

**INSTITUTO TECNOLÓGICO DE BUENOS
AIRES – ITBA**

ESCUELA DE GESTIÓN Y TECNOLOGÍA



Algoritmo de Predicción Churn de usuarios utilizando Machine Learning

AUTORES:

Binello, Matías (Leg. N° 60301)

Ferrari Aguilera, Rocío María (Leg. N° 58727)

DOCENTES TITULARES:

Gonzáles, Rubén D.

Nosetti, Inés.

Rodríguez Varela, Pablo.

**PROYECTO FINAL PRESENTADO PARA LA OBTENCIÓN DEL TÍTULO
DE LICENCIADO/A EN ANALITICA EMPRESARIA Y SOCIAL**

BUENOS AIRES

PRIMER CUATRIMESTRE, 2022



Índice

Resumen Ejecutivo	2
Objetivo del Proyecto	2
Medición de Valor y KPIs a impactar	3
Investigación de Metodologías y Fundamentación teórica	3
Alcances y Limitaciones del proyecto	4
Preguntas Iniciales	4
Etapas del proyecto	4
Plan de trabajo	5
Datos a utilizar	6
Herramientas a utilizar	6
Primera exploración y Limpieza de la Base	7
Análisis de Outliers	10
Conclusiones a Priori	13
Análisis Descriptivo	14
Análisis Estadístico	20
Hipótesis Planteada	22
Caso de Negocio	23
Investigación y Consideraciones	24
Desarrollo de modelos	26
Análisis y Comparación de resultados	29
Conclusiones	32
Próximos Pasos	32
Bibliografía	33



Resumen Ejecutivo¹

En el presente trabajo se llevó a cabo un análisis exhaustivo de la base de datos de usuarios de la empresa **XYZ**, a partir del cual se construyó un **modelo de predicción de Churn utilizando el algoritmo Random Forest**, con la finalidad de determinar aquellos usuarios que pasarán a ser inactivos al cabo de los siguientes 30 días. Teniendo en cuenta la definición de la empresa, un usuario es inactivo cuando no realiza al menos una transacción en los últimos 30 días.

El EDA (análisis exploratorio de los datos) mostró cómo se distribuyen las variables, su relación con el estado de inactividad y se determinaron los límites para limpiar la base y sacar los outliers y missings con R. La base a la cual se llegó incluye a un total de **373.811 usuarios**.

El modelo final seleccionado fue construido con Python con el cual se determinó que de los **193.656 usuarios activos** y con una tasa de Churn del **28%**, el costo de oportunidad es de **498.315,62 USD**, perdiendo **54.224 usuarios por mes**, sabiendo que el LTV (Lifetime Value) es de **15 USD** y el CAC (Costo de Adquisición de un cliente) es de **5,81 USD**. Con un **10%** de retención a través de campañas de marketing específicas para estos usuarios identificados, se estima que la cantidad de usuarios inactivos descendería a **48.801**, con un costo de oportunidad asociado de **461.768,86 USD**. Es decir, que **al reducir en un 2,8% los usuarios inactivos, se disminuyen las pérdidas un 2,05%**

Objetivo del Proyecto

- **Problema identificado**

XYZ es una startup argentina en **X** sector que actualmente está trabajando arduamente para adquirir nuevos usuarios de manera masiva. Sin embargo, no posee una estrategia de retención de usuarios desarrollada. Entendiendo el *churn* como usuarios inactivos que no realizan una transacción en los últimos 30 días.

- **Cómo se piensa resolver el problema**

La hipótesis que se plantea es que al desarrollar un algoritmo de predicción de *churn* de usuarios (o tasa de pérdida de usuarios), se podrán identificar a los usuarios que estén por ser inactivos para adoptar una política de marketing moderna, eficaz y personalizada, que lleve a retenerlos. El equipo de marketing podría entonces enfocar sus campañas a esos usuarios potencialmente inactivos y así ser más eficientes con el presupuesto, ya que se

¹ Todos los números debieron ser modificados debido a los requisitos de confidencialidad de la empresa. Asimismo, el nombre de la empresa también debió ser ocultado a pedido de la empresa.



tiene el conocimiento que retener a un usuario resulta menos costoso que adquirir uno nuevo.

Medición de Valor y KPIs a impactar

- Porcentaje de usuarios inactivos: Este dato es el fundamental para comprender no solo de donde se parte, sino que permitirá medir el valor de la solución en el tiempo.
- Costo de un usuario inactivo: Contando no solo lo que le cuesta a **XYZ** mantener a ese usuario inactivo en la aplicación sino que también sumando el costo de oportunidad de que este usuario no esté transaccionando.
- Lifetime Value (LTV) de un cliente

Investigación de Metodologías y Fundamentación teórica

Las empresas en el sector en que se encuentra la empresa han crecido de manera exponencial en los últimos años a nivel mundial. Según el World Retail Banking Report del año 2019, un 66,8% de clientes bancarios han utilizado o intentado utilizar una cuenta de compañías no tradicionales en los últimos tres años. Esto supone una gran atracción de usuarios para estas compañías que no paran de crecer y presupone un desafío enorme para reinventarse y retenerlos.

En primer lugar, se identificaron algunos puntos que justifican de manera teórica realizar un churn de usuarios para **XYZ**. Estos son:

1. La retención de usuarios reduce en cierta medida la necesidad de atraer a nuevos clientes, permitiendo a las organizaciones enfocarse en fortalecer las relaciones con usuarios existentes. La trayectoria de los clientes que ya adquirieron familiaridad con la aplicación y generaron cierto hábito de consumo ayuda a explicar esta situación.
2. Los clientes viejos, que están familiarizados con la empresa, tienden a comprar más cuando están satisfechos, esto puede lograrse realizando campañas de marketing personalizadas según los grupos de usuarios.
3. Los clientes a largo plazo son generalmente menos receptivos a campañas de marketing de la competencia.
4. La pérdida de clientes es un costo de oportunidad porque reduce las ventas y necesita de una adquisición de nuevos clientes para equiparar los gastos.



Al contar con una fundamentación teórica que respalde llevar a cabo un churn para una organización, faltaría averiguar el cómo. Para ello, se analiza el *paper* de una institución financiera radicada en Brasil, un país que ha crecido en gran medida en materia FinTech. En este estudio se analiza la propensión de un cliente a darse de baja de la institución. Fue llevado a cabo mediante 6 algoritmos distintos de Machine Learning: Decision Trees, K-nearest neighbours, Elastic Net, Logistic Regression, Support Vector Machines y Random Forest. Luego, se han comparado todos estos resultados para decidir qué modelo tiene mejor precisión para ser utilizado. Esto da una idea de las distintas maneras que existen para llevar a cabo un modelo de Churn, y la relevancia de no construir un solo modelo, sino comparar varios para detectar el mejor y más eficiente.

Alcances y Limitaciones del proyecto

Este proyecto tiene como objetivo el desarrollo de un algoritmo que permita detectar qué usuarios van a pasar a considerarse inactivos (según la definición de inactivos de la empresa **XYZ**). Se buscará identificar a los usuarios que el algoritmo reconozca como futuros usuarios inactivos sin una descripción detallada del por qué, sino que se proporcionará una idea general basada en las variables identificadas como relevantes en el comportamiento y perfilamiento de los usuarios.

Etapas del proyecto

Etapas 1: Análisis descriptivo (EDA)

Esta será una herramienta con la cual se trabajará para entender mejor el perfil y comportamiento de los usuarios. Consiste en las siguientes tareas:

- Obtener los datos: teniendo en cuenta las encriptaciones necesarias para no comprometer a la empresa.
- Detectar las variables relevantes que serán estudiadas.
- Limpiar la base de datos: sacar datos que están ensuciando la base.
- Importar datos a Rstudio: esta herramienta permitirá evaluar datos atípicos y el comportamiento de los mismos.
- Análisis estadístico: comprender el funcionamiento de las métricas de la empresa, así como los insights que ellos consideran pertinentes.
- Análisis multivariado: estudiar la correlación entre variables.
- Cálculos de KPIs
- Crear visualizaciones



Etapa 2: Algoritmo de predicción

El entregable final y más importante del proyecto, permitirá al equipo de Marketing o Growth obtener una lista de usuarios cada vez que se corra el algoritmo, la cual dirá cuál es la probabilidad de que un usuario comience a ser inactivo en el X período de tiempo que el equipo setee. Lo cual requiere:

- Investigar librerías y códigos Python
- Desarrollar modelos y métricas: entrenando y preparando distintos algoritmos de predicción.
- Comparar resultados
- Seleccionar el mejor modelo
- Eficientizar y mejorar el modelo final

Plan de trabajo

Teniendo en cuenta que el plazo del proyecto va desde el mes de marzo hasta junio inclusive, se creó el primer diagrama de Gantt que visualiza semanalmente, cada una de las tareas con sus respectivos plazos. En este caso, se toma como semana 1 a la semana del 28 de marzo y tiene como fecha de fin el viernes 1 de julio.

Tareas	28 mar	4 abr	11 abr	18 abr	25 abr	2 may	9 may	16 may	23 may	30 may	6 jun	13 jun	20 jun	27 jun
Obtener los datos														
Detectar variables relevantes														
Limpiar la base														
Importar datos a RStudio														
Análisis estadístico														
Análisis multivariado														
Cálculos de KPIs														
Crear visualizaciones														
Investigar librerías y códigos Python														
Desarrollar modelos y métricas														
Comparar resultados, seleccionar el mejor														
Eficientizar y mejorar el modelo final														

Figura 1: Diagrama de Gantt por semana



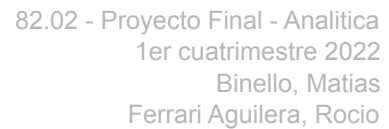
Datos a utilizar

Los datos disponibles para el desarrollo del modelo son los siguientes:

- user_id: Dato codificado para la protección de datos del usuario y la empresa
- primer_registro_app
- province
- locality
- zip_code
- cashback_enabled: Dato binario que nos cuenta si tiene habilitada la opción de cashback en la aplicación
- account_state: Habilitada/Pendiente de verificación/pausada
- verified_phone: Variable binaria siendo 1 si el celular está validado y 0 en caso contrario.
- gender
- edad
- nationality
- activity
- income_source
- civil_status
- volumen: Variable categórica que genera rangos del volumen de uso semanal. (Alto, Medio - Alto, Medio, Medio - Bajo, Bajo, Casi Nulo, Nulo)
- aum: Variable categórica que genera rangos según el AUM (assets under management) que poseen los usuarios de **XYZ** (Muy Alto, Alto, Medio - Alto, Medio, Medio - Bajo, Bajo, Casi Nulo, Nulo)
- estado: Variable dicotómica que indica Activo/Inactivo si el usuario realizó una transacción o actividad en los últimos 30 días.
- contactos_cx: Cantidad de contactos realizados al equipo de customer support
- tokens: Cantidad de criptomonedas que posee el usuario, no es un monto sino un contador de los distintos tipos de tokens que posee en cartera con dinero
- transacciones: cantidad de transacciones históricas realizadas por el usuario
- balance_fiat: monto de pesos que posee el usuario, pasado a dólares. Dato codificado para la protección de datos del usuario y la empresa
- balance_crypto: monto de cryptos que posee el usuario, pasado a dólares. Dato codificado para la protección de datos del usuario y la empresa

Herramientas a utilizar

- **R Studio**: Se utilizará este lenguaje para realizar el análisis descriptivo de las bases.
- **Python**: Lenguaje de programación en el cual se desarrollará el modelo predictivo.



En una primera instancia, se realizó una aproximación para entender mejor la base con la que se trata. Se cuenta con 886.145 usuarios y 23 variables, siendo estas del tipo numéricas y categóricas. Se puede ver que hay algunas variables que serán molestas debido a que poseen muchos datos nulos o de texto ingresado manualmente por el usuario, un gran foco de problemas de inconsistencia de datos. En la tabla siguiente se puede observar el tipo de dato asociado a cada variable y un breve *head* de la base que da indicio de la información de algunos usuarios.

Figura 2: Dimensión y tipos de variables de la base Usuarios

cashback_enabled	verified_phone	edad	contactos_cx
Min. : 0.0000	Min. : 0.000000	Min. : -78.00	Min. : 0.0000
1st Qu.:1.0000	1st Qu.: 0.000000	1st Qu.: 23.00	1st Qu.: 0.0000
Median :1.0000	Median : 0.000000	Median : 28.00	Median : 0.0000
Mean : 0.9963	Mean : 0.005147	Mean : 30.77	Mean : 0.3343
3rd Qu.:1.0000	3rd Qu.: 0.000000	3rd Qu.: 36.00	3rd Qu.: 0.0000
Max. :1.0000	Max. : 1.000000	Max. :2021.00	Max. :623.0000
NA's :2	NA's :2	NA's :2	NA's :2
tokens	antigüedad	transacciones	balance_fiat
Min. : 0.0000	Min. : 0.00	Min. : 0.00	Min. : -360107
1st Qu.: 0.0000	1st Qu.: 43.00	1st Qu.: 0.00	1st Qu.: 0
Median : 0.0000	Median : 84.00	Median : 1.00	Median : 0
Mean : 0.9003	Mean : 97.49	Mean : 23.16	Mean : 4728
3rd Qu.: 1.0000	3rd Qu.:128.00	3rd Qu.: 17.00	3rd Qu.: 0
Max. :17.0000	Max. :487.00	Max. :15216.00	Max. :24986663
NA's :2	NA's :2	NA's :2	NA's :2
balance_crypto			
Min. : -102741			
1st Qu.: 0			
Median : 0			
Mean : 2810			
3rd Qu.: 0			
Max. :30062864			
NA's :2			

Figura 3: Estadísticos de las variables numéricas.



