Escuela de Negocios. Master in Management + Analytics

A Machine Learning Approach for Churn Prediction in a Mobile App

Autoría: Roll, Ignacio
Año: *2024*

UNIVERSIDAD TORCUATO DI TELLA

Master in Management & Analytics

A Machine Learning Approach for Churn Prediction in a Mobile App

Roll, Ignacio (Legajo: 22A1608)

Abril 2024

Tutor: Guido de Caso

Resumen

En el contexto del año 2023, las empresas se encuentran enfrentando una realidad en la que los recortes de costos se han convertido en una necesidad imperante. En este escenario, la importancia de retener a los usuarios se vuelve aún más crucial, ya que adquirir nuevos clientes se ha vuelto cada vez más costoso. A lo largo del trabajo se trata el problema crítico de la pérdida de usuarios o "churn" en la industria de aplicaciones móviles. La tesis argumenta que en lugar de solo enfocarse en adquirir nuevos usuarios, las empresas de aplicaciones móviles deben centrarse en retener a los existentes para reducir la tasa de churn. La tasa de churn es un indicador crucial del éxito o fracaso de una empresa en retener a su base de clientes, y la presente tesis tiene como objetivo investigar los factores que influyen en la misma.

Para abordar este problema, se propone un enfoque basado en el análisis de datos y el aprendizaje automático para predecir el churn de los usuarios y luego poder ayudar a las empresas a tomar medidas para retenerlos. En resumen, la tesis destaca la importancia de la retención de usuarios y propone un enfoque para abordar el problema de la churn en la industria de aplicaciones móviles, donde se utilizarán los datos de una aplicación de delivery de comida.

Abstract

Retaining a customer is often less costly than acquiring them again. Throughout the work, the critical issue of user loss or "churn" in the mobile application industry is addressed. The thesis argues that instead of only focusing on acquiring new users, mobile application companies should concentrate on retaining existing ones to reduce the churn rate. The churn rate is a crucial indicator of a company's success or failure in retaining its customer base, and this thesis aims to investigate the factors that influence it.

To address this problem, an approach based on data analysis and machine learning is proposed to predict user churn and subsequently assist companies in taking measures to retain them. In summary, the thesis emphasizes the importance of user retention and proposes an approach to tackling the churn problem in the mobile application industry, where data from a food delivery application will be utilized.

Abril 2024

Índice

1. Introducción **7**

1.1. Churn en la industria 7

1.2. LTV, Retención y Churn 9

1.3. Retención vs Adquisición de Usuarios 10

1.4. Tipos de Churn en aplicaciones móviles 13

2. Contexto **13**

2.1. Inicio del proyecto 13

2.2. Industria de delivery en línea 13

2.3. Posibles razones del churn 14

2.4. Objetivo del trabajo 15

2.5. Mercado y competencia 16

3. Dataset **17**

3.1. Dimensiones 18

3.2. Data Cleansing 18

3.3. Highlights 19

3.4. Caso de estudio 23

4. Modelado **25**

4.1. Primer modelado 25

4.2. Propiedades en el tiempo 25

4.3. Propiedades estadísticas 26

4.4. Representación de usuario modelo 27

4.5. Análisis Descriptivo 27

4.6. Análisis Exhaustivo 29

4.7. Selección de features 33

4.8. Tipo de churn a considerar 34

5. Implementación **35**

5.1. Algoritmos de aprendizaje 35

5.2. Ensamble Learning 35

5.2.1. XGBoost 36

5.2.2. CatBoost 37

5.3. Hiperparametros 38

5.4. Entrenamiento, validación y test 40

5.5. Modelos estudiados y descartados 42

6. Resultados y discusión **43**

6.1. Métricas del modelo 43

6.2. Métricas y el negocio 45

6.3. XGBoost 45

6.4. CatBoost 52

6.5. Importancia e interpretación 57

7. Enfoque de negocio y discusión **59**
7.1. Escenarios a la hora de considerar enviar estrategias de Marketing 60
7.1.1. Grupo A: Usuario que es impactado y abandona 62
7.1.2. Grupo B: Usuario que no es impactado y abandona 63
7.1.3. Grupo C: Usuarios que es impactado y no abandona 64
7.1.4. Grupo D: Usuarios que no es impactado y no abandona 64
7.2. Usuarios pasivos o dormidos 65
7.3. Clientes leales frente a adquirir nuevos 66
7.4. Un buen producto y su posicionamiento 67
7.5. Engagement en Churn Prevention 68
8. Conclusiones **69**
9. Trabajo futuro **70**
10. Bibliografía **71**

Índice de Gráficos

Gráfico 1: Churn en diferentes industrias 7
Gráfico 2: Curvas de retención de una aplicación de un banco virtual en Argentina. 8
Gráfico 3: Vista previa del dataset inicial. 18
Gráfico 4: Cantidad de eventos según dispositivo en noviembre 2022. 20
Gráfico 5: Cantidad de eventos según hora del día y tipo de evento en noviembre de 2022. 20
Gráfico 6: Mapa satelital de calor según ubicaciones de eventos en el mes de noviembre. 21
Gráfico 7: Porcentaje de usuarios activos a lo largo de sus primeras semanas. 21
Gráfico 8: Cantidad de usuarios que realizan determinados eventos según rangos de cantidad. 22
Gráfico 9: Cantidad de transacciones acumuladas al momento del último evento del periodo. 23
Gráfico 10: Transacciones de Juan, usuario de estudio presente en el dataset. 24
Gráfico 11: Cantidad de eventos en el transcurso activo del usuario de estudio. 24
Gráfico 12: Muestra de la variable ‘install_time’. 29
Gráfico 13: Evolución acumulada de usuarios según ‘install_time’. 30
Gráfico 14: Evolución acumulada de usuarios según ‘install_time’ and Churn. 30
Gráfico 15: Instalaciones realizadas según hora en el día y Churn. 31
Gráfico 16: Muestra de la variable ‘deltatime_install_firstevent’. 31
Gráfico 17: Boxplot de Delta entre Instalación y Primer evento por Churn. 32
Gráfico 18: Distribución de Delta entre Instalación y Primer evento por Churn. 32
Gráfico 19: Distribución de Delta entre Instalación y Primer evento por Churn con Churn Rate. 33
Gráfico 20: Código de la implementación de la librería XGBoost. 37
Gráfico 21: Código con implementación de librería CatBoost. 38
Gráfico 22: Porción de código utilizada para optimizar hiperparametros con XGBoost. 40

Gráfico 23: Representación temporal de las muestras entrenamiento, validación y test. 41
Gráfico 24: Representación temporal de las muestras con Cut Off el 2023-01-31. 41
Gráfico 25: Eventos realizados por usuarios activos en el mes de noviembre de 2022. 42
Gráfico 26: Curva ROC utilizando XGBoost con el set de validación. 47
Gráfico 27: Distribución de churn utilizando XGBoost para el set de test. 48
Gráfico 28: Matriz de confusión utilizando XGBoost para el set de test. 49
Gráfico 29: Curva ROC utilizando XGBoost para el set de test. 50
Gráfico 30: Curva ROC con XGBoost y 2 meses de ventana para el set de test. 50
Gráfico 31: Distribución de churn con XGBoost y 2 meses de ventana para el set de test. 51
Gráfico 32: Matriz de confusión con XGBoost y 2 meses de ventana para el set de test. 51
Gráfico 33: Curva ROC utilizando CatBoost con el set de validación. 53
Gráfico 34: Distribución de churn utilizando CatBoost con el set de test. 54
Gráfico 35: Matriz de confusión utilizando CatBoost con el set de test. 54
Gráfico 36: Curva ROC utilizando CatBoost con el set de test. 55
Gráfico 37: Curva ROC utilizando CatBoost con 2 meses de ventana con el set de test. 55
Gráfico 38: Distribución de churn utilizando CatBoost con 2 meses de ventana para el set de test. 56
Gráfico 39: Matriz de confusión utilizando CatBoost con 2 meses de ventana para el set de test. 56
Gráfico 40: Importancia de variables utilizando XGBoost con un mes de ventana. 57
Gráfico 41: Importancia de variables utilizando CatBoost. 58
Gráfico 42: Representación de nuevos clientes vs actuales para una empresa promedio. 66

Índice de Tablas

Tabla 1: Escenario base de la empresa 11
Tabla 2: Escenario con 10% de aumento en retención 12
Tabla 3: Escenario con 10% de aumento en adquisición 12
Tabla 4: Columnas, descripción y valores de ejemplo del dataset. 18
Tabla 5: Porción de información extraída de usuario modelo Juan. 27
Tabla 6: descripción de las variables independientes luego del modelado. 27
Tabla 7: Algunas iteraciones de la búsqueda de hiperparametros utilizando XGBoost. 46
Tabla 8: Mejor combinación de hiperparametros para XGBoost. 46
Tabla 9: Métricas del modelo para diferentes umbrales utilizando XGBoost con el set de validación. 47
Tabla 10: Mejor combinación de hiperparametros para XGBoost con 2 meses de ventana. 50
Tabla 11: Mejores iteraciones de la búsqueda de hiperparametros utilizando CatBoost. 52
Tabla 12: Mejor combinación de hiperparametros para CatBoost. 52
Tabla 13: Métricas del modelo para diferentes umbrales utilizando CatBoost con el

set de validación. 53
Tabla 14: Mejor combinación de hiperparametros para CatBoost y 2 meses de ventana con test. 55
Tabla 15: Explicación de variables utilizando XGBoost. 58
Tabla 16: Explicación de variables utilizando CatBoost. 59
Tabla 17: Escenarios de una empresa a la hora de enviar estrategias de Marketing. 61

1. Introducción

1.1. Churn en la industria

Las empresas que presentan su negocio a través de una aplicación móvil, esmeran sus esfuerzos y recursos principalmente en la adquisición de clientes, pero dejan de lado la sostenibilidad y el cuidado de ellos. Los esfuerzos para mejorar la retención a menudo se dejan en un segundo plano o se descuidan por completo. Sin embargo, la retención debería tomar un lugar mucho más importante dado el impacto que tiene en el negocio. Según Tessitore (2023), las tasas de abandono según industria son las siguientes.

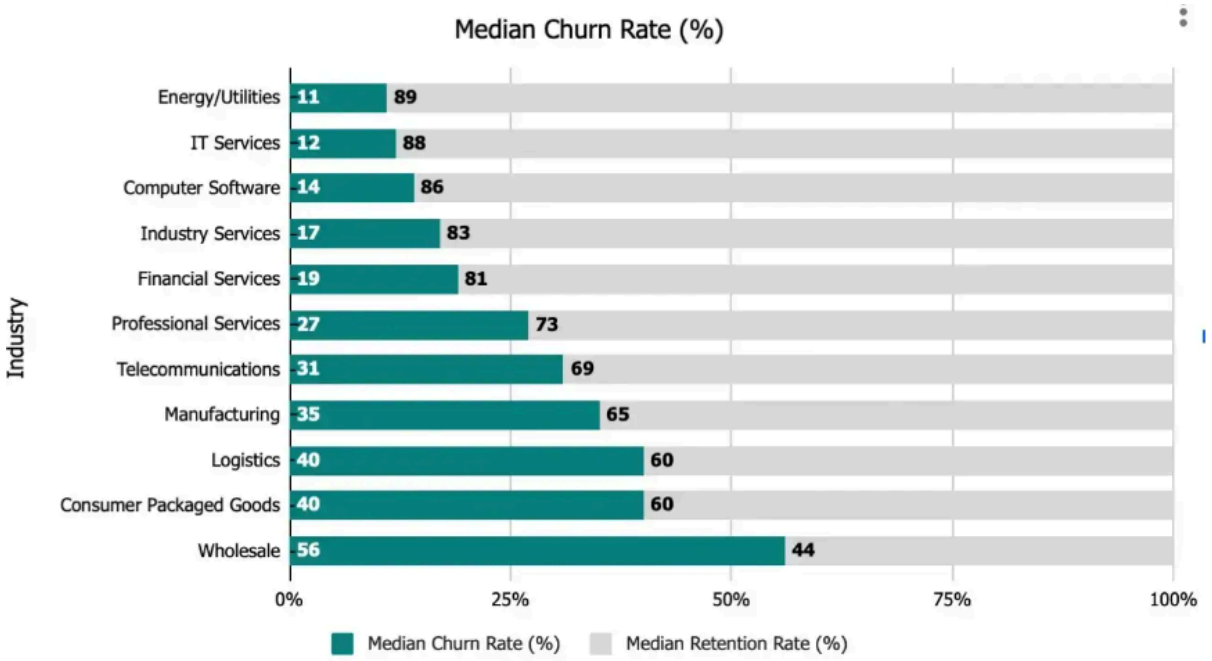

***Gráfico 1**: Churn en diferentes industrias*

Grandes empresas cuentan con aplicaciones móviles junto con una gran nómina de usuarios, y las mismas carecen de un adecuado sistema que logre identificar aquellos que podrían abandonarla, lo que termina incidiendo en severos impactos negativos para las mismas.

Los descuentos ofrecidos a los usuarios activos y la interacción / comunicación con ellos a lo largo de su ciclo de vida suelen carecer de sustento, por lo que no tienen el impacto esperado. Por un lado, los clientes que están por abandonar la plataforma, no reciben el correcto incentivo para permanecer en la misma. Por otro lado, suele ocurrir también que clientes promedio que generan una determinada ganancia para la empresa, reciban un descuento que no les genere ni un incentivo

ni un cambio de comportamiento y por lo tanto tampoco impacte de manera positiva en el ingreso de la compañía.

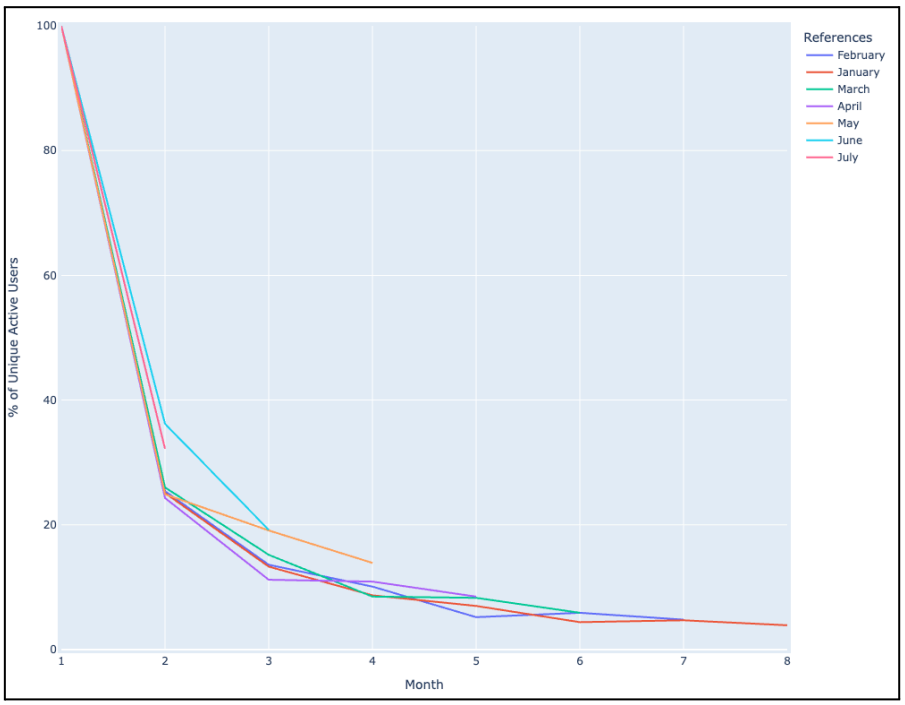

Gráfico 2: *Curvas de retención de una aplicación de un banco virtual en Argentina.*

Al observar las curvas de retención de una aplicación móvil que decide mantener su confidencialidad, se aprecia la caída al mes posterior a la primera activación del orden del 65 al 75%. A su vez, la retención continúa decreciendo de forma muy similar para todos los meses. En el caso del mes que se cuenta con mayor información, el cohort de Enero, se obtiene al mes 8 una retención de 4%. Esta caída tan fuerte al inicio y su continuación en todos los meses, atenta contra la sostenibilidad del negocio.

El churn es un factor importante y de gran preocupación en la industria de aplicaciones móviles porque los usuarios tienen una gran variedad de opciones y con facilidad pueden cambiar a otra aplicación si no están satisfechos o la competencia ofrece un producto o servicio mejor. Dado que la adquisición de nuevos usuarios es costosa y requiere importantes esfuerzos en marketing y publicidad, tratar el churn se vuelve un aspecto crítico para las empresas de aplicaciones móviles, considerando que los usuarios son su activo más valioso. La retención de usuarios y la reducción del churn son fundamentales para mantener una base de usuarios activa y satisfecha que genere ingresos a largo plazo.

1.2. LTV, Retención y Churn

Las aplicaciones móviles han revolucionado la forma en que interactuamos con las marcas y productos, y en este mundo altamente competitivo, es esencial para las empresas tener una estrategia sólida de retención de usuarios para maximizar el valor de vida del cliente (Lifetime Value) y minimizar la tasa de abandono (churn).

Profundizando un poco, el Lifetime value (LTV) es una métrica que representa la cantidad de ingresos que se espera que un cliente genere durante el tiempo que permanezca como cliente de una empresa. Se puede calcular de diversas formas, como por ejemplo multiplicando el valor promedio de un pedido por el número de pedidos que se espera que realice durante su "vida útil" como cliente. El LTV adquiere una gran importancia a la hora de ayudar a las empresas a entender el valor a largo plazo de sus clientes y a tomar decisiones sobre la inversión en adquisición y retención de clientes.

El LTV se calcula multiplicando el ARPU (ingreso promedio por usuario) por la duración media de la relación (L) y la tasa de retención (r):
LTV = ARPU * (L / (1 - r))

Por otro lado, la retención se refiere a la capacidad de retener a los clientes actuales que utilizan un producto o servicio (De Bock, 2011, p. 10). Se mide como un porcentaje del número total de clientes que se mantienen durante un período determinado. Una alta tasa de retención significa que una empresa está reteniendo a sus clientes existentes y puede estar haciendo un buen trabajo al proporcionar un buen servicio, una experiencia de usuario atractiva, productos de calidad y un valor agregado que los clientes no pueden encontrar en otros lugares.

La pérdida de clientes, conocida como churn, se lo define como la inversa a la retención y se refiere a la ocurrencia de un evento en el que un cliente deja de utilizar los productos o servicios de una empresa (Johny & Mathai, 2017, p. 5). Una alta tasa de churn significa que una empresa está perdiendo clientes y es posible que deba tomar medidas para mejorar la retención de usuarios.

Las tres métricas mencionadas se relacionan entre sí, donde el LTV se enfoca en el valor a largo plazo de un cliente, la retención se enfoca en la capacidad de una empresa para mantener a sus clientes existentes, y el churn se enfoca en la cantidad de clientes que abandonan una empresa durante un período de tiempo determinado. Todas estas métricas son importantes para comprender el rendimiento y la sostenibilidad a largo plazo de una empresa.

Destacando la estrecha relación sobre las métricas mencionadas, se estima que una disminución del 5 por ciento en la tasa de abandono de un negocio (o entonces de aumento en la tasa de retención), puede aumentar el LTV de un cliente en al menos un 35 y posiblemente hasta un 95 por ciento, dependiendo del dominio del negocio (Harvard Business School Press, 2011, p. 17).

Finalmente, con la explicación y la introducción de las principales definiciones, se continúa con el análisis de importancia de cada una de ellas y se lo enfrenta a la adquisición de usuarios.

1.3. Retención vs Adquisición de Usuarios

“La adquisición y la retención son como las dos alas de un pájaro: si falta alguna de las dos, el vuelo se vuelve imposible. De la misma manera, debes adquirir y luego retener a tus clientes para lograr el éxito.” (Ramirez, 2021).

Tanto la adquisición como la retención son importantes para una empresa, pero la retención de usuarios para muchos se suele considerar más crítica que la adquisición de nuevos usuarios. Hay varias razones para esto:

1. Costo: Las investigaciones indican que la adquisición de clientes en una empresa, es mucho más costosa que la retención. Según estudios, puede costar hasta 5 veces más adquirir un nuevo cliente que mantener uno existente (Huang & Kechadi, 2013, p. 5635–5647). La adquisición de usuarios puede requerir una inversión significativa en marketing y publicidad, mientras que la retención de usuarios se puede lograr a través de mejoras en la experiencia del usuario y la satisfacción del cliente.
2. Ingresos: Los usuarios existentes suelen generar más ingresos que los nuevos usuarios. Los usuarios que han utilizado un producto o servicio durante un período prolongado de tiempo son más propensos a realizar compras adicionales y a gastar más dinero que los nuevos usuarios.
3. Fidelidad: Los usuarios que se mantienen en una empresa durante un largo período de tiempo tienden a desarrollar una lealtad a la marca. Esto significa que son más propensos a recomendar la empresa a otros y a permanecer fieles a la empresa incluso si se presentan alternativas similares.

En resumen, aunque es importante adquirir nuevos usuarios para el crecimiento de la empresa, es igualmente importante mantener a los usuarios existentes satisfechos y comprometidos para garantizar un crecimiento sostenible y rentable.

Ejemplo práctico

Ahora, con un ejemplo práctico llevado a una empresa real se mostrará de forma hipotética, cómo afectaría un 10% de aumento en la retención de usuarios frente a un 10% de aumento en la adquisición de usuarios. La cantidad de empleados iniciales se desconoce pero se supone para poder plantear el caso y darle vida al ejemplo. La empresa cuenta con 500 usuarios en el periodo inicial, la cual logra incorporar 140 nuevos clientes por periodo inicialmente y cuenta con un 80% de retención.

En este estudio, analizamos dos posibles escenarios de la empresa Big Gym que cuenta con su aplicación BIGG de modelo de suscripción, durante un período de 12 meses. El escenario base representa la situación actual de la empresa, mientras que los escenarios 1 y 2 introducen cambios específicos para evaluar su impacto en el crecimiento de usuarios.

