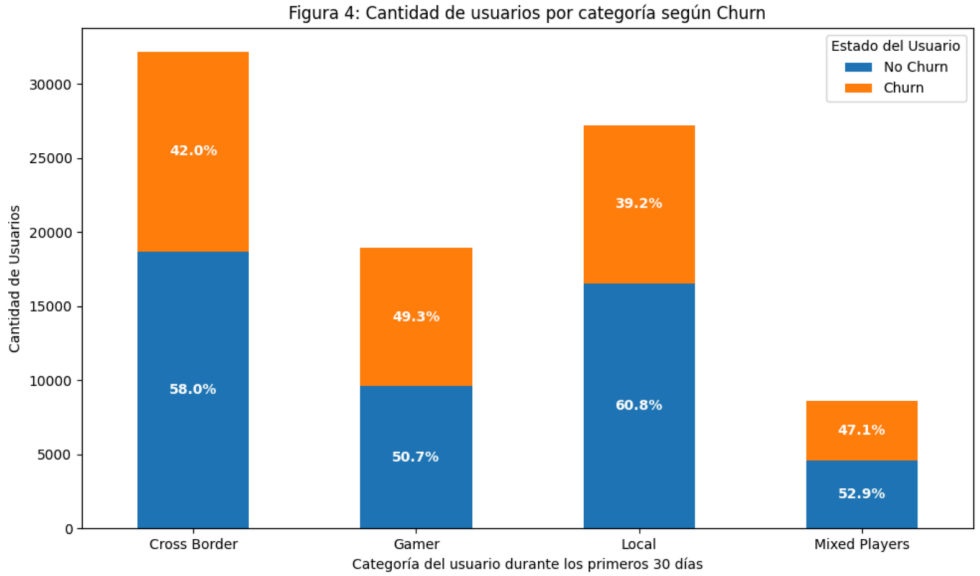
Por otro lado, los **Local Shoppers** son aquellos usuarios que también han realizado al menos una transacción genérica, pero que además interactúan con otros productos y servicios de la fintech como pagos con tarjeta, pago de servicios, ahorro, entre otros. Este grupo se caracteriza, además, por tener patrones de gasto más localizados, es decir, suelen operar dentro de un mismo país o incluso en una misma región. Finalmente, los *Mixed Players* son usuarios que muestran un comportamiento más variado, realizando un poco de todo, sin que

predomine una categoría particular. Debido a esta mezcla de actividades, no es posible asignarlos de forma clara a ninguna de las tres caracterizaciones anteriores.

Como muestra la Figura 4, la categoría **Cross Border** agrupa la mayor cantidad de usuarios y presenta una distribución relativamente equilibrada entre *churners* (42%) y no *churners* (58%). Este equilibrio puede sugerir una relación más estable con el producto o una mejor adecuación desde las primeras interacciones.

A su vez, la categoría **Local** muestra un número considerable de usuarios que hacen *churn* en términos absolutos, aunque su proporción respecto al total de usuarios del segmento es la más baja (39,2%) entre todos los segmentos. No obstante, estos usuarios cuyo comportamiento esta más limitado al entorno local podrían beneficiarse de intervenciones específicas para mejorar su retención.

Figura 4: Cantidad de usuarios por categoría según Churn



La categoría **Gamer**, es la categoría que registra la tasa más alta de *churn* (49,3%), además de presentar un nivel bajo de adopción de la tarjeta de débito. Esto sugiere que, además de que abandonan activamente el servicio, estos usuarios podrían no estar encontrando suficiente valor en el producto financiero, lo que plantea una oportunidad para adaptar las propuestas a sus intereses.

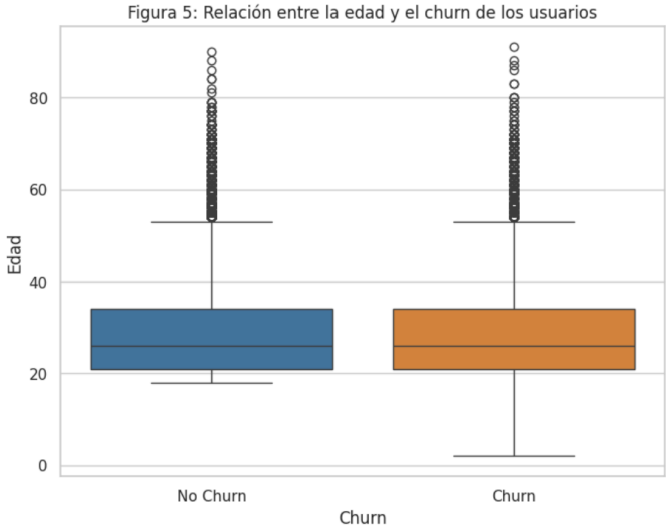
Por último, **Mixed Players** representa el menor volumen de usuarios, pero con una tasa de *churn* relativamente alta. A pesar de ser un segmento pequeño, su vulnerabilidad al abandono lo convierte en un grupo que requiere atención específica para evitar pérdidas tempranas.

Respecto a la relación entre la edad y el *churn* de los usuarios (Figura 5), se puede concluir que la distribución de edades entre los usuarios que hicieron *churn* y los que no es bastante similar.

Las medianas se ubican en un rango cercano (20-25 años), lo que indica que no hay diferencias relevantes en la edad típica de ambos grupos. Además, el rango intercuartil es amplio en los dos casos, lo que refleja una alta variabilidad en las edades y la ausencia de una concentración clara en un grupo etario específico.

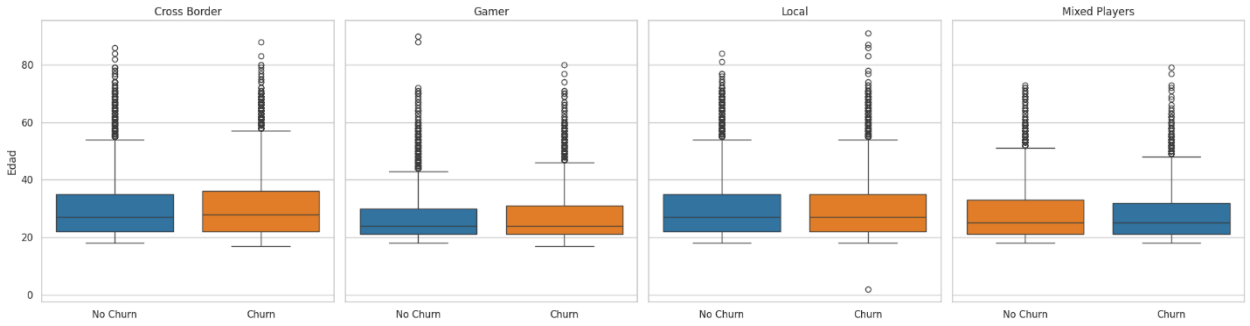
También se observan muchos valores atípicos, especialmente en edades mayores a 60 años. Esto muestra que, aunque la mayoría de los usuarios son jóvenes, existe una proporción no menor de personas mayores con comportamientos distintos. Sin embargo, como estos casos se presentan tanto en *churn* como en no *churn*, no se puede afirmar que la edad avanzada esté asociada directamente con la decisión de abandono.

Figura 5: Relación entre la edad y el churn de los usuarios



Al desagregar la información por segmento (Figura 6), se evidencian algunas diferencias en los perfiles etarios entre grupos. Los segmentos **Gamer** y **Mixed Players** concentran a los usuarios más jóvenes, con medianas que rondan los 23 a 25 años. En contraste, los segmentos **Cross Border** y **Local** agrupan usuarios con edades ligeramente mayores, con medianas cercanas a los 28-30 años. Este patrón parece estar vinculado al tipo de uso de la plataforma: mientras que los gamers hacen un uso más específico y lúdico, los usuarios locales o internacionales suelen utilizar la fintech para operaciones financieras más complejas, lo que podría estar asociado a una mayor edad y experiencia.

Figura 6: Distribución de Edad por Churn en Cada Segmento



2.1.2 Variables transaccionales de debit cards

El análisis del monto total transaccionado muestra una diferencia clara entre usuarios que abandonaron el producto y aquellos que permanecieron activos. La **media de transacciones** para usuarios que **no hicieron *churn*** es de **USD 163**, mientras que para los que **sí abandonaron** es de **USD 86,69**. Esta diferencia sugiere que un menor nivel de actividad económica está asociado con una mayor probabilidad de abandono.

En ambos grupos, el **mínimo transaccionado** es de **USD 0**, mientras que el **máximo** asciende a **USD 196.438,11** para usuarios activos y a **USD 78.452,28** para *churners*.

Desde la perspectiva de la **cantidad de transacciones**, se observa un patrón similar. Los usuarios que no abandonaron realizaron en promedio **16,26 transacciones**, frente a un promedio de **6,72** entre quienes sí realizaron *churn*. En ambos casos, el mínimo es de **1 transacción**, con un máximo de **989** para usuarios activos y **525** para *churners*.

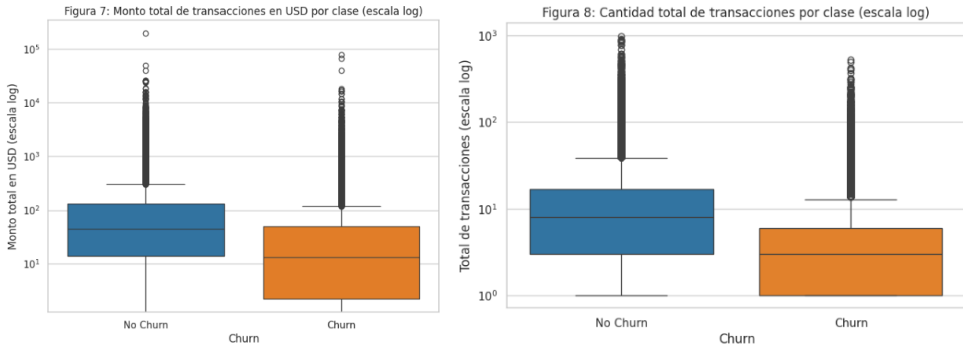
Al desagregar por tipo de moneda, se observa que la gran mayoría de las transacciones se realizaron en **pesos argentinos**. En esta moneda, los usuarios activos promediaron **14,68 transacciones**, mientras que los *churners* realizaron **5,92**. En **moneda extranjera**, los promedios bajan significativamente: **1,59** para usuarios activos y **0,79** para *churners*. En cuanto a los extremos, el máximo de transacciones en pesos argentinos fue de **989**, mientras que en moneda extranjera fue de **410**.

Estos resultados refuerzan la relación entre nivel de actividad —tanto en frecuencia como en volumen— y la probabilidad de *churn*. Un menor uso del producto, especialmente dentro del primer mes, se presenta como un fuerte indicador de abandono posterior.

Tal como se muestra en la Figura 7, existen diferencias claras en la distribución del **monto total transaccionado** entre usuarios *churners* y no *churners*. Los usuarios que **no abandonaron** el producto presentan una **mediana** en torno a los **USD 50–60** y un **rango intercuartílico** (IQR)

entre **USD 15 y 150**, lo que indica que la mayoría se concentra en ese rango de uso. Por el contrario, quienes **sí realizaron *churn*** tienen una mediana más baja, entre **USD 25 y 30**, y un IQR más acotado, lo que sugiere transacciones de menor valor.

Esta diferencia sugiere que los usuarios que abandonan el producto tienden, en sus primeros 30 días, a operar con montos más bajos, posiblemente reflejando un nivel de compromiso o adopción más débil desde el inicio. En ambos grupos se observan ***outliers*** —usuarios con montos transaccionados inusualmente altos—, por lo que se considera apropiado aplicar una transformación logarítmica a esta variable para reducir el impacto de estos valores extremos y mejorar su visualización y análisis.



De manera similar, la Figura 8 muestra diferencias en la **cantidad total de transacciones**. Los usuarios **no *churners*** tienen una mediana cercana a las **8 transacciones**, con un IQR entre **3 y 20**, lo que refleja una mayor frecuencia de uso del producto en su etapa inicial. En contraste, los *churners* presentan una mediana cercana a las **3 transacciones** y un rango intercuartílico mucho más estrecho.

Esta diferencia refuerza la idea de que la **baja frecuencia de uso en los primeros 30 días** es una señal temprana del riesgo de abandono. Nuevamente, ambos grupos exhiben *outliers* —usuarios con actividad muy elevada—, lo que también justifica el uso de escala logarítmica para esta variable.

En conjunto, estos análisis sugieren que tanto el **volumen** como la **frecuencia** de transacciones en el primer mes son indicadores importantes del comportamiento posterior, y deben ser considerados como *features* clave en la etapa de modelado.

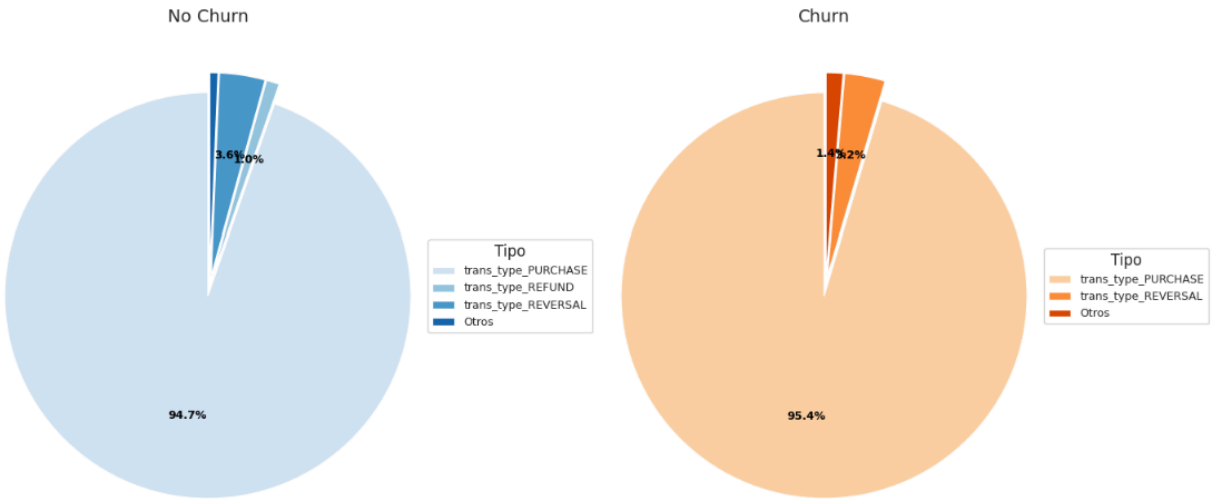
Por otra parte, como se observa en la Figura 9, la gran mayoría de las transacciones —tanto para usuarios que realizaron *churn* como para los que no— corresponden al tipo trans_type_PURCHASE, con proporciones superiores al **94%** en ambos grupos. Los tipos

REFUND y REVERSAL tienen una presencia marginal, aunque aparecen **ligeramente más frecuentes** en el grupo No *Churn*.

El resto de las categorías (CHARGEBACK, WITHDRAWAL, OFFLINE_PURCHASE, entre otras) representan **menos del 1% del total**, por lo que su peso es prácticamente irrelevante desde una perspectiva proporcional.

En una primera lectura, no se observa una relación significativa entre el tipo de transacción y el comportamiento de *churn*. Aun así, estas variables se conservarán para su evaluación posterior en la etapa de modelado, ya que podrían aportar información cuando se combinan con otros atributos del usuario o del contexto transaccional.

Fugura 9: Proporción de tipos de transacción por clase



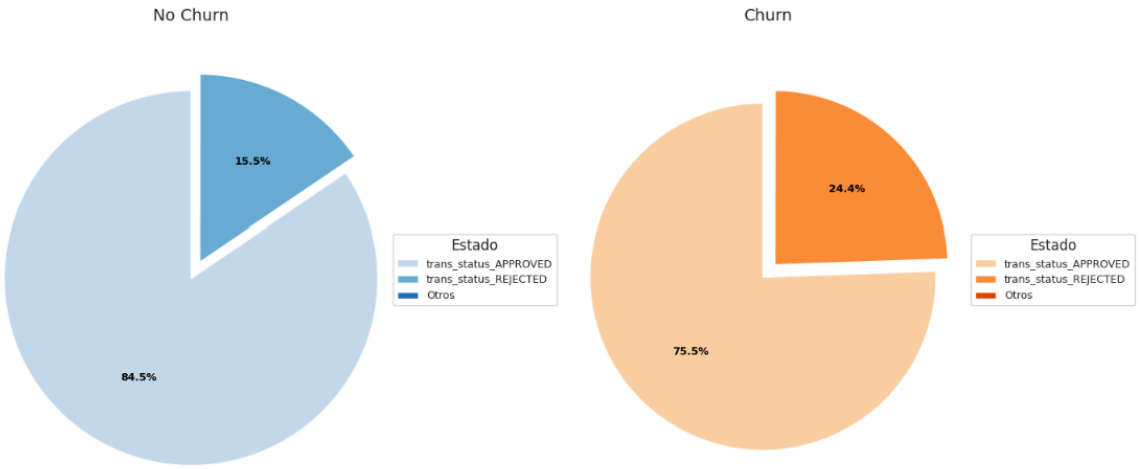
La Figura 10 muestra la distribución de transacciones según su estado. En ambos grupos, el estado predominante es trans_status_APPROVED, aunque con una diferencia notable: representa el **84,5%** de las transacciones en usuarios que **no hicieron *churn***, frente al **75,5%** en el grupo de ***churners***.