Existe un evidente problema con los datos que se observan, ya que, por ejemplo la variable edad cuenta con inconsistencias en los números, yendo de -78 años a 2021. Por esta razón, se filtró el campo edad para limitar a los usuarios con edades entre 18 y 99 años. Algo similar ocurre con las columnas balance fiat y crypto ya que hay dos cuentas que utiliza la empresa para realizar pagos que tienen balances negativos. También se procede a quitar estos usuarios.

Nuevo summary después de limpiar:

cashback_enabled	verified_phone	edad	contactos_cx
Min. :0.0000	Min. :0.00000	Min. :18.00	Min. : 0.000
1st Qu.:1.0000	1st Qu.:0.00000	1st Qu.:23.00	1st Qu.: 0.000
Median :1.0000	Median :0.00000	Median :28.00	Median : 0.000
Mean :0.9963	Mean :0.00502	Mean :30.84	Mean : 0.336
3rd Qu.:1.0000	3rd Qu.:0.00000	3rd Qu.:36.00	3rd Qu.: 0.000
Max. :1.0000	Max. :1.00000	Max. :99.00	Max. :623.000

tokens	antigüedad	transacciones	balance_fiat
Min. : 0.0000	Min. : 0.00	Min. : 0.00	Min. : 0
1st Qu.: 0.0000	1st Qu.: 43.00	1st Qu.: 0.00	1st Qu.: 0
Median : 0.0000	Median : 85.00	Median : 1.00	Median : 0
Mean : 0.9057	Mean : 97.73	Mean : 23.26	Mean : 4761
3rd Qu.: 2.0000	3rd Qu.:128.00	3rd Qu.: 17.00	3rd Qu.: 0
Max. :17.0000	Max. :487.00	Max. :15216.00	Max. :24986663

balance_crypto
Min. : 0
1st Qu.: 0
Median : 0
Mean : 2827
3rd Qu.: 0
Max. :30062864

880079 obs. of 23 variables

Figura 4: Nuevos estadísticos de las variables numéricas.

Con estos filtros se sacaron a unos 6.000 usuarios. Sin embargo, se debe limpiar la base con mayor detenimiento debido a que existen 3 tipos de usuarios identificados por **XYZ** que es necesario excluir ya que incluirlos se aleja del foco de este proyecto. Estos 3 tipos son:

- “Zombie Users” que generaron la cuenta, realizaron los pasos para identificarse como usuarios pero que nunca ingresaron dinero.
- “No Kyc Users” son aquellos que se bajaron la aplicación, se registraron con mail y teléfono pero nunca realizaron el KYC o autenticación de usuario.
- “Rat User”, aquel usuario que realizó el KYC y después generó el mínimo número de transacciones para poder retirar los \$300 de regalo que otorga la empresa, por la primera vez que se baja la aplicación.

387801 obs. of 23 variables

Por último, se decidió filtrar la variable antigüedad para que sea mayor a 30 días. Esto deja de lado a los nuevos usuarios que pueden entorpecer el análisis por sus comportamientos.

Al aplicar los filtros y excluir a estos usuarios se obtiene un total de 387.801 usuarios. Como se observa en la Figura 5, estos representan un 55,2% de los datos de la tabla original. A partir de esto, se puede decir que más de la mitad de los usuarios de **XYZ**, aproximadamente, no cumplen las condiciones de usuarios a estudiar.

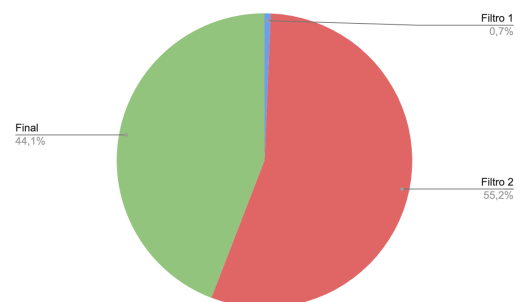


Figura 5: Composición de la base luego de filtrar.



Al volver a generar el resumen estadístico de la tabla se obtiene:

cashback_enabled	verified_phone	edad	contactos_cx	tokens
Min. :0.0000	Min. :0.000000	Min. :18	Min. : 0.0000	Min. : 0.000
1st Qu.:1.0000	1st Qu.:0.000000	1st Qu.:23	1st Qu.: 0.0000	1st Qu.: 1.000
Median :1.0000	Median :0.000000	Median :29	Median : 0.0000	Median : 2.000
Mean :0.9919	Mean :0.004778	Mean :31	Mean : 0.6563	Mean : 1.778
3rd Qu.:1.0000	3rd Qu.:0.000000	3rd Qu.:36	3rd Qu.: 1.0000	3rd Qu.: 2.000
Max. :1.0000	Max. :1.000000	Max. :94	Max. :623.0000	Max. :17.000
antiguedad	transacciones	balance_fiat	balance_crypto	
Min. : 31.0	Min. : 1.00	Min. : 0	Min. : 0	
1st Qu.: 64.0	1st Qu.: 5.00	1st Qu.: 0	1st Qu.: 0	
Median :104.0	Median : 20.00	Median : 0	Median : 0	
Mean :114.4	Mean : 50.92	Mean : 9229	Mean : 5947	
3rd Qu.:136.0	3rd Qu.: 58.00	3rd Qu.: 664	3rd Qu.: 187	
Max. :487.0	Max. :15216.00	Max. :7538142	Max. :30062864	

Figura 6: Resumen estadístico de la base final.

El resumen indica cómo se distribuyen los datos de las variables numéricas de la tabla final. La mínima cantidad de transacciones ahora es 1 y además, cambió toda la distribución de algunas variables que estaban relacionadas con los usuarios “zombies”, aquellos que no han validado su identidad “No KYC”, y usuarios “rat”.

Finalmente, se eliminó de la base las variables Estado_Civil e Income_Source ya que consistían principalmente de valores nulos que no aportan al análisis.

civil_status	cantidad
	382221
SINGLE	39788
MARRIED	6592
CO_HABITANT	3305
DIVORCED	1217
SEPARATED	521
CIVIL_UNION	284
WIDOWER	192

income_source	cantidad
	382182
PERSONAL_INCOME	48090
ASSETS_SALE	1486
INHERITANCE	1445
ECONOMIC_COMPENSATION	917

Figuras 7 y 8: Tablas de estado civil y fuente de ingresos con sus cantidades



Análisis de Outliers

A continuación se procederá a analizar los outliers o valores atípicos de aquellas variables numéricas no booleanas. Todo el tratamiento de outliers se realizó eliminando los datos superiores al percentil 0,99 salvo en ciertos casos que se indicarán a continuación. Esto fue debido a que se plantea que toda la información es relevante, pero, los valores extremos interfieren en gran medida en el análisis.

En primer lugar, se analizará la variable edad, y como se observa en las Figuras 9 y 10, se puede decir que la mayoría de los usuarios se concentran en un rango entre 20 y 30 años. Se observan casos atípicos, que se visualizan como círculos por encima de la caja. Al haber limpiado esta variable seteando un rango de 18 a 99 años de edad, no se decide trabajar con la limpieza de esta variable aún más, ya que se tiene la certeza de casos de usuarios con edades atípicas que se encuentran utilizando la aplicación. Es por esto, que se tomó la decisión de mantener los valores atípicos del campo “edad”.

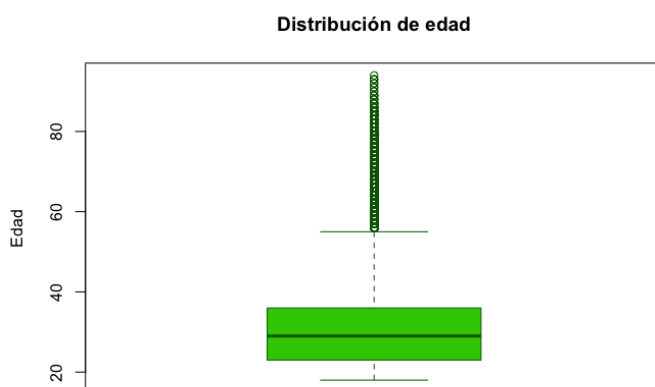


Figura 9: Diagrama de caja de la variable Edad.

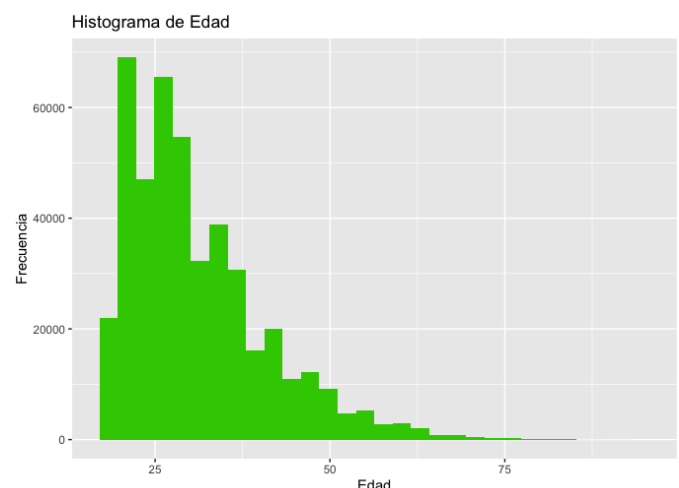


Figura 10: Histograma de Edad.

En segundo lugar, se analiza la variable cantidad de transacciones realizadas por el usuario. Al observar el primer gráfico de distribución de la muestra (Figura 11), se puede decir que en su mayoría, la variable se concentra en transacciones menores a 5.000. Los valores atípicos no permiten observar con detalle la mayor concentración de la muestra, haciendo que la caja del rango intercuartil se visualice como una línea. Es por esta razón, que se decidió acotar el rango de estudio eliminando los datos superiores al percentil 0,99. Al generar una

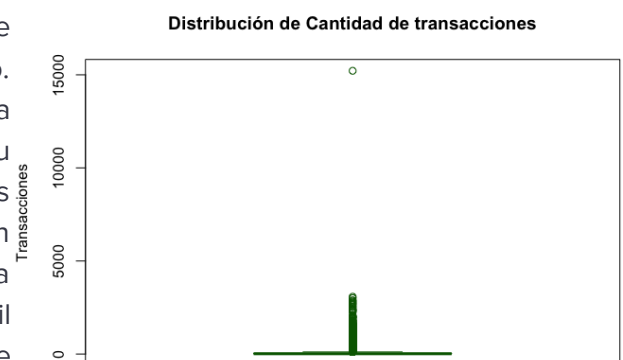


Figura 11: Gráfico de caja de la variable transacciones.



nueva base con este filtro, se graficaron las transacciones nuevamente, obteniendo los gráficos de distribución y de barras que se observan en las Figuras 12 y 13 respectivamente.

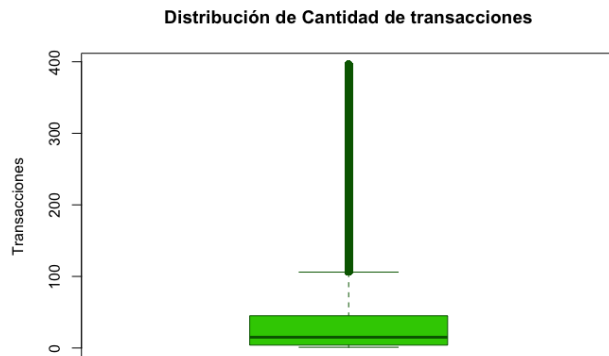


Figura 12: Nuevo Diagrama de caja de las Transacciones.

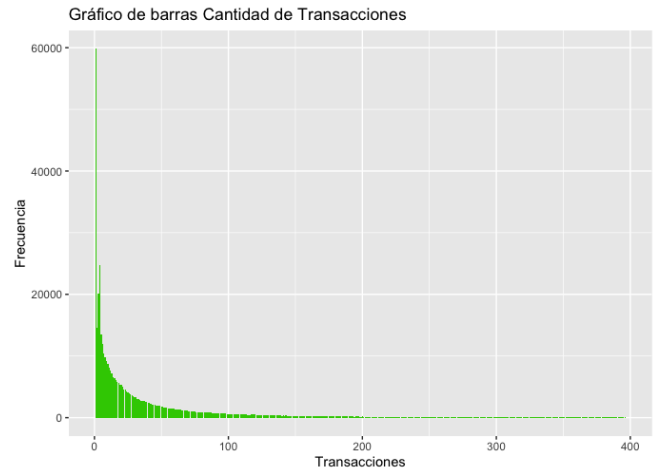


Figura 13: Gráfico de barras de transacciones.

En cuanto a los contactos a CX, se realizó el mismo procedimiento que para la cantidad de transacciones, eliminando aquellos valores mayores al percentil 0,99. Los valores resultantes poseen outliers. Sin embargo, se determinó que hasta 7 contactos es un número razonable, a diferencia de los 600 que había previamente. Se puede observar en la Figura 15 la distribución logarítmica que generan estos datos en la que existe una tendencia alrededor del 0, sin realizar contactos y que disminuyen conforme avanza la cantidad.

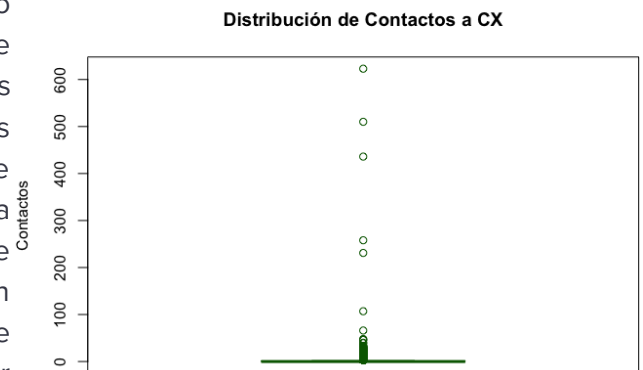


Figura 14: Distribución original de Contactos a CX.