Escenario Base

En el escenario base, la empresa cuenta con una tasa constante de retención de usuarios y una adquisición constante de nuevos usuarios, respecto su escenario base. Durante el período de 12 meses, se observa un aumento gradual en el número total de usuarios, como se muestra en la tabla.

Tabla 1: *Escenario base de la empresa*

Escenario BASE													
Mes	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
Retenidos	500	400	432	458	478	494	508	518	526	533	539	543	546
Nuevos		140	140	140	140	140	140	140	140	140	140	140	140
Final		540	572	598	618	634	648	658	666	673	679	683	**686**

Escenario 1: Aumento del 10% en Retención

En este escenario, se implementa una estrategia para mejorar la retención de usuarios en un 10%. Esto puede lograrse mediante la introducción de nuevas funciones en la aplicación, como recordatorios personalizados, contenido exclusivo para usuarios activos y programas de fidelización. Como resultado, se observa un incremento significativo en el número total de usuarios al final del período de 12 meses en comparación con el escenario base.

Tabla 2: *Escenario con 10% de aumento en retención*

Escenario 1		Aumento 10% en Retención											
Mes	0	1	2	3	4	5	6	7	8	9	10	11	12
Retenidos	500	450	531	604	670	729	782	830	873	911	946	978	1006
Nuevos		140	140	140	140	140	140	140	140	140	140	140	140
Final		590	671	744	810	869	922	970	1013	1051	1086	1118	**1146**

Escenario 2: Aumento del 10% en Adquisición de Usuarios

En este escenario, se enfoca en aumentar la adquisición de nuevos usuarios en un 10%. Esto puede lograrse mediante estrategias de marketing más agresivas, colaboraciones con influencers en el ámbito del fitness y campañas promocionales. Aunque se observa un aumento en el número de nuevos usuarios, el impacto en el crecimiento total de usuarios es menor en comparación con el escenario de aumento de retención.

Tabla 3: *Escenario con 10% de aumento en adquisición*

Escenario 2		Aumento 10% Adquisición de Usuarios											
Mes	0	1	2	3	4	5	6	7	8	9	10	11	12
Retenidos	500	400	443	478	505	528	545	559	571	580	587	593	597
Nuevos		154	154	154	154	154	154	154	154	154	154	154	154
Final		554	597	632	659	682	699	713	725	734	741	747	**751**

Análisis y Conclusiones

Luego de un periodo de 12 meses, un aumento del 10% en la retención de usuarios aumentó un 67% la base de usuarios, frente al incremento del 10% en la adquisición de usuarios que significó casi un 9,5% de incremento. En este ejemplo, se puede ver que el impacto de mejorar la retención fue casi 7 veces mayor que el impacto de la mejora en la adquisición.

La comparación entre los escenarios resalta la importancia de la retención de usuarios para el crecimiento sostenible de la aplicación BIGG. Aunque aumentar la adquisición de nuevos usuarios puede generar un crecimiento inicial, mejorar la retención resulta en un aumento acumulativo en el número total de usuarios a lo largo del tiempo. Esto sugiere que la empresa debería priorizar estrategias que fomenten la lealtad de los usuarios existentes, como mejorar la experiencia del usuario y ofrecer contenido personalizado.

1.4. Tipos de Churn en aplicaciones móviles

Según Henry (2023), existen dos maneras por las cuales un cliente puede abandonar una aplicación móvil, de manera voluntaria o de manera involuntaria.

- Voluntaria: es el tipo más común, donde un usuario decide por su propia cuenta abandonar la aplicación móvil. Esto se puede deber a motivos como la falta de interés, un cambio en sus necesidades, el precio de la suscripción, entre otras.
- Involuntaria: un cliente abandona la aplicación móvil por más que no tuvo esa intención. Esto se puede deber por ejemplo a un error de la aplicación que no permita su uso, un problema con los pagos en caso de suscripción, entre otras.

Es fundamental que las empresas comprendan los distintos tipos de churn y adopten estrategias efectivas para reducir su impacto en el crecimiento y la rentabilidad de la aplicación. Más adelante, se mencionará el tipo de churn a considerar según el contexto en el que se encuentra inmerso el caso.

2. Contexto

2.1. Inicio del proyecto

Grandes empresas cuentan con aplicaciones móviles sin control ni entendimiento de sus usuarios, y no presentan un adecuado sistema que les permita anticiparse si un usuario podría abandonar o dejar de usar la aplicación. En este trabajo se cuenta con los datos transaccionales de una plataforma móvil de delivery de comida, los cuales serán explicados más adelante.

2.2. Industria de delivery en línea

En la actualidad, las aplicaciones de delivery en línea han revolucionado los hábitos de consumo de la sociedad, reemplazando el pedido por teléfono o por web, lo que ha generado que los flyers y los imanes en las heladeras sean elementos del pasado.

Forbes (2023) informa que las aplicaciones de delivery están en alza. Según Forbes, 4 de cada 10 personas realizan compras de supermercados a través de aplicaciones, y el 90% aseguró que espera

incrementar sus pedidos bajo esta modalidad para el corriente año. Incluso, aquellos que aún no han usado este canal de compra planifican utilizarlo en los próximos meses (75%).

El cambio en los hábitos de consumo impulsado por estas aplicaciones ha transformado la forma en que las personas satisfacen sus necesidades de alimentación y servicios a domicilio. Por tanto, resulta crucial continuar explorando esta industria en constante evolución y comprender el comportamiento de sus usuarios. Considerando la importancia y este crecimiento acelerado en los últimos años, resulta inevitable que a lo largo de este trabajo se siga profundizando en su análisis.

2.3. Posibles razones del churn

Según Maan (2023), en su paper Customer Churn Prediction Model using Explainable Machine learning, existen numerosos desafíos para las empresas a la hora de predecir las razones por las cuales los clientes abandonan, considerando el entorno empresarial altamente competitivo de hoy en día. Algunas de las razones destacadas a continuación son:

- Servicios comparables disponibles en la competencia a precios más competitivos.
- Falta de herramientas y plataformas de BI para proporcionar visibilidad en las áreas de customer pain.
- La digitalización ha abierto y disponibilizado una vara más amplia de productos y servicios.
- La rotación accidental, a veces, puede ocurrir debido a un cambio repentino en la situación financiera del cliente por ejemplo.
- Falta de un modelo de predicción de churn de los clientes de la empresa.
- Falta de estrategias de retención de clientes y planes de incentivos para minimizar la rotación.

Teniendo en cuenta los factores anteriores, existe una necesidad significativa de desarrollar un modelo clasificador que pueda predecir con cierta precisión las rotaciones futuras en función de los datos históricos y los patrones de comportamiento del cliente.

2.4. Objetivo del trabajo

La industria de entrega de alimentos en línea es una de las más dinámicas y de mayor crecimiento en el mundo. Según Rocket Lab (2021), empresa de mobile app growth, se espera que esta industria alcance los 25.000 millones de euros a lo largo de 2023 sólo en Europa. Este crecimiento se debe en gran parte al aumento de la demanda de los consumidores de la conveniencia y comodidad, junto con la pandemia.

Sin embargo, este crecimiento también significa que la competencia en esta industria es cada vez más acentuada, lo que pone una mayor presión en las empresas para retener a sus clientes. Según Bejarano (2022), las empresas deben invertir cerca del 10% en Marketing, por lo que la inversión en marketing en Europa podría estar por encima de los 3000 millones de euros para finales del año 2023. Considerando estas cifras, muchas empresas están comenzando a buscar formas de retener a sus clientes existentes para reducir costos y mejorar los resultados. Este enfoque en la retención de clientes es particularmente importante en la industria de delivery en línea, donde a causa de la gran variedad de aplicaciones y oferta, el churn es particularmente alto.

Considerando este impacto y contexto, el objetivo principal de esta tesis es predecir la probabilidad de churn para una empresa de entrega de alimentos en línea en México utilizando algoritmos de machine learning. Esto implica la construcción de un modelo de clasificación binaria que pueda identificar con precisión a los clientes que están en riesgo de abandonar el servicio de entrega.

Por otro lado, con la idea de profundizar en el fenómeno del churn y desarrollar estrategias efectivas de retención, se establecen objetivos secundarios para abordar diferentes aspectos relevantes en el análisis y la discusión.

- Recopilar y analizar datos relevantes a través de interpretación de resultados: Se llevará a cabo una exhaustiva recopilación de datos relacionados con el comportamiento de los clientes en la plataforma. Estos datos serán sometidos a un análisis detallado para identificar patrones significativos que puedan ayudar a comprender mejor el fenómeno del churn. La interpretación de los resultados obtenidos de este análisis proporcionará una comprensión profunda de los factores que contribuyen al abandono de los clientes y guiará el desarrollo de estrategias efectivas de retención.

- Analizar el Churn y las Estrategias de Retención en la Industria: Con un enfoque en los posibles escenarios y estrategias de la empresa a la hora de tomar decisiones se examinará el fenómeno de churn, junto con la evaluación del valor del cliente y la consideración del costo de acciones de

retención. Se abrirá un espacio de discusión para abordar y analizar preguntas recurrentes relacionadas con el Churn y las estrategias de retención en el ámbito de las aplicaciones móviles, buscando darle aplicación a las probabilidades de churn obtenidas previamente.

El trabajo de tesis se enfocará en una aplicación de delivery en línea. Los datos que se presentan son para el país de México, y se considerarán los 31 estados mexicanos

2.5. Mercado y competencia

Considerando que el dataset e información disponible es de México, se analiza su situación.

La industria del delivery en línea en México ha experimentado un crecimiento significativo en los últimos años, y desde el inicio de la pandemia, ha experimentado un auge en la demanda. Según Ramos (2021), en su artículo en Marketing Ecommerce, las aplicaciones de delivery en línea más populares en el mercado mexicano son DiDi Food, Uber Eats y Rappi.

Uber Eats es una aplicación de origen estadounidense que ofrece servicios en más de 70 ciudades mexicanas, y está disponible las 24 horas del día. La aplicación ofrece envíos gratis mensuales por una tarifa fija y descuentos especiales, que incluyen descuentos en viajes en Uber.

DiDi Food es una división de repartos de DiDi, una empresa de origen asiático que ofrece servicios de comida a domicilio en más de 31 ciudades de México, incluyendo ciudades que no están disponibles en otras aplicaciones de entrega de alimentos, como Saltillo, Toluca, Ensenada, y Oaxaca. La aplicación ofrece promociones de descuentos de hasta el 50% en algunos casos.

Rappi es una aplicación colombiana que ofrece servicios de entrega de alimentos, compras en el supermercado y otros servicios adicionales en México. La aplicación ofrece envío gratis en la primera compra y también tiene una opción de suscripción que ofrece envío gratis en todos los pedidos.

En resumen, la industria de entrega de comida a domicilio en línea en México es altamente competitiva, con múltiples aplicaciones ofreciendo diferentes funcionalidades y servicios para los usuarios. Las mencionadas son solo algunas de las empresas de delivery en línea más importantes en México, pero existen muchas otras plataformas en el mercado, cada una con su propia propuesta de valor y enfoque en el mercado.

3. Dataset

Los datos provienen de un MMP (mobile measurement partner). Estas son empresas que ayudan a las aplicaciones a medir el rendimiento de las campañas en los canales de marketing publicitario, las fuentes de medios y las redes publicitarias. Atribuye, recopila y organiza los datos de las aplicaciones para ofrecer una evaluación uniforme de las métricas de rendimiento de las campañas. A partir de esto, se puede adquirir algunos eventos transaccionales de interés para el proyecto de una plataforma de delivery en línea recopilados por un MMP.

Estos MMP se encargan de recopilar toda la información de los usuarios a través de la plataforma. En este caso, los usuarios van a **ver** alguna publicidad por internet, y van a **ingresar** a la publicidad, e **instalan** la aplicación, **abren la app**, luego realizan el **sign up** y comienzan a **ver productos** y pantallas dentro de la aplicación, donde a veces **agregan productos a su carrito**, y luego **finalizan** la orden al realizar la compra (o cancelando).

Se cuenta con un dataset con toda la información histórica de esta aplicación para cada usuario por el periodo de septiembre 2022 a febrero 2023. El dataset contiene 4.157.509 registros.

Específicamente, los eventos que se tienen en cuenta para el estudio y análisis son:

- Registro exitoso → registration_complete
- Login completo → first_login
- Ver orden → select_vieworder
- Ver screen home → home_viewed
- Agregar a carrito → add_to_cart
- Confirmar primera orden → first_order_complete
- Realizar compra → purchase

Estos eventos se presentan en el dataset inicial, el cual luego se utilizará para análisis exploratorio de los datos, feature engineering y modelado del problema. También se cuenta con la información de fecha y hora de la atribución del usuario en la publicidad de internet, y de la instalación. Ambos datos no se cuentan como eventos particulares, sino que se obtienen de campos/variables del dataset. Las columnas presentes en el dataset inicial se detallarán en el siguiente apartado.

A continuación se visualiza una vista previa del dataset:

	event_name	event_time	install_time	attributed_touch_time	state	city	postal_code	device
1175210	purchase	2022-09-01 00:00:10	2022-08-31 23:42:03	2022-08-31 22:01:59	COA	Matamoros	27442	phone
1175220	add_to_cart	2022-09-01 00:00:35	2022-08-31 23:42:03	2022-08-31 22:01:59	COA	Matamoros	27442	phone
1175178	first_login	2022-09-01 00:00:47	2022-09-01 00:00:11	2022-08-31 23:57:55	MEX	Valle De Chalco Solidaridad	56610	phone
1175177	registration_complete	2022-09-01 00:01:17	2022-09-01 00:00:24	2022-08-31 20:19:24	YUC	Nueva Yucalpeten	97320	phone
1175176	first_login	2022-09-01 00:02:34	2022-09-01 00:02:01	2022-08-31 18:46:40	SON	Hermosillo	83296	phone
...	...	...	...	...	...	...	...	...
2098748	registration_complete	2023-01-19 23:59:47	2023-01-19 23:57:58	2023-01-19 00:43:28	GUA	Lagunillas	37669	phone
2098747	add_to_cart	2023-01-19 23:59:50	2022-11-19 21:17:00	2022-11-19 19:32:41	JAL	Tonala	45410	phone
2098746	add_to_cart	2023-01-19 23:59:53	2023-01-19 23:21:10	2023-01-19 12:41:09	NLE	Monterrey	64180	phone
2098745	add_to_cart	2023-01-19 23:59:54	2022-11-04 18:13:13	2022-11-03 20:59:40	BCN	Tijuana	22450	phone
2098744	add_to_cart	2023-01-19 23:59:55	2022-12-04 12:17:22	2022-12-04 09:39:11	VER	Cuitlahuac	94915	phone

Gráfico 3: *Vista previa del dataset inicial.*

3.1. Dimensiones

Como se mencionó anteriormente, se visualizan las dimensiones/columnas del dataset inicial:

Tabla 4: *Columnas, descripción y valores de ejemplo del dataset.*

Nombre	Descripción	Valor de ejemplo
event_name	Nombre del evento asociado	login_done, add_to_cart
event_time	Día y hora a la que se realiza el evento	2022-11-19T23:55:17.000Z
attributed_touch_time	Día y hora a la que fue atribuído el usuario por una publicidad en internet	2022-11-19T09:51:09.000Z
install_time	Día y hora en la que el usuario instaló la aplicación	2022-11-19T23:55:17.000Z
state	Estado en el que la persona realiza el evento	GUA (Guanajuato)
city	Ciudad en el que la persona realiza el evento	Mexico City
postal_code	Código postal del lugar donde la persona realiza el evento	98609
device	Tipo de dispositivo de la persona que realiza el evento	phone,tablet

Se cuenta con tres campos que hacen referencia al tiempo: install_time, event_time y attributed_touch_time, y los mismos son utilizados en el análisis de datos de aplicaciones móviles. ‘install_time’ se refiere al momento en que un usuario descarga e instala una aplicación en su dispositivo. ‘event_time’, por otro lado, se refiere al momento en que ocurre una acción específica dentro de la aplicación, como hacer click en un botón o completar una transacción. Por último, ‘attributed_touch_time’ se refiere al momento en que un usuario interactúa por primera vez con un anuncio o enlace que dirige a la descarga de la aplicación.

3.2. Data Cleansing

Una vez obtenido el dataset inicial, se procede con el proceso de limpieza de datos. El proceso de Data Cleansing es una etapa crucial en cualquier proyecto que involucre el uso de datos. Esta fase se

enfoca en proteger y asegurar la calidad de los datos utilizados, lo que resulta en información más precisa y confiable.

Durante el proceso de Data Cleansing, se evalúan diferentes atributos de calidad de los datos. Entre ellos se encuentran la validez, exactitud, completitud, consistencia y uniformidad.

- ➔ La validez se refiere a la revisión del tipo de datos utilizados en el conjunto de datos. Esto ayuda a asegurar que los datos sean coherentes y estén correctamente definidos en términos de formato y estructura.
- ➔ La exactitud, por su parte, mide qué tan cercano a la realidad es el dato observado.
- ➔ La completitud se enfoca en la cantidad de datos faltantes en cada registro. La presencia de datos faltantes podría afectar la calidad y la utilidad del dataset.
- ➔ La consistencia se enfoca en evitar contradicciones entre distintos registros del mismo conjunto de datos.
- ➔ La uniformidad se refiere a que todos los registros deben utilizar las mismas unidades de medida, criterios de medición y estructuras de datos.

Al enfocarse en la calidad de los datos, se puede entonces asegurar que la información será más precisa, confiable y útil para luego desarrollar los modelos y conseguir resultados de mayor confiabilidad. Una vez asegurados estos atributos en el dataset, se procede a realizar un estudio del mismo.

3.3. Highlights

A continuación, se expondrán algunos resultados de la fase exploratoria de datos.

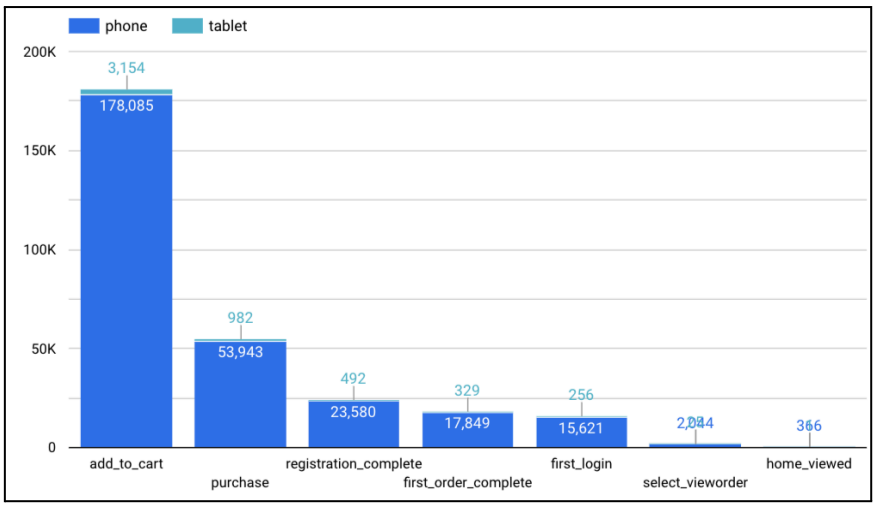

Gráfico 4: *Cantidad de eventos según dispositivo en noviembre 2022.*

Por un lado, como se podría esperar, se puede ver como el mayor porcentaje de los eventos se realizan desde dispositivos celulares. A su vez, a partir del gráfico se puede analizar que una compra promedio suele contener 3 productos.

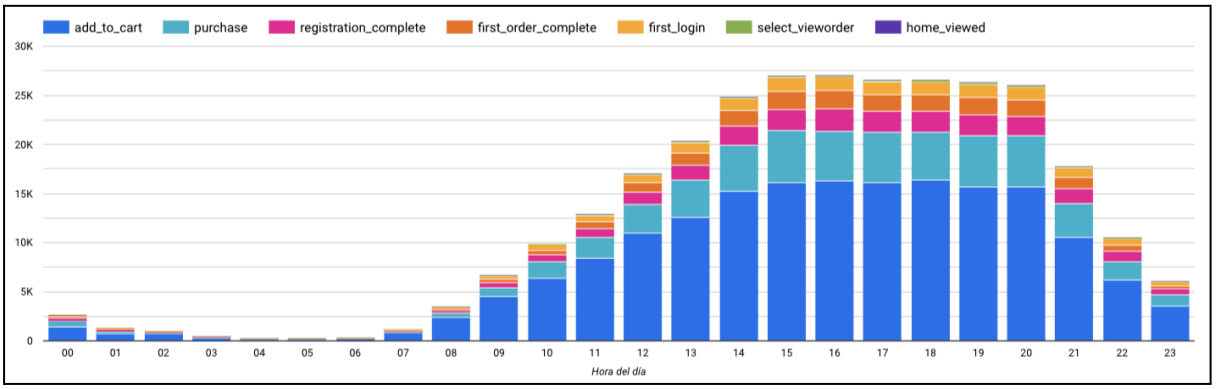

Gráfico 5: *Cantidad de eventos según hora del día y tipo de evento en noviembre de 2022.*

En el gráfico presentado, se puede ver que el mayor porcentaje de eventos se realiza entre las 14hs y las 20hs. A su vez, se puede analizar como el evento add_to_cart predomina en todas las horas del día, mientras que los registros bajan en proporción aún más en las horas menos demandantes.