Por su parte, las transacciones con estado REJECTED son significativamente más frecuentes en el grupo *churn*: **24,4%**, en comparación con sólo **15,5%** en el grupo no *churn*. Esta diferencia sugiere que una mayor proporción de transacciones rechazadas podría estar relacionada con el abandono temprano del producto, ya sea por fricción en la experiencia de uso o por fallos que afectan la percepción del servicio.

Los demás estados (PENDING, ERROR, DISMISSED, entre otros) tienen una representación muy baja y fueron agrupados bajo la categoría "Otros" por motivos de claridad visual y analítica. Dado su bajo peso relativo, se propone **descartar estas categorías menores** y conservar

únicamente las variables asociadas a los estados APPROVED y REJECTED para la etapa de modelado.

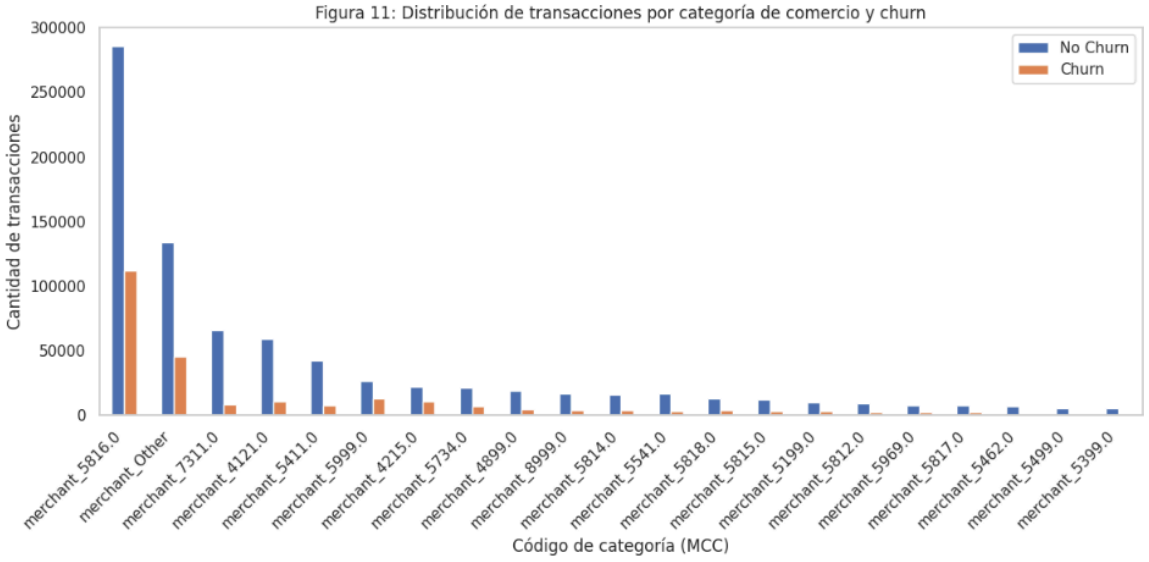
Figura 10: Proporción de estados de transacción por clase



La Figura 11 muestra la distribución de transacciones según la categoría de comercio (MCC), diferenciando entre usuarios que realizaron *churn* y los que no. En casi todas las categorías, el volumen de transacciones es mayor en el grupo de usuarios **no *churn***, lo que indica un uso más activo y diverso del producto entre quienes permanecen activos.

La categoría más frecuente para ambos grupos es **5816**, que corresponde a **suscripciones en plataformas digitales y de entretenimiento**, seguida por la **7311**, que incluye principalmente **compras dentro de Facebook**.

Figura 11: Distribución de transacciones por categoría de comercio y churn



Al analizar la **proporción de usuarios que realizaron al menos una transacción** en diferentes categorías MCC durante sus primeros 30 días, se observan diferencias claras: los usuarios que

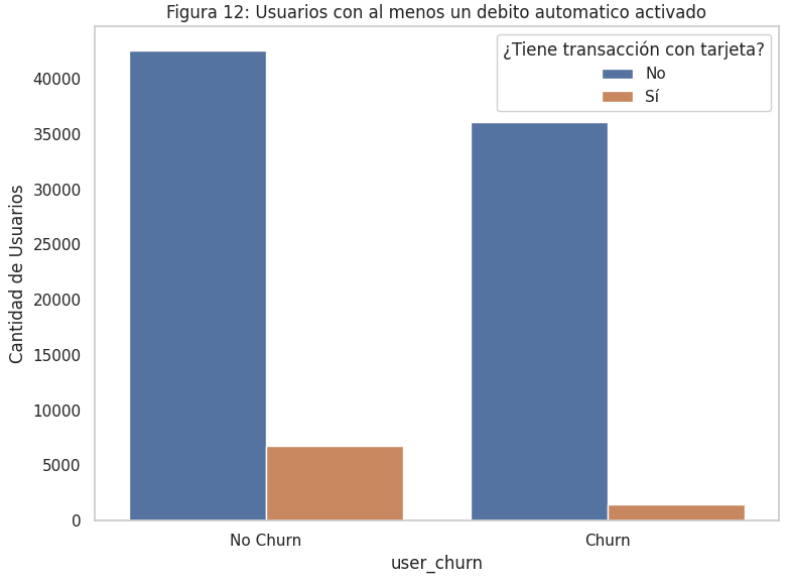
no hicieron *churn* muestran una mayor presencia en **supermercados (MCC 5411.0), plataformas de streaming (4899.0), transporte (4121.0)** y **estaciones de servicio (5541.0)**, con diferencias de entre **6 y 13 puntos porcentuales** respecto al grupo *churn*. Esto sugiere que el uso del producto para consumos cotidianos está asociado a una menor probabilidad de abandono.

Además, la variable merchant_Other, que agrupa transacciones fuera del top 20 de categorías más frecuentes, también presenta una diferencia notable: **59%** de los usuarios no *churn* transaccionaron en estas categorías, frente al **43%** de los *churners*. Esto podría indicar que un uso más **diverso o extendido** del producto contribuye a la retención.

A partir de este análisis, se decide **conservar las categorías con mayor poder discriminativo** y agrupar el resto bajo la categoría "Others", a fin de simplificar el modelo sin perder capacidad explicativa.

Por otro lado, una de las variables que puede considerarse indicativa de la fidelidad de los usuarios al servicio de tarjetas de débito es la presencia de débitos automáticos asociados. En la Figura 12 se observa que, si bien la mayoría de los usuarios no tienen activado este servicio, aquellos que sí lo tienen son, en su mayoría, usuarios que no realizaron *churn*: 6.771 usuarios, equivalentes a un 13,8% del total de usuarios activos, tienen al menos un débito automático. En contraste, solo 1.487 usuarios que hicieron *churn*, equivalentes a un 3,9% del total de *churners*, tenían débitos automáticos asociados.

Figura 12: Usuarios con al menos un debito automatico activado



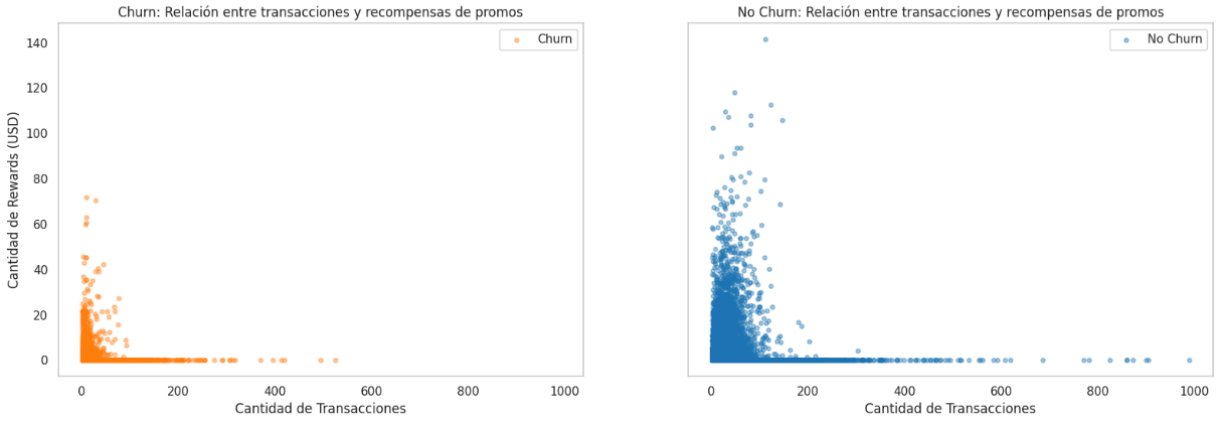
2.1.3 Variables transaccionales promociones

En lo que respecta a las variables asociadas a promociones, se observa que solo el **11%** de las transacciones realizadas por usuarios que **no hicieron *churn*** estuvieron vinculadas a algún tipo de promoción. En contraste, en el grupo ***churn***, esa proporción desciende al **4%**. Si bien ambos porcentajes son bajos, la diferencia es significativa y sugiere que el **acceso temprano a promociones** podría estar relacionado con una mayor retención del producto. Por este motivo, estas variables serán consideradas dentro del conjunto de *features* para el modelado.

La Figura 12 profundiza este análisis mostrando la relación entre la **cantidad de transacciones** y la **cantidad de recompensas obtenidas por promociones**, diferenciando entre ambos grupos. Se observa que, a igual número de transacciones, los usuarios que **no realizaron *churn*** tienden a recibir más recompensas. Además, este grupo muestra una mayor **dispersión vertical**, lo que indica un aprovechamiento más diverso y amplio de las promociones disponibles.

Por el contrario, en el grupo ***churn***, las recompensas son generalmente bajas y se concentran en usuarios con poca actividad transaccional. Esto podría reflejar una menor participación en promociones o una menor efectividad de las mismas en este segmento.

Figura 13: Relación entre Transacciones y Recompensas en los primeros 30 días

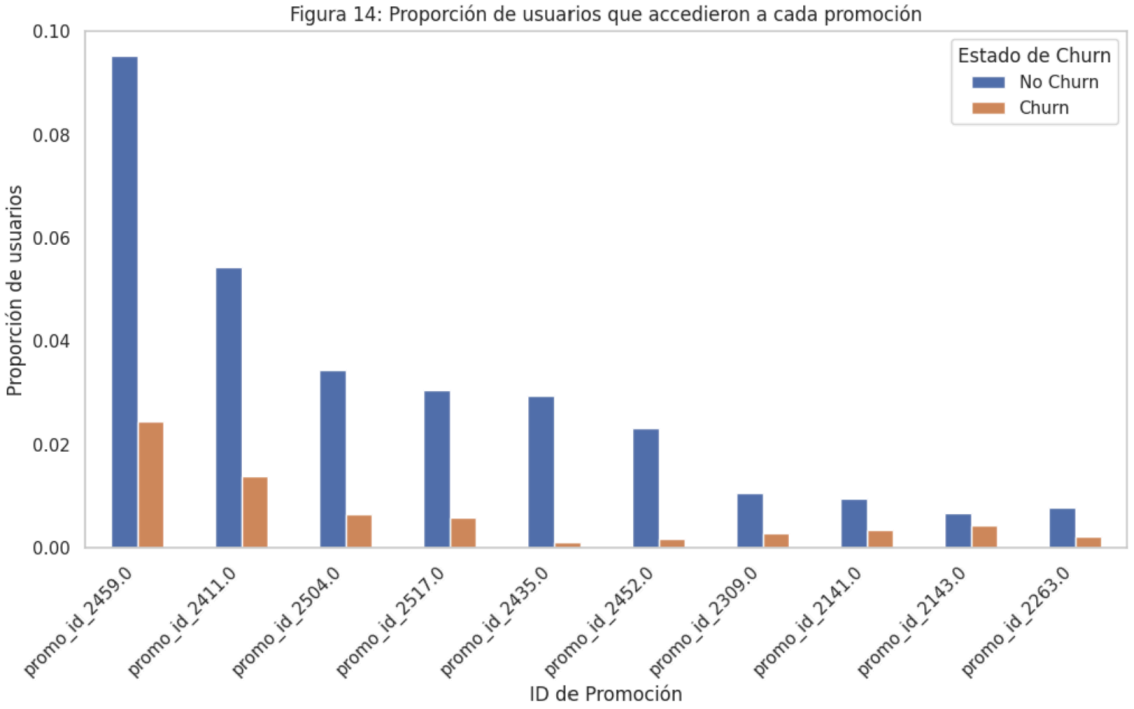


La Figura 14 muestra la proporción de usuarios que accedieron a las principales promociones durante sus primeros 30 días, diferenciando entre quienes realizaron *churn* y quienes no. Se identifican diferencias claras en el uso de ciertas promociones específicas: por ejemplo, la promo_id_2459.0 fue utilizada por el **10%** de los usuarios que permanecieron activos, frente a sólo el **2%** de los *churners*. Este patrón se repite en otras promociones como promo_id_2411.0, promo_id_2435.0 y promo_id_2504.0, que también muestran mayor participación entre usuarios que no abandonaron el producto.

En contraste, muchas otras promociones presentan una participación nula o marginal en ambos grupos, lo que sugiere una **baja relevancia práctica** o una **implementación limitada** durante el período analizado.

Dado este comportamiento, se propone **utililizar para el entrenamiento del del modelo únicamente aquellas promociones que superen cierto umbral mínimo de uso** —por ejemplo, más del 1% de participación en al menos uno de los grupos—. Las demás serán descartadas por su **baja representatividad** y escaso aporte potencial a la capacidad predictiva del modelo.

Figura 14: Proporción de usuarios que accedieron a cada promoción



2.1.4 Variables de tickets abiertos a atención al cliente

En relación con los tickets abiertos a atención al cliente vinculados al producto de tarjetas de débito, se observa que la **gran mayoría de los usuarios no realizaron ningún reclamo**, tanto en el grupo *churn* como en el no *churn*. Entre quienes **no hicieron *churn***, el número máximo de tickets abiertos fue de **14**, mientras que en el grupo *churn* fue de **12**.

Como se muestra en la siguiente tabla, aproximadamente el **97%** de los usuarios en ambos grupos **no abrieron ningún ticket** durante los primeros 30 días. Un **2%** abrió **un solo ticket**, y solo el **1% restante** registró **dos o más tickets**.

Tabla 1: Porcentaje de usuarios según grupo de tickets y *churn*

ticket_group	No Churn (%)	Churn (%)
0	96.80	96.90
1	1.70	1.80
2+	1.50	1.30

Entre quienes sí interactuaron con el soporte, el **tiempo promedio de resolución** muestra una distribución similar en ambos grupos: más del **60% de los casos** se resolvieron en **menos de un día**, y solo un pequeño porcentaje se extendió por **más de una semana**.

Tabla 2: Porcentaje de usuarios según tiempo de resolución de tickets y *churn*

res_time_group	No Churn (%)	Churn (%)
0 (sin tickets)	96.80	96.90
1. <=1 día	1.40	1.50
2. 1-3 días	0.20	0.20
3. 4-7 días	0.30	0.20
4. >7 días	1.30	1.20

Dado que no se identifican diferencias relevantes entre *churners* y no *churners* en estas variables, su **poder predictivo podría ser limitado, contrariamente a la hipótesis planteada respecto de usuarios *churners* y reclamos o tickets abiertos por una mala experiencia o deficiencia en el servicio**. Sin embargo, se recomienda conservarlas en el análisis preliminar para su **evaluación empírica posterior**, ya que podrían aportar valor en combinación con otras variables.

2.1.5 Variables relacionadas al programa de coins

El programa de **coins** tiene como objetivo fidelizar clientes e implica que, por cada transacción realizada por el usuario (no solo con el producto de tarjetas de débito), se acumulan cierta cantidad de *coins* que luego se pueden canjear por diferente tipos de premios.

En cuanto a las variables vinculadas a **coins**, se observan algunas diferencias entre usuarios *churn* y no *churn*. En promedio, los usuarios que **no realizaron *churn*** generaron **3.868 coins** durante los primeros 30 días de uso del producto, mientras que los que **sí realizaron *churn*** generaron **2.375 coins** en el mismo período.