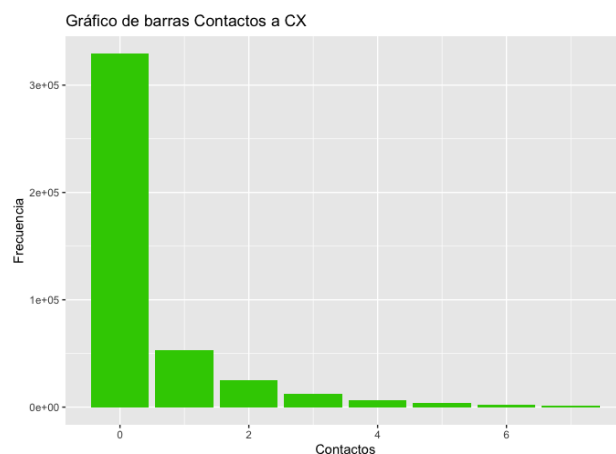


Figura 15: Gráfico de barras con eliminación de outliers.

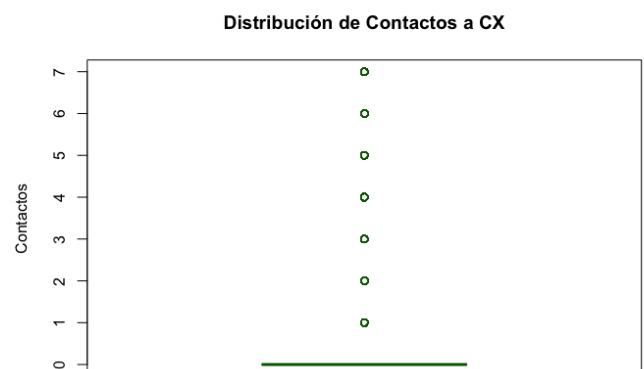


Figura 16: Nueva Distribución de Contactos a CX.



Con respecto al resto de las variables, se siguió con la metodología análoga, quitando datos de “balance fiat” y “balance crypto” con el percentil 0,99 y manteniendo la información original en los casos de “cantidad de tokens” y “antigüedad” que posee el usuario debido a que se consideran de relevancia y que no ensucian los datos.

Nuevo summary con tratamiento de outliers:

cashback_enabled	verified_phone	edad	contactos_cx	tokens
Min. :0.0000	Min. :0.000000	Min. :18.00	Min. :0.0000	Min. : 0.000
1st Qu.:1.0000	1st Qu.:0.000000	1st Qu.:23.00	1st Qu.:0.0000	1st Qu.: 1.000
Median :1.0000	Median :0.000000	Median :28.00	Median :0.0000	Median : 1.000
Mean :0.9924	Mean :0.004449	Mean :30.84	Mean :0.5391	Mean : 1.729
3rd Qu.:1.0000	3rd Qu.:0.000000	3rd Qu.:36.00	3rd Qu.:1.0000	3rd Qu.: 2.000
Max. :1.0000	Max. :1.000000	Max. :94.00	Max. :7.0000	Max. :17.000
antigüedad	transacciones	balance_fiat	balance_crypto	
Min. : 31.0	Min. : 1.00	Min. : 0.0	Min. : 0.0	
1st Qu.: 64.0	1st Qu.: 5.00	1st Qu.: 0.0	1st Qu.: 0.0	
Median :103.0	Median : 19.00	Median : 0.0	Median : 0.0	
Mean :113.1	Mean : 42.58	Mean : 4677.8	Mean : 1997.5	
3rd Qu.:136.0	3rd Qu.: 52.00	3rd Qu.: 416.4	3rd Qu.: 114.1	
Max. :487.0	Max. :396.00	Max. :172236.9	Max. :107595.4	

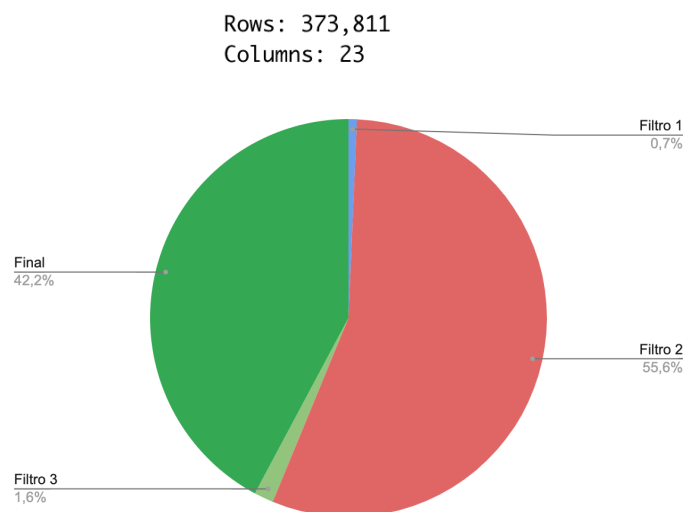


Figura 17 y 18: Summary de la nueva base y Gráfico de torta de filtros.

En estas visualizaciones se observa la nueva dimensión de la base, quedando con un total de 373.811 usuarios, manteniendo la misma cantidad de variables. Además, se muestra la proporción de usuarios que fueron dejados de lado para cada uno de los filtros realizados (Figura 18). En el caso del último filtro, representa nada más que un 1,6%, lo cual no varía en gran cantidad los números de la base como lo es el caso del filtro 2. En relación a los datos que tenía la base original, la base nueva cuenta con el 42,2% de los usuarios luego de la limpieza y el tratamiento de outliers.



Conclusiones a Priori

Habiendo limpiado la base y tratado los valores atípicos, se tiene información relevante sobre la cual se pueden inferir algunas conclusiones preliminares con las cuales trabajar la hipótesis del problema.

- La mayoría de los usuarios tiene habilitado el **cashback**.
- Los datos indican que la mayoría de los usuarios no tienen el **teléfono verificado**, lo cual sorprende, ya que al momento de crear la cuenta se debe verificar el teléfono con un mensaje SMS. Se cree que podría existir otra manera de validar la cuenta, por ejemplo a través de un código enviado al mail, que explique un poco más esta situación.
- Mirando las **edades**, se nota que la mayor porción de los usuarios ubicados en el percentil 25 al 75, son de 23 a 36 años.
- La media de **contactos a CX** es menor a 0.5, lo que significa que más de la mitad de los usuarios nunca se contactó con CX.
- La mayor parte de los usuarios tiene una baja cantidad de **tokens** o monedas distintas, siendo lo más frecuente 1 o 2. Se cree que puede ser por los altos rendimientos que se obtienen con cierto tipo de moneda como UST o DAI y el cashback que se obtiene en BTC con cada compra utilizando la tarjeta.
- Mirando la **cantidad de transacciones**, se encuentran los siguientes insights:
 - Existe al menos un usuario que tiene una sola transacción como mínimo.
 - Existe un máximo de 396 transacciones.
 - La mayor concentración de los datos se encuentran entre 4 y 45 transacciones.
- El bullet anterior da pie para hablar de la **antigüedad**. La mayoría de los usuarios de **XYZ** son relativamente nuevos debido a la magnitud con la que fue creciendo en estos últimos meses. Si se filtra la base para tener a los usuarios que tienen ya más de 100 días en la aplicación, la base de datos resultante sería más pequeña que la que se tiene actualmente. Dicho esto, se explica porqué las transacciones por usuario son tan variadas.
- Estudiar los **balances**, ya sea fiat o crypto, en términos de números no tendría sentido ya que estos valores fueron modificados para proteger a la empresa y a los usuarios. De igual manera, se utilizarán para el análisis ya que una vez que estén normalizados le serán útiles al modelo.



Análisis Descriptivo

Para comenzar con el análisis descriptivo de la base en estudio, se decidió evaluar la variable “género”. Como se visualiza en el gráfico de torta Figura 19, existe una gran cantidad de género masculino (71,19%) en contraposición con el género femenino (28,79%). Además, existe una baja proporción de datos missing (0,02%). En cantidad de usuarios, como se observa en la Figura 20, 68 usuarios no tienen llenado el campo género, 309.060 son de género masculino y 124.992 de género femenino.

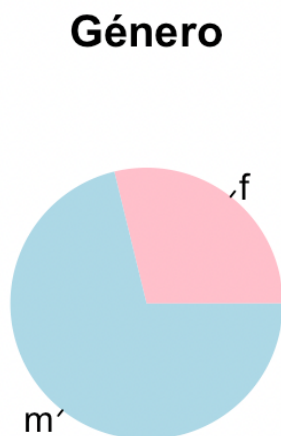


Figura 19: Gráfico de torta Género.

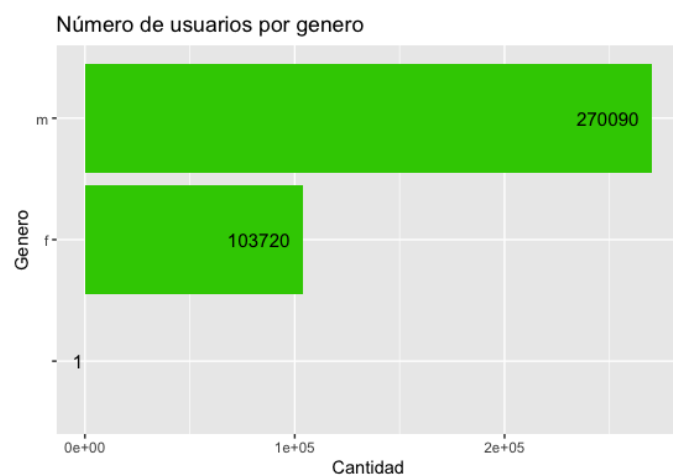


Figura 20: Gráfico de barras Género.

Resulta interesante ver la distribución de la variable género, ya que se puede ver una enorme diferencia entre lo que es género femenino y masculino. Al pensar en un por qué de la gran diferencia que se encuentra con esta variable, se podría remontar a que las empresas FinTech y todo lo relacionado con el mundo Crypto hace varios años que está fuertemente liderado y explorado en un ambiente más masculino que femenino. Sin embargo, esto ha ido cambiando con el pasar del tiempo, y es por eso que actualmente XYZ lleva una propuesta de equidad en sus empleados, siendo igualitarios a la hora de contratarlos, al igual que al momento de adquirir usuarios con campañas de marketing sin relación en el género. Igualmente, se están desarrollando campañas de marketing para explorar este sector aún más y poder captar un mayor número de usuarias.

Pasando a una variable distinta, se encuentra “provincia”, para la cual se decidió graficar de acuerdo a la cantidad de usuarios. En esta visualización (Figura 21), se nota claramente una diferencia entre las provincias del interior y Brasil, comparado con Buenos Aires (que vendría a representar Capital Federal) y la Provincia de Buenos Aires. Dentro del top 5 de Provincias, se encuentran también Córdoba y Santa Fe con una gran presencia. Cabe aclarar que las proporciones demográficas de la población también deben ser tenidas



en cuenta a la hora de sacar conclusiones en la cantidad de usuarios por provincia. Por ejemplo, difiere en gran medida la cantidad de personas que habitan en la provincia de La Pampa, o Formosa, comparadas con Buenos Aires, por lo cual se espera que la distribución se dé de esta manera.

Por último, cabe destacar que hay una leve inconsistencia de datos en esta variable, ya que hay missings y provincias repetidas por faltas de ortografía al no incluir la tilde, como en el caso de Entre Ríos.

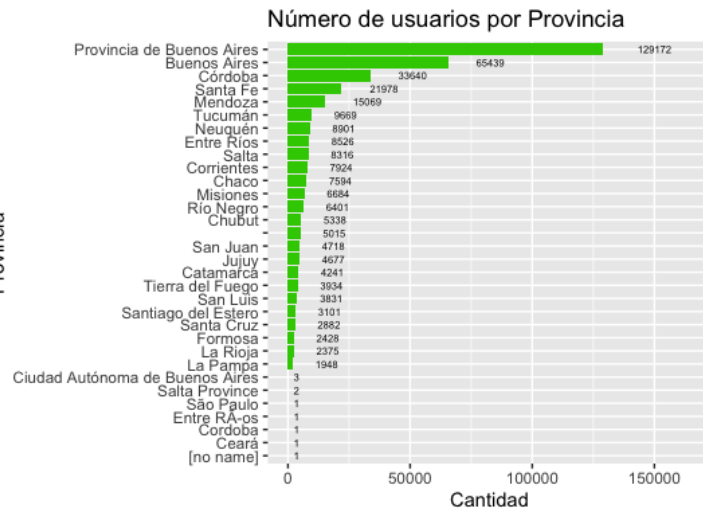


Figura 21: Gráfico de barras de usuarios por provincia

Debido a esto, se decidió agrupar las provincias en 6 zonas para deshacer este error de tipeo y para que la matriz de correlación se pueda estudiar con mayor facilidad. La distribución de usuarios en cada una de las zonas será explorada en un análisis bivariado más adelante. Las 6 zonas quedan definidas de la siguiente manera:

- **Sur:** Tierra del Fuego, Santa Cruz, Chubut, Río Negro, Neuquén
- **Centro:** La Pampa, Mendoza, San Luis, Córdoba, San Juan, La Rioja, Santa Fe, Entre Ríos
- **Norte:** Catamarca, Tucumán, Santiago del Estero, Chaco, Corrientes, Misiones, Formosa, Salta, Jujuy
- **CABA:** Buenos Aires, Ciudad Autónoma de Buenos Aires
- **Provincia de Buenos Aires**
- **Otros:** Todos los que quedan, que son aquellos de Brasil y campos nulos.