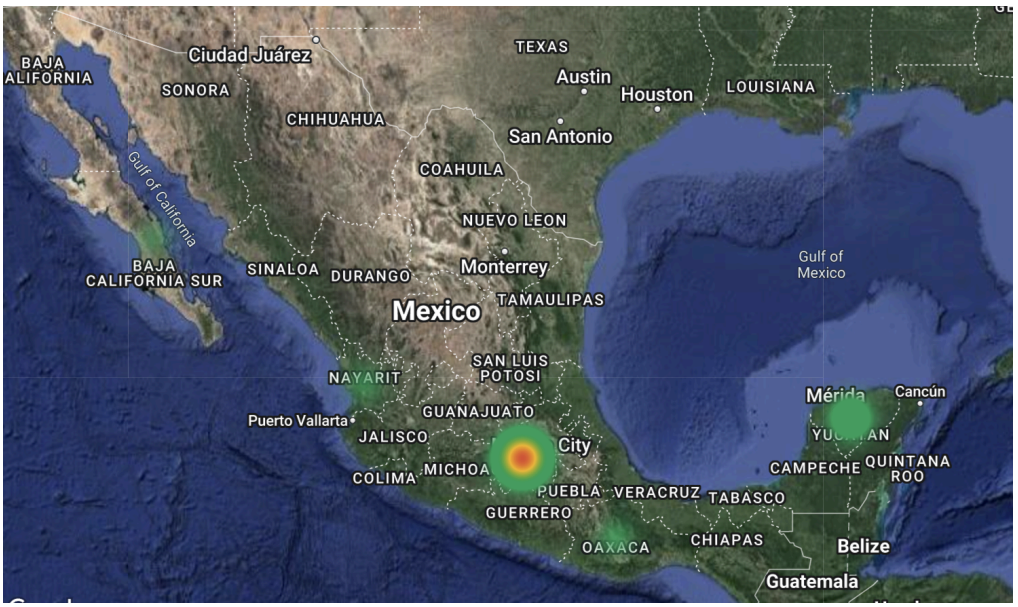

Gráfico 6: *Mapa satelital de calor según ubicaciones de eventos en el mes de noviembre.*

Siguiendo con el entendimiento del dataset, en el mapa de calor presentado podemos ver que la mayor cantidad de eventos provienen de 'Mexico City', y luego lo sigue el estado de 'Yucatan'.

A continuación, se interioriza un poco más en el comportamiento y entendimiento de los usuarios. Se busca entender las variables al alcance y detectar patrones para poder alimentar y nutrir los algoritmos de aprendizaje. Se continúa considerando la actividad de los usuarios en el mes de noviembre.

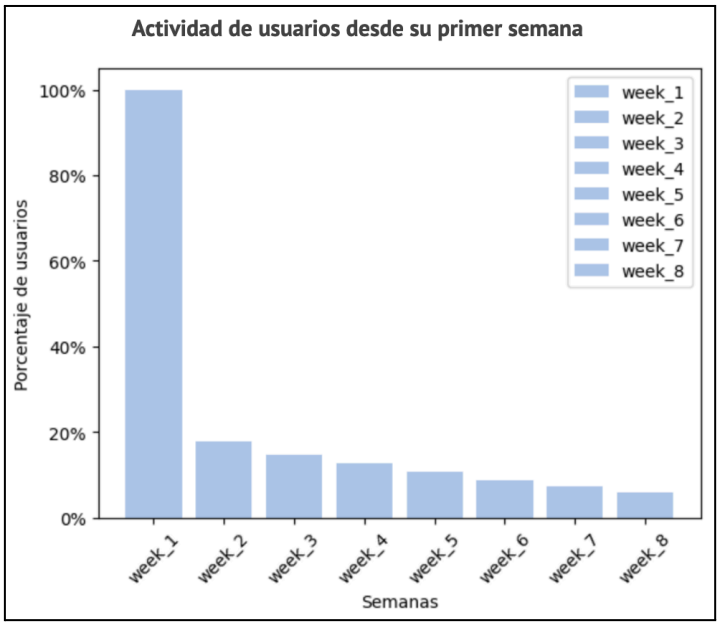

Gráfico 7: *Porcentaje de usuarios activos a lo largo de sus primeras semanas.*

Se puede analizar como más del 80% de los usuarios dejan de tener actividad luego de su primera semana de inicio. Algunas preguntas que se podrían realizar: ¿Son estos usuarios valiosos? ¿O solo ingresaron a la aplicación por una promoción o cupón de descuento?

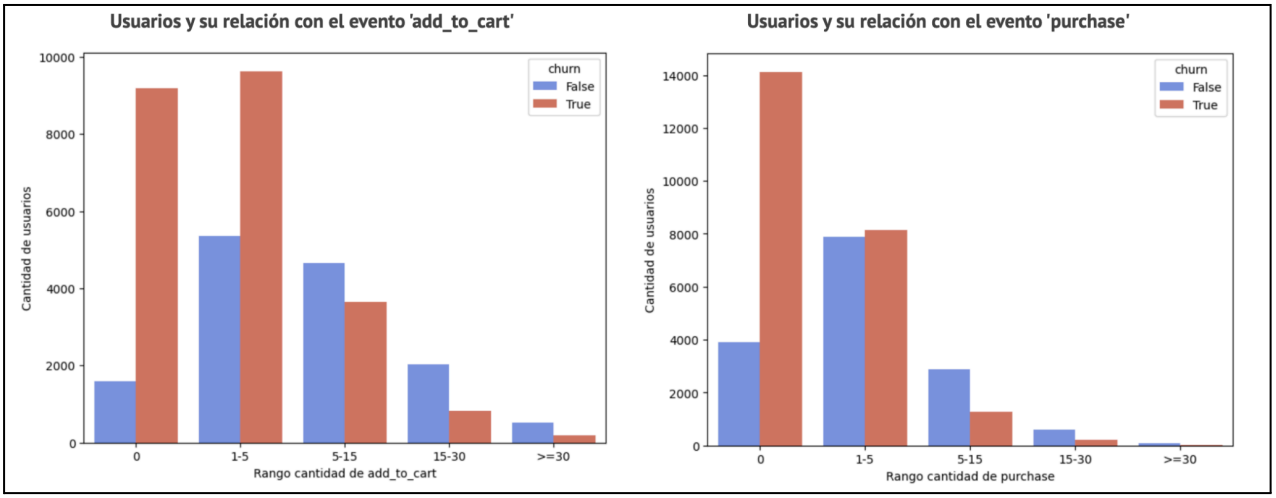

Gráfico 8: *Cantidad de usuarios que realizan determinados eventos según rangos de cantidad.*

A partir del gráfico, se puede observar que la mayor parte de los usuarios que no realizan ningún evento de agregar el carrito, tienden con gran probabilidad a abandonar la plataforma el siguiente periodo.

Por otro lado, realizar el evento compra está ligado a mantenerse activo en la plataforma por el siguiente periodo. A diferencia del evento agregar al carrito, al realizar al menos una compra baja en gran medida la tendencia a realizar Churn. Se puede interpretar también que, gran parte de los usuarios más valiosos (los que más compras realizan), suelen continuar en el siguiente periodo.

Con ambos eventos se detecta una relación con el churn, y se puede intuir que ambas podrían tener un gran importancia predictiva a la hora de correr un modelo.

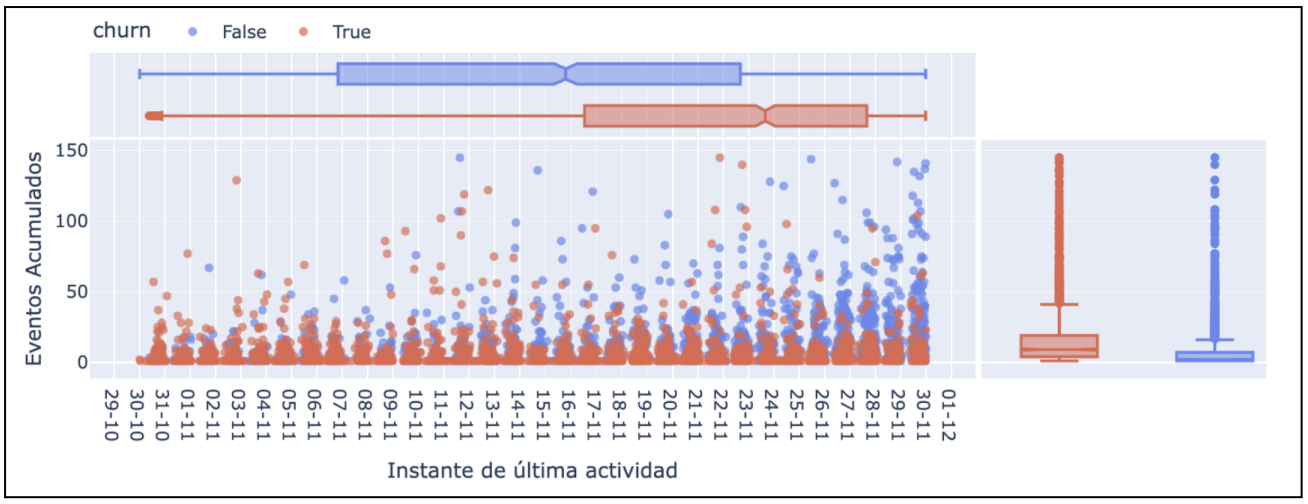

Gráfico 9: *Cantidad de transacciones acumuladas al momento del último evento del periodo.*

A partir de visualizar el gráfico, se puede analizar que una gran masa de usuarios realiza eventos hasta el fin del mes. Estos usuarios tienden a mantenerse activos en el siguiente periodo.

A su vez, se puede ver que los usuarios que más cantidad de eventos realizan, son usuarios que suelen mantenerse activos en el siguiente periodo. En contraste, los usuarios con menor cantidad de eventos acumulados tienden a abandonar, lo que se podría interpretar nuevamente que usuarios de menor valor son los que abandonan periodo a periodo.

Las variables presentadas en el gráfico se pueden considerar como atributos que potencialmente contribuirían a una clasificación más precisa por parte del modelo.

3.4. Caso de estudio

A continuación, se visualiza un usuario aleatorio en el dataset que realiza cantidad de transacciones promedio, junto con sus eventos a lo largo de su transcurso por la aplicación. A este usuario le damos un nombre ‘ Juan’, para poder citarlo a lo largo del trabajo.

	event_name	event_time	install_time	attributed_touch_time	state	city	postal_code	device
498044	first_login	2022-10-10 20:55:21	2022-10-10 20:55:08	2022-10-10 19:49:55	JAL	Zapopan	45110	phone
498039	add_to_cart	2022-10-11 15:11:41	2022-10-10 20:55:08	2022-10-10 19:49:55	QUE	El Mirador	76246	phone
498038	purchase	2022-10-11 15:12:24	2022-10-10 20:55:08	2022-10-10 19:49:55	QUE	El Mirador	76246	phone
498037	first_order_complete	2022-10-11 15:12:37	2022-10-10 20:55:08	2022-10-10 19:49:55	QUE	El Mirador	76246	phone
498086	add_to_cart	2022-10-12 08:52:49	2022-10-10 20:55:08	2022-10-10 19:49:55	QUE	El Mirador	76246	phone
...	...	...	...	...	...	...	...	...
2184043	purchase	2023-01-17 11:20:49	2022-10-10 20:55:08	2022-10-10 19:49:55	AGU	Vina Antigua	20908	phone
2133753	add_to_cart	2023-01-18 21:21:04	2022-10-10 20:55:08	2022-10-10 19:49:55	MIC	Morelia	58230	phone
2133741	purchase	2023-01-18 21:21:19	2022-10-10 20:55:08	2022-10-10 19:49:55	MIC	Morelia	58230	phone
2133549	add_to_cart	2023-01-18 21:26:35	2022-10-10 20:55:08	2022-10-10 19:49:55	MIC	Morelia	58230	phone
2133532	purchase	2023-01-18 21:27:00	2022-10-10 20:55:08	2022-10-10 19:49:55	MIC	Morelia	58230	phone

239 rows × 8 columns

Gráfico 10: *Transacciones de Juan, usuario de estudio presente en el dataset.*

En la Gráfico se puede observar al comportamiento de un determinado usuario en el tiempo, donde vio una publicidad y fue atribuido el día 2022-10-10 a las 19:49 en la ciudad de Zapopan. Luego este usuario realiza la instalación el mismo día a las 20:55 y segundos después completa el login. A las 15:11 del día siguiente, Juan agrega su primer artículo al carrito para luego ejecutar la compra y realizar confirmación. En los últimos datos de este usuario, podemos ver que algunos meses después continúa realizando compras de un único artículo. A continuación, se visualizan los eventos a lo largo del tiempo activo del usuario.

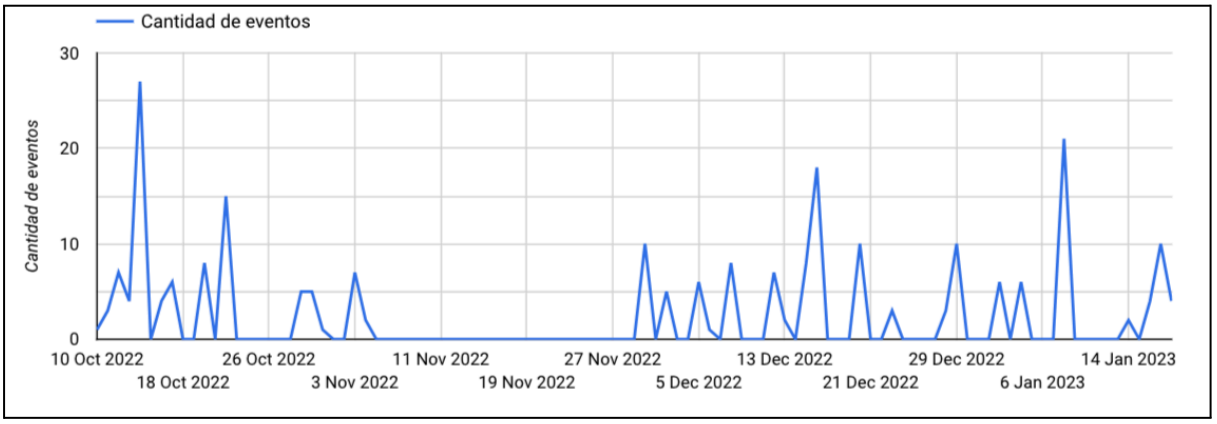

Gráfico 11: *Cantidad de eventos en el transcurso activo del usuario de estudio.*

Ahora se necesita llevar toda la información de este usuario y de los demás, a un único dataset que pueda recopilar cada historia de los usuarios en un mismo registro.

4. Modelado

4.1. Primer modelado

En la etapa de modelización, se busca comenzar a llevar todos los datos de un mismo usuario a un solo registro extrayendo la mayor información posible: esto se debe a que actualmente el dataset inicial presenta múltiples eventos/registros para un mismo usuario, dando la imposibilidad de poder comenzar con la experimentación de algoritmos de aprendizaje supervisado. En la búsqueda de extraer la mayor cantidad de información posible para cada usuario a partir de un único registro, se crearán luego diversas variables que aporten información y vayan describiendo el comportamiento del usuario.

Se extrae entonces toda la base de usuarios del data set inicial, junto con sus propiedades estáticas las cuales no varían a lo largo de cada evento/registro para el mismo usuario:

- *Attributed Touch Time*
- *Attributed Touch Type*
- *Install Time*
- *Device Category*

Una vez que se logra armar un dataset con un solo registro por usuario y continuando con la etapa de feature engineering, se procede a la técnica de agregación de datos en la búsqueda de extraer la mayor cantidad de información posible para cada usuario.

Se busca empezar a generar la mayor cantidad de variables posibles que puedan alimentar y ayudar a realizar un mejor entrenamiento para luego comenzar con la experimentación de diferentes algoritmos de aprendizaje supervisado.

4.2. Propiedades en el tiempo

Una vez extraída la información anterior, se comienza a extraer las propiedades relacionadas al tiempo del usuario:

- *Fecha del primer y último evento realizado por el usuario.*

- *Hora más temprano y más tarde que haya realizado un evento el usuario.*
- *Cantidad de eventos totales realizados en todo el periodo.*
- *Cantidad de cada evento en particular realizados en todo el periodo.*

Por otro lado, se crean columnas que van describiendo el comportamiento del usuario a través del tiempo:

- *Cantidad de eventos por semana a lo largo del tiempo desde su instalación.*

Siguiendo con el tiempo, se crean columnas que vayan describiendo la diferencia de tiempo entre los diferentes datos que se cuentan del usuario:

- *Diferencia entre atribución e instalación.*
- *Diferencia entre instalación y primer evento.*
- *Diferencia entre primer evento y último evento.*

Finalmente, del campo 'install time', se crean múltiples columnas para extraer la mayor información posible de la misma: mes, semana, semana del mes, día del mes, día de la semana.

4.3. Propiedades estadísticas

Ahora, se comienza a extraer las propiedades estadísticas del dataset inicial para nutrir al dataset donde luego entrenará el modelo. Se extrae la siguiente información:

- *Promedio de eventos por semana/mes.*
- *Número máximo de eventos realizados en un día.*
- *Desviación estándar del número de eventos por semana.*
- *Promedio de diferencia de tiempo entre eventos.*
- *Desviación estándar de diferencia de tiempo entre eventos.*
- *Mínima y máxima diferencia de tiempo entre un evento y otro.*

4.4. Representación de usuario modelo

Luego de haber realizado las transformaciones respectivas, se visualiza al usuario Juan con su información en un mismo registro:

Tabla 5: *Porción de información extraída de usuario modelo Juan.*

total_events	average_events_per_week	min_events_per_day	event_week_2	std_events_per_week	device	latest_event_time	oldest_event_time	first_login	purchase	
186	20.67	27	28.00	12.87	phone	2022-12-29T21:09:36.000Z	2022-10-10T20:55:21.000Z	1	35	20

4.5. Análisis Descriptivo

En esta sección se realizará un análisis detallado de las variables independientes luego de realizar el modelado e ingeniería de atributos, previamente a entrenar el modelo. Cada una de estas variables jugará un papel fundamental en la predicción del fenómeno de Churn. Se presenta una tabla con una descripción de cada variable, incluyendo su nombre, tipo de dato, una breve explicación de su significado, y los posibles valores que puede asumir, junto con sus límites.

Tabla 6: *descripción de las variables independientes luego del modelado.*

Variables Independientes	Tipo	Breve descripción	Valores
usuario	string	identificación del usuario	-
min_time_diff_sec	Float	diferencia minima en segundos entre dos eventos	Enteros entre 0 y 2.5M
deltatime_install_firstevent	Float	diferencia en segundos entre instalación y primer evento	Decimales entre 0 y 1.141490e+08
attributed_touch_time	Float	tiempo en el que el usuario se atribuyó	Valor de tiempo equivalente entre 2019-02-15 y 2022-11-30
deltatime_firstevent_lastevent	Float	diferencia en segundos entre primer evento y último evento	Decimales desde 0 en adelante
diff_from_latest	Entero	diferencia de tiempo en segundos desde el último evento registrado y el evento del usuario	Enteros entre 0 y 2.5M
std_events_per_week	Float	desviación estándar de eventos promedio por semana	Decimales entre 0 y 257.3
event_week_8	Entero	cantidad de eventos realizados en la 8va semana desde su instalación	Enteros entre 0 y 137
install_week_of_month	Entero	número de semana dentro del mes	Enteros entre 1 a 5
oldest_event_hour	Entero	hora en el que el usuario hizo el primer evento	Enteros entre 0 a 23
registration_complete	Booleana	se evalua si el usuario se registro completamente	True, False
event_week_3	Entero	cantidad de eventos realizados en la 3era semana	Enteros entre 0 a 441

		desde su instalación	
event_week_2	Entero	cantidad de eventos realizados en la 2da semana desde su instalación	Enteros entre 0 a 361
oldest_event_time	Entero	tiempo en el que se realizó el último evento	Valor de tiempo equivalente entre 2022-11-01 y 2022-11-30
first_login	Entero	cantidad de eventos 'first_login'	Decimales entre 0 y 17
install_hour	Entero	hora en la que se realizó la instalación	Enteros entre 0 a 23
first_order_complete	Float	se evalúa si el usuario realizó la última orden	True, False
install_time	Entero	tiempo en el que el usuario instaló la aplicación	Valor de tiempo equivalente entre 2019-04-10 y 2022-11-30
add_to_cart	Entero	cantidad de eventos 'add_to_cart'	Decimales entre 0 y 1065
event_week_4	Entero	cantidad de eventos realizados en la 4ta semana desde su instalación	Enteros entre 0 a 583
mean_time_diff_sec	Float	segundos promedio de diferencia entre eventos	Decimales entre 0 y 4.175286e+06
std_time_diff_sec	Float	desviación estándar de diferencia minima en segundos entre dos eventos	Decimales entre 0 y 5.904746e+06
select_vieworder	Entero	cantidad de eventos 'select_vieworder'	Enteros entre 0 y 35
latest_event_time	Float	tiempo en que el usuario realizo el último evento	Valor de tiempo equivalente entre 2022-11-01 y 2022-11-30
average_events_per_month	Entero	cantidad de eventos promedio por mes	Enteros entre 1 a 538.5
event_week_5	Entero	cantidad de eventos realizados en la 5ta semana desde su instalación	Enteros entre 0 a 221
event_week_6	Entero	cantidad de eventos realizados en la 6ta semana desde su instalación	Enteros entre 0 a 232
event_week_7	Entero	cantidad de eventos realizados en la 7ma semana desde su instalación	Enteros entre 0 a 133
event_week_1	Entero	cantidad de eventos realizados en la 1era semana desde su instalación	Enteros entre 1 a 469
latest_event_hour	Entero	diferencia en segundos entre instalación y attribution touch time	Enteros entre 1 a 12
install_month	Entero	número de mes del momento de la instalación	Enteros entre 1 a 12
max_events_per_day	Entero	máxima cantidad de eventos en un día	Enteros entre 1 a 575
min_events_per_day	Entero	mínima cantidad de eventos en un día	Enteros entre 0 a 15
install_weekday	Entero	díá en la semana en la que hizo la instalación	Enteros entre 0 a 6
deltatime_attrtouchtime_install	Float	diferencia en segundos entre instalación y y attribution touch time	Decimales entre 0 y 604793
install_week	Entero	semana en el año en la que se realizó la instalación	Enteros entre 1 a 53
home_viewed	Entero	cantidad de eventos 'home_viewed'	Enteros entre 0 a 52

purchase	Entero	cantidad de eventos 'purchase'	Enteros entre 0 a 154
max_time_diff_sec	Float	diferencia máxima en segundos entre dos eventos	Decimales entre 0 y 8.786385e+06
average_events_per_week	Float	cantidad de eventos totales promedios por semana	Decimales entre 0 y 58
install_day	Entero	día en el año en la que se realizó la instalación	Enteros entre 1 a 365
total_events	Entero	cantidad total de eventos realizados	Enteros entre 1 a 1077
device_phone	Booleana	se evalua si el usuario tiene teléfono movil	True, False
device_tablet	Booleana	se evalua si el usuario tiene tablet	True, False
attributed_touch_type_click	Booleana	se evalua si se atribuye la instalación del usuario a un click en un add	True, False
attributed_touch_type_impression	Booleana	se evalua si se atribuye la instalación del usuario a una impression en un add	True, False

4.6. Análisis Exhaustivo

A continuación, antes de proceder a la implementación de un modelo con el dataset ya preparado, se procede a realizar un análisis exhaustivo de algunas variables que se consideran importantes y aún no fueron analizadas. Estas variables van a entrenar al modelo como variables independientes, y también se va a estudiar su relación con la variable dependiente. Dado que se va a estudiar la actividad de los usuarios con los que se entrena el modelo, se va a considerar únicamente a los usuarios que instalaron la aplicación hasta fines de noviembre de 2022 o antes, esto luego será explicado más detalladamente.