Respecto a los **coins canjeados**, la media es de **2.538** para los usuarios que permanecieron activos y de **2.459** para quienes abandonaron el producto. Si bien la diferencia en esta última

variable es menos marcada, los usuarios no *churn* también muestran un nivel ligeramente superior de canje.

Es importante destacar que los coins canjeados durante los primeros 30 días **podrían haberse acumulado antes** del uso activo de la tarjeta de débito, ya que forman parte del balance general del usuario. Por lo tanto, estas variables deben analizarse de forma separada.

Aunque las diferencias no son sustanciales, el hecho de que los usuarios que permanecen activos **generen y canjeen más coins** sugiere un mayor nivel de interacción con el ecosistema de la plataforma. Por eso, estas variables serán consideradas en el modelado, ya que podrían aportar valor predictivo.

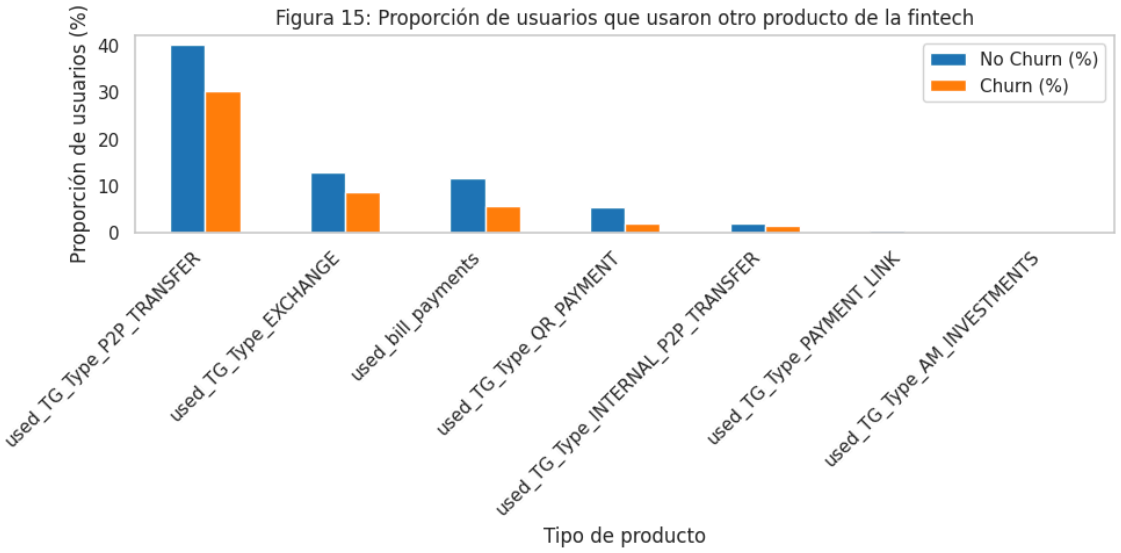
2.1.6 Variables relacionadas a otros productos de la fintech

La Figura 15 muestra la proporción de usuarios que realizaron al menos una transacción utilizando **productos alternativos** a la tarjeta de débito, desagregados por clase (*Churn* vs. No *Churn*). Se observa que los usuarios que **no realizaron *churn*** presentan una mayor adopción de estos productos, lo que sugiere un mayor grado de vinculación con el ecosistema de la fintech.

En particular, las **transferencias P2P** son las más utilizadas, con un **40% de adopción** en el grupo No *Churn* frente a un **30%** en el grupo *Churn*. Otras funcionalidades con diferencias relevantes son el **exchange de divisas**, el **pago de facturas** y los **pagos con QR**, todas con mayores tasas de uso entre los usuarios que permanecieron activos.

Este patrón indica que un uso más amplio de los servicios ofrecidos por la plataforma podría estar asociado a una **menor probabilidad de abandono**. En función de estos resultados, se recomienda **incluir en el modelo de predicción únicamente las variables** P2P_TRANSFER, EXCHANGE, BILL_PAYMENTS y QR_PAYMENT, por ser las de **mayor adopción** y con **diferencias más marcadas** entre los grupos.

Figura 15: Proporción de usuarios que usaron otro producto de la fintech



2.2 Preprocesamiento e ingeniería de datos

Como ya se mencionó anteriormente, el preprocesamiento de datos se integró principalmente durante la etapa de construcción de la base final. En ese momento, al realizar la agregación de las bases transaccionales por usuario, se generaron las variables dummy mediante one-hot-encoding. De esta manera, se transformaron las variables categóricas en variables de conteo, lo que permitió capturar información relevante sobre el comportamiento de cada usuario.

2.2.1 Tratamiento de valores nulos

La imputación de valores faltantes es un paso clave en la preparación de datos para modelos de machine learning, ya que la mayoría de los algoritmos (como regresión logística, SVM, XGBoost o árboles de decisión) requieren datasets completos para entrenar de forma efectiva. La presencia de nulos puede afectar el rendimiento, introducir sesgos o reducir la estabilidad del modelo.

En este caso, al tratarse de una base construida a partir de datos transaccionales, la cantidad de valores faltantes es relativamente baja. En general, los nulos se explican porque ciertos usuarios no aparecen en las tablas de promos, Kustomer o Coins —es decir, no realizaron transacciones asociadas a esas categorías.

Variables de promociones:

Las columnas asociadas a promociones tienen una gran proporción de valores nulos. De los 86.915 usuarios, 72.170 no figuran en la base de datos de promociones. Estos casos se imputaron con 0, dado que representan usuarios que no participaron en ninguna promoción

durante el período analizado. De estos, 31.997 corresponden al primer semestre y 40.153 al segundo semestre.

Variables de Kustomer support:

Lo mismo aplica para las columnas total_tickets, resolution_total_time y resolution_avg_time. Solo una minoría de usuarios abrió tickets vinculados al producto debit cards. En total, 33.843 usuarios con nulos pertenecen al primer semestre y 50.323 al segundo. También se imputaron con 0, por representar ausencia de interacción.

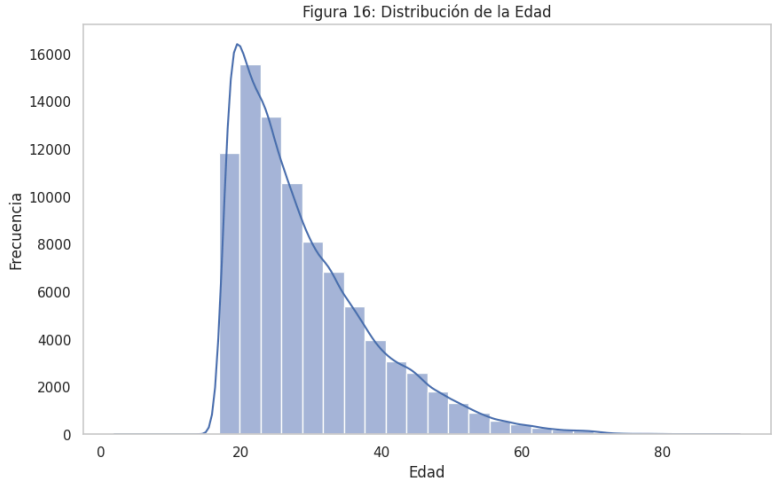
Variables de Coins:

En este caso, los valores faltantes son pocos: solo 4.923 usuarios no se encontraron en la base de transacciones de coins. Estos valores se imputaron con 0, ya que reflejan usuarios sin movimientos de este tipo en los primeros 30 días.

Variable edad:

Para 105 usuarios no se registró fecha de nacimiento, por lo tanto, no se pudo calcular la edad. Se analizó la distribución de esta variable mediante un histograma (Figura 16), el cual mostró una distribución asimétrica con sesgo positivo, concentrando la mayoría de los usuarios en torno a los 20 años. La media fue de 28,93 años y la mediana, de 26.

Figura 16: Distribución de la Edad



Dado el sesgo de la distribución y la presencia de valores extremos, se imputaron los valores faltantes con la mediana (26 años), por ser una medida más robusta frente a *outliers*.

2.2.2 Tratamiento de *outliers*

Durante el proceso de preprocesamiento de los datos, se implementó un tratamiento específico para los valores atípicos (*outliers*) presentes en las variables numéricas. El objetivo fue reducir el riesgo de que estos valores extremos distorsionen el comportamiento de los modelos de machine learning, particularmente aquellos sensibles a la escala de los datos, como la regresión logística.

El enfoque consistió en una combinación de detección selectiva de *outliers* y la aplicación de un escalado robusto basado en el rango intercuartílico (IQR). Este procedimiento se llevó a cabo en varias etapas.

En primer lugar, se identificaron todas las variables numéricas continuas del conjunto de datos. Para ello, se excluyeron tanto las variables binarias, como la variable objetivo user_churn y el identificador único de los users: user_id. Las variables binarias fueron definidas como aquellas que solo toman los valores 0 y 1, ya que estas no requieren tratamiento adicional en términos de escalado o detección de *outliers*.

Una vez determinadas las variables numéricas relevantes, se evaluó en cada una de ellas la presencia de *outliers* utilizando el criterio clásico propuesto por Tukey (1977), basado en el rango intercuartílico. Según esta metodología, un valor se considera un outlier si se encuentra fuera del rango comprendido entre: $Q1 - 1.5 * IQR$ y $Q3 + 1.5 * IQR$, donde $Q1$ y $Q3$ representan el primer y tercer cuartil, respectivamente, e IQR es la diferencia entre ambos. Se determinó que una variable debía ser tratada si más del 1% de sus observaciones se encontraban fuera de este rango.

Para las variables que presentaron una proporción significativa de valores atípicos, se aplicó una transformación mediante escalado robusto utilizando la función robust_scale de la librería sklearn.preprocessing. Esta transformación consiste en restar la mediana de la variable y dividir por su rango intercuartílico, es decir:

$$x_i^{scaled} = \frac{xi - mediana(X)}{IQR(X)} \quad (1)$$

Por último, se tomó la precaución de no modificar ciertas variables clave durante este proceso. La variable objetivo user_churn se mantuvo intacta para no alterar la naturaleza del problema de clasificación. Lo mismo se aplicó al identificador user_id y a todas las variables binarias, que

conservaron su codificación original en {0, 1}. Esto garantizó la interpretabilidad del modelo y evitó introducir distorsiones innecesarias.

Este enfoque metodológico permitió conservar la representatividad poblacional del conjunto de datos, ya que no se eliminaron observaciones, al mismo tiempo que se controló el efecto de los valores extremos sobre el entrenamiento de los modelos.

2.2.3 Transformación y creación de nuevas variables

En el conjunto de datos original, los campos relacionados con fechas estaban almacenados como variables de tipo object. Como paso inicial del preprocesamiento, estos campos fueron convertidos al tipo datetime (marca temporal), lo que permitió operar correctamente sobre ellos para la creación de variables derivadas. Asimismo, dado que las variables de tipo datetime no pueden ser utilizadas directamente en algunos modelos, se optó por derivar dos nuevas variables llamadas mes_creacion y days_between_user_and_card_creation. Estas variables representan el mes calendario en que el usuario creó su primera tarjeta y los días transcurridos entre la creación del user en la fintech y la creación de su primer tarjeta, lo cual permite capturar posibles patrones estacionales.

En relación con la edad del usuario, el dataset no contenía esta información de forma explícita. Sin embargo, se disponía de la fecha de nacimiento, por lo que se construyó una nueva variable denominada edad. Esta se calculó como la diferencia entre la fecha de creación de la primera tarjeta de débito y la fecha de nacimiento del usuario, lo que permitió obtener la edad que tenía el usuario al momento de solicitar su primera tarjeta.

Siguiendo un enfoque similar, se generó otra variable temporal que mide el tiempo de permanencia del usuario dentro de la plataforma antes de obtener su primera tarjeta de débito. Para calcular esta variable, se tomó la diferencia en días entre la fecha de creación del usuario en el sistema y la fecha de emisión de su primera tarjeta. Esta variable aporta información relevante sobre la trayectoria del usuario dentro del ecosistema de la Fintech antes de realizar una acción clave cómo solicitar una tarjeta de débito.

Por otro lado, considerando que las promociones se activan cuando el usuario realiza una transacción, se creó la variable *promo_ratio*, que representa la cantidad de promociones activadas sobre el total de transacciones realizadas. Esto permite obtener una mejor visión sobre el impacto de las promociones en la retención de usuarios. Además, esta variable puede servir para combinar ambas métricas en caso de detectarse multicolinealidad. El cálculo se

realiza dividiendo la cantidad de promociones recibidas por el usuario sobre la cantidad de transacciones realizadas por él.

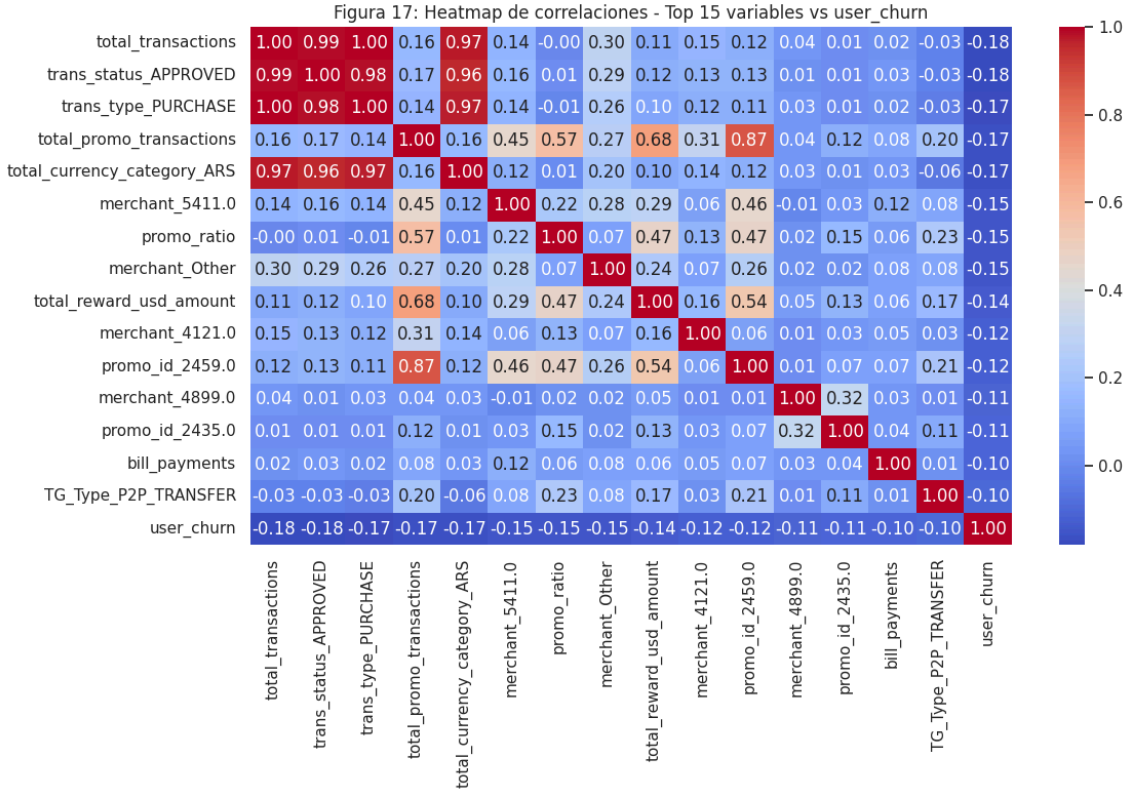
2.3 Selección de variables

Con el objetivo de identificar relaciones entre las variables disponibles y la variable objetivo user_churn, se realizó un **análisis de correlación de Pearson** sobre todas las variables numéricas. En la Figura 17 se presentan las **15 variables con mayor correlación (en valor absoluto)** con user_churn.

Si bien **ninguna variable muestra una correlación fuerte**, se destacan algunas relaciones **débiles pero consistentes**, con coeficientes cercanos a **-0,18**. Entre las más asociadas a **menor probabilidad de *churn*** se encuentran total_transactions, trans_type_PURCHASE, trans_status_APPROVED y total_promo_transactions. Todas ellas presentan correlaciones negativas, lo que sugiere que **una mayor actividad transaccional o una mayor exposición a promociones** se relaciona con una mayor retención.

Por otro lado, se observa una **alta correlación entre algunas variables entre sí**, especialmente entre total_transactions, trans_type_PURCHASE y trans_status_APPROVED, con valores cercanos a **1**. Esto indica la presencia de **multicolinealidad**, un fenómeno que puede afectar negativamente a ciertos modelos predictivos, en particular a los modelos lineales o árboles menos robustos frente a variables redundantes.