La distribución de los datos por cada zona, se visualiza en el siguiente gráfico:

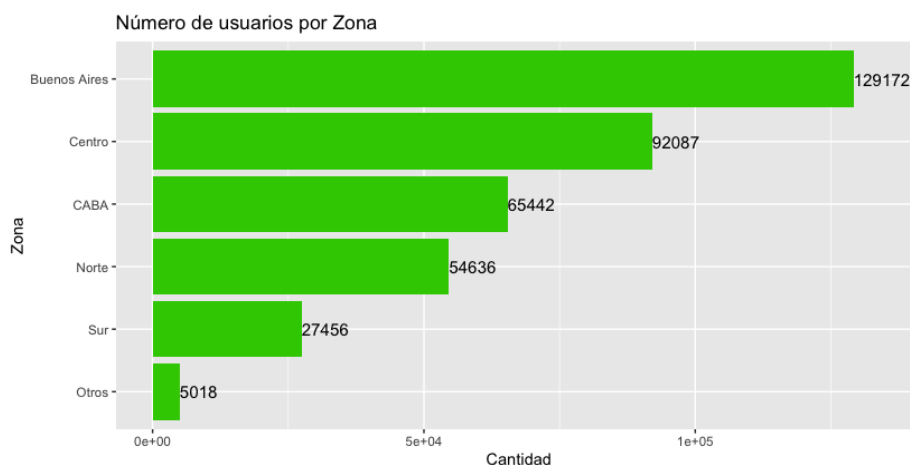


Figura 22: Gráfico de barras de usuarios por zona.



Siguiendo con el análisis descriptivo, se encuentra la variable más relevante en este estudio, que es la variable a predecir por el modelo: Estado. En este campo se identifica la variable dicotómica que puede tomar los valores de “Inactivo” o “Activo”. El estado activo está definido como un usuario que realiza al menos una transacción en los últimos 30 días, e inactivo en el caso contrario.

Al observar el gráfico de la Figura 23, se nota una diferencia entre los usuarios, siendo predominantemente Inactivos. Esto se puede deber a que muchos de los usuarios que se crean una cuenta en la aplicación, luego no realizan transacciones, o son usuarios que se han dado de baja del servicio.

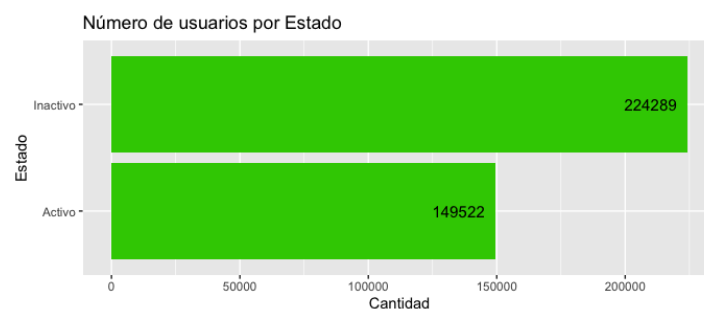


Figura 23: Cantidad de usuarios activos e inactivos.

Entrando en un análisis bivariado con la variable Estado, surge una primera impresión al ver la Figura 24, en la cual los usuarios masculinos tienen mayor tendencia a ser activos. Sin embargo, el foco de marketing en este caso no deberá centrarse en adquirir usuarios que tiendan a ser activos, sino en retener a los que no. Es por esto, que antes de analizar la significancia del dato, se considera que el género de los usuarios tendrá una gran relevancia para el algoritmo.

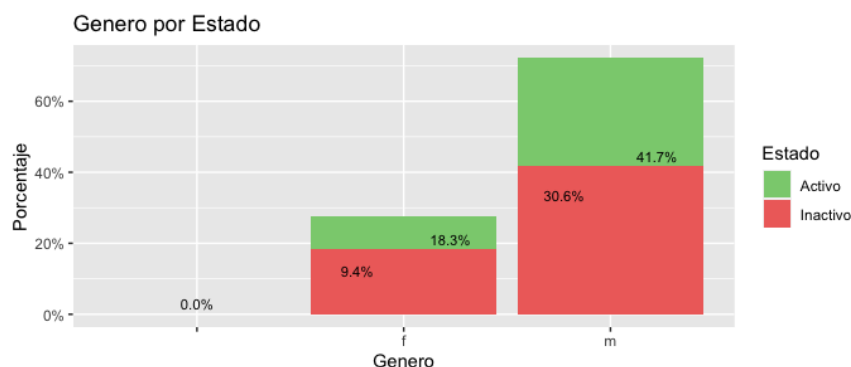
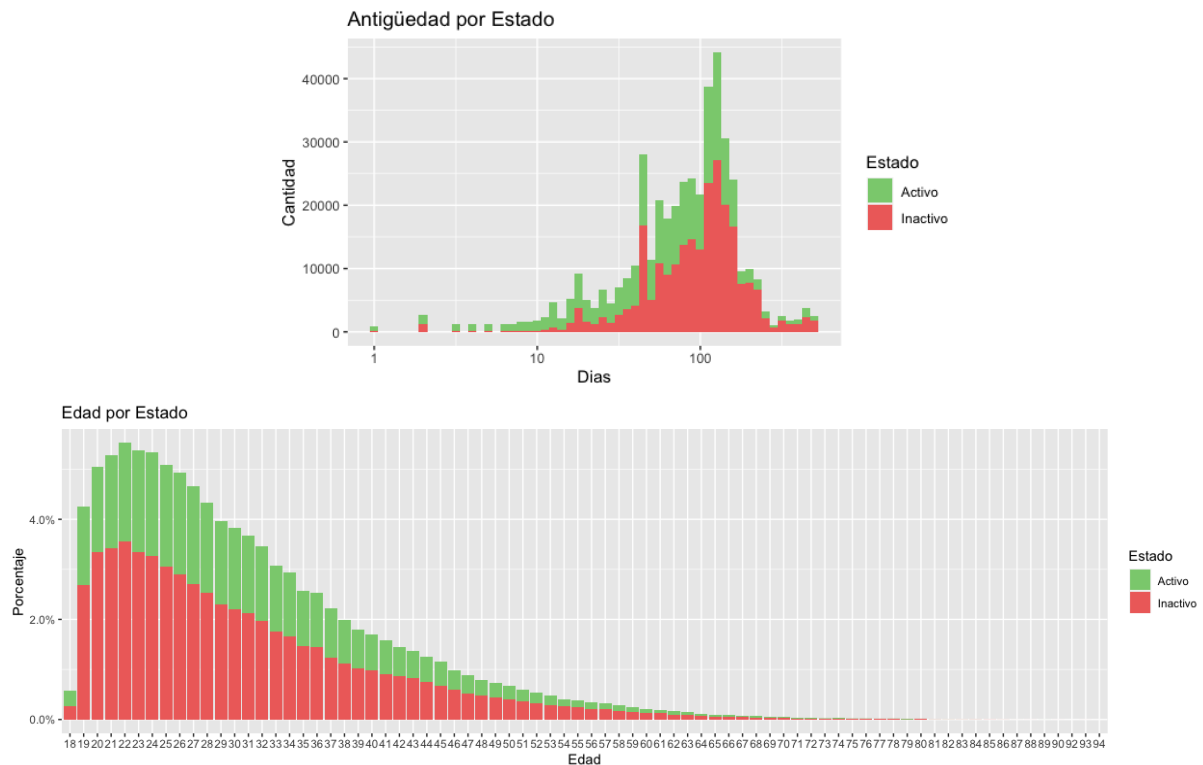


Figura 24: Gráfico de barras de Género por Estado

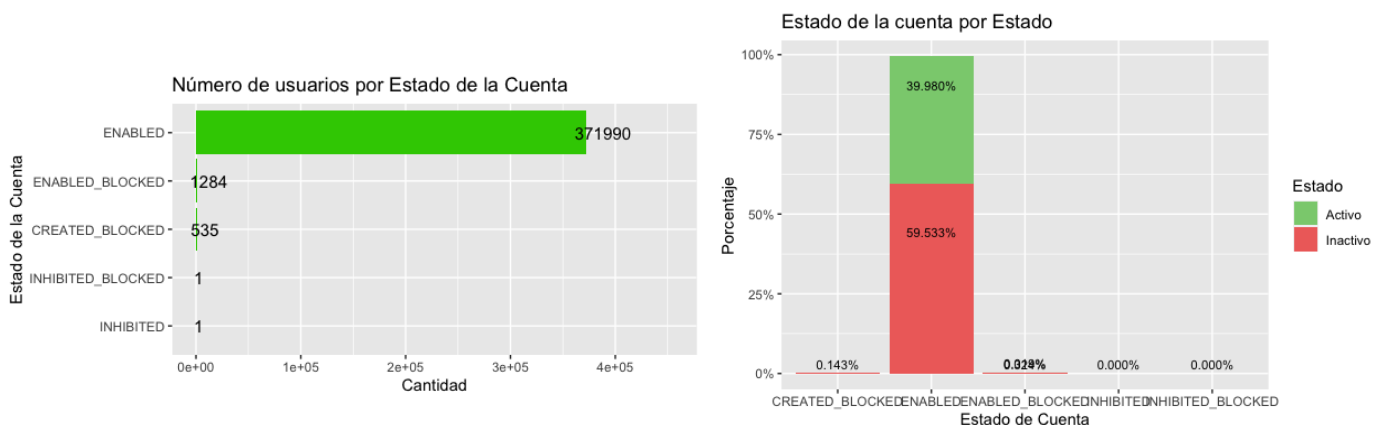
Observando la edad de los usuarios y la antigüedad, segmentada por el estado, no se obtiene una conclusión a priori, salvo por el hecho de que existe una clara tendencia de uso en personas de edad joven, como fue mencionado anteriormente. También se puede ver

el impacto de las campañas publicitarias en los picos del gráfico de antigüedad ya que para esos momentos se adquirieron grandes cantidades de usuarios nuevos.



Figuras 25 y 26: Histogramas de Antigüedad y Edad por Estado

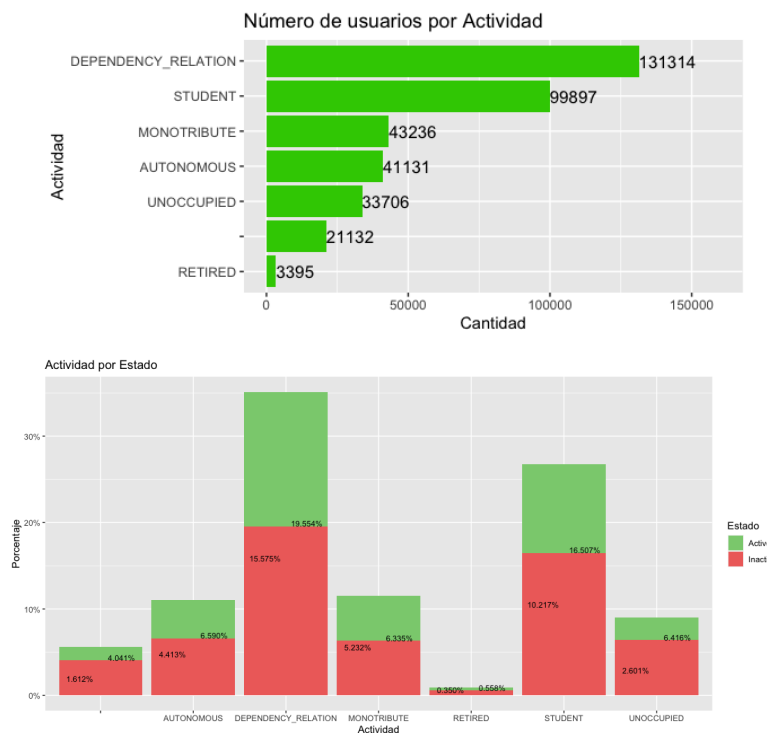
Pasando a analizar variables más de categorización del usuario, está el Estado de la cuenta, que permite diferenciar a aquellos usuarios a los que se les bloqueó la cuenta debido a un tema de compliance, los que tienen la cuenta inhabilitada por haberla pausado, o si están con la cuenta en condiciones normales como “ENABLED”. Como se observa, la mayor concentración de los datos se encuentra en esta categoría.



Figuras 25 y 26: Gráficos de barras de Estado de cuenta y Estado de Cuenta por Estado

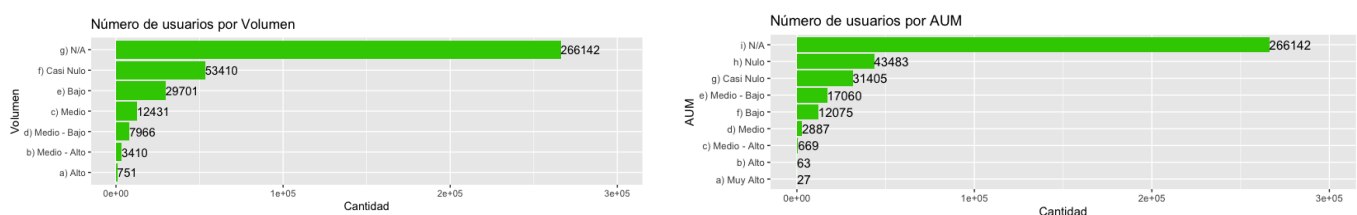


Luego, estudiando a los usuarios según la actividad que llevan a cabo, se pueden identificar dos grandes problemas. En primer lugar, los desocupados o campos nulos no son los grupos de mayor cantidad de usuarios y tienden a ser más inactivos en proporción a la cantidad de usuarios que las demás categorías. En segundo lugar, los usuarios en relación de dependencia o los estudiantes son los grupos con mayor cantidad de usuarios y presentan una distribución cercana al 40% de usuarios activos y 60% inactivos. El problema en estos casos, reside en que la cantidad de inactivos en ambos grupos es superior al resto de las categorías juntas, por lo que tal vez resulte conveniente para la empresa enfocarse en la retención de esos usuarios.



Figuras 27 y 28: Gráficos de barras de Actividad del usuario y Actividad por Estado

Mirando las distribuciones de AUM y Volumen, sucede algo peculiar en el que prácticamente todos los usuarios de los que no se tiene información son inactivos mientras que un porcentaje muy bajo de los que sí están categorizados resultan inactivos. No es una conclusión con la que se pueda trabajar en un principio, debido a la manera en que estas categorías son armadas por la empresa (están armadas para estudiar a los usuarios activos sin ser “ensuciadas” por los inactivos y por esta razón aparecen los N/A).