Variable 'Install_time'

install_time
2022-11-26 19:51:57
2022-11-07 14:59:45
2022-10-13 14:41:53
2022-10-08 21:00:13
2022-10-20 15:31:43

Gráfico 12: *Muestra de la variable 'install_time'.*

Esta variable representa el día y el horario en el que el usuario se instala la aplicación, y es de tipo 'datetime'. Del dataset que se presenta, el usuario que tiene la instalación más lejana es en '2019-04-10 00:31:56', y la más reciente es '2022-11-30 23:59:29'.

Ahora se visualiza como evolucionó la cantidad de usuarios según la fecha de instalación:

Gráfico 13: *Evolución acumulada de usuarios según 'install_time'.*

Se puede observar que la mayor parte de los usuarios instalaron la aplicación en el último periodo de tiempo. Ahora se procede a separar la curva según si los usuarios realizaron churn en el mes de diciembre de 2022.

Gráfico 14: *Evolución acumulada de usuarios según 'install_time' and Churn.*

Del gráfico se puede observar que los usuarios más antiguos se distribuyen de manera equitativa entre aquellos que han realizado churn y los que no lo han hecho. En contraste, a medida que nos acercamos a la fecha en la que se evalúa el churn, se observa una tendencia creciente de churn entre los usuarios más recientes al mes de diciembre que han instalado la aplicación.

A continuación, se visualiza las instalaciones según la hora en el día, y se evalúa su relación con la variable dependiente 'Churn'. También se grafica el Churn Ratio (Usuarios Churn / Total de usuarios), para poder evaluar mejor la tendencia.

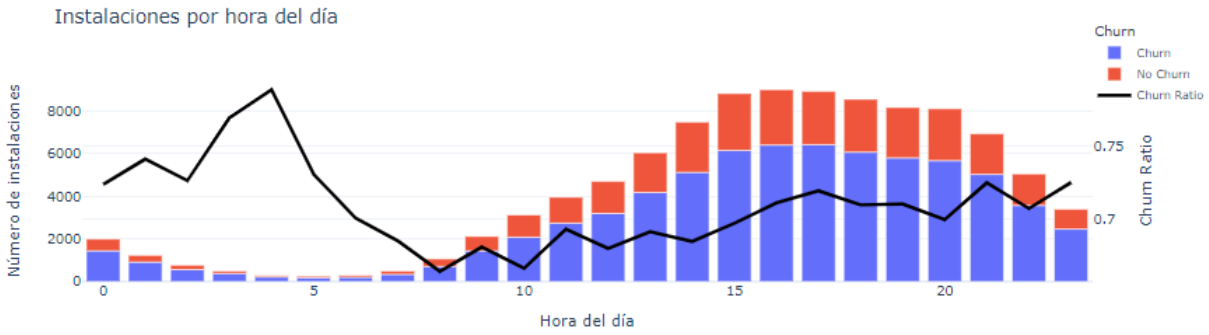

Gráfico 15: *Instalaciones realizadas según hora en el día y Churn.*

A partir del gráfico, se puede analizar que los usuarios que instalan la aplicación durante la mañana (8hs a 15hs aproximadamente) suelen tener menor tendencia a realizar Churn. Por otro lado, los usuarios con mayor tendencia a Churn se encuentra su instalación entre las 3hs y 6hs. Sin embargo, es importante tener en cuenta que las instalaciones durante esas horas son notablemente más bajas, lo que puede afectar la fiabilidad de este dato como representación completa del comportamiento de los usuarios. De todas formas, se espera que esta variable contribuya al aprendizaje del modelo y a mejorar la precisión de las predicciones.

Variable 'deltatime_install_firstevent'

deltatime_install_firstevent
49.0
81545.0
231.0
41.0
17.0

Gráfico 16: *Muestra de la variable 'deltatime_install_firstevent'.*

Esta variable representa la diferencia en segundos entre la instalación y el primer evento y es de tipo 'float'. Iniciando con el análisis de la variable, se transforma la misma a 'minutos' para una mayor comprensión y se visualiza un boxplot, pudiendo distinguirla con la variable dependiente Churn.

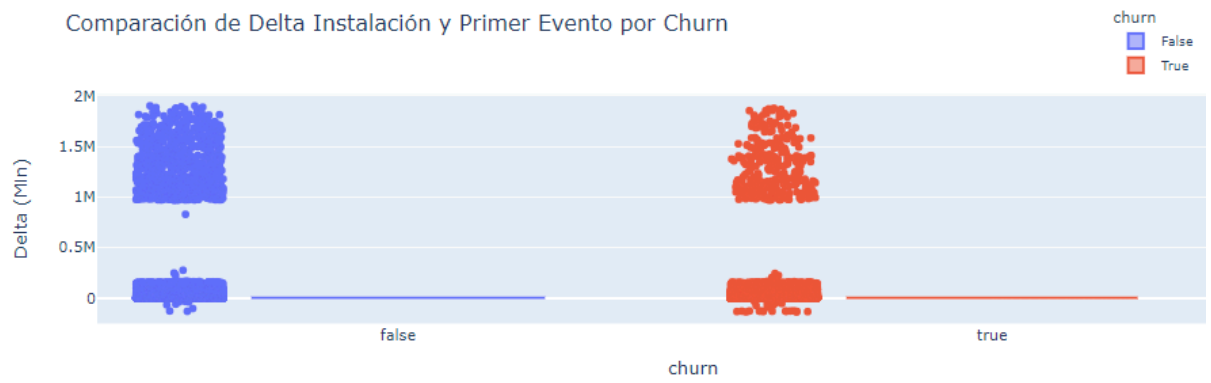

Gráfico 17: *Boxplot de Delta entre Instalación y Primer evento por Churn.*

A simple vista, no se logra distinguir una diferencia entre la distribución de la variable según Churn, donde ambas Medianas se encuentran en 2 minutos. Por otro lado, el promedio para los usuarios Churn es de 7697 minutos y para los usuarios no Churn es de 42649. De todas formas, estos promedios no son valores que tengan un gran peso ya que se ven altamente afectados por outliers, es decir, hay usuarios que han tardado más de 3 años en realizar su primer evento luego de la instalación, y esto causa que la media se aleje en gran medida de su mediana.

El valor mínimo general para la variable es aproximadamente 1 minuto y el valor máximo llega a un equivalente en minutos de 1319 días. A continuación, se visualiza su distribución en un gráfico de columnas para buscar mayor entendimiento.

Gráfico 18: *Distribución de Delta entre Instalación y Primer evento por Churn.*

Como se puede ver, la mayor parte de los usuarios tarda menos de 5 minutos en realizar el primer evento luego de la instalación. De hecho, podemos distinguir que el gráfico hace sentido a que la mediana de esta variable sea igual a 2.

Para entender si hay alguna tendencia respecto del Churn a lo largo del Delta, se procede a graficar el Churn Rate.

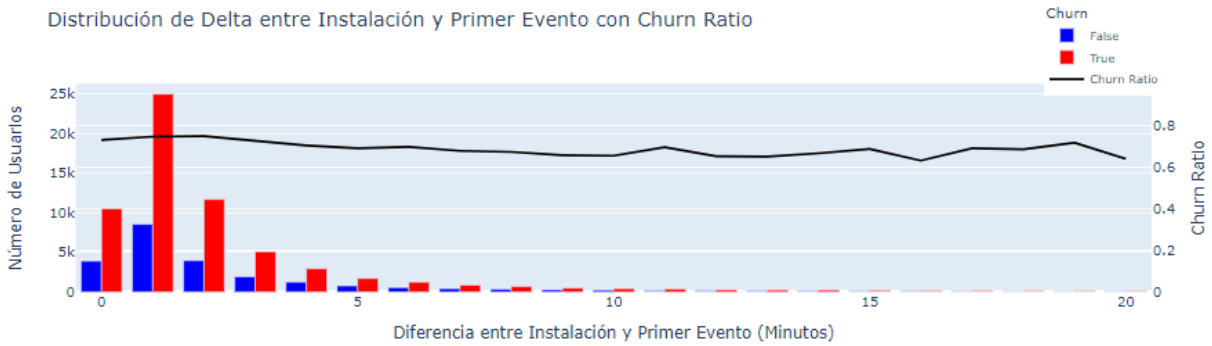

Gráfico 19: *Distribución de Delta entre Instalación y Primer evento por Churn con Churn Rate.*

Aunque se podría decir que para el valor de la mediana, 2 minutos, la cantidad de usuarios Churn es ligeramente mayor que para otros valores de la variable, en realidad no se visualiza una tendencia clara de la variable dependiente churn respecto de la estudiada. En caso de que exista algún patrón o relación no evidente a simple vista en el análisis gráfico, se espera que el algoritmo pueda identificarlo durante el proceso de implementación del modelo.

4.7. Selección de features

En general, la selección de características es un proceso importante que mejora el rendimiento de los algoritmos de aprendizaje automático y reduce la complejidad computacional o el tiempo de ejecución al reducir el número de variables independientes o variables explicativas. Dependiendo del problema y el contexto, una técnica puede ser más adecuada que otra.

Según Arrioja Landa Cosio (2021), el desempeño del modelo puede verse afectado negativamente por la alta dimensionalidad de los datos, lo que ha sido denominado como la "maldición de la dimensionalidad". La comprensión del efecto de la alta dimensionalidad en los datos y cómo afecta a los algoritmos es crucial. Sin embargo, el autor también señala que existen casos en los que tener dimensiones superiores puede ser beneficioso. Además, se destacan técnicas que permiten mitigar el impacto de la alta dimensionalidad en el análisis de datos.

Idris, Rizwan y Khan (2012, p. 1808-1819) han comparado el rendimiento de diferentes métodos de selección de características en la predicción de la rotación de clientes utilizando un algoritmo de

Random Forest. Concluyen que el preprocesamiento apropiado de datos y características es vital para la clasificación.

La estrategia adoptada para mejorar el rendimiento del modelo comenzó con un enfoque inicial que incorporaba todas las características disponibles, sirviendo como punto de referencia. Se realiza en cada algoritmo de entrenamiento que luego se presentarán, para evaluar la importancia de cada característica durante el proceso de entrenamiento.

Durante este análisis, se identificaron características con una contribución baja o nula a la capacidad del modelo para predecir el objetivo. Entre estas características se encontraban variables como 'device', 'state' y 'postal_code', que parecían tener un impacto limitado en la predicción.

Basándose en estos hallazgos, se optó por descartar las características menos importantes y se volvió a ejecutar los algoritmos de entrenamiento utilizando solo las características seleccionadas. El objetivo era comparar el rendimiento de este nuevo modelo con el benchmark original, buscando mejoras significativas en la precisión de la predicción y la complejidad del modelo.

4.8. Tipo de churn a considerar

Luego de haber definido los tipos de churn, se considerarán entonces únicamente los casos de churn de tipo voluntario; cuando un usuario deja de tener actividad / utilizar la plataforma por un periodo de tiempo determinado, entonces se lo definirá como churn. En la industria de aplicaciones móviles, específicamente de delivery en línea, se suele considerar a un usuario inactivo a partir de transcurrir más de un mes sin realizar actividad en la misma.

Cabe destacar qué, si bien se podría haber considerado el evento "uninstall" como un churn, y luego desarrollar modelos que entrenen del mismo, se toma la decisión de evitar este camino dado que gran parte de estos eventos se pueden deber al tipo de churn involuntario, lo que implicaría determinado ruido en el estudio. Siguiendo con un ejemplo, un usuario puede eliminar la aplicación a causa de no tener suficiente memoria en el teléfono pero tiempo después podría volver a instalarla cuando requiera hacer uso de la misma.

5. Implementación

5.1. Algoritmos de aprendizaje

En este apartado se realizará la implementación de diferentes algoritmos de aprendizaje para buscar predecir el churn de los usuarios en la aplicación de delivery de comida.

Existen dos tipos principales de algoritmos de aprendizaje: supervisado y no supervisado.

Los algoritmos de aprendizaje supervisado se utilizan cuando se dispone de datos etiquetados previamente, es decir, los datos ya tienen las respuestas correctas. En el contexto del aprendizaje supervisado, las "entradas" se refieren a los datos de entrada, como características o variables. Las "salidas" son las respuestas o etiquetas correspondientes a esas entradas, representando las respuestas correctas asociadas a los datos de entrenamiento. El objetivo de estos algoritmos es encontrar una relación entre las entradas y las salidas a partir de estos datos etiquetados para poder hacer predicciones o clasificaciones precisas en nuevos datos.

Por otro lado, los algoritmos de aprendizaje no supervisado se utilizan cuando no se dispone de datos etiquetados previamente, es decir, no se sabe cuáles son las respuestas correctas. Se suelen utilizar con el objetivo de encontrar patrones o agrupamientos en los datos. Por ejemplo, se podría utilizar un algoritmo de aprendizaje no supervisado para agrupar a los clientes de un supermercado en diferentes segmentos según sus costumbres o hábitos de compra.

Considerando el contexto actual, se optará por utilizar algoritmos de aprendizaje supervisado debido a la disponibilidad de datos etiquetados y a su capacidad para ajustarse de manera más adecuada a la situación. Sin embargo, en algunas ocasiones, un único modelo puede resultar insuficiente para lograr los resultados deseados en las predicciones. Es en este punto donde entra en juego el ‘Ensamble Learning’..

5.2. Ensamble Learning

El ensamble learning es una técnica que tiene como objetivo combinar las predicciones generadas por múltiples modelos individuales, con el fin de obtener un resultado global más preciso y robusto. Al integrar las predicciones de varios modelos, se busca aprovechar las fortalezas y compensar las debilidades de cada uno, logrando un mejor rendimiento predictivo. Su estrategia se basa en la fuerza de la unidad, ya que las combinaciones eficientes diferentes modelos pueden generar

modelos más precisos y robustos. Las tres principales clases de métodos de aprendizaje en conjunto son:

- ➢ Bagging: técnica que construye diferentes modelos en paralelo utilizando subconjuntos aleatorios de datos y combina deterministicamente las predicciones de todos los predictores. Con estos algoritmos se consigue que los errores se compensen debido a que cada modelo se entrena con subconjuntos que eligen muestras de con repetición de una manera totalmente aleatoria (Kuhn & Johnson, 2013, p. 192-193).
- ➢ Boosting: técnica iterativa, secuencial y adaptativa, donde cada predictor corrige el error de su modelo predecesor, adaptándose y mejorando de esta manera el rendimiento general del modelo (Kuhn & Johnson, 2013, p. 203-208).
- ➢ Stacking: técnica que implica combinar las predicciones de múltiples algoritmos de aprendizaje automático, como los dos antes mencionados.

Como se puede analizar, para problemas de predicción con estas características, se pueden utilizar diversos modelos de aprendizaje. Sin embargo, considerando el contexto y su potencia predictiva, se hará enfoque en los modelos de Boosting.

En un primer análisis, se evaluará la performance con un algoritmo moderno de gran uso en la actualidad, XGBoost.

5.2.1. XGBoost

XGBoost es una librería popular de aprendizaje automático y un algoritmo de ensamble basado en árboles de decisión. Es conocido por su potencia, versatilidad y alto rendimiento en una amplia gama de problemas de aprendizaje supervisado y su gran capacidad para manejar grandes conjuntos de datos, características de alta dimensionalidad y una variedad de tipos de variables.

El algoritmo de XGBoost utiliza un enfoque de ensamble en el que se construyen múltiples árboles de decisión en secuencia. Cada árbol se ajusta a los errores residuales del árbol anterior, lo que permite corregir gradualmente los errores y mejorar el rendimiento predictivo.

XGBoost ofrece gran cantidad de hiperparámetros para ajustar y optimizar el rendimiento del modelo, los cuales serán expuestos más adelante.

A continuación se visualiza la primera implementación de la librería:

```python
# Importar las bibliotecas necesarias
import xgboost as xgb
# Definir los parámetros del modelo
params = {
    'max_depth': 15,
    'learning_rate': 0.07,
    'n_estimators': 40,
    'min_child_weight': 6,
    'subsample': 0.55,
    'objective': 'binary:logistic',
    'eval_metric': 'logloss'
}
# Entrenar el modelo
model = xgb.XGBClassifier(**params)
model.fit(x_train, y_train)
# Predecir sobre los conjuntos de entrenamiento y validación
y_train_pred = model.predict(x_train)
y_valid_pred = model.predict(x_valid)
```

Gráfico 20: *Código de la implementación de la librería XGBoost.*

Para la elección adecuada de hiperparametros se realizarán técnicas de validación que se expondrán luego. Con esta primer implementación, se obtiene como predicción una variable booleana True/False, ya que el modelo se encarga de categorizar de manera automática con un umbral de 0.5; esto luego será evaluado y se analizará el mejor umbral según los objetivos buscados de negocio.

El tiempo promedio de entrenamiento para este algoritmo fue de **15 segundos por iteración.**

5.2.2. CatBoost

Considerando el contexto del problema, se exploran nuevas librerías y se hace uso entonces de CatBoost. El mismo, también basado en el método de gradient boosting, el cual es un potente algoritmo de aprendizaje automático que fue ganando popularidad debido a su capacidad para manejar características categóricas de manera eficiente.

Según John (2023), CatBoost se destaca por sus características únicas que lo diferencian de otros algoritmos:

- Árboles simétricos: A diferencia de XGBoost, CatBoost construye árboles simétricos y balanceados. Esto permite una implementación eficiente en la capacidad de procesamiento, reduce el tiempo de predicción y controla el sobreajuste, ya que la estructura del árbol actúa como una forma de regularización.

- Impulso ordenado: CatBoost utiliza el concepto de impulso ordenado, lo que significa que entrena el modelo en un subconjunto de datos y calcula los residuos en otro subconjunto. Esto evita la filtración de datos objetivo y reduce el riesgo de sobreajuste en conjuntos de datos pequeños o ruidosos.
- Soporte nativo de mayor tipo de características: CatBoost admite características de variedad de tipos, como numéricas, categóricas y de texto. Esto elimina la necesidad de realizar un preprocesamiento exhaustivo de los datos y ahorra tiempo y esfuerzo en la etapa de preparación de los datos.

A continuación se visualiza la primera implementación de la librería:

```python
from catboost import CatBoostClassifier
params = {
    'iterations': 150,
    'learning_rate': 0.07,
    'depth': 6,
    'l2_leaf_reg': 5,
    'loss_function': 'Logloss',
    'random_seed': 42
}
# Entrenar el modelo
model = CatBoostClassifier(**params)
model.fit(x_train, y_train)

# Realizar predicciones en el conjunto de entrenamiento
y_train_pred = model.predict(x_train)
# Realizar predicciones en el conjunto de validación
y_valid_pred = model.predict(x_valid)
```

Gráfico 21: *Código con implementación de librería CatBoost.*

El tiempo promedio de entrenamiento para este algoritmo fue de **3,5 segundos.** Se puede ver una gran mejoría en tiempos, lo que permite luego también realizar una búsqueda más exhaustiva de hiperparametros.

5.3. Hiperparametros

Algunos de los hiperparametros más conocidos que se utilizan en ambas librerías expuestas son:

➢ Máxima Profundidad (max_depth): Especifica la profundidad máxima de cada árbol en el ensamble. Un valor demasiado alto puede permitir que el modelo capture relaciones más complejas en los datos, pero también puede llevar a un mayor riesgo de sobreajuste.

- ➢ Tasa de aprendizaje (learning_rate): Controla la tasa de aprendizaje utilizada para actualizar los pesos de los árboles en cada iteración. Un valor más bajo significa una tasa de aprendizaje más lenta, lo que puede mejorar la generalización del modelo, pero requerirá más iteraciones para converger a un buen resultado.
- ➢ Número de árboles (n_estimators): Indica el número de árboles a construir en el ensamble. Cuanto mayor sea este valor, más complejo será el modelo y mayor será el riesgo de sobreajuste. Es importante encontrar un equilibrio entre un número suficiente de estimadores para capturar las relaciones en los datos y evitar el sobreajuste.
- ➢ Proporción de muestra (subsample): Controla la proporción de muestras utilizadas para entrenar cada árbol individualmente.

Utilizar hiperparámetros en un modelo de aprendizaje automático es crucial para encontrar el equilibrio adecuado entre el sesgo y la varianza. El sesgo se refiere a las suposiciones simplificadas que hace el modelo sobre los datos, mientras que la varianza se refiere a la sensibilidad del modelo a las fluctuaciones en los datos de entrenamiento.

El sobreajuste, también conocido como Overfitting, hace referencia a cuando el modelo se adapta demasiado a los datos de entrenamiento y pierde la capacidad de generalizar correctamente a nuevos datos (datos de validación / test). En este caso, el sesgo es bajo, ya que el modelo se ajusta bien a los datos de entrenamiento, pero la varianza es alta, lo que significa que el modelo es muy sensible a pequeños cambios en los datos.