Figura 17: Heatmap de correlaciones - Top 15 variables vs user_churn



En resumen, el análisis sugiere que, aunque el poder predictivo individual de las variables es limitado, algunas —principalmente las vinculadas a **actividad transaccional** y **promociones**— presentan señales relevantes. A su vez, la **redundancia entre variables altamente correlacionadas** deberá ser tratada con cuidado para evitar impactos negativos en el rendimiento del modelo.

Como próximos pasos, se propone **Reducir la multicolinealidad**, conservando sólo aquellas variables que aporten información diferencial y **seleccionar un conjunto reducido de variables relevantes**, priorizando interpretabilidad y capacidad explicativa.

2.3.1 Selección de variables por multicolinealidad

La presencia de variables altamente correlacionadas puede afectar negativamente el rendimiento de ciertos modelos predictivos, especialmente aquellos sensibles a la redundancia de información.

Para detectar posibles casos de colinealidad severa, se calculó la matriz de correlación de Pearson entre todas las variables numéricas del conjunto de datos. Se identificaron como candidatos a eliminación todos los pares de variables cuya correlación absoluta superaba el umbral de 0.90, valor comúnmente utilizado en la literatura como indicativo de colinealidad elevada.

En cada par de variables altamente correlacionadas, se conservó aquella con mayor correlación absoluta con la variable objetivo user_churn, y se eliminó la otra por considerarse redundante y de menor valor explicativo para el problema de predicción.

Como resultado de este proceso, se eliminaron las siguientes variables:

- 'trans_type_PURCHASE', 'trans_status_APPROVED' y 'total_currency_category_ARS', todas altamente correlacionadas con 'total_transactions', que fue la variable conservada por su mayor asociación con la variable objetivo.
- 'total_transaction_origin_amount', eliminada por su alta correlación con 'total_transaction_accounting_usd_amount'.

Este procedimiento permitió reducir la redundancia entre variables sin comprometer la capacidad explicativa del conjunto de datos, mejorando así la estabilidad y eficiencia de los modelos entrenados posteriormente.

2.3.2 Selección de variables por baja varianza

En esta etapa se aplicó un filtro de **baja varianza** con el objetivo de eliminar aquellas variables que presentaban valores prácticamente constantes a lo largo de la muestra. Este tipo de variables, al no mostrar variación significativa entre los casos, **aportan escasa capacidad discriminativa** y suelen tener un impacto nulo o irrelevante en los modelos predictivos.

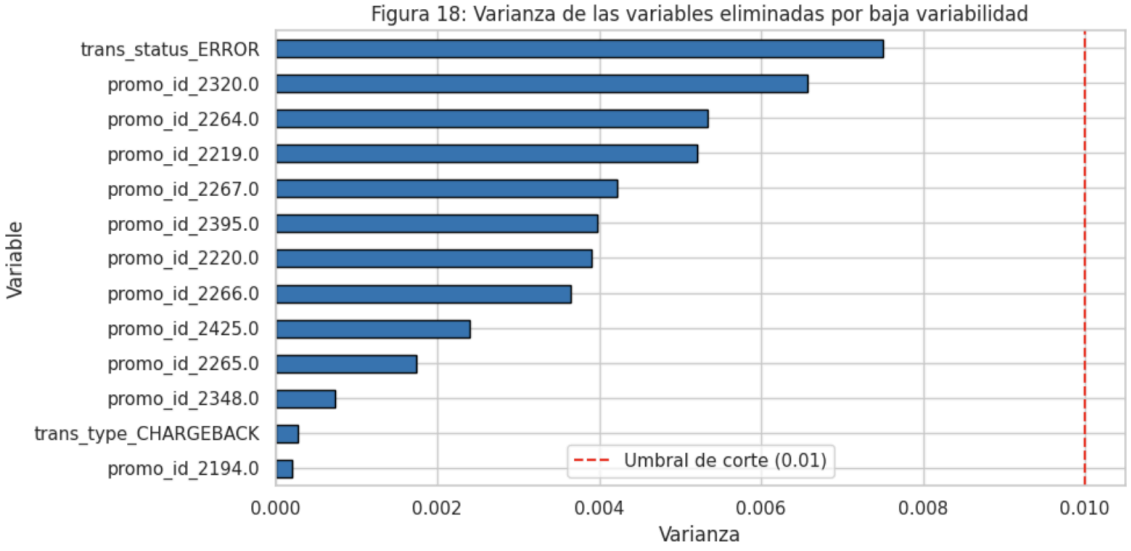
Para asegurar un tratamiento adecuado, se excluyeron previamente las **variables binarias (dummies)** del análisis, ya que su naturaleza dicotómica puede llevar a varianzas naturalmente bajas sin que esto implique falta de valor predictivo. De esta forma, el filtro de varianza se aplicó exclusivamente sobre variables numéricas continuas (tipos float o int).

Se utilizó un umbral de **0.01**, criterio comúnmente adoptado en la literatura para identificar variables con muy poca variabilidad. Las variables que presentaron una varianza inferior a dicho valor fueron eliminadas del conjunto de datos.

Como resultado, se descartaron **13 variables**. La mayoría de ellas correspondían a identificadores específicos de promociones con presencia marginal en la muestra, así como a ciertos tipos de transacciones extremadamente infrecuentes, como trans_type_CHARGEBACK o trans_status_ERROR. En la Figura 18 se presentan las varianzas de las 13 variables que fueron eliminadas. Tal como se observa, todas se encuentran significativamente por debajo del umbral establecido de 0.01 (indicado con la línea roja punteada), lo que justifica su exclusión del modelo.

Estas variables corresponden en su mayoría a promociones puntuales o tipos de transacción extremadamente infrecuentes, como trans_status_ERROR o trans_type_CHARGEBACK, cuya baja presencia en la muestra impide que aporten valor discriminativo a la predicción del *churn*. En contextos donde la varianza es tan reducida, la inclusión de estas variables no solo es innecesaria, sino que puede introducir ruido en el modelo y dificultar su interpretabilidad.

Figura 18: Varianza de las variables eliminadas por baja variabilidad



2.3.3 Selección de variables mediante la aplicación de LASSO (regresión logística con penalización L1)

LASSO (*Least Absolute Shrinkage and Selection Operator*) es una técnica de regularización que busca ajustar un modelo predictivo penalizando la complejidad del mismo. A diferencia de una regresión logística tradicional, que estima los coeficientes βj minimizando únicamente el error de predicción, LASSO incorpora una penalización sobre la magnitud de dichos coeficientes. Esta penalización se ve reflejada en el hiperparámetro lambda:

$$\sum_{i=1}^{n}\left(y_i-\hat{y}_i\right)^2+\lambda\sum_{j=1}^{k}\left|\beta_j\right| \qquad (2)$$

Dado que LASSO utiliza una penalización sobre la magnitud de los coeficientes, es fundamental que todas las variables estén en una **misma escala** para evitar que el modelo favorezca o penalice injustamente a aquellas con unidades numéricas mayores. Por este motivo, antes de aplicar el algoritmo, se realizó una **estandarización de todas las variables numéricas**. Este

procedimiento consiste en transformar cada variable para que tenga **media cero y desviación estándar uno**, asegurando que la penalización actúe de forma equitativa sobre todas las características.

Como resultado del proceso de selección automática de variables mediante regresión logística penalizada con LASSO, se descartaron 10 variables cuyo coeficiente resultó igual a cero. Estas variables fueron consideradas por el modelo como irrelevantes o redundantes una vez ajustado por el resto de las características disponibles. La mayoría corresponde a tipos de transacciones poco frecuentes o promociones con escasa incidencia en la muestra, lo que refuerza la idea de que su aporte al modelo predictivo es limitado. Este procedimiento permitió reducir aún más la dimensionalidad del conjunto de datos, conservando únicamente las variables con mayor capacidad explicativa sobre la probabilidad de *churn*.

Un caso particular fue la variable merchant_5816.0, que si bien presentaba una alta frecuencia en la muestra, fue descartada por el modelo LASSO. El análisis de correlación evidenció una relación muy fuerte con total_transactions (coeficiente de 0.87), lo que sugiere que su efecto ya se encuentra explicado por esta última. Además, su correlación con la variable objetivo user_churn fue prácticamente nula (-0.06), lo cual refuerza la decisión del algoritmo de no incluirla en el modelo. Este tipo de situaciones destaca la capacidad de LASSO para identificar y eliminar variables redundantes o poco informativas, incluso cuando su presencia en los datos es elevada.

Para facilitar la trazabilidad del proceso y ofrecer una visión integral de la selección de variables, en el Apéndice B se incluye una tabla resumen con todas las variables evaluadas, indicando su inclusión o descarte según cada uno de los tres métodos aplicados: correlación, varianza y LASSO.

3. Metodología

En esta sección se detallan los procedimientos implementados para la evaluación, comparación y selección de modelos de machine learning aplicados a un problema de clasificación binaria, cuyo objetivo es predecir el abandono (*churn*) de usuarios.

A partir del conjunto de datos ya preprocesado, se definió una estrategia de validación respetando la dimensión temporal de los datos. El primer semestre del período analizado se destinó al desarrollo del modelo y el correspondiente ajuste de hiper parámetros, aplicando

validación cruzada con un 80% de los datos para el conjunto de entrenamiento (train) y un 20% para el conjunto de prueba (test).

Posteriormente, se utilizó un nuevo set de datos correspondiente al segundo semestre como conjunto de evaluación *out-of-sample*, lo que permitió validar el desempeño del modelo sobre una muestra completamente nueva y cronológicamente posterior, en línea con el comportamiento real de los usuarios (James, Witten, Hastie, & Tibshirani, 2013).

Este enfoque tiene como objetivo preservar la estructura temporal de los datos, asegurando que el modelo aprenda únicamente sobre información disponible previa al momento de la predicción. Esta condición es fundamental para replicar condiciones reales de aplicación del modelo y evitar distorsiones que puedan surgir del entrenamiento con datos que, en la práctica, aún no habrían ocurrido.

En este sentido, se contempló especialmente el riesgo de *data leakage*, entendiendo que la inclusión de variables con información futura puede llevar a una sobrestimación artificial del desempeño del modelo (Kaufman et al., 2012).

Con el fin de identificar el modelo con mejor rendimiento, se entrenaron y evaluaron múltiples algoritmos de clasificación supervisada, considerando distintas configuraciones y combinaciones de parámetros. Para ello, se aplicó un proceso de optimización de hiperparámetros mediante Random Search, con el objetivo de maximizar el desempeño predictivo y garantizar el poder de generalización de cada modelo.

La selección del modelo final se basó en una comparación sistemática utilizando métricas como exactitud, precisión, recall, F1-score y el AUC-ROC, que permiten evaluar distintos aspectos del desempeño en tareas de clasificación. Estas métricas serán detalladas en la sección de resultados (4. Resultados).

3.1 Modelos

3.1.1 Regresión logística

La regresión logística es un modelo estadístico utilizado para resolver problemas de clasificación binaria, en los que la variable dependiente puede tomar únicamente dos posibles valores, típicamente codificados como 0 y 1. Este modelo permite estimar la probabilidad de que ocurra un evento en función de un conjunto de variables predictoras. A diferencia de la regresión lineal, que puede producir predicciones fuera del rango [0,1], la regresión logística emplea una transformación logística para asegurar que las probabilidades predichas se

mantengan en un rango válido. La forma funcional del modelo se expresa mediante la función sigmoide:

$$P(x) = \frac{1}{1+e^{-(\beta_0+\beta_1 x_1+\beta_2 x_2+\cdots+\beta_n x_n)}} \quad (3)$$

Donde:

- $P(x)$ es la probabilidad de que la variable dependiente y valga 1, es decir la probabilidad de que el evento ocurra, dado el vector de variables explicativas x.
- $\left(x_1, x_2, \ldots, x_n\right)$ es el conjunto de variables independientes o características predictoras del modelo.
- β_0 es el intercepto o constante del modelo y representa el valor base del log-odds cuando todas las variables independientes son iguales a 0.
- $\beta_1, \beta_2, \ldots, \beta_n$ son los coeficientes del modelo, que miden el efecto de cada variable independiente sobre el log-odds del evento:
    - Si $\beta_i > 0$, un aumento en x_i aumenta la probabilidad del evento, ceteris paribus.
    - Si $\beta_i < 0$, un aumento en x_i disminuye la probabilidad del evento, ceteris paribus.

En la regresión logística, los coeficientes del modelo se estiman mediante el criterio de máxima verosimilitud. Esto significa que el modelo elige aquellos valores para los parámetros que hacen que las predicciones se parezcan lo más posible a lo que realmente ocurrió. En otras palabras, busca los valores que mejor explican los datos observados, maximizando la coincidencia entre lo que el modelo predice y lo que efectivamente pasó.

Este modelo se destaca por su capacidad de estimar directamente la probabilidad de ocurrencia de un evento y por ofrecer una interpretación sencilla y clara de sus coeficientes, los cuales permiten analizar el efecto de cada variable independiente sobre el log-odds del evento. El modelo también es flexible, ya que no exige que las variables predictoras sigan una distribución normal, y puede adaptarse a relaciones no lineales mediante la inclusión de transformaciones adecuadas.

No obstante, para que las estimaciones obtenidas sean válidas e interpretables, la regresión logística se basa en ciertos supuestos fundamentales. Entre ellos, se requiere que la variable dependiente sea binaria y que las observaciones sean independientes entre sí. Además, se asume una relación lineal entre las variables independientes y el logit de la probabilidad del evento, así como la ausencia de multicolinealidad severa entre los predictores. El cumplimiento de estos supuestos permite garantizar la estabilidad de las estimaciones y la adecuada interpretación de los coeficientes (James, Witten, Hastie, & Tibshirani, 2021).

En el contexto de esta tesis, la regresión logística fue considerada como punto de partida por su simplicidad y alta interpretabilidad. Esta característica resulta valiosa para el negocio, ya que permite comprender el efecto individual de cada variable sobre la probabilidad de *churn* y facilita la toma de decisiones basadas en evidencia. Además, en contextos con una dimensión de datos moderada, como el presente caso, este modelo puede alcanzar desempeños comparables al de algoritmos más complejos, manteniendo una buena interpretabilidad y menor costo computacional.

3.1.2 Árboles de decisión

Los árboles de decisión son un modelo de aprendizaje supervisado que puede utilizarse tanto para problemas de clasificación como de regresión. En el caso de una tarea de clasificación binaria, el objetivo es predecir la clase de una observación dividiendo el espacio de atributos en regiones definidas por reglas sobre las variables explicativas. A diferencia de la regresión logística, este modelo no asume una relación funcional específica entre las variables independientes y la variable de salida, lo que lo hace más flexible para capturar relaciones complejas (James, Witten, Hastie, & Tibshirani, 2021).

La construcción de un árbol de decisión se basa en un proceso llamado partición recursiva binaria, mediante el cual se divide el espacio de atributos en subregiones cada vez más homogéneas. En cada paso, el algoritmo selecciona la variable y el punto de corte que mejor separen las clases, según algún criterio de impureza como el índice Gini o la Entropía. Este proceso continúa de forma sucesiva hasta alcanzar un criterio de detención, como una profundidad máxima o una mínima cantidad de observaciones por nodo (Tan, Steinbach, & Kumar, 2019). Una vez construido el árbol, se predice la clase de una observación en función de la región del espacio de atributos en la que cae. La probabilidad condicional de que una observación x_i pertenezca a una clase C_k, se estima como la proporción de observaciones de la clase k dentro del nodo correspondiente. Esta probabilidad se calcula como:

$$P(x_i) = \frac{n_k}{n} \tag{4}$$

donde n_k representa la cantidad de observaciones de la clase k en el nodo donde cae x_i y n es el total de observaciones en ese nodo.

Tanto el índice de Gini como la entropía son medidas que permiten cuantificar la impureza de un nodo, es decir, cuán mezcladas están las clases dentro de él. El índice de Gini se define como:

$$Gini(t) = 1 - \sum_{k=1}^{K} p_k^2 \tag{5}$$

donde p_k representa la proporción de observaciones de la clase k en el nodo t. Esta medida toma valores cercanos a cero cuando las observaciones pertenecen mayoritariamente a una sola clase.

Por su parte, la entropía se expresa como:

$$Entropy(t) = -\sum_{k=1}^{K} p_k(p_k) \tag{6}$$

Al igual que el índice de Gini, su valor es mínimo cuando el nodo es puro y máximo cuando las clases están equilibradas. En ambos casos, el algoritmo selecciona la división que genere la mayor reducción en la impureza, lo que permite mejorar la capacidad del modelo para discriminar entre clases.