Figuras 29 y 30: Cantidad de usuarios por Volumen y por AUM

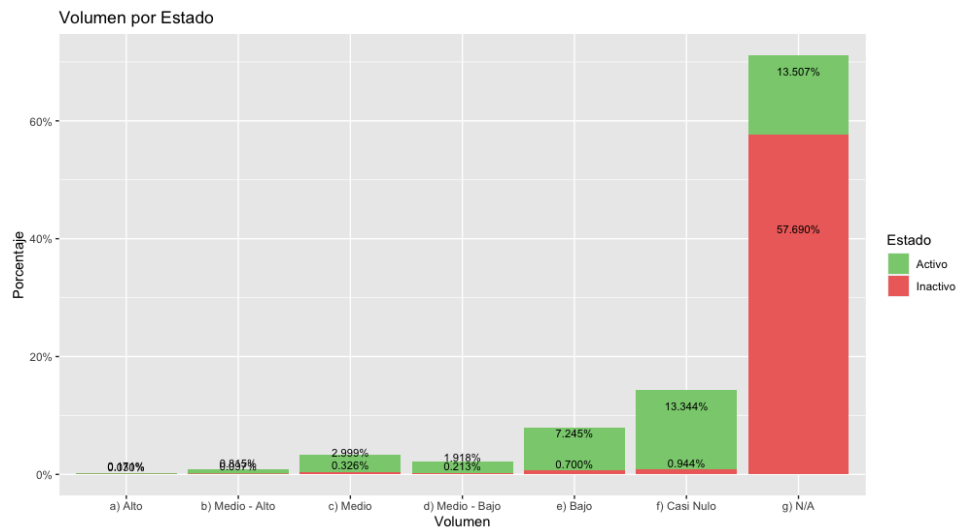


Figura 31: Gráfico de barras de Volumen por estado

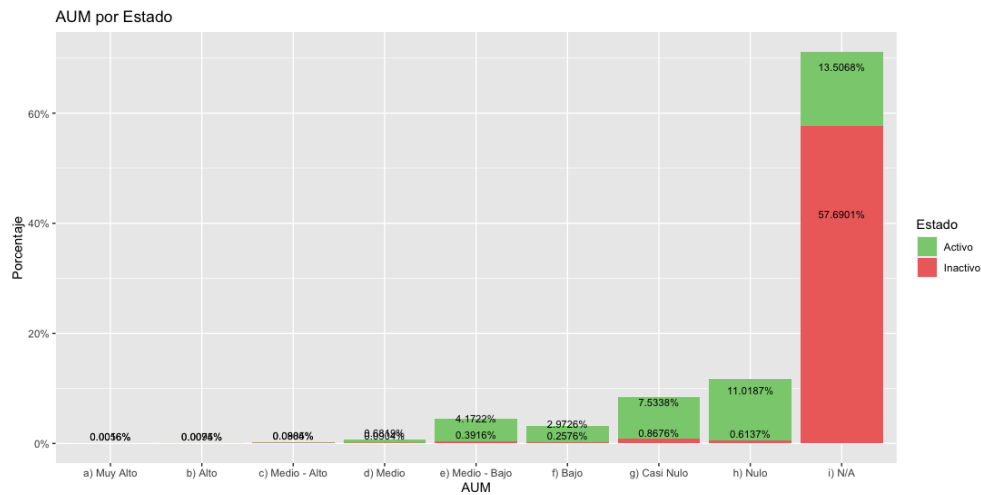


Figura 32: Gráfico De barras de AUM por estado

Por último, retomando el análisis de las provincias, esta vez con las zonas conformadas y la variable estado incluida, se observa un dato que previamente podía pasar desapercibido. La zona “Centro” posee más cantidad de usuarios que “CABA”, esto, si bien es comprensible por la cantidad de provincias que abarcan esta zona, también sucede gracias a Neuquén y el Crypto Valley que se está conformando en esa provincia, el cual genera una gran cantidad de usuarios. Cabe aclarar que los usuarios de Brasil, incluidos en la zona “Otros”, estaban inactivos al momento de descargar la base, ya que en ese momento la aplicación aún se encontraba en estado Beta cerrada.

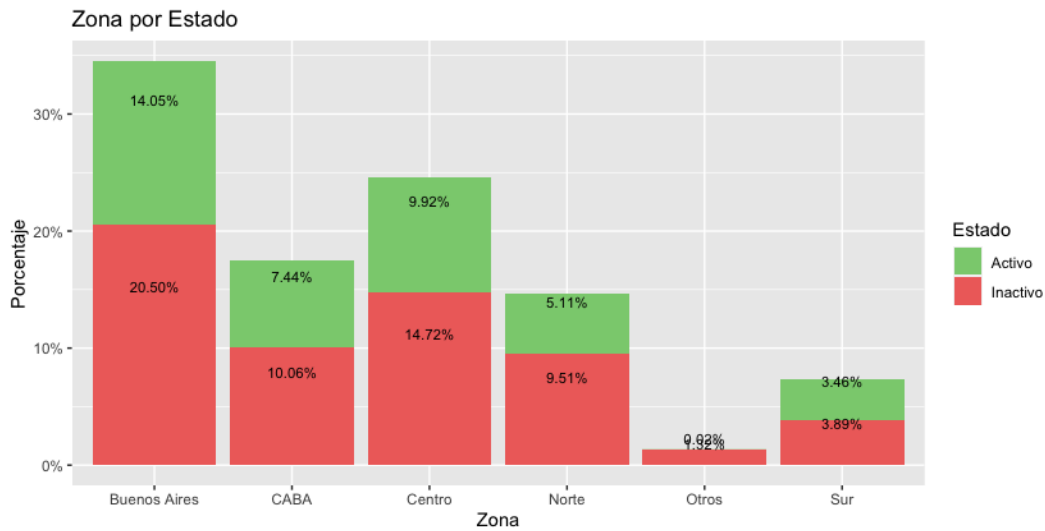


Figura 33: Gráfico de barras de zona por estado

Análisis Estadístico²

Para comenzar con el análisis estadístico de las variables y poder realizar una matriz de correlación con el fin de evaluar qué variables serán relevantes para el modelo, se decidió agrupar las variables VOLUMEN y AUM en menos categorías como fue realizado con la variable Provincia. Esto se debe a que las categorías iniciales eran muchas y al momento de realizar la matriz de correlación se encontraron dificultades para construirla por la gran cantidad de variables que tenía la base y el nivel de procesamiento que requiere RStudio para correrla.

En ambos casos se utilizó el mismo criterio de disminuir a 4 categorías, con las siguientes composiciones para el VOLUMEN:

- 4: a) Alto y b) Medio - Alto
- 3: c) Medio y d) Medio - Bajo
- 2: e) Bajo y f) Casi Nulo
- 1: g) N/A

Y las siguientes composiciones para el AUM:

- 4: a) Muy Alto y b) Alto
- 3: c) Medio - Alto, d) Medio y e) Medio - Bajo
- 2: f) Bajo, g) Casi Nulo, h) Nulo
- 1: i) N/A

²Se eliminó de la base aquellas variables que iban a presentar una correlación obvia, siendo estas Zip_Code y Localidad, explicados por la variable Provincia.



Se utilizaron números del 1 al 4 para convertir la variable en numérica en vez de categórica y así poder estudiar mejor el comportamiento en la matriz de correlaciones sin generar tantas dummies en el proceso.

A su vez, se decidió realizar un procedimiento similar para la variable NACIONALIDAD, ya que se contaba con 76 nacionalidades distintas. Es por esto que se crearon 2 grupos de acuerdo a la cantidad de usuarios que representaban para la base:

- Argentina
- Otros (que incluye a campos nulos)

Previamente, se había considerado el uso de más categorías pero para reducir la dimensión de la base y ante la diferencia de proporción entre Argentina y el resto, se optó por esta solución.

Con todo esto realizado, la base resultante es de 30 variables (incluyendo dummies) y conforma la matriz de correlación que se ve a continuación:

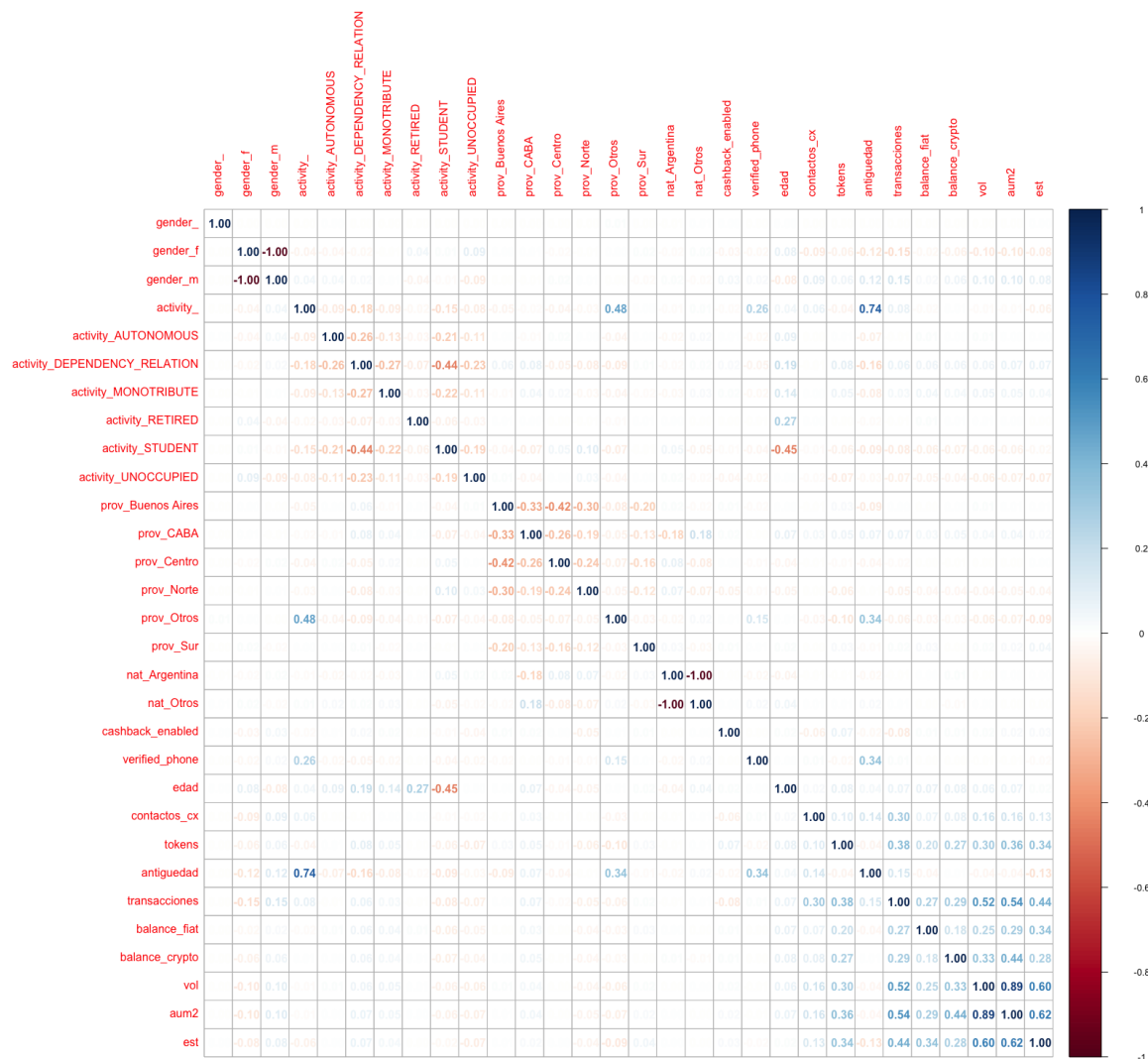


Figura 34: Matriz de correlación.



Esta compleja matriz, permite notar que no existe una correlación mayor al 60% (salvando las obvias de las dummy) excepto por el 89% que se ve entre AUM y VOLUMEN y ANTIGÜEDAD con el campo de actividad nulo (ACTIVITY_) con un 74% de correlación.

Se considera que la primera relación (AUM-VOLUMEN), por más que podría ser pertinente sacar una de ellas, se decide mantener ambas variables, ya que hablan de una tendencia similar pero no exactamente igual, del dinero guardado en la aplicación y el dinero transaccionado semanalmente a través de la aplicación.

Con respecto a la segunda relación (ANTIGÜEDAD y ACTIVITY_), también se eligió mantener ambas variables, ya que fue asociada a un cambio en los requisitos que pide **XYZ** al registrarse como usuario. Antes, no era obligatorio completar el campo actividad y hace unos meses se volvió campo obligatorio a completar.

Hay algunas relaciones como por ejemplo VOL y AUM con TRANSACCIONES, que son de esperar, ya que a mayor volumen de dinero en la wallet, se pueden realizar una mayor cantidad de transacciones. Otra relación esperable, es la de EDAD con la ocupación ACTIVITY_STUDENT, ya que a menor edad, mayor es la probabilidad de que sean personas universitarias o estudiando, con un 45% de correlación.

También se aprecian relaciones que no son tan esperables, como el caso del 48% entre la zona OTROS y ACTIVITY_ con campos nulos. Esto se debe a que la zona OTROS es una zona compuesta mayoritariamente por Brasil y campos nulos. Por lo cual, podría estar pasando que se relacionan los campos nulos de ambas variables ya que son personas que no completaron aquellos datos.

Si bien hay relaciones marcadas como se analizaron, no existe una correlación fuerte entre variables para descartar alguna de ellas porque explican lo mismo. Es por este motivo, que se decidió mantener todas las variables que se ven en la matriz de correlación para los futuros análisis y experimentación.

Hipótesis Planteada

Los usuarios pudieron ser perfilados y analizados correctamente con el EDA, por lo tanto, el modelo predictivo de Machine Learning que se adopte, podrá catalogar a un usuario como activo o inactivo para los 30 días siguientes a cuando se corra el modelo.