Por otro lado, utilizando hiperparametros equivocados y ajustando insuficientemente, se corre el riesgo de underfitting. Esto significa que el modelo no se ajusta lo suficiente a los datos de entrenamiento y tiene dificultades para capturar las relaciones y patrones subyacentes en los datos. Para este caso, el sesgo es alto y la varianza es baja, ya que el modelo pierde la sensibilidad a las fluctuaciones en los datos.

Encontrar el equilibrio adecuado entre el sesgo y la varianza implica ajustar los hiperparámetros de manera óptima. Esto implica realizar un adecuado ajuste de los hiperparámetros para obtener un modelo que se ajuste bien a los datos de entrenamiento sin sobre ajustarlos. Es un proceso iterativo en el que se prueban diferentes combinaciones de hiperparámetros y se evalúa el rendimiento del modelo mediante diferentes técnicas en datos de validación.

A continuación se visualiza una porción del código que se utiliza para buscar la mejor combinación de hiperparametros para el caso de XGBoost:

```python
# Se crea un dataframe vacío para almacenar los resultados
results = pd.DataFrame(columns=['max_depth', 'learning_rate', 'n_estimators', 'min_child_weight', 'subsample', 'auc_valid', 'auc_train

# Se itera sobre las combinaciones de hiperparámetros
for max_depth in max_depth_range:
    for learning_rate in learning_rate_range:
        for n_estimators in n_estimators_range:
            for min_child_weight in min_child_weight_range:
                for subsample in subsample_range:
                    print("Iteración: max_depth " + str(max_depth) + ", learning_rate " + str(learning_rate) + ", n_estimators " + st
                    # Se definen los parámetros del modelo
                    params = {
                        'max_depth': max_depth,
                        'learning_rate': learning_rate,
                        'n_estimators': n_estimators,
                        'min_child_weight': min_child_weight,
                        'subsample': subsample,
                        'objective': 'binary:logistic',
                        'eval_metric': 'logloss'
                    }
                    # Se entrena el modelo
                    model = xgb.XGBClassifier(**params)
                    model.fit(x_train, y_train)
                    # Se predice sobre train, y se calcula AUC
                    y_train_pred = model.predict_proba(x_train)[:, 1]
                    auc_train = roc_auc_score(y_train, y_train_pred)
                    # Se predice sobre validación, y se calcula AUC
                    y_valid_pred = model.predict_proba(x_valid)[:, 1]
                    auc_valid = roc_auc_score(y_valid, y_valid_pred)
                    results = results.append({'max_depth': max_depth, 'learning_rate': learning_rate, 'n_estimators': n_estimators, '
```

Gráfico 22: *Porción de código utilizada para optimizar hiperparametros con XGBoost.*

5.4. Entrenamiento, validación y test

Como se mencionó anteriormente, el periodo a trabajar es mensual, y continuando con la lógica se separan los datos para realizar el entrenamiento, luego la validación y el testeo. Para no perder el orden temporal, y buscando predecir la probabilidad de que un usuario abandone la plataforma en el siguiente periodo, se separan los datos de la siguiente manera:

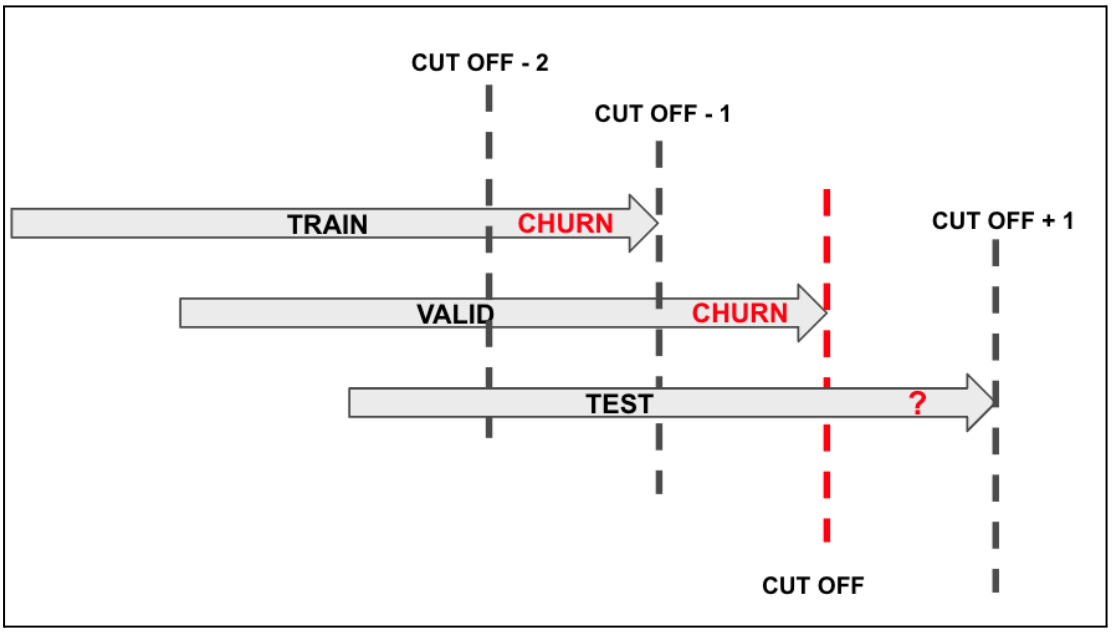

***Gráfico 23**: Representación temporal de las muestras entrenamiento, validación y test.*

Como se puede ver en la imágen, al realizar un primer Cut Off del día 31 de enero de 2023, la estructura quedaría de la siguiente manera:

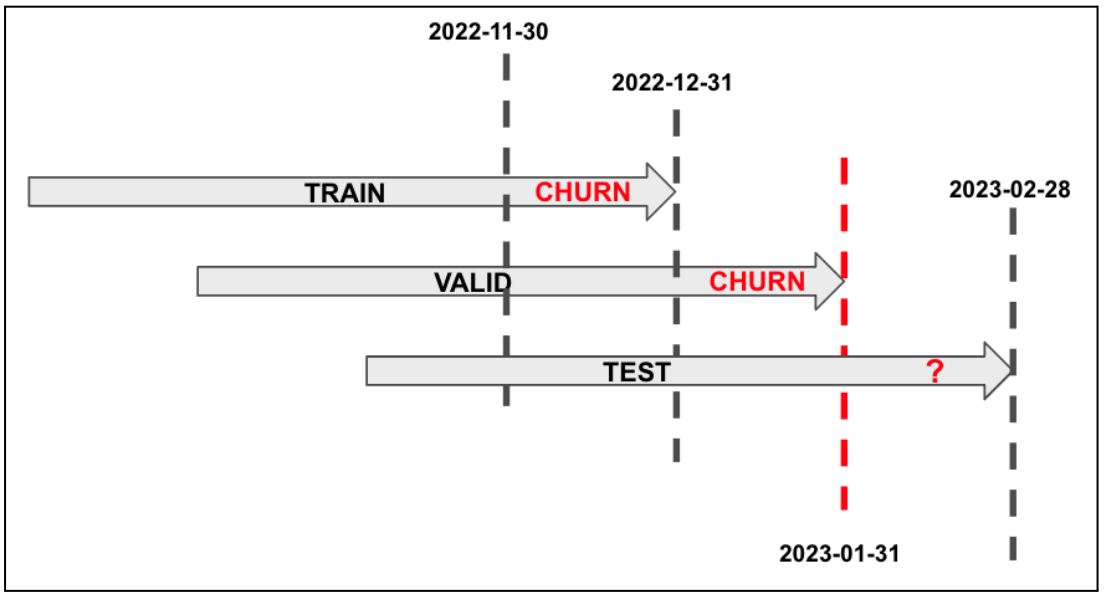

***Gráfico 24**: Representación temporal de las muestras con Cut Off el 2023-01-31.*

Los modelos entrenados y luego validados, se utilizarán para predecir los datos de test al final del trabajo. Estos datos se encuentran fuera del dataset inicial para evitar cualquier tipo de Data Leakage y conseguir resultados los más cercanos a la realidad.

Dado que se cuenta con datos desde septiembre de 2022, el periodo de entrenamiento para este caso es de 3 meses. Otra variable a tener en cuenta es la ventana de tiempo con la que se contempla si un usuario ha tenido actividad. Si se toma, por ejemplo, una ventana de tiempo de un mes para entrenar, entonces se entrenará con todas las transacciones dentro del periodo de tiempo dado, de los usuarios que han tenido actividad en el último mes.

Se muestra a continuación la cantidad de eventos realizados por los usuarios considerados en esta ventana de tiempo:

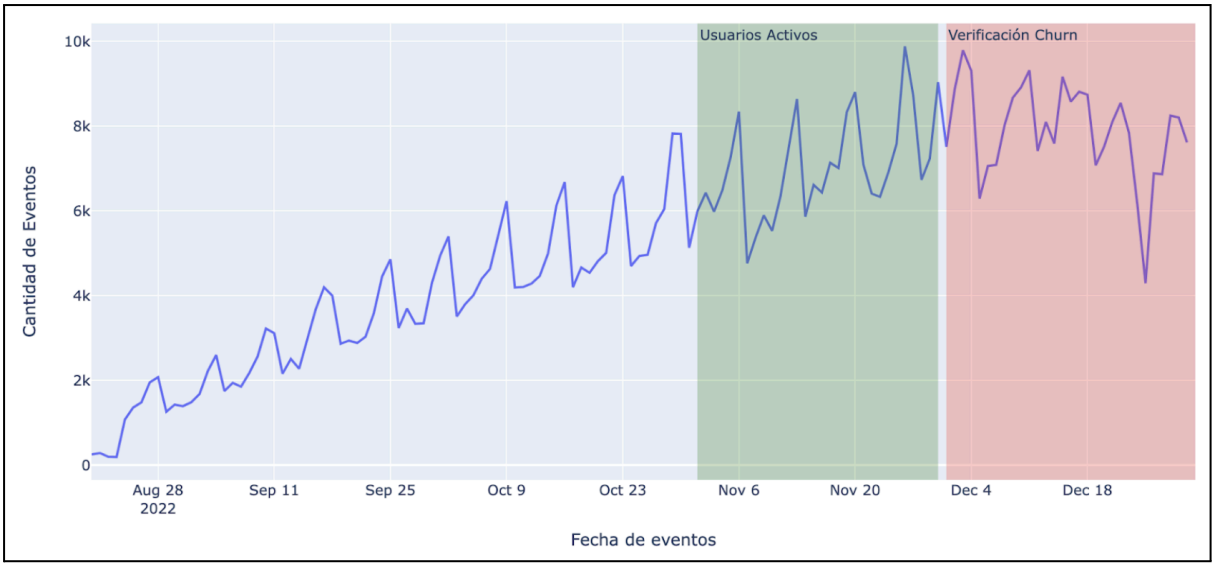

Gráfico 25: *Eventos realizados por usuarios activos en el mes de noviembre de 2022.*

Se realizan pruebas con diferentes ventanas de tiempo, y se expondrán luego los resultados.

5.5. Modelos estudiados y descartados

Al haber establecido el enfoque del problema se acudió a bibliografía relacionada a las diversas maneras de utilizar métodos estadísticos y de machine learning para la predicción de churn. Como primera instancia, se descartan los algoritmos para problemas de regresión, principalmente en los que no se obtiene un output binario o entre 0 y 1.

Otros modelos estudiados y descartados:

- Redes Neuronales: esta implementación ha sido explorada principalmente debido a su gran poder predictivo. Desafortunadamente, el algoritmo desarrollado estuvo algunas horas intentando converger a un mínimo local pero no lo logró y se terminó descartando.
- Algoritmos de Bagging: se utilizan algoritmos de Boosting sobre Bagging debido a su mejor rendimiento para casos donde se observa menor varianza en los datos de entrenamiento pero un mayor sesgo. A su vez, tienen un mayor enfoque en instancias difíciles de clasificar, dándole una mayor importancia a instancias mal clasificadas para mejorar el rendimiento en ellas, lo que se considera conveniente para estos casos por la naturaleza de los datos.

6. Resultados y discusión

Como se ha mencionado previamente, este trabajo utilizará un enfoque de aprendizaje automático centrado en técnicas de clasificación. Para evaluar estas técnicas y determinar las mejores interpretaciones de los modelos desarrollados, es necesario contar con un conjunto de métricas que validan su eficacia. Estas métricas también pueden ayudar a lograr un equilibrio necesario entre el sesgo y la varianza del modelo.

A continuación, se detallan las métricas seleccionadas por el autor en función de la experiencia relevada en otros trabajos similares. En una segunda instancia, se abordarán aquellas métricas que son relevantes desde una perspectiva económica o del negocio. Esto significa que, si bien el modelo utilizará métricas comunes en el aprendizaje automático, también se considerarán métricas específicas para evaluar la eficiencia del negocio en cuestión.

6.1. Métricas del modelo

La evaluación exhaustiva de un modelo es fundamental para medir su desempeño y determinar su eficacia en la tarea de clasificación. En este contexto, se utilizan diversas métricas para cuantificar y comparar el rendimiento del modelo. Algunas de estas métricas incluyen precision, recall, f1 score y accuracy. Estas medidas permiten evaluar diferentes aspectos de la capacidad del modelo para clasificar de manera más objetiva las instancias positivas y negativas:

- Precision: métrica que indica la proporción de predicciones positivas correctas realizadas por el modelo en relación con todas las predicciones positivas. Es útil para evaluar la exactitud de las predicciones positivas del modelo.

$$Precision = \frac{Verdaderos\ positivos}{Verdaderos\ positivos + Falsos\ positivos}$$

- Recall: también conocido como sensibilidad o tasa de verdaderos positivos, mide la proporción de instancias positivas correctamente identificadas por el modelo en relación con todas las instancias positivas reales. Se suele utilizar para evaluar la capacidad del modelo para detectar de manera completa los casos positivos.

$$Recall = \frac{Verdaderos\ positivos}{Verdaderos\ positivos + Falsos\ negativos}$$

- F1 score: métrica que combina la precisión y el recall. Es útil cuando se desea encontrar un equilibrio entre la precisión y el recall. El F1 score proporciona una evaluación general del rendimiento del modelo al considerar tanto los falsos positivos como los falsos negativos.

$$F1\ score = \frac{2\ x\ Precision\ x\ Recall}{Precision + Recall}$$

- Accuracy: La exactitud es una métrica que calcula la proporción de predicciones correctas realizadas por el modelo en relación con todas las predicciones realizadas. Es una medida general del rendimiento del modelo, pero puede contener cierto sesgo en presencia de desequilibrios de clase en los datos.

$$Accuracy = \frac{Predicciones\ Correctas}{Predicciones\ Totales}$$

Continuando con las métricas, una que destaca por su importancia y utilidad es el "Área Bajo la Curva" (AUC, por sus siglas en inglés). El AUC proporciona una medida agregada del rendimiento global del modelo a través de la curva ROC. La curva ROC representa la relación entre la tasa de verdaderos positivos y la tasa de falsos positivos a medida que se varía el umbral de clasificación.

La elección de utilizar AUC como métrica de evaluación es especialmente relevante debido a varias razones. En primer lugar, el AUC es una medida robusta que no se ve afectada por desequilibrios en la distribución de las clases. Esto lo hace especialmente adecuado en casos donde los datos presentan un sesgo hacia una clase dominante, como en este caso donde tenemos mayor parte de la clase positiva (usuarios que abandonan la aplicación en el siguiente periodo). Además, el AUC proporciona una evaluación global del modelo, considerando todas las posibles configuraciones de umbrales de clasificación, lo que lo convierte en una métrica más completa y confiable.

Para realizar la elección óptima de hiperparametros iterando sobre el set de validación, se utilizará a la métrica AUC como métrica de evaluación y optimización. Se busca obtener una medida objetiva y precisa del rendimiento del modelo en la tarea de clasificación, permitiendo comparar diferentes modelos y seleccionar aquellos que presenten un mejor desempeño en términos de la capacidad de discriminación y la capacidad para minimizar los errores de clasificación.

6.2. Métricas y el negocio

Considerar las métricas para el negocio adecuadamente desempeña un papel fundamental al evaluar el impacto del churn y la rotación de clientes en una empresa. Cada usuario que abandona la aplicación representa una pérdida potencial en términos de ingresos y costo de adquisición. Por lo tanto, es crucial comprender y monitorear las métricas correspondientes.

Con el fin de mitigar esta pérdida, se pueden ejecutar diferentes estrategias de retención, como el envío de promociones y cupones de descuento a los usuarios con alto riesgo de abandono, como una medida preventiva para evitar el churn. Esto se realiza ya que en la mayor parte de los casos, el costo asociado a enviar promociones adicionales a usuarios es considerablemente menor que el costo de perder clientes, y más cuando estos pueden ser potencialmente valiosos. Para realizar de una manera adecuada estas estrategias, es importante tener en cuenta las métricas del modelo anteriormente presentadas.

Por lo mencionado y considerando este tipo de industria, se puede asumir cierta cantidad de falsos positivos, es decir, usuarios clasificados como churn y que no tendrán una intención real de abandonar. Entonces, luego de haber conseguido el modelo que optimice AUC, se buscará un umbral de decisión que tenga mayor en cuenta a la métrica ‘Recall’ dejando un poco de lado ‘Precision’, es decir, buscando que el modelo detecte la mayoría de los casos de churn, tomando luego las medidas proactivas necesarias y resultando en la menor pérdida de oportunidades de retención.

6.3. XGBoost

Utilizando el set de validación, se ejecuta el algoritmo de XGBoost presentado anteriormente. A continuación, se muestran los resultados de algunas de las iteraciones realizadas en busca de optimizar el AUC:

Tabla 7: *Algunas iteraciones de la búsqueda de hiperparametros utilizando XGBoost.*

	max_depth	learning_rate	n_estimators	min_child_weight	subsample	auc_valid	auc_train
0	15.0	0.04	12.0	6.0	0.55	0.751378	0.837019
1	15.0	0.04	12.0	6.0	0.85	0.750662	0.862796
2	15.0	0.04	30.0	6.0	0.55	0.753036	0.856352
3	15.0	0.04	30.0	6.0	0.85	0.752479	0.883879
4	15.0	0.04	55.0	6.0	0.55	0.753026	0.874046
5	15.0	0.04	55.0	6.0	0.85	0.752505	0.903053
6	15.0	0.07	12.0	6.0	0.55	0.752458	0.843886
7	15.0	0.07	12.0	6.0	0.85	0.749391	0.871670
8	15.0	0.07	30.0	6.0	0.55	0.750724	0.869753
9	15.0	0.07	30.0	6.0	0.85	0.750981	0.899826
10	15.0	0.07	55.0	6.0	0.55	0.748783	0.891840

Comparar el AUC de validación con el AUC de train resulta de gran utilidad para evaluar la capacidad de generalización del modelo predictivo. Si existe una correspondencia cercana entre ambos AUC, esto sugiere que el modelo ha logrado generalizar de manera efectiva y puede tener un buen rendimiento en datos nuevos y no vistos previamente. En este caso, se puede intuir que en la mayor parte de las iteraciones el modelo podría estar sobreajustando (overfitting) y que no estaría generalizando correctamente los datos.

Se continúa iterando entre hiperparametros, buscando capturar patrones relevantes en los datos de manera efectiva, y se ejecutan 750 iteraciones con un tiempo de ejecución **total de 200 minutos**. Finalmente se obtiene la mejor combinación que optimiza el AUC en validación:

Tabla 8: *Mejor combinación de hiperparametros para XGBoost.*

max_depth	learning_rate	n_estimators	min_child_weight	subsample
3	0.07	90	6	0.55

Se visualiza la curva ROC para esta mejor combinación de hiperparametros:

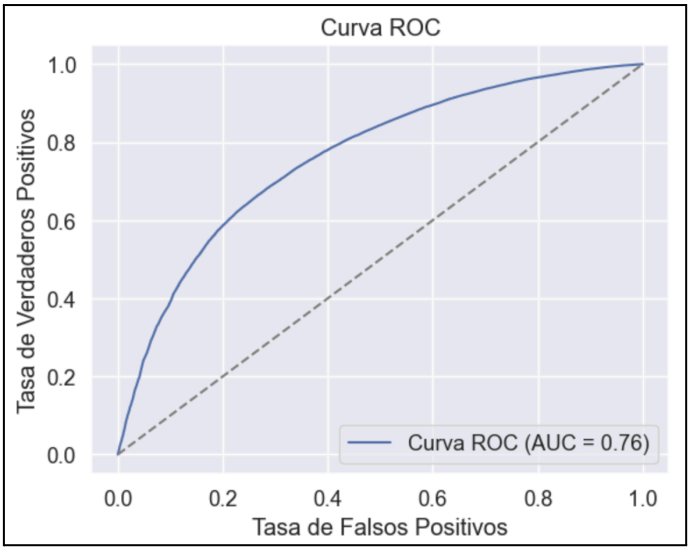

Gráfico 26: *Curva ROC utilizando XGBoost con el set de validación.*

Una vez entrenado el mejor modelo, se busca el umbral de decisión adecuado:

Tabla 9: *Métricas del modelo para diferentes umbrales utilizando XGBoost con el set de validación.*

	Umbral	Recall	Precision	F1-score	Accuracy
0	0.10	0.983367	0.705888	0.821838	0.709581
1	0.13	0.963952	0.723069	0.826313	0.723969
2	0.16	0.942420	0.737603	0.827527	0.732414
3	0.19	0.919673	0.749977	0.826201	0.736444
4	0.22	0.897194	0.761237	0.823643	0.738290
5	0.25	0.873459	0.771353	0.819237	0.737444
6	0.28	0.846363	0.781788	0.812795	0.734434
7	0.31	0.818336	0.792358	0.805137	0.730183
8	0.34	0.787301	0.804104	0.795614	0.724470
9	0.37	0.750886	0.816408	0.782277	0.715294
10	0.40	0.710758	0.828664	0.765195	0.702878
11	0.43	0.675473	0.838619	0.748256	0.690403
12	0.46	0.641416	0.848516	0.730573	0.677746
13	0.49	0.610113	0.857188	0.712848	0.665186
14	0.52	0.572709	0.866558	0.689637	0.648874
15	0.55	0.536111	0.873915	0.664549	0.631331

A partir de analizar la tabla, se revela el trade-off entre las métricas de recall y precision. Al observar estos dos valores, se puede notar que a medida que uno aumenta, el otro tiende a disminuir. Esto se debe a la naturaleza del problema y a las decisiones de clasificación que el modelo realiza.