En este caso, se utilizó el índice de Gini como criterio de división para la construcción del árbol, dado que es el valor por defecto en la implementación de DecisionTreeClassifier de *scikit-learn*

y ofrece resultados muy similares a los obtenidos con Entropía. Además, presenta una ventaja computacional al ser más eficiente, ya que no requiere el cálculo de logaritmos (Tan et al., 2019).

Una de las principales ventajas de los árboles de decisión es su interpretabilidad, ya que permiten representar de forma explícita las reglas de decisión que conducen a una determinada predicción. Además, el modelo es capaz de capturar relaciones no lineales e interacciones entre variables sin necesidad de transformaciones previas, y puede trabajar con variables numéricas y categóricas sin requerir estandarización.

A diferencia de otros modelos, no impone supuestos fuertes sobre la distribución de los datos, ni requiere que exista una relación funcional específica entre los atributos y la variable objetivo. Sin embargo, los árboles presentan un riesgo elevado de sobreajuste si no se controla adecuadamente su complejidad.

Para mitigar este riesgo, es común aplicar estrategias tales como establecer una profundidad máxima (max_depth), definir un número mínimo de observaciones requerido para realizar una división (min_samples_split), o fijar un mínimo de observaciones que deben tener las hojas terminales (min_samples_leaf) (James et al., 2021; Tan et al., 2019).

Dado que el comportamiento de los usuarios puede estar influido por múltiples factores que interactúan de manera no lineal, los árboles de decisión permiten capturar reglas específicas de abandono sin necesidad de suposiciones funcionales, como sí ocurre en el caso de la regresión logística. Su estructura visual también aporta claridad a la hora de comunicar patrones relevantes al equipo de negocio.

3.1.3 Random Forest

El modelo de Random Forest extiende la lógica de los árboles de decisión individuales mediante un enfoque de ensamblado conocido como bagging (bootstrap aggregating). Esta técnica consiste en generar múltiples subconjuntos del conjunto de entrenamiento mediante muestreo aleatorio con reemplazo (*bootstrap*), y entrenar un árbol de decisión independiente sobre cada uno de ellos. Además, en cada división dentro de un árbol, el algoritmo selecciona un subconjunto aleatorio de variables predictoras en lugar de evaluar todas, lo que introduce una mayor diversidad entre los árboles.

En tareas de clasificación, cada árbol produce una predicción de clase, y el modelo final se construye mediante **voto mayoritario**, es decir, asignando a cada observación la clase que haya sido más veces predicha por los árboles del conjunto. Esta estrategia permite **reducir la**

varianza del modelo sin incrementar el sesgo, mejorando así su capacidad de generalización respecto a un único árbol (Breiman, 2001).

Una de las principales ventajas del modelo Random Forest es su alta capacidad predictiva, incluso en contextos con muchas variables y relaciones no lineales. Gracias al mecanismo de agregación, el modelo es menos propenso al sobreajuste que un árbol individual, y presenta buena robustez frente al ruido y a la inclusión de variables irrelevantes. Otra ventaja es que puede aplicarse sin necesidad de escalar los datos y que maneja sin inconvenientes tanto variables numéricas como categóricas.

Sin embargo, este tipo de modelos también presenta ciertas limitaciones. En comparación con un árbol de decisión único, Random Forest es menos interpretable, ya que las predicciones resultan del consenso entre múltiples árboles. Por otro lado, a medida que se incrementa la cantidad de árboles, el modelo puede volverse más costoso computacionalmente, especialmente en conjuntos de datos de gran tamaño.

Para mitigar el riesgo de sobreajuste y mejorar la capacidad de generalización, el modelo permite ajustar diversos hiperparámetros. Entre ellos, la cantidad de árboles utilizados en el bosque, la profundidad máxima permitida para cada árbol, la cantidad mínima de observaciones necesarias para dividir un nodo o para formar una hoja, y la proporción de variables consideradas en cada partición.

Random Forest fue incluido por su capacidad de mejorar la precisión sin perder robustez, al combinar múltiples árboles y reducir el riesgo de sobreajuste. Este modelo resulta especialmente adecuado para problemas como el presente, donde se busca modelar y predecir el comportamiento humano —en este caso, el abandono temprano de usuarios— a partir de múltiples variables que pueden interactuar de forma compleja y no lineal. Además, su tolerancia al ruido y su capacidad para detectar relaciones sutiles entre variables, sin perder capacidad de generalización, lo convierten en una herramienta eficaz para anticipar decisiones de los usuarios.

3.1.4 XGBoost

XGBoost (Extreme Gradient Boosting) es un algoritmo de aprendizaje supervisado basado en árboles de decisión, diseñado específicamente para optimizar el rendimiento predictivo mediante un enfoque de ensamblado conocido como boosting por gradiente. A diferencia de Random Forest, que construye múltiples árboles en paralelo y combina sus predicciones,

XGBoost lo hace de forma secuencial, agregando nuevos árboles que corrigen los errores cometidos por los anteriores (Chen & Guestrin, 2016).

El modelo se construye como una suma de funciones aditivas, donde cada función corresponde a un árbol de decisión. La predicción para una observación x_i se expresa como:

$$\hat{y}_i = \phi(x_i) = \sum_{k=1}^{K} f_k(x_i), \quad f_k \in F \tag{7}$$

Donde $\hat{y}_i$ representa la predicción del modelo para la observación x_i, K es la cantidad total de árboles, y cada f_k es una función que pertenece al conjunto F. De esta manera, el modelo combina múltiples árboles, cada uno ajustado sobre los errores residuales de las predicciones anteriores, logrando una mejora progresiva en el ajuste del conjunto.

El entrenamiento de XGBoost se basa en la minimización de una función objetivo regularizada, compuesta por una función de pérdida que mide el error de predicción y un término de regularización que penaliza la complejidad de los árboles. En tareas de clasificación binaria, la función de pérdida utilizada es la log-loss o binary cross-entropy, la cual penaliza con mayor intensidad aquellas predicciones que asignan una alta probabilidad a una clase incorrecta. Esta combinación permite controlar tanto el ajuste del modelo como su capacidad de generalización. A su vez, XGBoost ajusta la contribución de cada nuevo árbol mediante un parámetro de tasa de aprendizaje (*learning rate*), lo que reduce su impacto progresivamente en la predicción final y ayuda a mitigar el sobreajuste.

XGBoost se destaca por su rendimiento predictivo, sobre todo en conjuntos de datos estructurados. Su capacidad para corregir errores de forma progresiva y sus mecanismos de regularización lo hacen muy efectivo para evitar el sobreajuste. Además, es un modelo eficiente en términos de tiempo de entrenamiento, puede manejar datos faltantes y permite conocer la importancia de las variables utilizadas.

Entre sus desventajas, puede requerir más tiempo de ajuste que otros modelos debido a la cantidad de hiperparámetros que ofrece. También es menos interpretable que un árbol de decisión simple, ya que sus predicciones se basan en la combinación de muchos árboles.

Para regularizar el modelo y evitar el sobreajuste, los hiperparámetros más importantes son la tasa de aprendizaje (*learning_rate*), que reduce el peso de cada árbol nuevo, la profundidad máxima de los árboles (*max_depth*), el número mínimo de observaciones requeridas para dividir un nodo (*min_child_weight*), y la cantidad de árboles a entrenar (*n_estimators*). También es útil ajustar la proporción de datos y variables usados en cada iteración (*subsample* y *colsample_bytree*), lo que agrega aleatoriedad y mejora la generalización.

XGBoost fue incorporado al análisis por su gran desempeño en tareas de clasificación. Al construir árboles de forma secuencial y corregir los errores de los modelos anteriores, este algoritmo resulta especialmente útil para capturar patrones complejos y relaciones no lineales que pueden surgir en el comportamiento humano, como el abandono temprano de un servicio. Su capacidad para aplicar regularización y controlar el sobreajuste lo convierte en una alternativa robusta frente a modelos más simples, especialmente cuando se busca maximizar la precisión sin comprometer la generalización. Si bien requiere un ajuste más fino de hiperparámetros y es menos interpretable, su inclusión se justifica por el potencial de mejorar la capacidad predictiva del modelo en este caso.

3.2 Selección de hiperparámetros

La elección adecuada de hiperparámetros es una etapa muy importante en este tipo de estudios, especialmente para mejorar el rendimiento y la capacidad de generalización de los modelos de machine learning. Dos de los enfoques más utilizados para este fin son Grid Search y Random Search.

Grid Search consiste en evaluar de forma exhaustiva todas las combinaciones posibles dentro de un conjunto predefinido de valores, mientras que Random Search selecciona aleatoriamente combinaciones dentro de ese espacio, lo que permite explorar más configuraciones cuando los recursos computacionales son limitados (Bergstra & Bengio, 2012).

Como paso previo a la búsqueda de hiperparámetros, se definieron manualmente algunos valores comúnmente utilizados, con el objetivo de evitar el sobreajuste y obtener métricas iniciales razonables. Esta instancia permitió identificar configuraciones iniciales para utilizar como punto de partida.

En este caso, se optó por emplear Random Search, ya que ofrece un equilibrio eficaz entre eficiencia computacional y capacidad exploratoria. Esta técnica ha demostrado ser particularmente efectiva cuando solo una parte de los hiperparámetros influye

significativamente en el desempeño del modelo, lo que permite obtener buenos resultados con un número reducido de evaluaciones (Géron, 2022).

Para la optimización se utilizó como métrica el F1-score, al tratarse de una medida que combina precisión y recall de forma balanceada. Aunque el conjunto de datos no presenta un desbalance marcado entre clases, se buscó emplear una métrica que no favoreciera un tipo particular de error, permitiendo así una evaluación más equilibrada del rendimiento del modelo (Witten et al., 2016).

Dado que los datos presentan una estructura temporal, se implementó una estrategia de validación cruzada que preserva la cronología mediante la técnica *TimeSeriesSplit*. Este enfoque resulta adecuado para evitar el uso inadvertido de información futura durante el entrenamiento, lo cual podría inducir sesgos en la evaluación (Hyndman & Athanasopoulos, 2018).

Tabla 3: Rango de hiperparámetros del modelo de Árbol de Decisión

Hiperparámetros	Valores evaluados
max_depth	Entero aleatorio entre 5 y 14
min_sample_split	Entero aleatorio entre 10 y 30
min_sample_leaf	Entero aleatorio entre 5 y 20
ccp_alpha	Distribución uniforme continua [0.00, 0.02]

Tabla 4: Rango de hiperparámetros del modelo Random Forest

Hiperparámetros	Valores evaluados
n_estimators	Entero aleatorio entre 100 y 250
max_depth	Entero aleatorio entre 4 y 9
min_sample_split	Entero aleatorio entre 20 y 40

min_sample_leaf	Entero aleatorio entre 8 y 20
max_features	[None,'sqrt', 'log2']

Tabla 5: Rango de hiperparámetros del modelo XGBoost

Hiperparámetros	Valores evaluados
n_estimators	Entero aleatorio entre 250 y 350
max_depth	[2, 3, 4, 5]
learning_rate	Distribución uniforme continua [0.03, 0.08]
subsample	[0.6, 0.7, 0.8]
colsample_bytree	[0.6, 0.7, 0.8]
gamma	Distribución uniforme continua [0.6, 2.0]
min_child_weight	Entero aleatorio entre 4 y 10
reg_alpha	Distribución uniforme continua [0.5, 1.1]
reg_lambda	Distribución uniforme continua [0.3, 0.9]

4. Resultados

Tal como se mencionó en la sección anterior, se entrenaron cuatro algoritmos y para evaluar su desempeño se utilizaron múltiples métricas que permiten capturar distintos aspectos de la capacidad predictiva (Han, Kamber & Pei, 2011).

Por un lado, la matriz de confusión permite observar la cantidad de verdaderos positivos, verdaderos negativos, falsos positivos y falsos negativos, brindando así información sobre el tipo de errores cometidos por los modelos. A raíz de esta matriz, se calcularon las siguientes métricas:

- La precisión se define como la proporción de predicciones positivas que fueron correctas, es decir:

$$Precision = \frac{TP}{TP+FP} \quad (8)$$

- La sensibilidad o recall mide la capacidad del modelo para identificar correctamente los casos positivos reales:

$$Recall = \frac{TP}{TP+FN} \quad (9)$$

- El F1-score es la media armónica entre precisión y recall, útil como medida resumen cuando se desea equilibrar ambos aspectos:

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \quad (10)$$

- La exactitud o accuracy indica la proporción total de predicciones correctas sobre la totalidad de observaciones:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (11)$$

Finalmente, se consideró el área bajo la curva ROC (AUC ROC) como la medida principal del modelo. Esta métrica evalúa el rendimiento del clasificador en todos los umbrales posibles y resume su habilidad para distinguir entre clases positivas y negativas. Un valor de AUC cercano a 1 indica una excelente capacidad de discriminación.

Estas métricas fueron calculadas sobre el conjunto de validación cruzada (primer semestre) para los cuatro modelos y sobre el conjunto de evaluación out-of-sample (segundo semestre), en el caso de Regresión Logística, Random Forest y XGBoost, con el fin de obtener seguridad sobre la capacidad predictiva y el poder de generalización de los modelos.

4.1 Desempeño de los modelos

Figura 19: Matriz de confusión por modelo y conjunto (Train/Test)
(con valores absolutos y porcentaje por clase)

De los 28.624 usuarios del set de entrenamiento, 12.794 hicieron *churn* y 15.830 no realizaron *churn*. Para los 12.794 casos que realmente abandonaron el servicio en M1, la regresión logística predijo 70,7% de forma correcta, frente a un 60.3% en Árboles de decisión, 58.6% en Random Forest y 69,8% en XGBoost.

Desde la perspectiva del recall sobre la clase positiva (*churn*), se observan diferencias significativas entre los modelos evaluados. Esta métrica, que mide la proporción de usuarios que efectivamente realizaron *churn* y fueron correctamente detectados, es especialmente relevante en contextos donde se busca implementar estrategias de retención proactiva. Un modelo con alto recall minimiza la cantidad de usuarios que abandonan sin haber sido identificados a tiempo.

En este sentido, la regresión logística fue el modelo con mejor desempeño, logrando un recall de 0.719, seguido por XGBoost, con 0.672. Ambos modelos mostraron una buena capacidad de detección temprana de *churn*, con niveles bajos de falsos negativos. En cambio, el árbol de decisión y el Random Forest presentaron valores más bajos, con recalls de 0.599 y 0.579 respectivamente, lo que evidencia una menor sensibilidad para captar la clase de mayor interés.

Sin embargo, dado que un valor elevado de recall no necesariamente implica un buen desempeño global, no se considera adecuado evaluar los modelos únicamente en función de esta métrica de forma aislada.

Tabla 6: Comparación de desempeño de los modelos en entrenamiento y test

Modelo	Conjunto	Accuracy	Precision (1)	Recall (1)	F1-score(1)	AUC ROC
Regresión logística	Train	0.674	0.619	0.707	0.66	0.743
Regresión logística	Test	0.676	0.617	0.719	0.665	0.741
Árboles de decisión	Train	0.676	0.648	0.603	0.625	0.738
Árboles de decisión	Test	0.677	0.651	0.599	0.624	0.734
Random Forest	Train	0.688	0.674	0.586	0.627	0.759
Random Forest	Test	0.680	0.663	0.579	0.618	0.750
XGBoost	Train	0.726	0.691	0.698	0.695	0.805
XGBoost	Test	0.705	0.668	0.675	0.672	0.774

La regresión logística mostró un desempeño estable y consistente, con métricas similares entre entrenamiento y test. Obtuvo un F1-score de 0.665 y un AUC ROC de 0.741 en test, lo que indica que, a pesar de su simplicidad, logra un rendimiento razonable en la tarea de clasificación binaria. Su comportamiento similar en los datos de entrenamiento y validación sugiere que no presenta sobreajuste.

El árbol de decisión, ajustado mediante búsqueda aleatoria de hiperparámetros, alcanzó un F1-score de 0.624 y un AUC ROC de 0.734. Si bien sus métricas son comparables a las de la regresión logística, no logró superarla en capacidad discriminativa, evidenciando un rendimiento más limitado para diferenciar correctamente entre ambas clases.

El modelo de Random Forest mejoró ligeramente los resultados anteriores, alcanzando un F1-score de 0.618 y un AUC ROC de 0.750 en el conjunto de test. Esto evidencia una mayor capacidad para captar patrones complejos sin incurrir en sobreajuste, ya que las métricas entre entrenamiento y test se mantuvieron relativamente estables.

Finalmente, el modelo de XGBoost fue el que obtuvo los mejores resultados generales. En el conjunto de test alcanzó un F1-score de 0.672 y un AUC ROC de 0.774, lo que indica una ventaja respecto al resto de los modelos, tanto en precisión como en capacidad discriminativa. Si bien mostró un rendimiento superior en entrenamiento, las métricas en test también fueron más altas, lo que justifica su selección para la etapa final de evaluación.