Caso de Negocio

Luego de haber realizado todo el estudio descriptivo de la base, se tiene la información necesaria para poder mostrar y comprender mejor el problema que se está tratando. Es por eso que el caso de negocio planteado para el modelo predictivo de **CHURN** toma de base a los **193.656 usuarios activos** al momento de descargar los datos. A una tasa de **CHURN del 28%** (dato otorgado por la empresa), se realiza el siguiente cálculo:

$$\text{Costo de Oportunidad} = \text{Tasa de CHURN} * \text{Usuarios} * (\text{LTV} - \text{CAC})$$

Realizando esta cuenta, con un **LTV de 15 USD** y un **CAC de 5,81** por usuario, el **costo de oportunidad sería de 498.315,62 USD**, perdiendo **54.224 usuarios por mes** (suponiendo una cantidad de usuarios activos que no varíe, lo mismo que decir, ingresa la misma cantidad de usuarios que churnea cada mes).

Se estima que con un **10% de retención**, a través de campañas de marketing dirigidas específicamente a estos usuarios, ya teniendo sus perfiles y comportamientos estudiados, a un **costo de 2,45 USD por usuario**, la cantidad de usuarios que dejan de usar la aplicación descendería a **48.801**, dejando un costo de oportunidad de **461.768,86 USD**. Para verlo con una mayor claridad, la propuesta es **reducir en un 2,8% a los usuarios que dejan de usar XYZ, disminuyendo las pérdidas un 2,05%**.

Por otro lado, evaluando un caso aún más ambicioso al lograr reducir el costo de retener a un usuario a **\$2 USD**, con una **retención efectiva del 15%** del **CHURN**, resultaría en una **reducción de los usuarios inactivos del 4,2% y del costo de oportunidad de 3,29%**. Lo más interesante de todo esto es que con el paso del tiempo, y manteniendo como hipótesis que los costos se mantengan constantes en el tiempo, el mayor entendimiento del cliente permitirá realizar campañas cada vez más personalizadas, por lo que la eficacia de las campañas de marketing aumentarán, logrando disminuir el churn y el costo de oportunidad aún más.

Este escenario se podría considerar una vez que la primera propuesta sea implementada y se pueda tener más certeza de la efectividad de las campañas de Producto y los conocimientos adquiridos por Marketing. De esta manera, se disminuiría la tasa de **CHURN** inicial y aumentaría el porcentaje de usuarios que utilicen **XYZ** de manera habitual, con bajos riesgos de pasar a ser inactivos.



Investigación y Consideraciones

Una vez que la información se encuentra limpia, consistente, válida y lista para ser trabajada, se empieza con la experimentación del problema. La idea de esta etapa consiste en encontrar un modelo que determine si un usuario tiene propensión hacia el Churn a partir del conocimiento de las características de aquellos que ya han hecho Churn, quedando definido como aquellos usuarios que son inactivos por no haber realizado al menos una transacción en los últimos 30 días. Para ello, se investigaron modelos de Machine Learning particularmente de aprendizaje supervisado, que sean algoritmos de clasificación, capaces de generalizar datos de usuarios nunca antes vistos y predecir su estado, con un subconjunto de testeo, a partir de un conjunto conocido, el entrenamiento.

Las técnicas de aprendizaje supervisado se seleccionaron de acuerdo a una lista de consideraciones o criterios:

- Que presenten un buen desempeño para problemas de clasificación binaria
- Que permitan el manejo de grandes bases de datos, ya que actualmente **XYZ** superó el millón de usuarios
- Sean sencillas de interpretar, por ejemplo a partir de la obtención de métricas que se puedan comparar
- Sean técnicas utilizadas anteriormente para otros modelos de predicción, ya que se pueden considerar confiables

A continuación, se presentan 3 técnicas que fueron seleccionadas por cumplir con estos criterios:

1. **Árbol de Decisión:** es una estructura tipo grafo con nodos que representan una prueba sobre un atributo. Cada rama representa el resultado del test y cada hoja es una categoría o etiqueta. Compone un conjunto de reglas que permiten llegar a una conclusión. Tienen la ventaja de presentar alta exactitud y ser buenos para encontrar relaciones no lineales que se pueden adaptar a múltiples problemas.
2. **Random Forest:** consiste en un gran número de árboles de decisión que funcionan como una estructura en la cual cada árbol arroja una predicción, y aquella que más se repita será la predicción resultante. Tiene la ventaja de que no se ve afectado por errores individuales que puede presentar algún árbol.
3. **XGBoost:** es una extensión de los árboles de decisión que busca convertir aquellos árboles que resultaron malos predictores en buenos predictores a partir de un método iterativo que aumenta el peso de las observaciones más difíciles de clasificar y crea nuevos árboles para obtener la predicción final. La ventaja que presenta es que al hacer esto en paralelo, lo hace más rápido y posee buen desempeño y alta



adaptabilidad.

Con el fin de encontrar el mejor modelo que prediga el estado de los usuarios, se debe contestar la pregunta “mejor en cuanto a qué?”. Para ello se obtienen métricas que permiten cuantificar el desempeño del modelo de clasificación binaria. Las métricas elegidas y la manera de calcularlas son:

1. **Accuracy:** Mide el ratio de predicciones correctas sobre el total de predicciones.

$$accuracy = \frac{TP+TN}{FP+FN+TP+TN}$$

2. **Precisión:** Mide la proporción de predicciones correctas sobre el total de predicciones positivas. Se asocia a la calidad de la predicción.

$$precision = \frac{TP}{TP + FP}$$

3. **Sensibilidad (o Recall):** La sensibilidad, tasa de aciertos o tasa positiva real (TPR), es la proporción de la cantidad total de instancias pertinentes que se recuperaron realmente. Mide qué proporción de positivos reales se identificó correctamente.

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

4. **F1:** El puntaje F1 es una medida de la precisión de una prueba, es la media armónica de precisión y recuperación. Puede tener una puntuación máxima de 1 (precisión y recuerdo perfectos) y una mínima de 0. En general, es una medida de la precisión y robustez del modelo.

$$F1 = \frac{2*TP}{2*TP + FN + FP}$$

Siendo:

TP: True positive o verdaderos positivos.

TN: True negative o verdaderos negativos.

FP: False positives o falsos positivos. (Error de tipo I)

FN: False negatives o falsos negativos. (Error de tipo II)

		Actual Values	
		Yes	No
Predicted Values	Yes	True Positive	False Positive
	No	False Negative	True Negative

Figura 35: Matriz de confusión.



Desarrollo de modelos

Para mostrar el detalle del desarrollo de los modelos, se procede a comentar el código y exponer algunas decisiones que se fueron tomando para cada uno de los modelos.

En primer lugar, se importaron las librerías y se particionó el dataset en 2. Por un lado, el conjunto de testeo compuesto por el 30% de los datos de la base original, y por otro lado el conjunto de entrenamiento con el 70% restante. No se determinó una seed en particular para que cada corrida sea completamente aleatoria.

```
df.dropna()  
dftrain, dftest = train_test_split(df, test_size = 0.30)
```

Durante este proceso, se decidió dejar de lado las variables “cantidad de transacciones” ya que representaba un valor muy dependiente de la antigüedad del usuario, en el que se acumulaba el valor y terminaba explicando el modelo en relación a este número, por otro lado, las variables “balance_fiat” y “balance_crypto” también se eliminaron ya que se concluyó que, con la base actual siendo una foto estática y sin la posibilidad de ir variando mes a mes, explicaban nuevamente todo el modelo (una vez quitadas las transacciones), siendo un usuario con dinero en la aplicación activo y sin dinero inactivo.

El conjunto de entrenamiento fue llamado “x” excluyendo a la variable a predecir, que se encuentra en el dataset “y”. El conjunto de testeo fue llamado “x_test” y por otro lado, se creó “y_test” que incluye únicamente a la variable a predecir.

```
x=dftrain.drop(["est","balance_fiat","balance_crypto","transacciones"], axis=1)  
y=dftrain["est"]  
x_test=dftest.drop(["est","balance_fiat","balance_crypto","transacciones"], axis=1)  
y_test=dftest["est"]
```

El desarrollo de los tres modelos (árbol de decisión, XGBoost y Random Forest) fue realizado siguiendo la misma metodología. Primero, se crearon los modelos modelo_t para el árbol de decisión, modelo_xgb para XGBoost y modelo_rf para Random Forest, con parámetros aleatorios, como se visualiza en los siguientes códigos respectivamente:

```
modelo_t = DecisionTreeClassifier(max_depth=5,max_features=None)  
modelo_t.fit(x, y)
```

```
modelo_xgb = xgb.XGBClassifier(objective = "binary:logistic", random_state=123, n_estimators = 100, learning_rate = 0.1,subsample=0.5)  
modelo_xgb.fit(x,y)
```

```
modelo_rf = RandomForestClassifier(n_estimators = 1000, max_depth = None, max_features = 0.3, oob_score = False, n_jobs = -1, random_state = 123)  
modelo_rf.fit(x, y)
```

Luego, se calcularon algunas métricas como el Score utilizando cross validation de la siguiente manera:



```
kf = KFold(n_splits=10)
score = modelo_t.score(x,y)
print("Métrica del modelo", score)
scores = cross_val_score(modelo_t, x, y, cv=kf, scoring="accuracy")
print("Métricas cross_validation", scores)
print("Media de cross_validation", scores.mean())
preds = modelo_t.predict(x_test)
score_pred = metrics.accuracy_score(y_test, preds)
print("Métrica en Test", score_pred)
```

Estas dan un primer indicio del funcionamiento y precisión de cada modelo. Para que no sea tan extenso el código que se muestra, se procede a mostrar únicamente los pasos realizados con el Árbol de Decisión. Sin embargo, como ya se explicó anteriormente, este proceso fue llevado a cabo para los otros dos modelos de manera análoga.

Para evaluar el impacto de cada variable sobre el modelo, se calculó el “feature importance” que devuelve un número de 0 a 1 con el peso de importancia que cada variable tiene sobre los modelos. Es decir, da un porcentaje de qué tanto impacta la variable en la toma de decisión para la predicción del estado del usuario.

```
# get importance
importance = modelo_t.feature_importances_
# summarize feature importance
for i,v in enumerate(importance):
    print('Feature: %0d, Score: %.5f' % (i,v))
# plot feature importance
plt.bar([x for x in range(len(importance))], importance)
plt.show()
```

Además, para que los modelos sean lo más completos y eficientes posible, se creó la matriz de hiperparámetros que devuelve valores óptimos que deben asignarse a los parámetros de cada modelo con el fin de determinar con mayor precisión el estado de cada usuario.

```
param_grid_tc = {
    'max_depth': [3,5,10,None],
    'max_features': [0.1,0.3,0.5,None],
    'min_samples_split': [2,5,10],
    'min_samples_leaf': [5,10,15]
}
```

Los resultados de esta matriz fueron visualizados a partir del siguiente código:

```
resultados = pd.DataFrame(grid_tc.cv_results_)
resultados.filter(regex = '(param*|mean_t|std_t)') \
    .drop(columns = 'params') \
    .sort_values('mean_test_score', ascending = False) \
    .head(4)
```

También se creó un array con los hiperparámetros que mejor se ajustaban al modelo,



que servirá de base para la construcción de cada uno de los modelos definitivos.

```
print(grid_tc.best_params_, ":", grid_tc.best_score_, grid_tc.scoring)

# Construyo el modelo y ajusto los datos.
modelo_final = grid_tc.best_estimator_
# Realizo las predicciones
y_pred = modelo_final.predict(x)
predicciones = [round(value) for value in y_pred]
# Evalúo las predicciones
precision_train = accuracy_score(y, predicciones)
# Repito el proceso con datos de evaluacion
y_pred = modelo_final.predict(x_test)
predicciones = [round(value) for value in y_pred]
# Evalúo las predicciones
precision_test = accuracy_score(y_test, predicciones)
print(modelo_final)
print('Precisión Decision Tree Classifier train/test {0:.5f}/{1:.5f}'
      .format(precision_train, precision_test))
```

Por último, se calcularon las métricas para cada modelo. El análisis de los mismos se encuentra en el apartado siguiente.

```
kf = KFold(n_splits=10)
score = modelo_final.score(x,y)
print("Métrica del modelo", score)
scores = cross_val_score(modelo_final, x, y, cv=kf, scoring="accuracy")
print("Métricas cross_validation", scores)
print("Media de cross_validation", scores.mean())
preds = modelo_final.predict(x_test)
score_pred = metrics.accuracy_score(y_test, preds)
print("Métrica en Test", score_pred)
```

```
y_pred = modelo_t.predict(x_test)
x_pred = modelo_t.predict(x)
# Confusion Matrix
confusion_matrix(y_test, y_pred)
# Accuracy
accuracy_score(y_test, y_pred)
# Recall
r1 = recall_score(y_test, y_pred)
r2 = recall_score(y, x_pred)
print(r1, r2)
# Precision
p1 = precision_score(y_test, y_pred)
p2 = precision_score(y, x_pred)
print(p1)
print(p2)
# F1
f1 = f1_score(y_test, y_pred)
f2 = f1_score(y, x_pred)
print(f1)
print(f2)
```



Análisis y Comparación de resultados

La matriz de hiper parámetros, devolvió la recomendación de utilizar los siguientes valores para cada modelo, dentro de las posibles combinaciones que se pueden observar en las fotos debajo de la ecuación.:

1. Árbol de decisión:

{'max_depth': 10, 'max_features': None, 'min_samples_leaf': 15, 'min_samples_split': 10}

```
param_grid_tc = {  
    'max_depth': [3,5,10,None],  
    'max_features': [0.1,0.3,0.5,None],  
    'min_samples_split': [2,5,10],  
    'min_samples_leaf': [5,10,15]  
}
```