En este sentido, encontrar un equilibrio óptimo entre recall y precision puede ser un desafío. Para tomar una decisión final sobre qué modelo o umbral de decisión elegir, el F1-score es una métrica útil y conveniente ya que combina tanto el recall como la precision en un solo valor que proporciona una medida equilibrada del rendimiento del modelo.

Por lo expuesto, se elige finalmente un umbral de 0,16. Cabe destacar que con esta elección de umbral, se consigue uno de los valores más altos posibles para el Recall, métrica la cual se destacó de importancia para este caso.

Se predice en el set de test, y utilizando el umbral de decisión mencionado se obtienen las siguientes métricas:

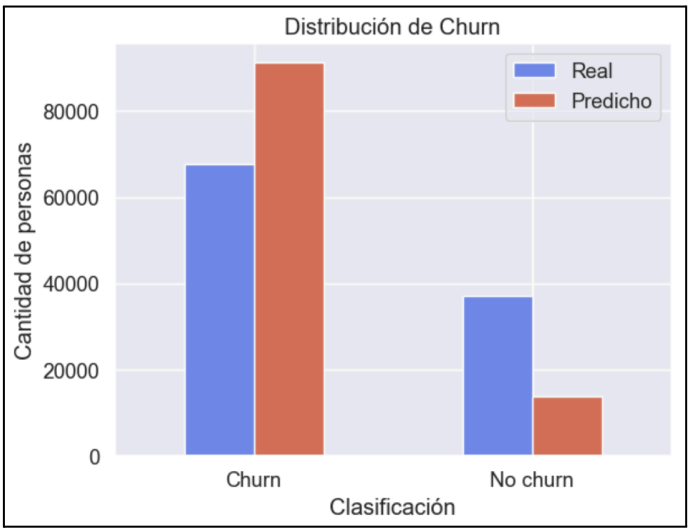

Gráfico 27: *Distribución de churn utilizando XGBoost para el set de test.*

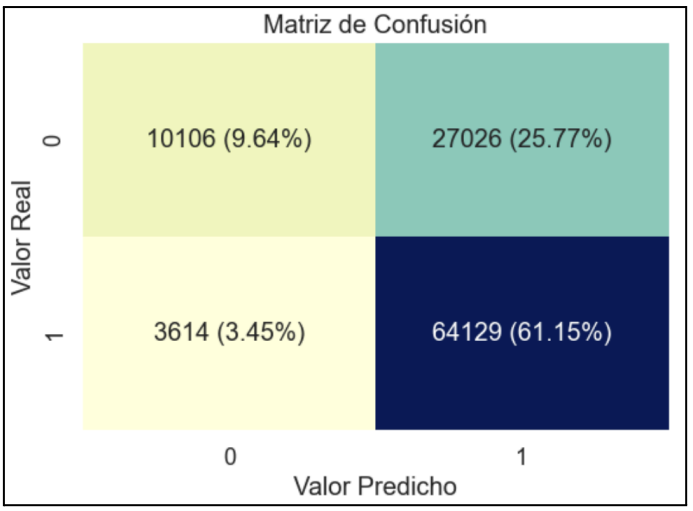

Gráfico 28: *Matriz de confusión utilizando XGBoost para el set de test.*

A partir de estos resultados, se puede ver que en el set de prueba, el modelo logra clasificar correctamente el comportamiento del más del 70% de los usuarios (usuarios que se encontraron activos en el mes de enero). A su vez se destaca que, de la base total de usuarios, el 64,6% estuvieron activos en el mes de enero y luego abandonaron la aplicación en el periodo de febrero, y que de este porcentaje, el 61,15% se logró predecir su comportamiento correctamente. Entonces, considerando únicamente estos usuarios que realmente abandonaron la aplicación en el siguiente periodo, se puede decir que el modelo tuvo un 95% de predicciones correctas, o también llamado Recall.

Finalmente, la curva ROC para el set de test con el modelo entrenado por la librería XGBoost, resulta en la siguiente:

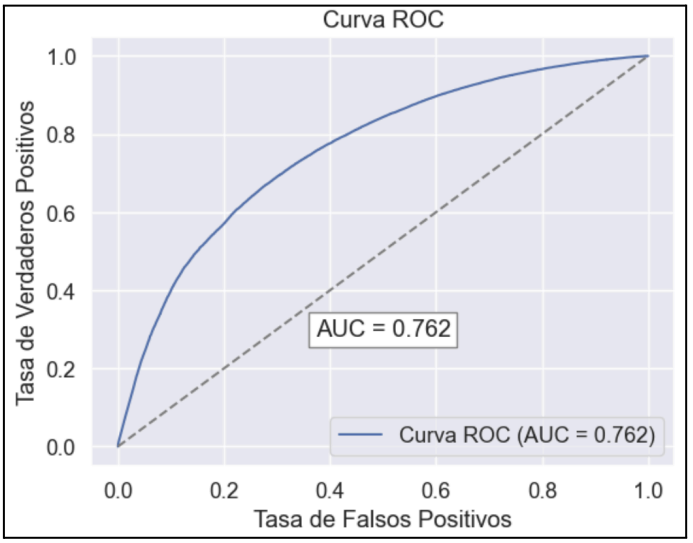

Gráfico 29: *Curva ROC utilizando XGBoost para el set de test.*

Siguiendo con el análisis, se entrena para obtener un modelo considerando una ventana de actividad de 2 meses. La mejor combinación de hiperparametros es la siguiente:

Tabla 10: *Mejor combinación de hiperparametros para XGBoost con 2 meses de ventana.*

max_depth	learning_rate	n_estimators	min_child_weight	subsample
3.0	0.05	60.0	6.0	0.80

Se visualiza la curva ROC para este caso con la mejor combinación de hiperparametros y utilizando el set de test:

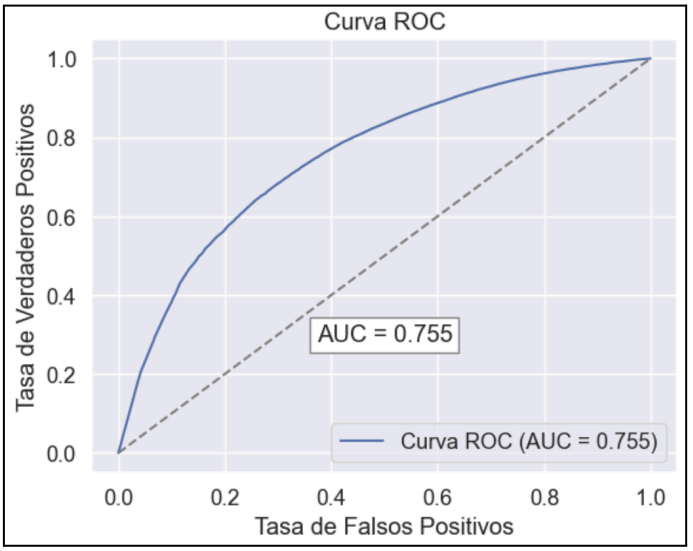

Gráfico 30: *Curva ROC con XGBoost y 2 meses de ventana para el set de test.*

Al considerar una ventana de activación de dos meses en lugar de una, la métrica AUC se vió afectada negativamente. Esto es un resultado diferente a lo que se hubiera esperado ya que se estaría considerando una perspectiva más completa y representativa del comportamiento de los usuarios. Esto podría ser resultado a causa de:

- Pérdida de información relevante: Al extender la ventana de activación a dos meses, existe la posibilidad de incluir datos que no son relevantes para el evento objetivo. Esto podría dificultar la capacidad del modelo para identificar patrones significativos, afectando negativamente la métrica AUC.

- Cambio en la distribución de los datos: Al ampliar la ventana de activación, es probable que la distribución de los datos también se vea afectada, lo que a su vez impactaría negativamente en la capacidad del modelo para generalizar y obtener una mejor métrica AUC.

El umbral que maximiza el F1 score para este caso es 0,25. Se exponen entonces los resultados conseguidos.

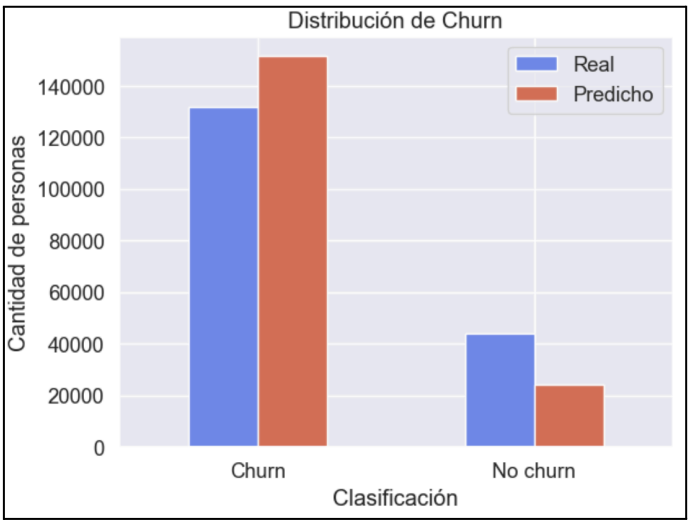

Gráfico 31: *Distribución de churn con XGBoost y 2 meses de ventana para el set de test.*

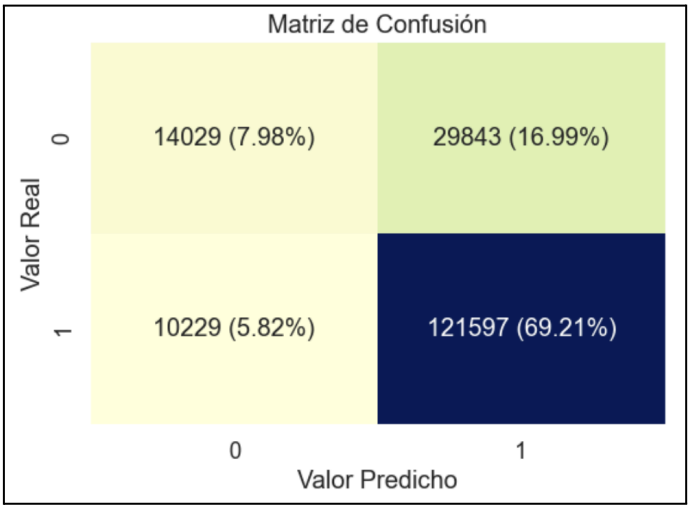

Gráfico 32: *Matriz de confusión con XGBoost y 2 meses de ventana para el set de test.*

6.4. CatBoost

Utilizando el set de validación, se ejecuta el algoritmo de la librería CatBoost presentada anteriormente. A continuación, se muestran los resultados de algunas de las iteraciones realizadas en busca de optimizar el AUC, ordenadas según el mayor valor en validación:

Tabla 11: *Mejores iteraciones de la búsqueda de hiperparametros utilizando CatBoost.*

depth	learning_rate	iterations	l2_leaf_reg	auc_valid	auc_train
2.0	0.04	350.0	5.0	0.790898	0.798707
2.0	0.04	250.0	5.0	0.790316	0.797492
3.0	0.07	250.0	3.0	0.789384	0.801593
2.0	0.07	250.0	3.0	0.788086	0.799177
3.0	0.07	350.0	3.0	0.788009	0.803015
8.0	0.04	250.0	3.0	0.787932	0.811993
2.0	0.07	350.0	3.0	0.787644	0.800247
2.0	0.04	350.0	3.0	0.787233	0.798752
3.0	0.04	250.0	3.0	0.787161	0.799491
3.0	0.04	350.0	3.0	0.787042	0.800781

A partir de estas iteraciones, se visualiza la mejor combinación de hiperparametros que optimiza el AUC:

Tabla 12: *Mejor combinación de hiperparametros para CatBoost.*

max_depth	learning_rate	n_estimators	l2_leaf_reg
2	0.04	350	5

Luego de ejecutar el algoritmo con la mejor combinación de hiperparametros, se visualiza la curva ROC:

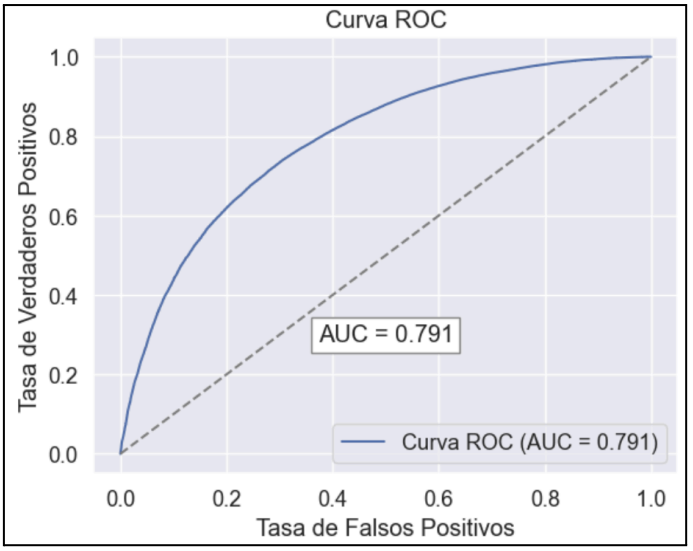

Gráfico 33: *Curva ROC utilizando CatBoost con el set de validación.*

Una vez entrenado el mejor modelo, se busca el umbral de decisión adecuado:

Tabla 13: *Métricas del modelo para diferentes umbrales utilizando CatBoost con el set de validación.*

	Umbral	Recall	Precision	F1-score	Accuracy
0	0.10	0.998630	0.690299	0.816320	0.693885
1	0.13	0.995326	0.700953	0.822597	0.707571
2	0.16	0.989961	0.711848	0.828179	0.720199
3	0.19	0.981997	0.722592	0.832556	0.730942
4	0.22	0.971464	0.733044	0.835579	0.739579
5	0.25	0.959462	0.743669	0.837895	0.747119
6	0.28	0.944778	0.754550	0.839017	0.753044
7	0.31	0.928272	0.764622	0.838537	0.756497
8	0.34	0.909634	0.773980	0.836342	0.757507
9	0.37	0.887537	0.783658	0.832369	0.756497
10	0.40	0.863279	0.793840	0.827105	0.754160
11	0.43	0.836734	0.804409	0.820253	0.750207
12	0.46	0.807746	0.814114	0.810918	0.743417
13	0.49	0.778730	0.823459	0.800470	0.735559
14	0.52	0.746777	0.833094	0.787578	0.725604
15	0.55	0.712593	0.843058	0.772355	0.713871

Continuando con lo expuesto anteriormente, se elige un umbral de 0,28 y se vuelve a ejecutar el modelo entrenado pero esta vez con el set de test, obteniendo las siguientes métricas:

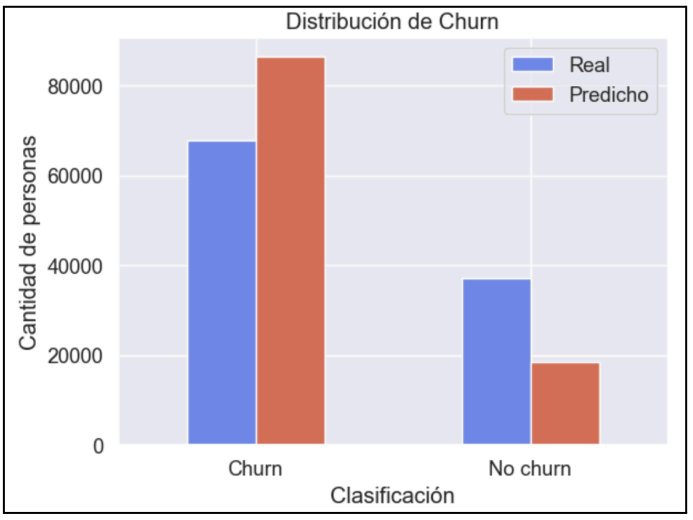

Gráfico 34: *Distribución de churn utilizando CatBoost con el set de test.*

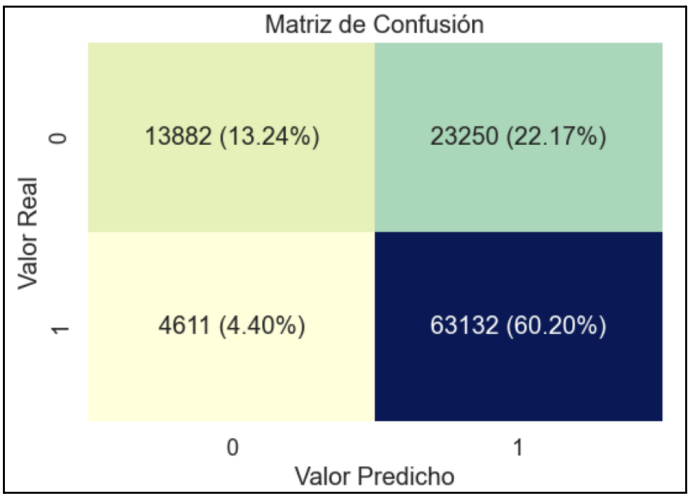

Gráfico 35: *Matriz de confusión utilizando CatBoost con el set de test.*

Con estos resultados, se puede ver que el modelo logra clasificar correctamente el comportamiento del más del 73,5% de los usuarios del set de test. A su vez se destaca que, se ha logrado predecir correctamente a un 96% de los usuarios que realmente realizaron churn, resultados aún mejores que los de XGBoost (63132 usuarios predecidos positivos de los 67743 totales que realmente abandonaron).

Finalmente, la curva ROC para el set de test con el modelo entrenado y validado, resulta en la siguiente:

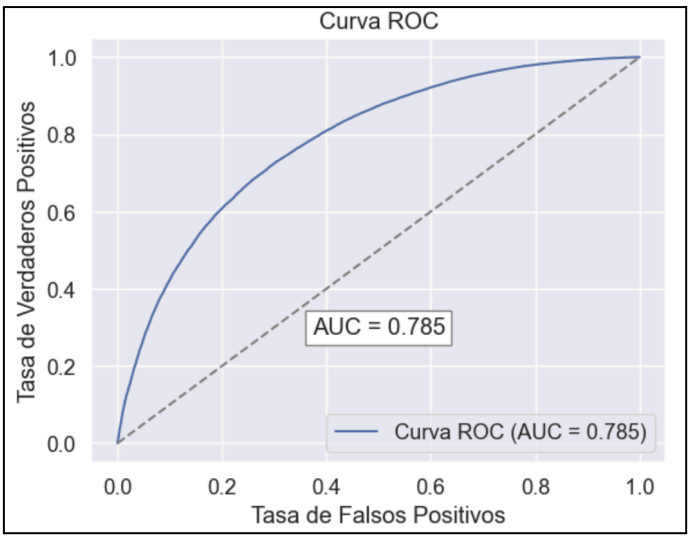

Gráfico 36: *Curva ROC utilizando CatBoost con el set de test.*

Como se podía prever, con el uso de esta librería se ha logrado mejorar aún más las métricas, obteniendo un AUC en test aún mayor que con la librería XGBoost (0,785 sobre 0,762).

Siguiendo con el análisis, se entrena para obtener un modelo considerando una ventana de actividad de 2 meses. La mejor combinación de hiperparametros es la siguiente:

Tabla 14: *Mejor combinación de hiperparametros para CatBoost y 2 meses de ventana con test.*

max_depth	learning_rate	n_estimators	l2_leaf_reg
2.0	0.07	250.0	5.0

Se visualiza la curva ROC para este caso con la mejor combinación de hiperparametros:

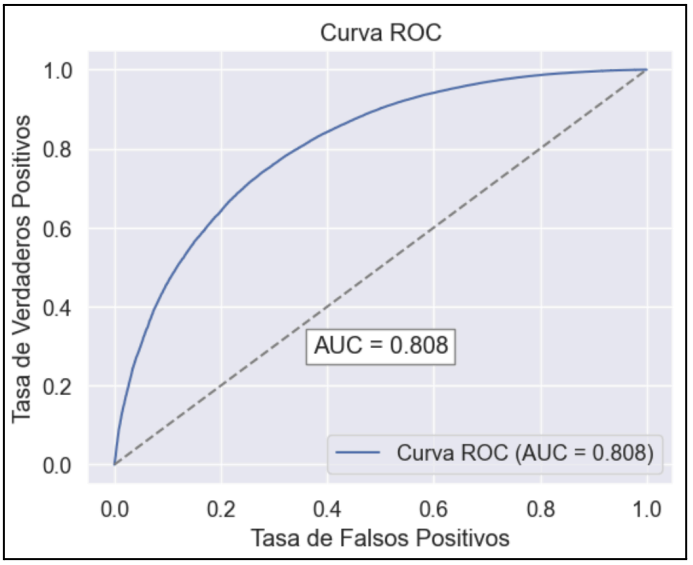

Gráfico 37: *Curva ROC utilizando CatBoost con 2 meses de ventana con el set de test.*

Ahora si, como era de esperar, el AUC que arroja el modelo de machine learning entrenado con usuarios que hayan realizado algún evento en una ventana de 2 meses, es mayor respecto del modelo entrenado con una ventana de actividad de un mes. Al utilizar una ventana de 2 meses para predecir el churn, se obtiene una perspectiva más completa y representativa del comportamiento del usuario en comparación con la otra. Esto se debe a que un período de 2 meses permite capturar de manera más exhaustiva los patrones y tendencias de uso a lo largo del tiempo, lo que proporciona una visión más sólida de la intención de abandono.