En líneas generales, los resultados obtenidos a lo largo del proceso de validación cruzada muestran que los distintos modelos considerados —Regresión Logística, Árboles de decisión, Random Forest y XGBoost— alcanzaron niveles de desempeño relativamente similares, con

diferencias acotadas en métricas como el F1-score y el AUC ROC. Esta convergencia puede atribuirse, en parte, a las características del conjunto de datos utilizado: una estructura tabular, un volumen de observaciones moderado y una relación no excesivamente compleja entre las variables predictoras y la variable objetivo.

Tal como señalan Hastie, Tibshirani y Friedman (2009), en escenarios con baja complejidad y conjuntos de datos de tamaño limitado, los modelos estadísticos simples pueden alcanzar rendimientos comparables a los de algoritmos más sofisticados, sin que la complejidad adicional se traduzca necesariamente en una mejora significativa en la capacidad predictiva. Esta observación refuerza la importancia de seleccionar el modelo no solo por su sofisticación técnica, sino también por su adecuación al problema, su interpretabilidad y su estabilidad en contextos reales.

No obstante, en función de estos resultados, se seleccionaron los modelos de Regresión Logística, Random Forest y XGBoost como los tres candidatos más prometedores para ser evaluados sobre el conjunto out-of-sample (segundo semestre), con el objetivo de determinar cuál de ellos generaliza mejor a nuevos datos.

Asimismo, los modelos de Random Forest y XGBoost fueron entrenados utilizando los hiperparámetros óptimos obtenidos durante la etapa de ajuste. Para una descripción detallada de cada parámetro y su valor final, puede consultarse el Apéndice C.

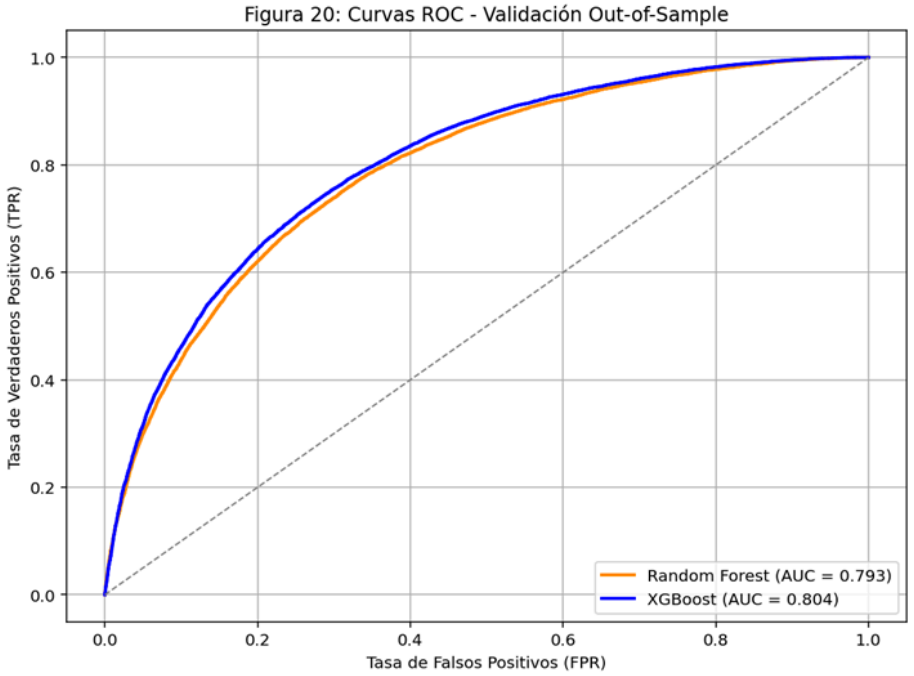
Tabla 7: Comparación de desempeño de los modelos en el set de validación

Modelo	Accuracy (1)	Precision (1)	Recall (1)	F1-score(1)	AUC ROC
Regresión Logística	0.70	0.63	0.68	0.65	0.761
Random Forest	0.72	0.68	0.65	0.67	0.793
XGBoost	0.73	0.68	0.69	0.69	0.804

Los tres modelos fueron entrenados sobre la totalidad de los datos del primer semestre y evaluados sobre el segundo semestre, simulando una situación real de predicción futura. Los resultados muestran que tanto el modelo de Random Forest como el de XGBoost alcanzaron un nivel similar de precisión global, con un *accuracy* del 72% y 73% respectivamente. Sin embargo, al observar métricas más discriminantes, se destaca el mejor desempeño del modelo de XGBoost, que obtuvo un F1-score de 0.73 y un AUC ROC de 0.804, superando levemente a

Random Forest (F1-score: 0.67, AUC ROC: 0.793). El algoritmo de Regresión Logística, si bien tuvo un buen desempeño en terminos generales, no logró superar a los otros dos modelos y se puede observar una caída en la métrica de recall respecto a lo observado en la etapa de entrenamiento y testeo.

Figura 20: Curvas ROC - Validación Out-of-Sample



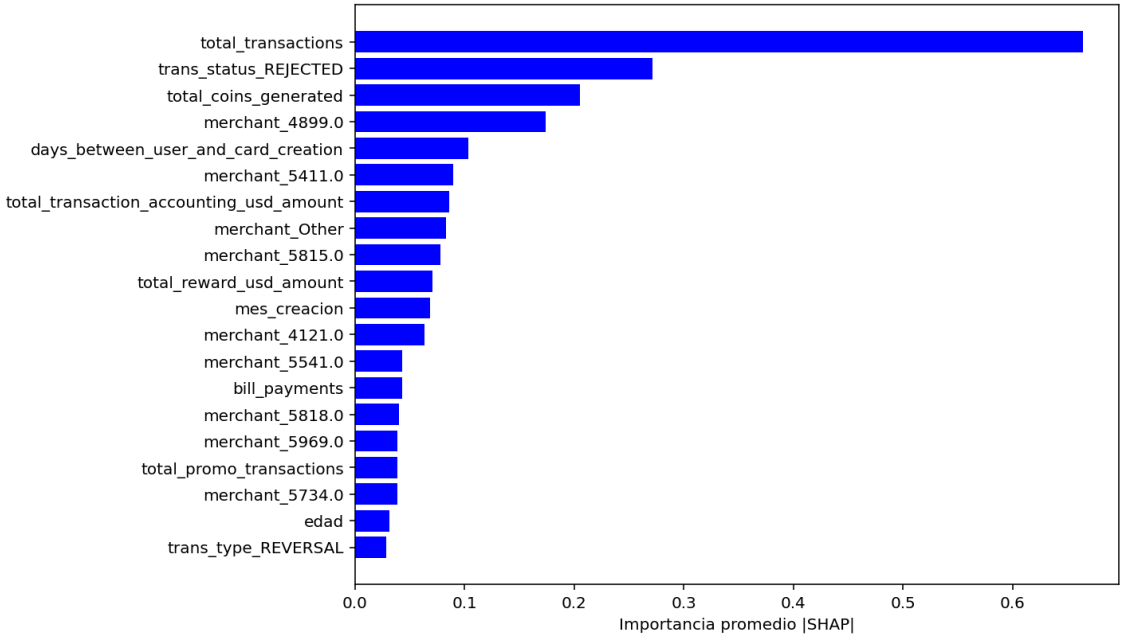
Esta diferencia, aunque sutil, sugiere que XGBoost tiene una mayor capacidad para distinguir entre usuarios que abandonan y aquellos que no, incluso cuando las proporciones de clase están balanceadas. Además, la estabilidad entre las métricas de entrenamiento, test y out-of-sample refuerza su capacidad de generalización.

En función de estos resultados, **el algoritmo de XGBoost fue seleccionado como el modelo final del estudio**.

4.2 Importancia de las variables en la predicción

Tal como se mencionó en la sección anterior, con el objetivo de identificar en qué medida y en qué dirección afecta cada variable a la predicción del modelo XGBoost, se aplicó la metodología de **valores SHAP** (*SHapley Additive exPlanations*). Esta metodología permitió entender la contribución de cada variable explicativa en la predicción individual de cada observación.

Figura 21: Top 20 features más importantes según SHAP



Según se puede observar en la Figura 21, la variable con mayor importancia en las predicciones del modelo es total_transactions, que representa el total de transacciones realizadas por cada usuario durante sus primeros 30 días. Esta variable presentó un valor SHAP promedio superior a 0.6, lo que indica una alta influencia en la probabilidad de abandono. Le sigue en importancia la variable trans_status_REJECTED, que captura la cantidad de transacciones rechazadas experimentadas por el usuario en el mismo período, con una importancia cercana a 0.3.

Asimismo, se destacan entre las variables más relevantes aquellas que agrupan las transacciones por categoría de comercio (merchant_XXXX). Esto sugiere que el tipo de comerciante con el que interactúa el usuario es un factor importante para entender su probabilidad de retención o abandono del servicio.

Por otro lado, tanto la cantidad de promociones recibidas como el monto total de recompensas percibidas por cada usuario también muestran ser variables explicativas importantes del comportamiento de *churn*. En particular, el monto de las promociones (total_reward_usd_amount) parece aportar más información que la cantidad de transacciones de promociones recibidas (total_promo_transactions), lo cual sugiere que el valor económico de las recompensas influye más que su frecuencia.

Finalmente, variables temporales como el mes de creación del usuario (mes_creacion) y los días transcurridos entre el alta en el ecosistema de la fintech y la solicitud de la primera tarjeta (days_between_user_and_card_creation) también contribuyen a la predicción. Además, se

observa que los usuarios que utilizan el servicio de pago de facturas (bill_payments) presentan patrones relevantes para explicar su comportamiento futuro.

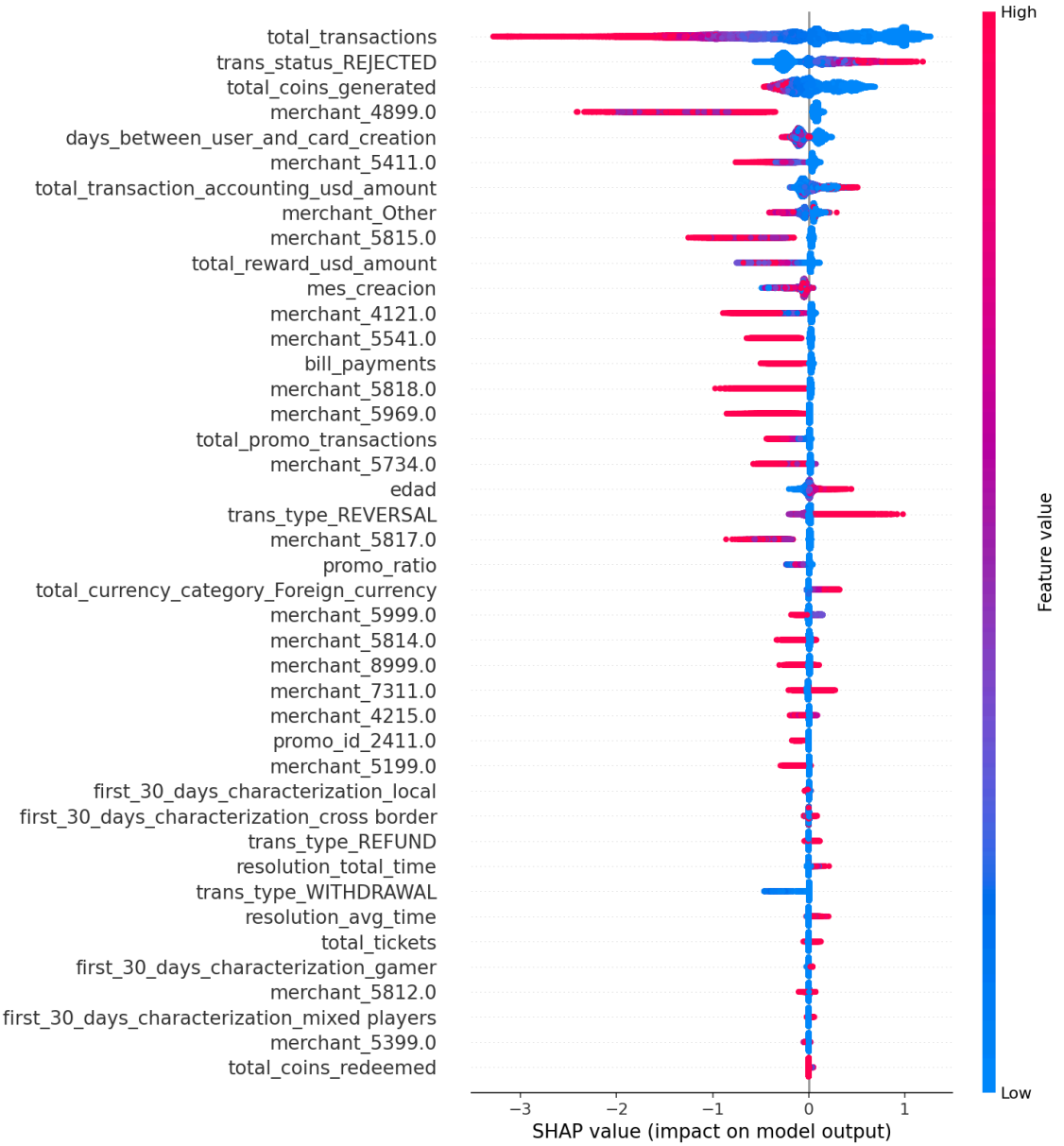
Para poder analizar no solamente la importancia en términos absolutos sino también entender la contribución de cada variable en la predicción del modelo en dirección y magnitud, se presenta el siguiente gráfico donde cada punto representa una observación del conjunto de datos de validación.

El eje x representa el valor SHAP asociado a una variable para una observación específica. Este valor indica el efecto que dicha variable tuvo sobre la predicción del modelo. Valores positivos significan que la variable contribuyó a aumentar la probabilidad de *churn* (clase 1), mientras que valores negativos indican una contribución hacia el no abandono (clase 0).

El color indica el valor real que toma la variable para esa observación. Los puntos en rojo corresponden a valores altos de la variable, mientras que los puntos en azul corresponden a valores bajos.

En el eje y se presenta las variables ordenadas de arriba hacia abajo según su importancia global. A efectos prácticos de visualización e interpretación del gráfico, se estableció un umbral de 0,001 para la magnitud promedio de dichos valores. En consecuencia, las variables cuyo valor SHAP medio absoluto se encuentra por debajo de este umbral fueron excluidas del gráfico, dado que su contribución al resultado del modelo es marginal y su inclusión podría dificultar la claridad de la interpretación general.

Figura 22: Importancia de variables según valores SHAP (umbral ≥ 0.001)



En primer lugar, como ya se había observado anteriormente, el total de transacciones realizadas por el usuario es la variable con mayor influencia en el modelo. La Figura 22 muestra las variables más importantes según los valores SHAP, y permite identificar cuatro variables con un peso claramente superior al resto:

- **total_transactions:** A mayor cantidad de transacciones durante los primeros 30 días, mayor es la propensión del usuario a permanecer en la plataforma. Esta variable refuerza la hipótesis de que un mayor nivel de actividad inicial está asociado con una mejor adopción del producto y, por ende, una mayor retención.

- **trans_status_REJECTED:** La cantidad de transacciones rechazadas experimentadas por los usuarios es un fuerte predictor de abandono. Una alta exposición a este tipo de experiencias negativas incrementa significativamente la probabilidad de *churn*, probablemente por generar frustración o desconfianza en el servicio.
- **total_coins_generated:** El total de coins acumulados a través del programa de fidelización se asocia con menor propensión al *churn*, sugiriendo que los usuarios que se ven beneficiados por incentivos económicos tienden a permanecer más en la plataforma.
- **merchant_4899.0:** Una mayor cantidad de transacciones con esta categoría de comercio, asociada a servicios como Netflix, se vincula con una mayor probabilidad de retención. Esto sugiere que cuando la tarjeta se utiliza para gastos frecuentes y cotidianos, como suscripciones digitales, se fortalece la relación del usuario con el producto.

Además de las cuatro variables principales, se destacan otros factores relevantes que contribuyen a explicar el comportamiento de los usuarios. Entre ellos, se encuentran las categorías de comercios asociados al consumo cotidiano, como merchant_5411.0 (supermercados) y merchant_5815.0 (servicios de streaming de música), que muestran una asociación con mayor retención. En contraste, otras categorías como merchant_5818.0, vinculada a juegos en línea, presentan una mayor probabilidad de abandono del servicio.