2. XGBoost:

{'learning_rate': 0.1, 'max_depth': 5, 'n_estimators': 100, 'subsample': 1}

```
param_grid = {'n_estimators' : [10,50,100,500,1000],  
    'max_depth' : [None,3,5],  
    'subsample' : [0.3,0.5,1],  
    'learning_rate' : [0.5,0.1,0.01]  
}
```

3. Random Forest:

{'criterion': Gini, 'n_estimators': 50, 'max_depth': 10, 'max_features': 0.3,
'min_samples_leaf': 10, 'min_samples_split': 15}

```
param_grid_rf = {  
    'max_depth': [3,5,10,None],  
    'max_features': [0.1,0.3,0.5,None],  
    'n_estimators': [10,50,100,500,1000],  
    'min_samples_split': [2,5,10],  
    'min_samples_leaf': [5,10,15],  
    'criterion': ['gini', 'entropy'],  
}
```

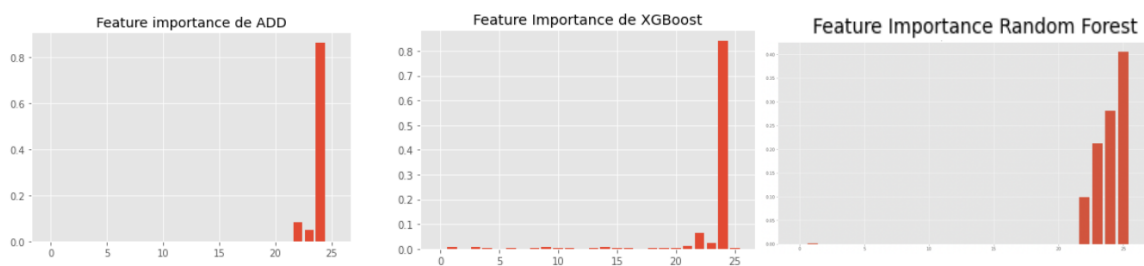
Luego, al realizar los feature importance, se encontraron los siguientes resultados para cada uno de los modelos:

#	Variable	ADD	XGBoost	Random Forest
1	Genero Nulo	0	0	0
2	Genero F	0,036	0,713	0,348
3	Genero M	0	0	0,35
4	Activity Nulo	0	0,677	0,102
5	Activity Autonomous	0	0,148	0,41
6	Activity Dependency Relation	0	0	0,49



7	Activity Monotribute	0	0,156	0,377
8	Activity Retired	0	0	0,088
9	Activity Student	0	0,29	0,367
10	Activity Unoccupied	0	0,778	0,265
11	Zona Buenos Aires	0	0,29	0,57
12	Zona CABA	0	0,253	0,496
13	Zona Centro	0	0	0,55
14	Zona Norte	0	0,204	0,429
15	Zona Otros	0,024	0,653	0,13
16	Zona Sur	0	0,515	0,328
17	Nationality Argentina	0	0,219	0,214
18	Nationality Otros	0	0	0,215
19	Cashback Enabled	0	0,246	0,094
20	Verified Phone	0	0,136	0,035
21	Edad	0,036	0,391	14,814
22	Contactos a CX	0,111	0,992	2,307
23	Tokens	8,454	6,227	10,696
24	Antigüedad	4,755	2,478	21,911
25	Volumen	86,504	84,209	22,956
26	AUM	0,08	0,426	21,46

Figura 36: Resultados de Feature Importance



Figuras 37, 38 y 39: Gráficos de Feature Importance

Evaluando los resultados provistos por los Feature Importance de cada modelo, se decidieron transformar algunas de las variables y quitar aquellas que no son de comportamiento del usuario sino de perfilamiento. Esto se debe a que no presentan importancia para ninguno de los modelos y por ende, podrían estar ensuciando el proceso de clasificación. Es por esto, que se decidió realizar algunas transformaciones de variables para seguir manteniendo los datos, pero con un mayor aporte de información.

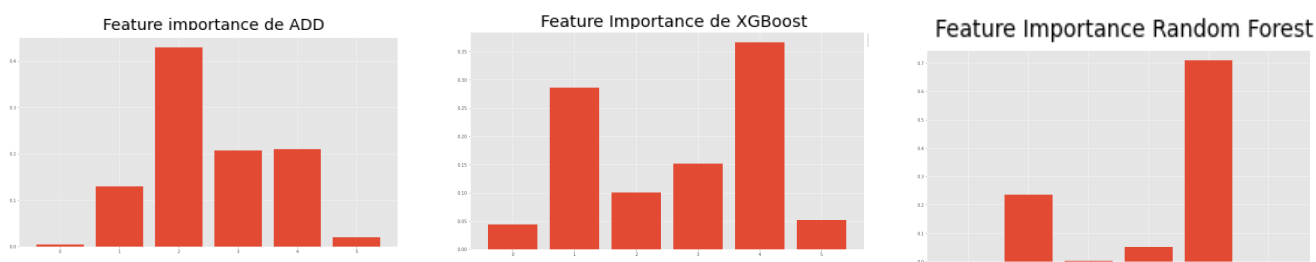


Los cambios propuestos fueron realizando un promedio de TRANSACCIONES, AUM, VOLUMEN Y CONTACTOS_CX por mes. De esta manera, se evalúa el comportamiento de cada usuario en un tiempo determinado, y no de una manera general o histórica. Además, se dejaron de lado las variables de perfilamiento a excepción del género masculino, que es una variable booleana 1 cuando es masculino y 0 cuando se trate de un usuario femenino.

Entonces, las variables que quedaron luego de las transformaciones y última selección son las siguientes:

- Género Masculino
- Volumen por mes
- AUM por mes
- Tokens
- Transacciones por mes
- Contactos a CX por mes

A continuación se muestran los resultados para cada uno de los modelos:



Figuras 40, 41 y 42: Nuevos Gráficos de Feature Importance

Métricas Test	ADD	XGBoost	Random Forest
Accuracy	84,50%	83,52%	82,08%
Recall	77,51%	72,79%	77,20%
Precision	86,39%	83,97%	81,67%
F1	81,70%	77,98%	79,37%

Figura 43: Scores del conjunto Test

Observando cada uno de los modelos y los resultados obtenidos, se puede decir que en términos generales, los modelos que mejor performan son el Árbol de Decisión y XGBoost por las métricas y los gráficos. Queda evidenciado que el modelo de Random Forest no presenta tanta importancia en sus variables como los otros dos modelos. Con un mayor detalle, se ve que si bien las métricas del Árbol de Decisión resultaron ser las más altas para cada parámetro evaluado, las variables que más importancia tienen entre los tres modelos es el de XGBoost.

Sumado a esto, se sabe que teóricamente el modelo de XGBoost es el que mejor se adapta para casos de clasificación de CHURN de usuarios, y la métrica F1, es una de las que



mejor se adapta a estos modelos, por presentar clases desbalanceadas. Es por esto, que se decide continuar con el modelo de XGBoost como modelo final, ya que tanto F1 como el Recall (Sensibilidad), para controlar los falsos positivos que tanto importan a la hora de clasificar usuarios, sobrepasan el 70% y por ende, aceptables para su selección.

Es a raíz de esto, que se puede concluir que este modelo cumple con predecir de una manera efectiva los estados de los usuarios y es un buen modelo para elegir como final. Con estos parámetros luego se podrá construir el XGBoost que se extrapolará con los datos actuales de **XYZ** para saber realmente qué usuarios tienen tendencia a ser inactivos.

Conclusiones

En primer lugar, cabe decir que se ha podido abordar el problema correctamente. Esto se debe a que se llegó a la solución de determinar un modelo de predicción que le servirá a la empresa a enfocarse en aquellos usuarios que pasarán a ser inactivos.

Por otro lado, se pudo monetizar el proyecto, obteniendo un costo de oportunidad aproximado, que le brindará beneficios a **XYZ** no solo a un nivel económico, sino también en una mejor retención de usuarios y de disponibilidad de información para actuar y tomar decisiones que estén 100% basadas en datos.

Además, se evaluaron distintas alternativas que podrían serles útil en un futuro para determinar el algoritmo que mejor les convenga. En este caso, fue seleccionado el modelo de XGBoost por los motivos ya explicados, pero quedan presentadas las métricas de los otros modelos si es que quisieran explorar otras alternativas o bien, utilizar más de una.

Asimismo, al realizar este proyecto, se fueron presentando algunos inconvenientes que se pudieron resolver, generando un mayor conocimiento para el equipo. Por ejemplo, se descubrieron inconsistencias en la base y en algunos datos, entonces se decidió estandarizar variables como Provincia y Actividad; se generaron variables compuestas que permitieron un mejor análisis de los datos; se fue viendo como iban modificándose las métricas tanto del conjunto de entrenamiento como del de testeo; entre otros. Todos y cada uno de estos problemas, no terminaron siendo problemas realmente, sino más bien desafíos que tuvo que enfrentar el equipo y, se puede concluir, que la realización del proyecto ayudó considerablemente a un mejor entendimiento de modelos de predicción y su implementación en el mundo real.

Próximos Pasos

En cuanto a los próximos pasos del proyecto, se identificaron 2 grandes verticales sobre las cuales trabajar. Por un lado, el desarrollo de un perfilamiento de los usuarios, no solo para entender quién es el usuario que es activo, el que va a churnear y demás, sino un perfilamiento de “quien es el usuario que no sirve”, incluyendo en esta consideración a los “Rat users”, “Zombie Users” y “No Kyc Users”. Este análisis genera una mejor comprensión en el equipo de marketing para generar campañas más eficientes y cost effective a la hora de



retener usuarios. Además, al tener un perfil de aquellos usuarios que transaccionan seguido y generan profit para la empresa, no se detendrá en intentar adquirir usuarios que no generarán revenue a la empresa. Este primer escenario consiste entonces de las siguientes etapas:

- Implementación mensual: correr el código del modelo seleccionado para evaluar el CHURN del mes siguiente.
- Perfilamiento específico: evaluando las características de los usuarios a los cuales se quiere retener.
- Campañas de retención: realizadas de manera interdisciplinaria en conjunto con el equipo de Marketing.

La otra vertical a atacar se basa en una mejora del modelo. Al ser este trabajo desarrollado en un ámbito educativo, siendo los datos publicables, no se tuvo acceso a todos los datos y variables que hubiese sido ideal trabajar. Se propone poder contar con datos de comportamiento de usuarios mes a mes en vez de una foto sacada en el día de extracción de los datos. Se cree que así se podrá construir un mejor modelo, con un mejor funcionamiento mes a mes. Este segundo escenario consta entonces de una sola etapa que será la de optimización de campañas, para en un futuro utópico, contar con un 100% de retención.

Bibliografía

Capgemini E, World Banking Report (2019)

<https://www.capgemini.com/es-es/wp-content/uploads/sites/16/2019/10/World-Retail-Banking-Report-2019.pdf> (Consultado: 24/03/2022)

Martinez Heras, Jose. *Precision, Recall, F1, Accuracy en clasificación*. (2020, Octubre)

<https://www.iartificial.net/precision-recall-f1-accuracy-en-clasificacion/> (Consultado: 10/05/2022)

Shin, Terence. *Comprensión de la Matriz de Confusión y Cómo Implementarla en Python* (2020, Mayo)

<https://www.datasource.ai/es/data-science-articles/comprension-de-la-matriz-de-confusion-y-como-implementarla-en-python> (Consultado: 13/05/2022)

Soporte Visual

Presentación PREZI <https://prezi.com/view/0RzjcXKBoGIhWK1mcNC/>

Algoritmo de predicción Churn de usuarios utilizando Machine Learning

Binello Matias
Ferrari Aguilera Rocio

ITBA



PROYECTO

**BUSINESS
CASE**

DATOS

**PRÓXIMOS
PASOS**

PROYECTO

¿En qué
consiste?

¿Qué es
CHURN?

¿Cuál es el
alcance?

¿Por qué
considerarlo?

BUSINESS CASE

193.656

Usuarios
Activos

28%

CHURN actual

54.224

Usuarios que se van a ir
el mes que viene

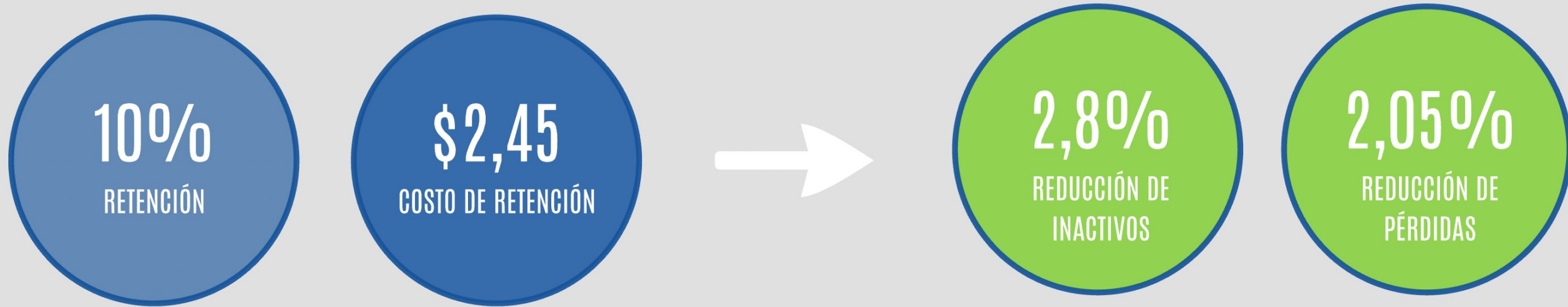
ARPU = 15 USD

Costo de Adquisición = 5,81 USD

Costo de Oportunidad = 498.315 USD

CO = USUARIOS INACTIVOS * (ARPU-CAC)

PRIMER ESCENARIO / CORTO PLAZO



\$36.546,76 USD
REDUCCION DE PERDIDAS

SEGUNDO ESCENARIO / LARGO PLAZO

15%
RETENCIÓN

\$2
DE RETENCIÓN



4,2%
REDUCCIÓN DE
INACTIVOS

3,29%
REDUCCIÓN DE
PÉRDIDAS

\$58.428,24 USD
REDUCCIÓN DE PÉRDIDAS

DATOS

A diagram illustrating the relationship between data and analysis. A large light gray circle with a thick white border is centered on the left. To its right, three smaller blue circles are arranged vertically. The top blue circle contains the text 'MODELO', the middle one contains 'OTRAS VARIABLES', and the bottom one contains 'EDA'.