Siguiendo con los mismos procedimientos, se busca el umbral que maximiza el F1 score y el mismo es 0,3. Se exponen entonces los resultados conseguidos:

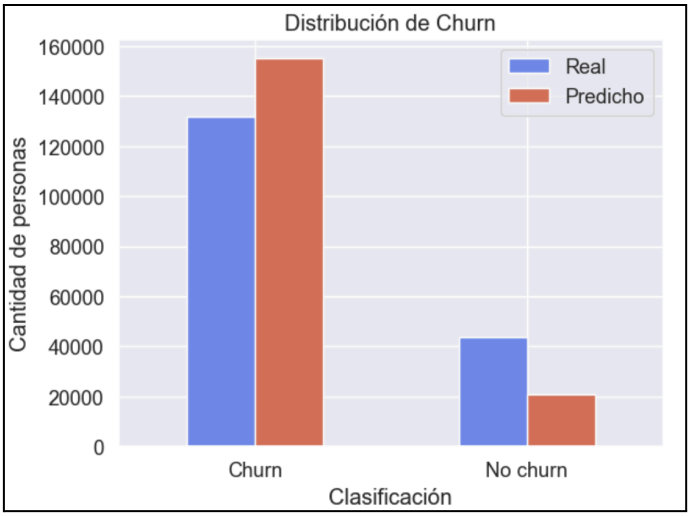

Gráfico 38: *Distribución de churn utilizando CatBoost con 2 meses de ventana para el set de test.*

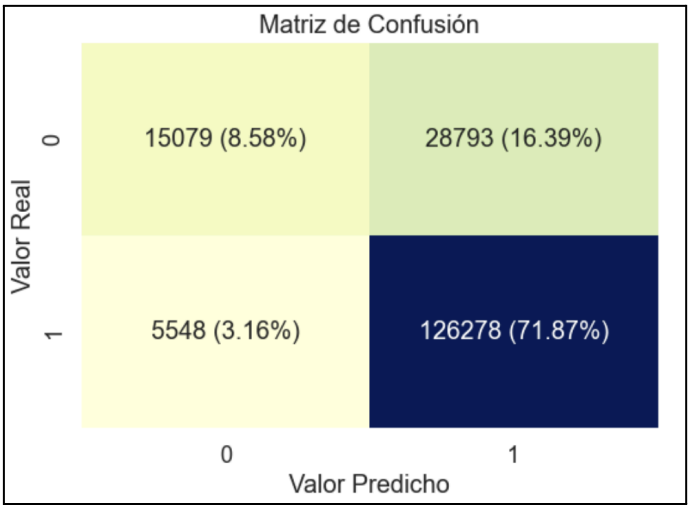

Gráfico 39: *Matriz de confusión utilizando CatBoost con 2 meses de ventana para el set de test.*

6.5. Importancia e interpretación

Los modelos generados por algoritmos tan potentes como los utilizados en este trabajo tienen la ventaja de ser muy eficaces a la hora de predecir, pero suelen ser una 'caja negra' en sus resultados, en qué variables se centran y cómo son utilizadas. A continuación, para mostrar qué variables son las más importantes, se computa cuánto disminuye el error debido a cada una de éstas, y se seleccionan las variables que hayan aportado y alimentado al poder predictivo del modelo.

Para el caso del primer algoritmo utilizado, XGBoost, la importancia de variables con un mes de ventana fue la siguiente:

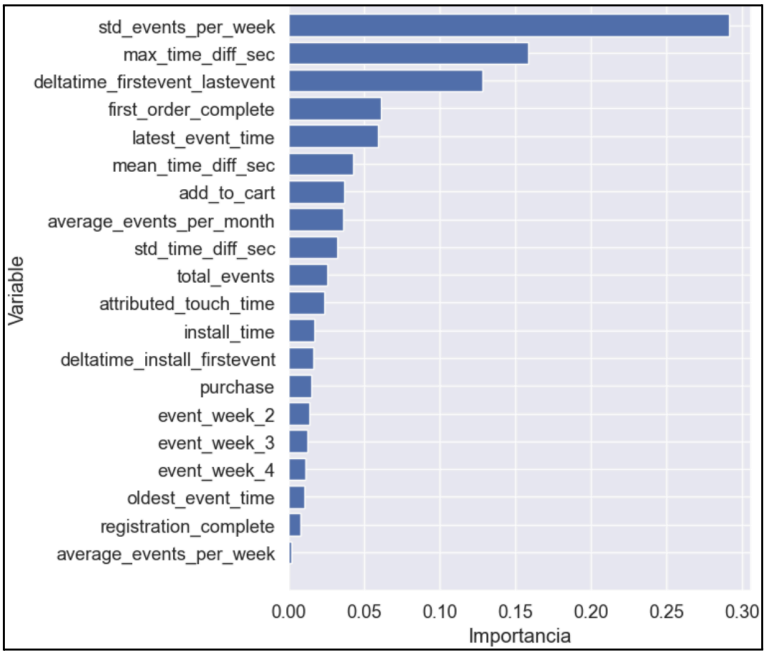

Gráfico 40: *Importancia de variables utilizando XGBoost con un mes de ventana.*

Estas variables se detallan en la tabla 13 a continuación:

Tabla 15: *Explicación de variables utilizando XGBoost.*

Variable	Explicación
std _events_per_week	desviación estándar de eventos totales
max_time_diff_sec	máxima diferencia en segundos entre eventos
deltatime_firstevent_lastevent	diferencia en segundos entre primer evento y último evento
first_order_complete	variable booleana que representa si el usuario completo la primer orden
latest_event_time	tiempo en que el usuario realizo el último evento
mean_time_diff_sec	segundos promedio de diferencia entre eventos
add_to_cart	cantidad de eventos 'add_to_cart'
average_events_per_month	cantidad de eventos totales promedios por mes
std_time_diff_sec	desviación estandar de diferencia de tiempo entre eventos
total_events	cantidad total de eventos realizados
attributed_touch_time	tiempo en el que el usuario se atribuyó
install_time	tiempo en el que el usuario instaló la aplicación
deltatime_install_firstevent	diferencia en segundos entre instalación y primer evento
purchase	cantidad de eventos 'purchase'
event_week_2	cantidad de eventos realizados en la 2da semana desde su instalación
event_week_3	cantidad de eventos realizados en la 3era semana desde su instalación
event_week_4	cantidad de eventos realizados en la 4ta semana desde su instalación
oldest_event_time	tiempo en que el usuario realizo el primer evento
registration_complete	cantidad de eventos 'registration_complete'
average_events_per_week	cantidad de eventos totales promedios por semana

Como era de esperar, varias de las variables que contribuyeron en el aprendizaje del modelo fueron creadas como resultado de haber visualizado su potencial al exponerlas en la exploración de datos. Esto ocurrió con variables como: total_events, latest_event_time, purchase, actividad a lo largo de las primeras semanas del usuario (event_week_2/3/4), entre otras.

Luego, se visualiza la importancia de variables para la librería CatBoost:

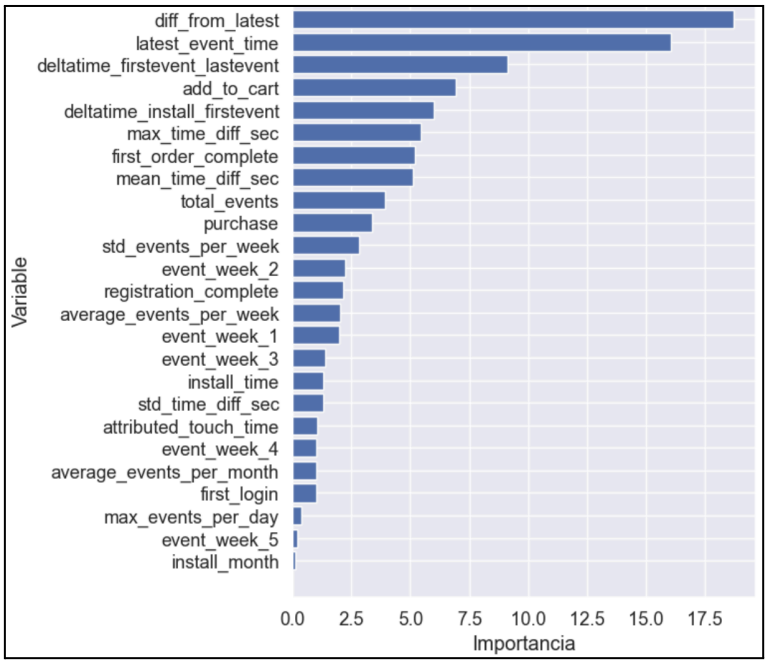

Gráfico 41: *Importancia de variables utilizando CatBoost.*

Estas variables se detallan en la tabla 14 a continuación:

Tabla 16: *Explicación de variables* utilizando CatBoost.

Variable	Explicación
diff_from_latest	diferencia de tiempo en segundos desde el último evento registro y el evento del usuario
latest_event_time	tiempo en que el usuario realizo el último evento
deltatime_firstevent_lastevent	diferencia en segundos entre primer evento y último evento
add_to_cart	cantidad de eventos 'add_to_cart'
deltatime_install_firstevent	diferencia en segundos entre instalación y primer evento
max_time_diff_sec	máxima diferencia en segundos entre eventos
first_order_complete	variable booleana que representa si el usuario completo la primer orden
mean_time_diff_sec	segundos promedio de diferencia entre eventos
total_events	cantidad total de eventos realizados
purchase	cantidad de eventos 'purchase'
std_events_per_week	desviación estándar de eventos promedio por semana
event_week_2	cantidad de eventos realizados en la 2da semana desde su instalación
registration_complete	cantidad de eventos 'registration_complete'
average_events_per_week	cantidad de eventos totales promedios por semana
event_week_1	cantidad de eventos realizados en la 1era semana desde su instalación
event_week_3	cantidad de eventos realizados en la 3era semana desde su instalación
install_time	tiempo en el que el usuario instaló la aplicación
std_time_diff_sec	desviación estandar de diferencia de tiempo entre eventos
attributed_touch_time	tiempo en el que el usuario se atribuyó
event_week_4	cantidad de eventos realizados en la 4ta semana desde su instalación
average_events_per_month	cantidad de eventos promedio por mes
first_login	cantidad de eventos 'first_login'
max_events_per_day	máxima cantidad de eventos en un día
event_week_5	cantidad de eventos realizados en la 5ta semana desde su instalación
install_month	número de mes del momento de la instalación

Los modelos basados en las metodologías Boosting han demostrado ser herramientas muy útiles y capaces de encontrar patrones complejos dentro de grandes cantidades de información que no se podrían descubrir fácilmente si una persona se propusiera hacerlo. Sin embargo, dentro de estas dos herramientas la que ha demostrado ser mucho más eficaz al momento es CatBoost y que XGBoost no ha podido mostrar el mejor potencial en esta ocasión a pesar de haber sido el algoritmo al que más tiempo computacional consumió.

7. Enfoque de negocio y discusión

En esta sección se busca brindar un espacio para darle sentido de negocio y aplicación a las probabilidades obtenidas anteriormente en el modelo. Para realizar esto, se expondrán algunas preguntas recurrentes relacionadas con el Churn y las estrategias de retención en el ámbito de las aplicaciones móviles, buscando analizarlas y llevarlas a discusión.

Algunas preguntas de las cuales se hablará y discutirá serán:

- ¿Es lógico aplicar las mismas estrategias de marketing a usuarios con diferentes niveles de intención de abandono de la aplicación? ¿Y a aquellos con diferentes LTV reportados?
- ¿Cuáles podrían ser los efectos de una mala estrategia de retención?
- ¿Es posible que sea demasiado tarde para aplicar la estrategia de Marketing, y sea irreversible la intención de darse de baja de la aplicación?
- ¿Tiene sentido aplicar estrategias de marketing o retención a usuarios que ya son leales a la plataforma?
- ¿Es posible que el verdadero desafío no radique en retener a los usuarios, sino en la estrategia de adquisición de los mismos? ¿Existe la posibilidad de que esté atrayendo a mi aplicación clientes que nunca generarán un valor significativo para mi empresa?
- ¿Es conveniente retener a usuarios que no aportan valor? ¿Y existe alguna forma de evitar tomar esa acción?
- ¿Cuál es el rol de un producto y su posicionamiento en una estrategia de retención?

Poder prever si un cliente abandonará o no durante un período de tiempo determinado le da a la empresa tiempo para reaccionar y posiblemente evitar el evento de abandono. Por lo tanto, contar con la predicción de abandono es importante para cualquier negocio basado en suscripciones (Ge, He, Xiong, & Brown, 2017, p. 106–111).

Para los siguientes análisis se considera que ya se conocen las probabilidades de ocurrencia del evento de churn, dando lugar a una discusión sobre los eventos y acciones que se pueden realizar con esta información, con un enfoque centrado en el negocio para identificar estrategias efectivas de retención de clientes buscando optimizar la rentabilidad de la empresa.

7.1. Escenarios a la hora de considerar enviar estrategias de Marketing

Primeramente, al buscar reducir la tasa de abandono en una aplicación, es crucial comprender los posibles escenarios que la empresa podría enfrentar al evaluar diversas estrategias de retención.

La empresa puede categorizar a los usuarios como posibles usuarios a abandonar, y por eso se los impactará con estrategias de retención. En este caso a los usuarios con alta posibilidad de abandonar se los etiqueta como "Envía Estrategias", y en caso contrario cómo "No envía Estrategias". Luego de

enviar las estrategias de Marketing a este primer grupo y transcurrir el período mensual, se conocerá el resultado real dando por consiguiente a los 4 escenarios a continuación.

Tabla 17: *Escenarios de una empresa a la hora de enviar estrategias de Marketing.*

Resultado real	**Accionar de la empresa para el envío de estrategias**	
	Envía Estrategias	No envía Estrategias
Realiza Churn	**A:** Usuario que es impactado por estrategias y abandona la aplicación.	**B:** Usuario no es impactado por estrategias y abandona la aplicación.
No realiza Churn	**C:** Usuario es impactado por estrategias y no abandona la aplicación.	**D:** Usuario no es impactado por estrategias y no abandona la aplicación.

Para los escenarios del cuadro, la empresa enviaría estrategias de retención a los grupos A y C, y por lo tanto los grupos B y D quedarían sin recibir comunicaciones relacionadas a retención.

La empresa va a clasificar a los usuarios de cierta manera, y en función a eso enviará una o varias estrategias de Marketing para evitar que el usuario abandone la plataforma. Antes de poder clasificarlos, la empresa deberá evaluar la probabilidad de darse de baja de cada usuario en particular y se presentarán diversos casos.

Por un lado, se encontrará con clientes con una intención muy alta de darse de baja, y para estos casos una estrategia de marketing enviando por ejemplo un simple email, difícilmente impedirá que el usuario realice churn. Asimismo, también puede ocurrir que por más que el usuario tenga gran intención de abandonar la plataforma, el usuario representa un bajo valor de vida para la empresa que no valga la pena enviarle una promoción ya que el resultado final podría incurrir en pérdidas.

Por otro lado, se presentarán clientes con menor intención a darse de baja y enviando una promoción podría lograrse retenerlo, pero también el usuario podría haberse mantenido activo incluso sin haber recibido incentivo alguno, por lo que sería una pérdida de oportunidad de ganancia para la empresa.

Por esto mismo, es importante contar con diferentes tipos de estrategias para buscar la permanencia de la mayor parte de los clientes en la aplicación.

7.1.1. Grupo A: Usuario que es impactado y abandona

La desvinculación de usuarios de una plataforma, a pesar de los esfuerzos por retenerlos mediante estrategias de marketing, plantea un desafío significativo. La identificación y clasificación de usuarios potenciales de churn, seguida de la decisión de dirigirles una estrategia de marketing, implica un gasto adicional para la empresa. La implementación de estrategias de marketing conlleva costos significativos para las empresas, que pueden aumentar considerablemente cuando dichas estrategias pasan de ser simples correos electrónicos a promociones o descuentos (Kuhn & Johnson, 2013, p. 523).

Dado este grupo de usuarios que fueron impactados y las estrategias finalmente no dieron resultados, da lugar entonces a evaluar el proceso de adquisición de usuarios, siendo este mismo un paso crucial en el análisis de la retención. En numerosas ocasiones, el origen del problema radica en la fase de adquisición, donde puede resultar muy difícil retener a usuarios que nunca estuvieron verdaderamente comprometidos con el servicio de la plataforma. Muchas veces, estos usuarios se registran únicamente motivados por descuentos específicos u otras ventajas momentáneas, lo que conlleva a una baja retención a corto plazo. Este tema se va a profundizar en una sección más adelante.

Con este grupo de usuarios, la empresa debe plantearse y evaluar si no está incurriendo en altos costos al enviar comunicaciones o promociones y, finalmente, no lograr retenerlos. A su vez, es importante resaltar que estos usuarios pueden haber tomado la decisión de abandonar por diferentes motivos, pero la empresa debe asegurarse de que no haya sido a causa de las mismas estrategias de marketing. Esto podría dar lugar a los usuarios 'pasivos o dormidos', cuya situación se abordará más adelante.

Otro aspecto crucial a tener en cuenta es que, incluso cuando se detecta la intención del usuario de darse de baja, a menudo esta detección se produce tarde o la decisión de implementar una estrategia correctiva se toma demasiado tarde. Esto se debe a que el usuario ya ha tomado la decisión mucho antes, lo que puede resultar en que sea casi imposible revertir su determinación. Para estos casos, es esencial anticiparse a la pérdida del engagement del cliente. Dada la naturaleza de estos problemas, donde la decisión de abandonar el producto puede ocurrir mucho antes de que se manifieste, es crucial implementar estrategias preventivas para retener a los usuarios antes de que decidan abandonar. Esto implica no solo detectar señales de posible abandono, sino también actuar proactivamente por ejemplo buscando adquirir usuarios con un mayor valor de vida o con

mayor interés en el fin del producto, siendo entonces más posible que se logre prevenir su pérdida. A este punto se le va a dedicar una sección más adelante.

Por último pero no menos importante, dentro de este conjunto de usuarios pueden presentarse los que fueron identificados correctamente y se podría haber evitado su abandono, pero se los impactó con una comunicación o promoción no lo suficientemente efectiva o incentivadora. Por lo que entonces, este grupo de usuarios formaría parte del efecto de una mala estrategia de retención.

7.1.2. Grupo B: Usuario que no es impactado y abandona

En primer lugar, es importante reconocer que existen causas involuntarias que pueden llevar a los usuarios a dejar de utilizar la plataforma, como se ha mencionado anteriormente. Además, es posible que las herramientas de detección de intenciones de abandono no sean completamente efectivas, lo que podría resultar en la falta de identificación de usuarios en riesgo, por lo que este grupo de usuarios estarían formando parte del efecto de malas estrategias de retención.

Es importante que la empresa evalúe a estos usuarios, y se asegure que realmente no sean usuarios que en su momento le generaron gran ganancia a la empresa, y quizá ahora los está perdiendo por no haber establecido comunicaciones con ellos, ya sea por no mostrar las novedades o los nuevos features de la aplicación por ejemplo, generando la pérdida de interés por parte del usuario.

Asimismo, en algunos casos, la empresa puede optar por no enviar estrategias de retención a usuarios identificados como posibles candidatos a abandonar la aplicación si estos representan un valor bajo en términos de rentabilidad. En estos escenarios, es crucial analizar el valor de vida del cliente (LTV), ya que aunque pueda parecer una mala métrica perder un cliente al no enviarle comunicaciones o promociones, muchas veces habría resultado peor intentar retener a alguien que no genera valor para el negocio. Este enfoque subraya la importancia de revisar la estrategia de adquisición de clientes, como ya se mencionó anteriormente. En las próximas secciones, volveremos a abordar este tema en mayor profundidad.

7.1.3. Grupo C: Usuarios que es impactado y no abandona

En muchas ocasiones, al enviar estrategias de marketing a usuarios con posible intención de abandonar, se logra finalmente retenerlos y mantenerlos activos en la plataforma. Sin embargo, es importante tener en cuenta que la tasa de respuesta a menudo se sobreestima, ya que incluye a clientes que siempre responden, y que no hubieran abandonado la plataforma incluso sin haber recibido mensaje (Kuhn & Johnson, 2013, p. 524). Esto ocasiona que el beneficio estimado total de enviar estrategias finalmente no sea el beneficio real, dado que se encuentra el beneficio base incrustado de esos usuarios que son leales a la plataforma.

Dado lo mencionado, es crucial reconocer que, aunque el usuario no abandone la plataforma y fue un usuario impactado por estrategias, su retención puede no ser óptima. Se puede ocurrir que el usuario no abandone pero tampoco genere beneficios adicionales, lo que resulta en la pérdida de recursos destinados a mantener su lealtad. Esto resalta la importancia de evaluar no solo la retención en sí misma, sino también el valor que cada usuario aporta a la empresa, para optimizar así las estrategias de retención y maximizar el LTV.

Como menciona Tralice, en su tesis titulada 'Predicción de Churn de Seguros con LightGBM', la verdadera búsqueda consiste en encontrar a los clientes que cambian de parecer por el hecho de estar gestionados (o en este caso recibir una campaña de marketing), que no necesariamente es la gente que más probabilidad de churn tiene. Es por eso que en la práctica pueden ocurrir inconsistencias a lo que se estudió previamente al realizar el modelo. El modelo identifica los clientes más cercanos a darse de baja pero no toma en cuenta aquellos que ya están decididos a abandonar el servicio y no cambiarán de parecer, situación que ya se discutió previamente. Por más que el modelo aumente su grado de precisión, si no se considera correctamente estos casos, el rendimiento del modelo podrá quedar expuesto y empezará a quedar obsoleto y perderá valor.

7.1.4. Grupo D: Usuarios que no es impactado y no abandona

En este grupo de usuarios se encuentran aquellos que realizan transacciones de forma esporádica en la plataforma, con intervalos de tiempo que pueden abarcar varias semanas entre una transacción y otra. Estas transacciones pueden ocurrir debido a descuentos ofrecidos por la aplicación o simplemente por casualidad. Debido a esta baja fidelidad, la empresa decide no implementar

estrategias de retención ni dirigirles acciones de marketing, basándose en el análisis del valor de vida útil (LTV).