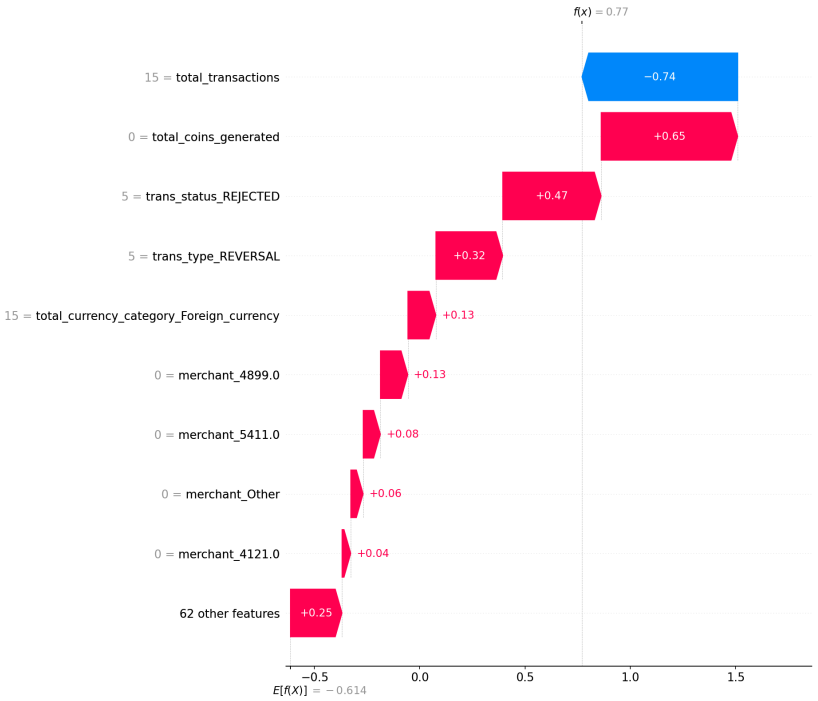
Por otro lado, la segmentación de usuarios durante los primeros 30 días revela que aquellos identificados como “locales” tienden a permanecer en la plataforma, mientras que los categorizados como “gamers” exhiben un mayor riesgo de *churn*. Este patrón se encuentra en línea con lo observado en el uso de comercios y tipos de transacción, ya que quienes integran el producto en su economía diaria muestran una relación más estable con el servicio. En esta línea, las transacciones de tipo WITHDRAWAL y bill_payments también reflejan un uso más funcional e integrado del producto, lo que parece estar asociado a una mayor fidelidad.

Las variables relacionadas con promociones y recompensas también aportan a la predicción del modelo, aunque con menor peso relativo. En particular, el monto total de recompensas recibidas (total_reward_usd_amount) parece tener un efecto más claro en reducir el riesgo de *churn* que la mera cantidad de transacciones promocionales (total_promo_transactions) o el ratio de promociones (promo_ratio). Esto sugiere que lo relevante no es la frecuencia con la que se otorgan incentivos, sino el impacto económico real que estos generan para el usuario.

Respecto a las variables de tiempo de resolución de tickets abiertos a backoffice, se destacan resolution_total_time y resolution_avg_time. Aunque su importancia relativa es baja, se observa que mayores tiempos de resolución tienden a estar asociados con una mayor propensión al abandono. Este hallazgo puede interpretarse como un indicio de que una atención lenta o procesos operativos extensos afectan negativamente la experiencia del usuario, incluso cuando no son el factor principal del modelo.

Por último, se destaca la variable edad, que, si bien tiene una importancia menor, permite observar una ligera tendencia según la cual los usuarios de mayor edad presentan una mayor propensión al *churn* en comparación con los usuarios más jóvenes.

Figura 23: Valores SHAP para un usuario en caso de Verdadero Positivo (ID Usuario = 16835004)

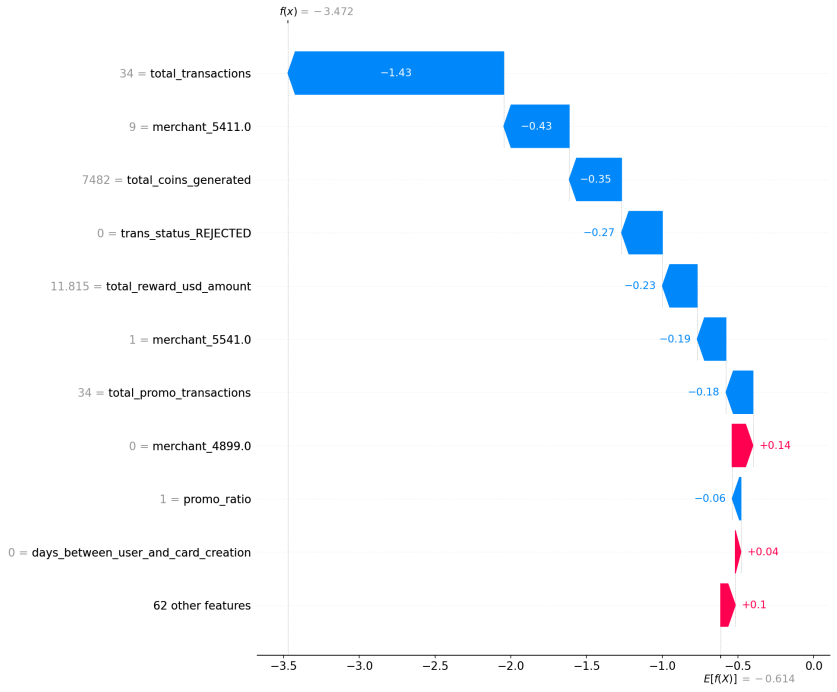


La Figura 23 muestra que el valor promedio de la predicción del modelo, considerando todos los usuarios, es de -0.614. Este valor representa el punto de partida del modelo antes de incorporar las características particulares del caso analizado. A partir de ese valor base, cada variable del usuario ajusta la predicción hacia arriba o hacia abajo, según su impacto en el modelo.

En el caso del usuario con ID 16835004, al considerar sus características individuales, el resultado final de la predicción alcanza un valor de 0.77. Dado que f(x) > 0.5, el modelo predice que este usuario realizará *churn*, lo cual efectivamente ocurrió, siendo entonces un caso de verdadero positivo.

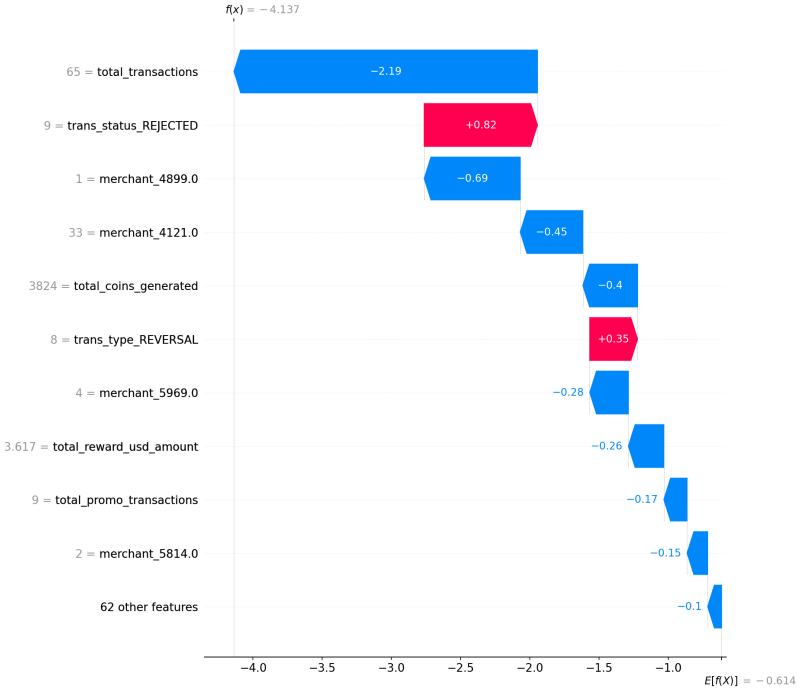
Las variables clave que impulsaron esta predicción fueron trans_type_REVERSAL y trans_status_REJECTED. Aunque el usuario realizó 15 transacciones en sus primeros 30 días, lo cual de manera aislada podría asociarse a una mayor probabilidad de retención, 5 de esas transacciones fueron reversadas y otras 5 rechazadas. Estas dos variables, junto con el hecho de que el usuario utilizó la tarjeta en el extranjero, empujan la predicción hacia el *churn*, superando la influencia positiva de la cantidad total de transacciones.

Figura 24: Valores SHAP para un usuario en caso de Falso Negativo (ID Usuario = 18143696)



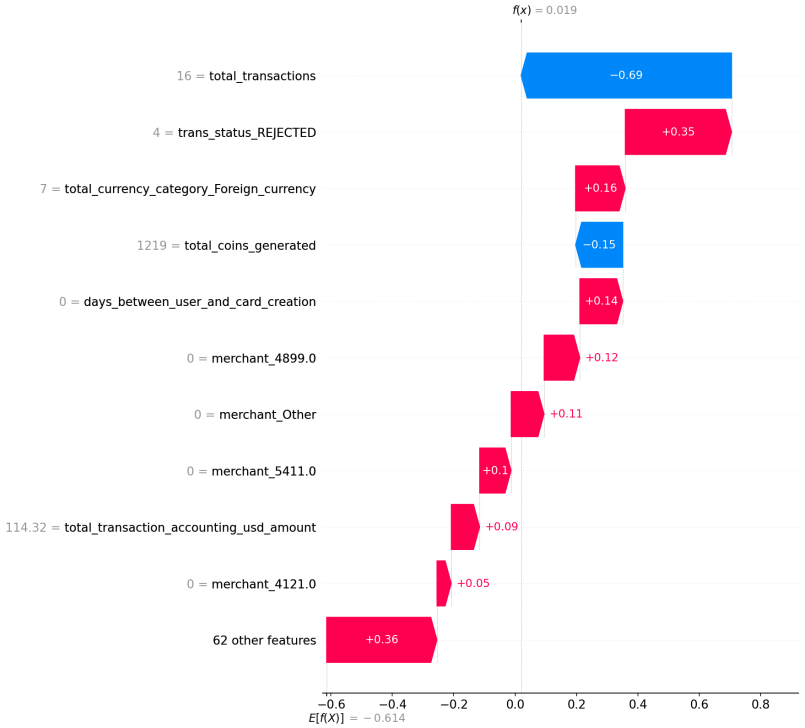
Respecto al user id 18143696, el modelo predijo que este usuario no realizaría *churn* cuando verdaderamente sí lo hizo. Las variables que más empujaron a la predicción de la clase negativa fueron la cantidad total de transacciones realizadas, el número de transacciones con el MCC 5411 (supermercados) y el monto total recibido en concepto de promociones. Un aspecto relevante de este caso es que el usuario no tuvo ninguna transacción rechazada, y la variable trans_status_REJECTED fue una de las más influyentes para predecir que no habría *churn*. Este comportamiento resulta contraintuitivo, ya que, en sus primeros 30 días, el usuario mostró características típicas de un cliente que la fintech logra retener.

Figura 25: Valores SHAP para un usuario en caso de Verdadero Negativo (ID Usuario = 17475111)



El usuario con ID 17475111 representa un caso exitoso de retención. Aunque este usuario experimentó 9 transacciones rechazadas y 9 transacciones reversadas, las variables que impulsaron la predicción hacia la retención fueron principalmente la cantidad total de transacciones, ya que realizó más de una transacción por día en los primeros 30 días. Además, el usuario realizó transacciones con códigos de comercio (MCC) asociados a la clase negativa, lo que refuerza la predicción de no *churn*. También tuvieron un impacto positivo en la predicción las variables relacionadas con las promociones, lo que contribuyó a identificar correctamente que este usuario permanecería utilizando la debit card.

Figura 26: Valores SHAP para un usuario en caso de Falso Positivo (ID Usuario = 18156785)



Por último, respecto al user id 18156785, el modelo predijo que este usuario realizaría *churn* cuando, en realidad no lo hizo. Las variables que más empujaron a la predicción de la clase positiva (*churn*) fueron el estado de las transacciones rechazadas (trans_status_REJECTED) y el monto total de coins generadas mediante el programa de fidelización de rewards. A su vez, otras variables que empujaron a la predicción de *churn* fueron la cantidad de transacciones hechas en moneda extranjera, la cantidad de días entre que se creó el usuario y que se creó su primer tarjeta de débito y que no se realizaron transacciones con los MCC 4899 y 5411.

5. Conclusiones

5.1 Sugerencias del negocio

A partir de los *insights* obtenidos en la Sección 4.2, se identificó que:

Las transacciones rechazadas son un predictor clave de abandono y estrategias para mejorar la experiencia inicial

La cantidad de transacciones rechazadas durante los primeros 30 días de uso de la tarjeta de débito impacta directamente en el abandono de los usuarios. La hipótesis central es que el rechazo de estas transacciones se debe a un mal onboarding hacia los usuarios al momento de obtener la tarjeta. El 95% de las transacciones rechazadas durante este periodo se deben a la falta de saldo en la billetera del usuario, lo cual resulta evidente dado que la tarjeta es prepaga. Este hecho destaca la importancia de una introducción efectiva a las condiciones de uso del producto. Si el usuario no comprende correctamente cómo funciona la tarjeta, es probable que sufra una mala experiencia, lo que podría llevarlo a desestimar el producto y no continuar utilizándolo.

Se recomienda implementar un proceso de onboarding claro y detallado, que no solo explique las condiciones de uso, sino que también oriente sobre cómo garantizar un uso adecuado de la tarjeta. Además, se ha identificado que la única forma de notificación de transacciones rechazadas actualmente es mediante notificaciones push, las cuales no siempre son recibidas por los usuarios. Es fundamental ampliar los métodos de comunicación para asegurar que los usuarios sean notificados de manera clara la situación, considerando la posibilidad de incorporar canales como el correo electrónico o WhatsApp.

Asimismo, se propone una estrategia adicional para mejorar la experiencia de los primeros días del usuario: otorgar un crédito limitado para las primeras transacciones. Esta medida permitiría que los usuarios realicen al menos cinco transacciones aprobadas, con un monto limitado para evitar abusos, lo que podría aumentar la confianza y la satisfacción del usuario en el uso del producto desde sus primeras interacciones.

El uso en comercios específicos como indicador de retención y promociones como herramienta para mejorar la retención

Por otra parte, una observación clave es que los usuarios que logran ser retenidos son aquellos que utilizan con mayor frecuencia los códigos de comercio (MCC) mencionados anteriormente.

En consecuencia, una vez que los usuarios solicitan la tarjeta, se recomienda implementar un sistema de comunicaciones que los fomente a utilizarla en comercios específicos que tengan un alto *engagement*. Además, sería de gran ayuda lanzar promociones diseñadas para incentivar estas transacciones, dirigiendo a los usuarios hacia los MCC que favorecen la retención.

En este sentido, se identificó que la variable referente al monto en dólares recibido por promociones es un factor que empuja hacia la predicción de no abandono. Por lo tanto, las promociones deben ser diseñadas para maximizar el uso de la tarjeta en estos comercios clave, lo que podría contribuir a mejorar la retención de los usuarios.

Transacciones en moneda extranjera como predictor de abandono y estrategias específicas para usuarios que utilizan la tarjeta en el exterior

Por último, se observó que una variable importante que impulsa la predicción de abandono es la realización de transacciones en moneda extranjera. Este patrón sugiere que algunos usuarios solicitan la tarjeta para utilizarla en viajes puntuales al extranjero, pero al regresar dejan de usarla. Esto nos lleva a la segunda hipótesis: ciertos usuarios que viajan al extranjero tienden a abandonar la tarjeta cuando regresan a Argentina.

Para abordar esta situación, se propone segmentar a estos usuarios y diseñar estrategias específicas para ellos. Por ejemplo, se podría ofrecer un cashback en pesos argentinos por cada transacción realizada en moneda extranjera. De esta forma, a su regreso, se les podría inducir a realizar transacciones locales mediante comunicaciones de marketing que los motiven a usar la tarjeta en los MCC con mayor *engagement*. También se podrían incluir promociones específicas para incentivar el pago de facturas, con el objetivo de fomentar el uso local del producto y aumentar su retención. Otra propuesta interesante es que, por cada compra en el ámbito local, el usuario genere millas para poder canjearlas por viajes en alguna aerolínea.

Implementación operativa del modelo de predicción de *churn*

Para que las estrategias propuestas puedan aplicarse de forma efectiva, es necesario integrar el modelo predictivo dentro de los procesos internos de la empresa. Su utilidad no se limita a anticipar el abandono, sino que permite gestionar de forma activa el riesgo de *churn* en tiempo real, combinando tecnología, automatización y conocimiento del negocio.

Una alternativa concreta de implementación consiste en establecer un *pipeline* semanal, en la que el modelo se ejecute automáticamente sobre los usuarios que cumplieron 30 días desde la emisión de la tarjeta. Esto permitiría clasificar a los usuarios según su probabilidad de abandono —por ejemplo, en niveles alto, medio y bajo— y vincular cada segmento con las

acciones más adecuadas según los patrones identificados: comunicaciones educativas en casos de transacciones rechazadas, promociones dirigidas hacia comercios específicos (MCCs) o incentivos de retención para usuarios que utilizaron la tarjeta en el exterior.

Los resultados del modelo también podrían integrarse en un *dashboard* interno que facilite el monitoreo de usuarios en riesgo por parte de los equipos de marketing y producto, así como la detección de patrones agregados para ajustar campañas de manera dinámica. En paralelo, el modelo podría vincularse con el CRM de la empresa para asignar tareas manuales de seguimiento a usuarios con alto riesgo, habilitando intervenciones personalizadas por parte del equipo *customer success* o *customer support.*

No puede descartarse el uso de la probabilidad output de cada usuario como una variable que integre un perfilamiento de cada usuario, como herramienta de decisión ya sea para una acción de venta, marketing o de riesgo crediticio.

Por último, se sugiere implementar un circuito de retroalimentación que permita evaluar el impacto de las acciones disparadas: si los usuarios revirtieron su comportamiento, si aprovecharon las promociones ofrecidas o si realizaron nuevas transacciones. Esta instancia no solo es positiva para validar la utilidad del modelo, sino también para retroalimentar futuros entrenamientos, mejorar su precisión y adaptarlo a posibles cambios en el comportamiento de los usuarios.

5.2 Limitaciones y futuras mejoras

Uno de los mayores desafíos de este trabajo fue la consolidación de una base de datos por usuario que contuviera variables significativas para explicar el comportamiento de los mismos. Hasta la fecha, la empresa no contaba con una base de datos integrada por usuario que permitiera analizar de manera efectiva sus interacciones con la tarjeta, así como su relación con otras variables clave.

La ausencia de una estructura organizada y documentada de las bases de datos complicó aún más este proceso, ya que las fuentes de datos no estaban bien definidas. Esto generó dificultades adicionales en la limpieza, transformación y análisis de los datos. Además, la empresa hoy en día no logra capturar el comportamiento completo del usuario desde su primer contacto con el producto hasta el posible abandono o retención.

A raíz de estas limitaciones, se recomienda que se inicie un proceso de estructuración y centralización de las bases de datos por usuario. Una forma de llevar a cabo esta tarea, es

mediante el diseño y creación de un modelo de datos a nivel usuario. En este sentido, es fundamental no solo registrar las transacciones realizadas con la tarjeta, sino también incorporar datos adicionales que permitan captar de manera más precisa el ciclo de vida del usuario. Esta información debe organizarse por periodos, permitiendo así una visión temporal más clara y el análisis de tendencias de comportamiento a lo largo del tiempo.

Por otra parte, si bien el algoritmo de XGBoost ha demostrado un buen desempeño, alcanzando un AUC ROC de 0.804 en el conjunto de validación, se considera que existen oportunidades para mejorar esta métrica en el futuro. A medida que se obtengan más variables explicativas por incorporación de información nueva de los usuarios a través del tiempo, es probable que se logre una mayor precisión en las predicciones. Por lo tanto, se recomienda continuar refinando el modelo incorporando nuevas variables y explorando diferentes técnicas de ingeniería de características, lo que podría resultar en un desempeño aún más robusto.

Referencias

- Finnovista. (2024). *Radar Argentina 2024: Mapeo del ecosistema Fintech argentino*.
- Global Processing. (2024). *GP Insight: Segundo semestre 2024*. Informe interno.
- Tukey, J. W. (1977). *Exploratory data analysis*. Addison-Wesley.
- James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An introduction to statistical learning: With applications in R*. Springer.
- Kaufman, S., Rosset, S., Perlich, C., & Stitelman, O. (2012). Leakage in data mining: Formulation, detection, and avoidance. ACM Transactions on Knowledge Discovery from Data (TKDD), 6(4), 1–21.
- He, H., & Ma, Y. (2017). *Imbalanced Learning: Foundations, Algorithms, and Applications*. Wiley.
- Tan, P.-N., Steinbach, M., & Kumar, V. (2019). *Introduction to data mining* (2nd ed.). Pearson.
- Breiman, L. (2001). Random forests. *Machine Learning, 45*(1), 5–32
- Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794.
- Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research, 13*(Feb), 281–305.
- Géron, A. (2022). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow* (3rd ed.). O'Reilly Media.
- Hyndman, R. J., & Athanasopoulos, G. (2018). *Forecasting: Principles and Practice* (2nd ed.). OTexts.
- Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J. (2016). *Data Mining: Practical Machine Learning Tools and Techniques* (4th ed.). Morgan Kaufmann.
- Han, J., Kamber, M., & Pei, J. (2011). *Data mining: Concepts and techniques* (3rd ed.). Morgan Kaufmann.
- Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: Data mining, inference, and prediction* (2nd ed.). Springer.

Apéndice

Apéndice A. Detalle de columnas y tipos de datos

Columna/Feature	Tipo de dato	Descripción
user_id	string - key	Identificador interno unico para cada usuario
user_external_id	string - key	Identificador externo unico para cada usuario
total_transaction_accounting_usd_amount	float	Monto total de la transacciones con debit cards en dolares convertido al tipo de cambio oficial
total_transaction_origin_amount	float	Monto total de la transacciones con debit cards en moneda de origen
total_transactions	int	Cantidad de transacciones con debit cards
user_churn	int	Variable binaria que indica si un usuario abandonó el servicio (1) o sigue activo (0).
edad	int	Edad del usuario
total_currency_category_ARS	int	Cantidad total de transacciones con debit cards realizadas en pesos argentinos
total_currency_category_Foreign_currency	int	Cantidad total de transacciones con debit cards realizadas en monedas diferentes al peso argentino
total_card_payment_purpose	int	Cantidad de transacciones generadas por débitos automáticos.
first_card_creation_date	date	Fecha de creación de la primer debit card
user_created_date	date	Fecha de creación del usuario
semestre	string - key	Indica si el usuario corresponde al primer semestre del 2024 o el segundo semestre. Es una variable key para separar el dataset entre train/test y validation.
days_between_user_and_card_creation	int	Cantidad de dias que pasaron entre que se creo el usuario dentro de la plataforma y que creo la primer debit card
trans_type_CHARGEBACK	int	Cantidad de contracargos
trans_type_DEBIT_ADJUSTMENT	int	Cantidad de transacciones que corresponden a ajustes de montos en transacciones
trans_type_OFFLINE_PURCHASE	int	Cantidad de transacciones de compra con debit cards que se hicieron offline
trans_type_PURCHASE	int	Cantidad de transacciones de compra con debit cards que se hicieron en linea
trans_type_REFUND	int	Cantidad de transacciones de debit cards reembolsadas
trans_type_REVERSAL	int	Cantidad de transacciones de debit cards reversadas
trans_type_WITHDRAWAL	int	Cantidad de transacciones de retiros de efectivo realizadas con la debit card

trans_status_APPROVED	int	Cantidad de transacciones de debit cards aprobadas
trans_status_DISMISSED	int	Cantidad de transacciones de debit cards no realizadas por time outs
trans_status_ERROR	int	Cantidad de transacciones de debit cards no realizadas por algun tipo de error
trans_status_PENDING	int	Cantidad de transacciones de debit cards pendientes de aprobacion
trans_status_REJECTED	int	Cantidad de transacciones de debit cards rechazadas
trans_status_WAITING_MANUAL_APPROVAL	int	Cantidad de transacciones de debit cards pendientes de aprobacion manual
merchant_xxxx	int	Cantidad de transacciones de debit cards realizadas en cotegorias de comercios en especifico
total_promo_transactions	int	Cantidad total de recompensas recibidas por promociones activas
total_reward_usd_amount	float	Monto total de las recompensas recibidas por promociones activas
promo_id_xxxx	int	Cantidad total de recompensas recibidas con determinadas promociones
total_tickets	int	Cantidad total de tickets abiertos a customer support
resolution_total_time	float	Tiempo de resolucion de tickets (en dias)
resolution_avg_time	float	Tiempo promedio de resolucion por ticket (en dias)
total_astrocoins_generated	int	Cantidad total de astrocoins generados
total_astrocoins_redeemed	int	Cantidad total de astrocoins canjeados
first_30_days_characterization_cross border	int	Variable binaria que indica si el usuario tiene perfil de cross border (transacciones en diferentes lugares del mundo)
first_30_days_characterization_gamer	int	Variable binaria que indica si el usuario tiene perfil de gamer (transacciones con merchants de gambling)
first_30_days_characterization_local	int	Variable binaria que indica si el usuario tiene perfil de local (transacciones dentro de Argentina)
first_30_days_characterization_mixed players	int	Variable binaria que indica si el usuario tiene perfil mixto (no se le puede asignar un segmento en especifico)
bill_payments	int	Variable binaria que indica si el usuario tambien realiza transacciones en el producto relacionado a pago de facturas
TG_Type_AM_INVESTMENTS	int	Variable binaria que indica si el usuario invierte su saldo de balance en la billetera
TG_Type_EXCHANGE	int	Variable binaria que indica si el usuario realiza cambios de moneda dentro de la billetera
TG_Type_INTERNAL_P2P_TRANSFER	int	Variable binaria que indica si el usuario realiza transferencias a otras personas que tambien estan dentro de la plataforma

TG_Type_P2P_TRANSFER	int	Variable binaria que indica si el usuario realiza transferencias a otras personas que no necesariamente estan dentro de la plataforma
TG_Type_PAYMENT_LINK	int	Variable binaria que indica si el usuario tambien realiza transacciones en el producto relacionado a pagos por medio de links de pago
TG_Type_QR_PAYMENT	int	Variable binaria que indica si el usuario tambien realiza transacciones en el producto relacionado a pagos con QR

Apéndice B. Comparación de selección de variables por método: correlación, varianza y LASSO

Variable	Correlación	Varianza	LASSO
user_id	✅	✅	✅
total_transaction_accounting_usd_amount	✅	✅	✅
total_transaction_origin_amount	❌	✅	✅
total_transactions	✅	✅	✅
user_churn	✅	✅	✅
edad	✅	✅	✅
total_currency_category_ARS	❌	✅	✅
total_currency_category_Foreign_currency	✅	✅	✅
total_card_payment_purpose	✅	✅	✅
first_card_creation_date	✅	✅	✅
user_created_date	✅	✅	✅
trans_type_CHARGEBACK	✅	❌	✅
trans_type_DEBIT_ADJUSTMENT	✅	✅	❌
trans_type_OFFLINE_PURCHASE	✅	✅	✅
trans_type_PURCHASE	❌	✅	✅
trans_type_REFUND	✅	✅	✅
trans_type_REVERSAL	✅	✅	✅
trans_type_WITHDRAWAL	✅	✅	✅
trans_status_APPROVED	❌	✅	✅
trans_status_DISMISSED	✅	✅	❌
trans_status_ERROR	✅	❌	✅
trans_status_PENDING	✅	✅	❌
trans_status_REJECTED	✅	✅	✅
trans_status_WAITING_MANUAL_APPROVAL	✅	✅	✅
merchant_4121.0	✅	✅	✅
merchant_4215.0	✅	✅	✅
merchant_4899.0	✅	✅	✅
merchant_5199.0	✅	✅	✅
merchant_5399.0	✅	✅	✅
merchant_5411.0	✅	✅	✅
merchant_5462.0	✅	✅	✅

merchant_5499.0	✅	✅	✅
merchant_5541.0	✅	✅	✅
merchant_5734.0	✅	✅	✅
merchant_5812.0	✅	✅	✅
merchant_5814.0	✅	✅	✅
merchant_5815.0	✅	✅	✅
merchant_5816.0	✅	✅	❌
merchant_5817.0	✅	✅	✅
merchant_5818.0	✅	✅	✅
merchant_5969.0	✅	✅	✅
merchant_5999.0	✅	✅	✅
merchant_7311.0	✅	✅	✅
merchant_8999.0	✅	✅	✅
merchant_Other	✅	✅	✅
days_between_user_and_card_creation	✅	✅	✅
semestre	✅	✅	✅
total_promo_transactions	✅	✅	✅
total_reward_usd_amount	✅	✅	✅
promo_id_2141.0	✅	✅	✅
promo_id_2142.0	✅	✅	✅
promo_id_2143.0	✅	✅	✅
promo_id_2194.0	✅	❌	✅
promo_id_2218.0	✅	✅	✅
promo_id_2219.0	✅	❌	✅
promo_id_2220.0	✅	❌	✅
promo_id_2221.0	✅	✅	✅
promo_id_2263.0	✅	✅	✅
promo_id_2264.0	✅	❌	✅
promo_id_2265.0	✅	❌	✅
promo_id_2266.0	✅	❌	✅
promo_id_2267.0	✅	❌	✅
promo_id_2309.0	✅	✅	✅
promo_id_2320.0	✅	❌	✅
promo_id_2348.0	✅	❌	✅
promo_id_2349.0	✅	✅	✅
promo_id_2352.0	✅	✅	✅
promo_id_2366.0	✅	✅	✅
promo_id_2395.0	✅	❌	✅
promo_id_2396.0	✅	✅	✅
promo_id_2411.0	✅	✅	✅
promo_id_2425.0	✅	❌	✅
promo_id_2435.0	✅	✅	✅
promo_id_2452.0	✅	✅	✅
promo_id_2459.0	✅	✅	✅
promo_id_2475.0	✅	✅	✅
promo_id_2504.0	✅	✅	✅

promo_id_2517.0	✅	✅	✅
promo_id_2536.0	✅	✅	✅
promo_id_2540.0	✅	✅	❌
promo_id_2555.0	✅	✅	❌
promo_id_2559.0	✅	✅	❌
promo_id_2569.0	✅	✅	❌
promo_id_2570.0	✅	✅	❌
promo_id_2578.0	✅	✅	❌
total_tickets	✅	✅	✅
resolution_total_time	✅	✅	✅
resolution_avg_time	✅	✅	✅
user_external_id	✅	✅	✅
total_astrocoins_generated	✅	✅	✅
total_astrocoins_redeemed	✅	✅	✅
first_30_days_characterization_cross border	✅	✅	✅
first_30_days_characterization_gamer	✅	✅	✅
first_30_days_characterization_local	✅	✅	✅
first_30_days_characterization_mixed players	✅	✅	✅
bill_payments	✅	✅	✅
TG_Type_AM_INVESTMENTS	✅	✅	✅
TG_Type_EXCHANGE	✅	✅	✅
TG_Type_INTERNAL_P2P_TRANSFER	✅	✅	✅
TG_Type_P2P_TRANSFER	✅	✅	✅
TG_Type_PAYMENT_LINK	✅	✅	✅
TG_Type_QR_PAYMENT	✅	✅	✅
mes_creacion	✅	✅	✅
promo_ratio	✅	✅	✅
used_bill_payments	✅	✅	✅
used_TG_Type_AM_INVESTMENTS	✅	✅	✅
used_TG_Type_EXCHANGE	✅	✅	✅
used_TG_Type_INTERNAL_P2P_TRANSFER	✅	✅	✅
used_TG_Type_P2P_TRANSFER	✅	✅	✅
used_TG_Type_PAYMENT_LINK	✅	✅	✅
used_TG_Type_QR_PAYMENT	✅	✅	✅

Apéndice C. Hiperparámetros seleccionados

Modelo	Hiperparámetro	Descripción	Valor óptimo
Árboles de decisión	max_depth	Profundidad máxima del árbol	10
	min_samples_split	Mínimo de muestras para dividir un nodo	20
	min_samples_leaf	Mínimo de muestras en una hoja	19
	ccp_alpha	Parámetro de poda que controla la complejidad del árbol	0,0007

Random Forest	n_estimators	Número de árboles en el bosque	104
	max_depth	Profundidad máxima de cada árbol	6
	min_samples_split	Mínimo de muestras para dividir un nodo	27
	min_samples_leaf	Mínimo de muestras en una hoja	15
	max_features	Número máximo de variables consideradas por división	None
XGBoost	n_estimators	Número de áboles boosteados	250
	max_depth	Profundidad máxima de cada árbol	5
	learning_rate	Tasa de aprendizaje (shrinkage)	0,0514
	subsample	Proporción de muestras utilizadas en cada árbol	0,8
	colsample_bytree	Proporción de variables usadas por árbol	0,8
	gamma	Umbral mínimo de ganancia para realizar una división	1,3689
	min_child_weight	Peso mínimo de observaciones en una hoja	8
	reg_alpha	Término de regularización L1 (Lasso)	1,0774
	reg_lambda	Término de regularización L2 (Ridge)	0,7427