El análisis del valor de vida útil en este grupo de usuarios es crucial, ya que muchas veces se decide no impactarlos debido a su alta probabilidad de abandono, combinada con un bajo valor de vida útil para la empresa. Sin embargo, es importante destacar que estos usuarios pueden permanecer activos sin abandonar, aunque con transacciones poco frecuentes.

Por último, cabe destacar que dentro de estos usuarios existe un segmento cuyo valor para la empresa es considerable y, aunque no sean propensos al abandono, podrían beneficiarse de estrategias de marketing personalizadas para aumentar su compromiso y valor en el corto plazo y su lealtad en el largo plazo, lo que a su vez contribuiría a incrementar el beneficio de la empresa. Sin embargo, estas acciones se encuentran más relacionadas con el incremento del engagement que con la prevención del churn, lo cual un mayor análisis estaría fuera del alcance de este trabajo.

7.2. Usuarios pasivos o dormidos

Según la tendencia actual, los usuarios están volviéndose más selectivos, tomando decisiones informadas sobre qué aplicaciones permanecen en sus dispositivos y cuáles no (Mobio Global US, 2023).

Los usuarios 'pasivos' o 'dormidos' son aquellos que, aunque continúan utilizando la plataforma o servicio, lo hacen de manera muy limitada o sin interactuar de manera significativa. Esta categoría de usuarios puede representar un desafío particular para las empresas y podrían ser los más sensibles ante una mala estrategia de retención, ocasionando un efecto no deseado. Cuando se utilizan modelos de respuesta para la retención de clientes, existe la posibilidad de que el simple hecho de contactar a estos usuarios pueda desencadenar el abandono, ya que puede recordarles que podrían encontrar una mejor oferta con otra empresa (Kuhn & Johnson, 2013, p. 523).

Según Anjali (2016), una de las principales razones por las cuales un usuario desinstala una aplicación es recibir notificaciones irrelevantes. Además, es más probable que lo haga si el usuario ha estado inactivo por un tiempo, ya que estas notificaciones le recuerdan que posiblemente ya no necesita la aplicación.

En resumen, comprender el comportamiento del usuario es crucial en el mercado actual de aplicaciones y servicios digitales. Los usuarios cada vez más selectivos demandan aplicaciones útiles y

relevantes que respeten su tiempo y privacidad. Identificar y atender adecuadamente a los usuarios pasivos o dormidos, así como optimizar las estrategias de retención, son elementos clave para mantener la relevancia y la competitividad. En última instancia, equilibrar la personalización y la no intrusión es fundamental para fomentar la lealtad y el compromiso a largo plazo. Como parte de una estrategia adecuada, buscando evitar resultados indeseados, podrían incluir el envío de alguna promoción atractiva o incentivos significativos para mantenerse activos, como por ejemplo un descuento en el costo del delivery o de algún producto.

7.3. Clientes leales frente a adquirir nuevos

Dado que el costo de adquirir un nuevo cliente puede exceder sustancialmente el costo de retener a un cliente existente, es importante para una aplicación mantener a sus clientes. En un mercado en maduración, un negocio debería enfocarse más en retener clientes en lugar de centrarse en adquirir nuevos clientes (Ahn, Han, & Lee, 2006, p. 552–568).

A continuación, se visualiza el gasto de un típico presupuesto de Marketing para una empresa promedio, según Harvard Business School Press. (2011, p. 6)

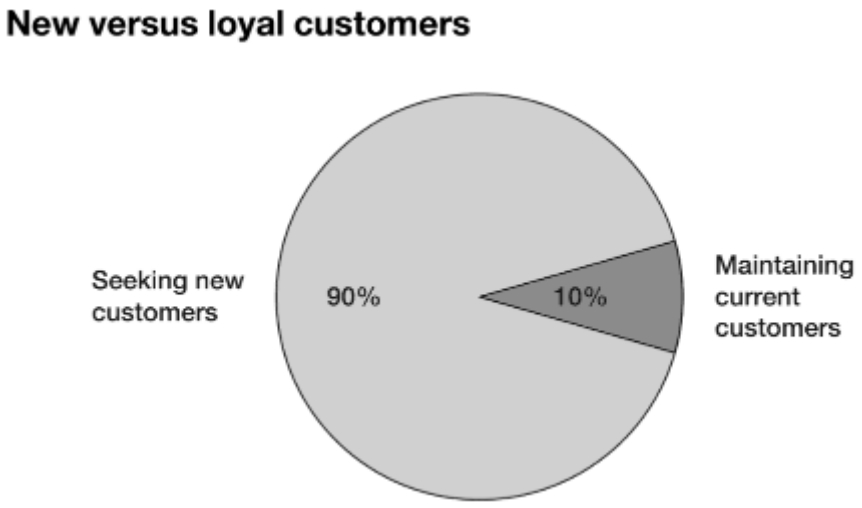

Gráfico 42: *Representación de nuevos clientes vs actuales para una empresa promedio.*

Según este documento, las empresas enfocan su energía y presupuesto en conseguir mayormente nuevos clientes, ofreciendo precios bajos de entrada e incentivos para registrarse en la aplicación, pero el gasto en Marketing y publicidad termina ascendiendo a enormes cifras. Siguiendo con la misma línea, dentro de una empresa, los mayores incentivos suelen ir a empleados que traen nuevos

clientes en lugar de a los empleados que se esmeran en tratar mantener la lealtad y la satisfacción de los mismos. Todo esto lleva a la noción de que todo cliente es un buen cliente, lo que no es ciertamente preciso.

Por un lado, el verdadero cliente leal se termina convirtiendo en un 'Apóstol', y es alguien que incluso recomienda la compañía y el servicio a otros, generando nuevos negocios y aumentando significativamente las ganancias para la empresa.

A su vez, cuanto más dura la relación con el cliente, más rentable tiende a ser el mismo. Según un estudio que menciona el documento, lograr extender la relación de un cliente de 5 a 6 años, puede resultar tanto como de 25% a 85% de incremento en la ganancia para la empresa (p. 17).

Según esta bibliografía, las compañías no prestan adecuada atención a los clientes leales ya qué realmente no aprecian que tanto valor tienen estos clientes. Las empresas deberían calcular el LTV para poder abrir los ojos ante la situación.

7.4. Un buen producto y su posicionamiento

Siguiendo con la discusión, se destaca la importancia de lograr un producto sólido, ya que en caso contrario cualquier estrategia de marketing puede carecer de sentido. Es difícil que exista una estrategia de retención adecuada para ofrecer un servicio que el cliente a simple vista pierde el interés.

Por otro lado, una excelente predicción de usuarios que pueden abandonar la plataforma sería de poca o nula utilidad si no se cuenta con un adecuado equipo de Marketing que tome diferentes estrategias y decisiones. Por esto mismo, el equilibrio entre producto, posicionamiento y predicción se vuelve crucial para el éxito de una empresa.

"Un gran posicionamiento potencia todas tus estrategias de marketing y ventas. Un posicionamiento sólido se siente como si estuviéramos haciendo trampa. Nos permite avanzar junto con las fuerzas de los mercados en los que operamos, haciendo que todo lo que hacemos en marketing y ventas sea más fácil. No importa en qué dirección nos enfrentemos, el viento sopla a nuestro favor." (April Dunford, 2019)

Además, el posicionamiento efectivo no sólo genera un impacto inicial por atraer mayores usuarios, sino que también fomenta la lealtad a largo plazo. Los clientes que perciben un valor diferenciado en un producto o servicio y se identifican con la marca tienden a mantener una relación continua con la

empresa (Reichheld, 2003). Esta lealtad se traduce en la recomendación activa a otros donde los usuarios actúan como defensores de la marca, compartiendo experiencias positivas con amigos, familiares y colegas, contribuyendo a la estabilidad y el crecimiento sostenido de la empresa

En este sentido, un buen posicionamiento, junto con un buen producto para acompañarlo, se convierten en el cimiento sobre el cual se construye la retención de usuarios. Al ofrecer una propuesta de valor clara y relevante, los usuarios terminan siendo más propensos a permanecer comprometidos con la empresa a lo largo del tiempo.

7.5. Engagement en Churn Prevention

El Customer Engagement (CE) ha sido definido de diversas maneras. Aluri, Price, & McIntyre (2019, p. 78–100) definen la participación del cliente como el nivel de presencia física, cognitiva y emocional de un cliente en la relación con una organización de servicios. Por otro lado, Brodie et al. (2011, p. 252-271) definen la participación del cliente como "la intensidad de la participación y conexión de un individuo con las ofertas y actividades de la organización iniciadas ya sea por el cliente o por la organización".

En su tesis titulada 'Modeling Customer Engagement with Churn and Upgrade Prediction' (2022), Lampinen utiliza el aprendizaje automático para prever el abandono del cliente a partir del engagement en modelos de suscripción. Destaca la importancia de entender el engagement del cliente mediante el análisis de su comportamiento, combinando experiencia de negocio con técnicas avanzadas de aprendizaje automático para predecir y evitar el abandono.

En el contexto de la predicción del churn, la relación entre el Customer Lifetime Value (LTV), el churn y el engagement desempeña un papel crucial en la gestión de clientes.

Aunque en este trabajo no se cuenta con un cálculo explícito del LTV, es importante reconocer su relevancia teórica. Si la empresa dispusiera de datos y modelos para calcular el LTV, sería de gran utilidad para predecir el churn antes de que ocurra la pérdida de engagement, y evitar llegar al punto mencionado en una sección anterior, donde el usuario ya tomó la decisión de darse baja.

La conexión entre el LTV, el churn y el engagement del cliente radica en el hecho de que los clientes altamente comprometidos tienden a generar un LTV más alto y tienen una menor probabilidad de abandonar la empresa. Por lo tanto, si se pudiera prever el churn antes de que se produzca una

disminución significativa en el engagement, se podrían tomar medidas proactivas para retener a estos clientes y maximizar su LTV.

Las estrategias de retención de clientes basadas en la predicción del churn y el engagement pueden incluir la segmentación de clientes en función de su probabilidad de churn y su nivel de engagement. Por ejemplo, los clientes con un alto riesgo de churn pero un alto nivel de engagement podrían ser objeto de campañas de reactivación o de programas de fidelización personalizados para fortalecer su relación con la empresa y prolongar su ciclo de vida como clientes rentables.

En resumen, aunque no se disponga de un cálculo específico del LTV en este estudio, se reconoce su importancia potencial para la predicción del churn y la gestión del engagement del cliente. Como se mencionará más adelante, integrar el análisis del LTV en futuras investigaciones podría proporcionar una visión más completa y estratégica de cómo prevenir la pérdida de clientes y maximizar el valor a largo plazo para la empresa.

En adición a lo mencionado, se resalta el trabajo que se lleva a cabo a lo largo de la tesis de Delgado, titulada 'Predictive Customer Lifetime value modeling: Improving customer engagement and business performance', donde se expone cómo obtener el LTV puede ayudar en gran medida a las campañas de engagement, y puede ser de gran insight a la hora de decidir el envío de promociones.

8. Conclusiones

Se parte con este trabajo buscando entender el papel fundamental que toma la retención de usuarios en la industria de aplicaciones móviles, y si es posible aumentar la misma a partir de diferentes herramientas.

A modo conclusión del trabajo, se destaca la importancia de realizar un enfoque basado en el análisis de datos y el aprendizaje automático para lograr predecir el comportamiento de usuarios que podrían abandonar la aplicación, utilizando los resultados de estos modelos para poder realizar diferentes estrategias y tomar medidas para retenerlos. Estas herramientas podrían ahorrar grandes cantidades de dinero y permitirían a las empresas no solo enfocarse en la adquisición de los usuarios.

A su vez, es crucial destacar que, junto con las predicciones y el análisis de datos, el enfoque de negocio es un pilar esencial para el éxito empresarial. La combinación de una visión estratégica sólida y la capacidad de prever el comportamiento del usuario es clave para dirigir eficazmente una empresa en el competitivo mercado de las aplicaciones móviles. Estos elementos, en conjunto con

un buen posicionamiento, conforman una estrategia integral que permite establecer relaciones duraderas con los clientes y generar valor a largo plazo.

Asimismo, a partir del análisis exploratorio de datos, se puede concluir que los usuarios más valiosos para la aplicación tienden a mantenerse activos en la misma periodo a periodo. De todas maneras, se considera de suma importancia poder ejecutar estos modelos y conocer usuarios con gran probabilidad de churn, para buscar disminuir las grandes cifras de abandono que cuentan las compañías de este tipo de industria.

A su vez, se destaca que la elección de la herramienta con la cual se entrena el modelo puede ser determinante en la calidad de los resultados, siendo CatBoost ampliamente superior que XGBoost en la mayoría de ellos y también se lo podría atribuir a su modernidad.

Finalmente, se enfatiza la importancia de haber realizado análisis y exploración de datos en las primeras etapas del trabajo ya que luego de realizar *feature engineering* e implementar estas variables, se logró ver que contribuyeron en el aprendizaje de los modelos en gran medida, generando muy buenos resultados.

9. Trabajo futuro

Uno de los principales desafíos en este proyecto fue la gestión de una extensa base de datos. Procesar una cantidad tan vasta de información y realizar cálculos para generar variables no fue una tarea trivial. En futuras versiones de este estudio, o en caso de buscar mejorarlo, se podría considerar, en el mejor escenario, utilizar computadoras con una capacidad de memoria mayor. Si eso no fuera factible, otra alternativa sería adaptar el modelo y determinar cuál es la variable de tendencia más relevante para la predicción.

En consonancia con la exploración de nuevas variantes, también se podrían probar otros algoritmos de clasificación. En este proyecto se optó por utilizar XGBoost y CatBoost, principalmente debido a su alto poder predictivo. Sin embargo, también se podrían considerar otros tipos de algoritmos, como por ejemplo, las Máquinas de Vectores de Soporte, que presentan un nivel de complejidad similar a los utilizados en este contexto.

Por otro lado, hubiera sido de gran utilidad contar con la disponibilidad de otros eventos dentro de la aplicación, lo cual habría proporcionado al modelo una base más sólida para realizar predicciones más precisas. Por ejemplo, la inclusión de eventos como la validación de datos o la utilización de cupones habría sido de gran ayuda. Además, la posibilidad de buscar mejoras al agregar información

detallada de compras en caso de obtenerlas, o mediante la incorporación de datos georeferenciados, podría potenciar aún más el modelo. Asimismo, se podría potenciar el modelo mediante el entrenamiento con una mayor cantidad de datos, especialmente si se dispone de registros más antiguos. De esta manera, se obtendría una perspectiva más completa y enriquecida que permitiría al modelo capturar patrones y tendencias de manera más efectiva.

Por último, se sugiere para un futuro trabajo poder realizar diferentes estrategias de retención a partir de los resultados obtenidos del modelo. A estos resultados se los podría combinar junto con un modelo de aprendizaje no supervisado para la segmentación de usuarios, donde entonces se podría realizar dos tipos de estrategias a modo de ejemplo:

- Ofrecer descuentos tentadores a usuarios con gran probabilidad de churn y que pertenecen a un segmento de mayor valor para la empresa. Cabe destacar que es importante acompañar a estos descuentos tentadores con un modelo de segmentación para disminuir la cantidad de usuarios que los recibe, ya que la empresa posiblemente incurra en grandes pérdidas si se los envía a toda la base de usuarios por igual.
- Ofrecer descuentos de menor valor para la empresa, como por ejemplo, descuentos de porcentajes menores. También se podría estudiar el envío de diferentes mensajes como email, SMS o push notifications, mostrando novedades o diferentes características de la aplicación buscando evitar que estos usuarios abandonen la misma.

Estas estrategias pueden complementarse con el análisis del Valor de Vida del Cliente (LTV), permitiendo la personalización de descuentos para usuarios con mayor LTV. De esta manera, se evita perder a clientes que podrían generar ingresos más significativos para la compañía. Integrar el análisis del LTV en futuras investigaciones proporciona una visión más completa y estratégica para prevenir la pérdida de clientes y maximizar el valor a largo plazo para la empresa.

10. Bibliografía

1. Aluri, A., Price, B. S., & McIntyre, N. H. (2019). Using machine learning to cocreate value through dynamic customer engagement in a brand loyalty program. Journal of Hospitality & Tourism Research.
2. Ames, Iowa. (2019). Customer Churn: A Study of Factors Affecting Customer Churn using Machine Learning . Iowa University. Recuperado de:

https://dr.lib.iastate.edu/server/api/core/bitstreams/963c8e0d-4209-4137-9d05-ac20968963f9/content

3. Anjali, Jain. 2016. 10 reasons why users uninstall your mobile app. Recuperado de: 10 reasons why users uninstall your mobile app - CleverTap
4. Arrioja Landa Cosio, N. (2021, octubre). La maldición de la dimensionalidad. Recuperado de Feature Selection Techniques in Machine Learning (Updated 2024) (analyticsvidhya.com)
5. Bejarano, Jason. (2022). ¿Cuánto Invertir en Publicidad? Recuperado de: https://www.linkedin.com/pulse/cu%C3%A1nto-invertir-en-publicidad-jerson-bejarano/?originalSubdomain=es
6. Brodie, R. J., Hollebeek, L. D., Juric, B., & Ilic, A. (2011). Customer engagement: Conceptual domain, fundamental propositions, and implications for research. Journal of Service Research, 14(3).
7. Cuervo Sánchez, C.A. (2021). Effects of Artificial Intelligence on Marketing Strategies. Recuperado de: https://dialnet.unirioja.es/descarga/articulo/7705935.pdf
8. De Bock, K. W., & Poel, D. V. d. (2011). An empirical evaluation of rotation-based ensemble classifiers for customer churn prediction. Expert Systems with Applications, 38.
9. Dunford, April. (2019). Obviously Awesome. How to Nail Product Positioning So Customers Get It, Buy It, Love It.
10. Forbes. (2023). 4 de cada 10 personas compran en supermercados a través de apps de delivery. Recuperado de: Según una encuesta, 4 de cada 10 personas compran en supermercados a través de apps de delivery - Forbes Argentina
11. Ge, Y., He, S., Xiong, J., & Brown, D. E. (2017). Customer churn analysis for a software-as-a-service company. En 2017 Systems and Information Engineering Design Symposium (SIEDS).
12. Harvard Business School Press. (2011). Focusing on your customer. Recuperado de: Focusing on Your Customer - Harvard Business Review - Google Books
13. Henry, Isabelle. (2023). ¿Cómo luchar eficazmente contra el churn?. Recuperado De: ¿Cómo luchar eficazmente contra el churn? | Actito
14. Huang, Y. & Kechadi, T. (2013). An effective hybrid learning system for telecommunication churn prediction. Expert Systems with Applications, 40.
15. Idris, A., Rizwan, M., & Khan, A. (2012). Churn prediction in telecom using random forest and PSO based data balancing in combination with various feature selection strategies. Computers and Electrical Engineering, 38(6).

16. J.-H. Ahn, S. P. Han, and Y.-S. Lee. (2006). Customer churn analysis: Churn determinants and mediation effects of partial defection in the korean mobile telecommunications service industry. Telecommunications Policy.
17. John, Brian. (2022). How to Implement Customer Churn Prediction. Recuperado de: https://neptune.ai/blog/how-to-implement-customer-churn-prediction
18. John, Brian. (2023). When to Choose CatBoost Over XGBoost. Recuperado de: https://neptune.ai/blog/when-to-choose-catboost-over-xgboost-or-lightgbm
19. Kuhn, M., & Johnson, K. (2013). Applied predictive modeling (Vol. 26). Recuperado de: applied-predictive-modeling-max-kuhn-kjell-johnson_1518.pdf (wordpress.com)
20. Lampinen, S. (2022). Modeling Customer Engagement with Churn and Upgrade Prediction [Tesis de maestría, Universidad de Helsinki, Programa de Maestría en Ciencia de Datos]. Recuperado de https://helda.helsinki.fi/server/api/core/bitstreams/59a2a22f-de66-45ee-9162-1c02771ee697/content
21. Maan, J., & Maan, H. (2023). Customer Churn Prediction Model using Explainable Machine Learning. Recuperado de: https://arxiv.org/abs/2303.00960
22. Mobio Global US. 2023. Uninstalling Apps: Challenges & Solutions. Recuperado de: Uninstalling Apps: Challenges & Solutions | Mobio Group | by Mobio Global US | Medium
23. Ramirez, Sandra. (2021). Adquisición de clientes VS. Retención. Recuperado de: https://www.linkedin.com/pulse/adquisici%C3%B3n-de-clientes-vs-retenci%C3%B3n-d%C3%
24. Ramos, Mariano. (2021). Diferencias y funciones de las principales apps de comida a domicilio en México. Recuperado de: https://marketing4ecommerce.mx/didi-food-uber-eats-rappi-la-funciones-de-las-apps-de-comida-a-domicilio-en-mexico-y-sus-principales-diferencias/
25. Reichheld, F., & Teal, T. (1996). The Loyalty Effect: The Hidden Force Behind Growth, Profits, and Lasting Value. Harvard Business School Press.
26. Reichheld, Frederick. (2003). The One Number You Need to Grow. Recuperado de: The One Number You Need to Grow (hbr.org)
27. Rocket Lab. (2021). El delivery de alimentos y su valor en el 2023. Recuperado de: https://www.interempresas.net/Alimentaria/Articulos/356492-El-delivery-de-alimentos-alcanzara-un-valor-de-25-mil-millones-de-euros-en-Europa-en-2023.html
28. Tessitore, Sabrina. (2023). What's the Average Churn Rate by Industry?. Recuperado de: https://customergauge.com/blog/average-churn-rate-by-industry

29. Tralice, F. (2019). Predicción de Churn de Seguros con LightGBM (Tesis de maestría, Universidad Torcuato Di Tella, Master in Management + Analytics). Recuperado de: Predicción de Churn de Seguros con LightGBM (utdt.edu)
30. Material de cátedras.