MODELO

**OTRAS
VARIABLES**

EDA

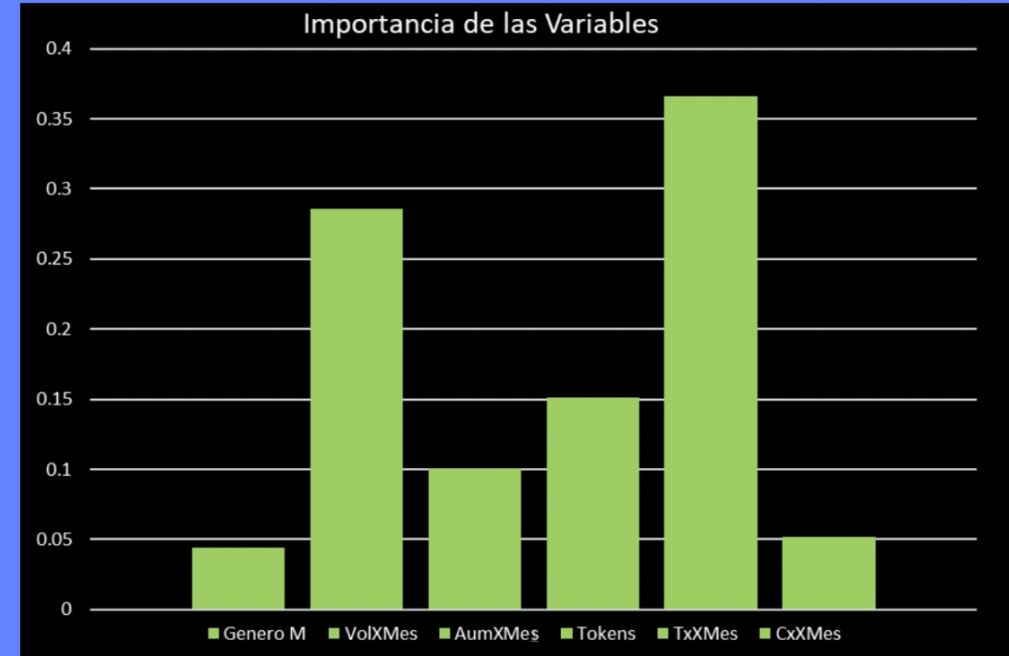
MODELO XGBOOST

'learning_rate': 0.5 - 'max_depth': 5 - 'n_estimators': 500 - 'subsample': 1

Variables

- Género masculino
- Volumen por mes
- AUM por mes
- Tokens
- Transacciones por mes
- Contactos a CX por mes

Feature Importance



Métricas Test

77,98%
F1

72,79%
Sensibilidad

83,52%
Accuracy

83,97%
Precision



Prezi

OTRAS VARIABLES ESTUDIADAS

Localidad

*Cashback
Habilitado*

Nacionalidad

Edad

*Estado
Civil*

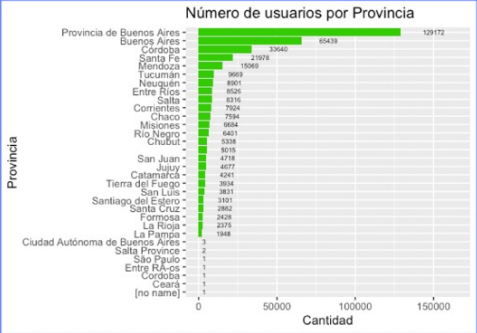
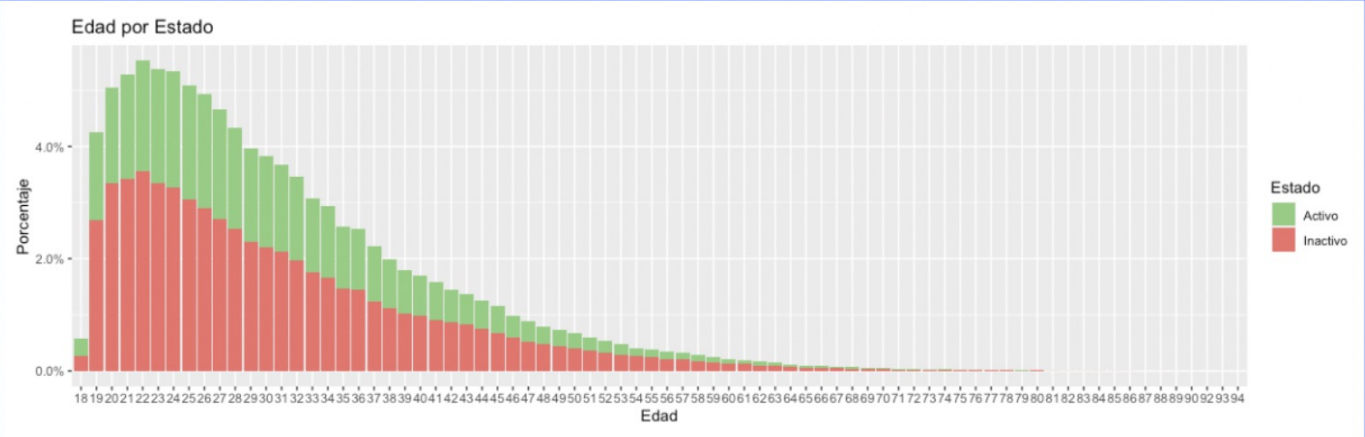
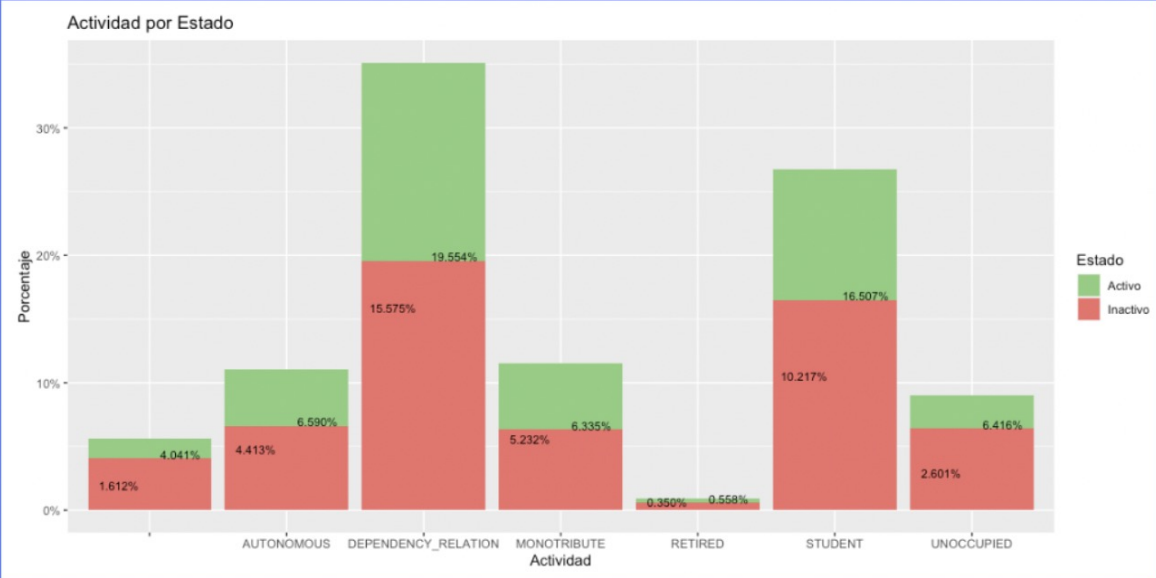
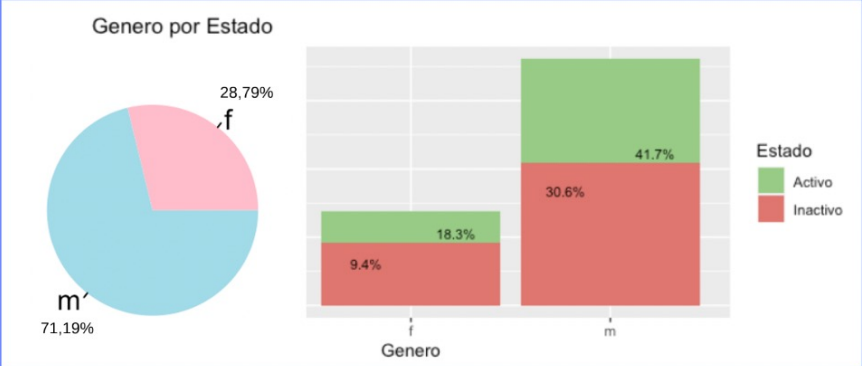
Antigüedad

Provincia

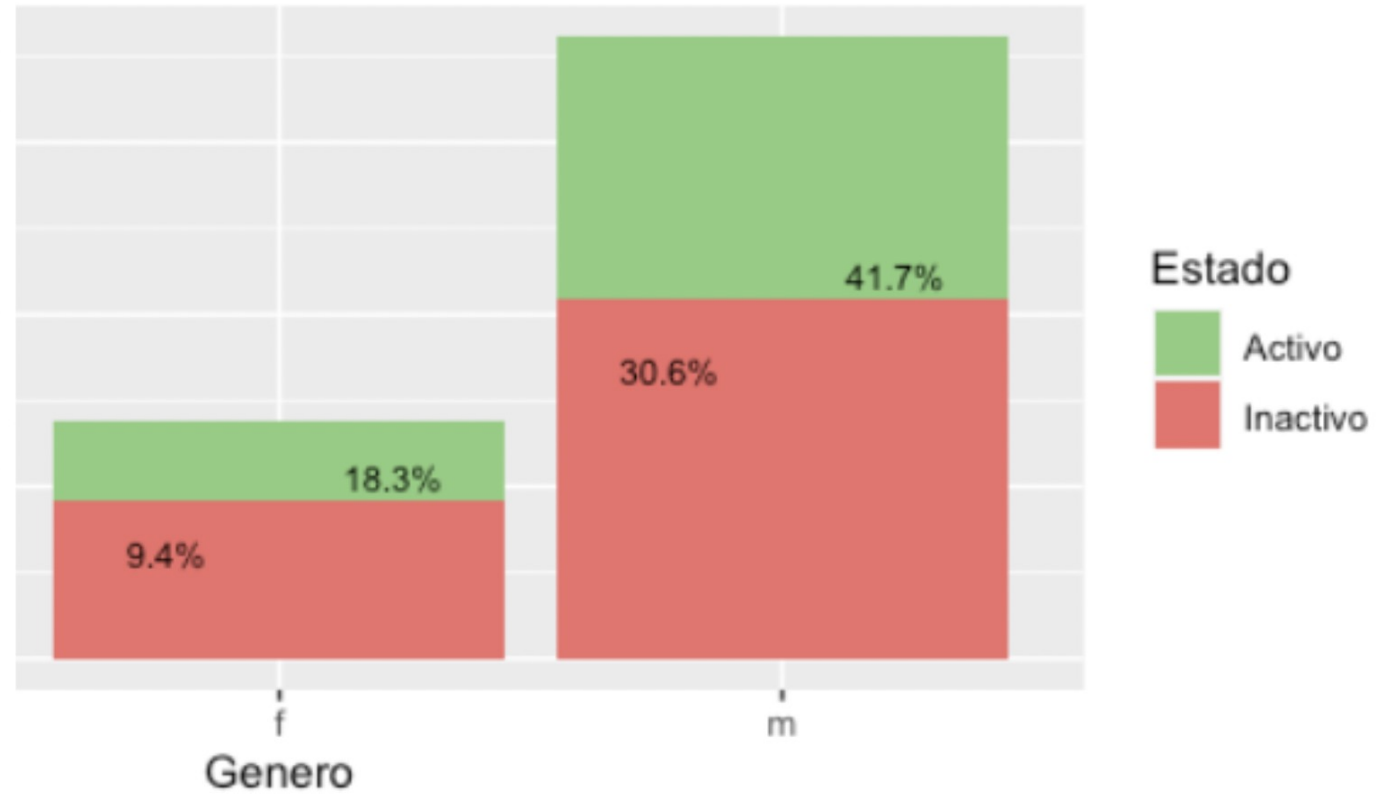
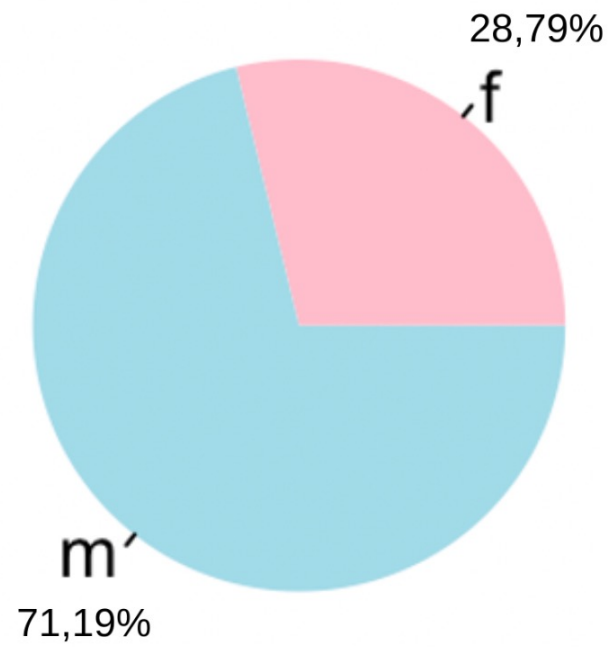
*Teléfono
Verificado*

Actividad

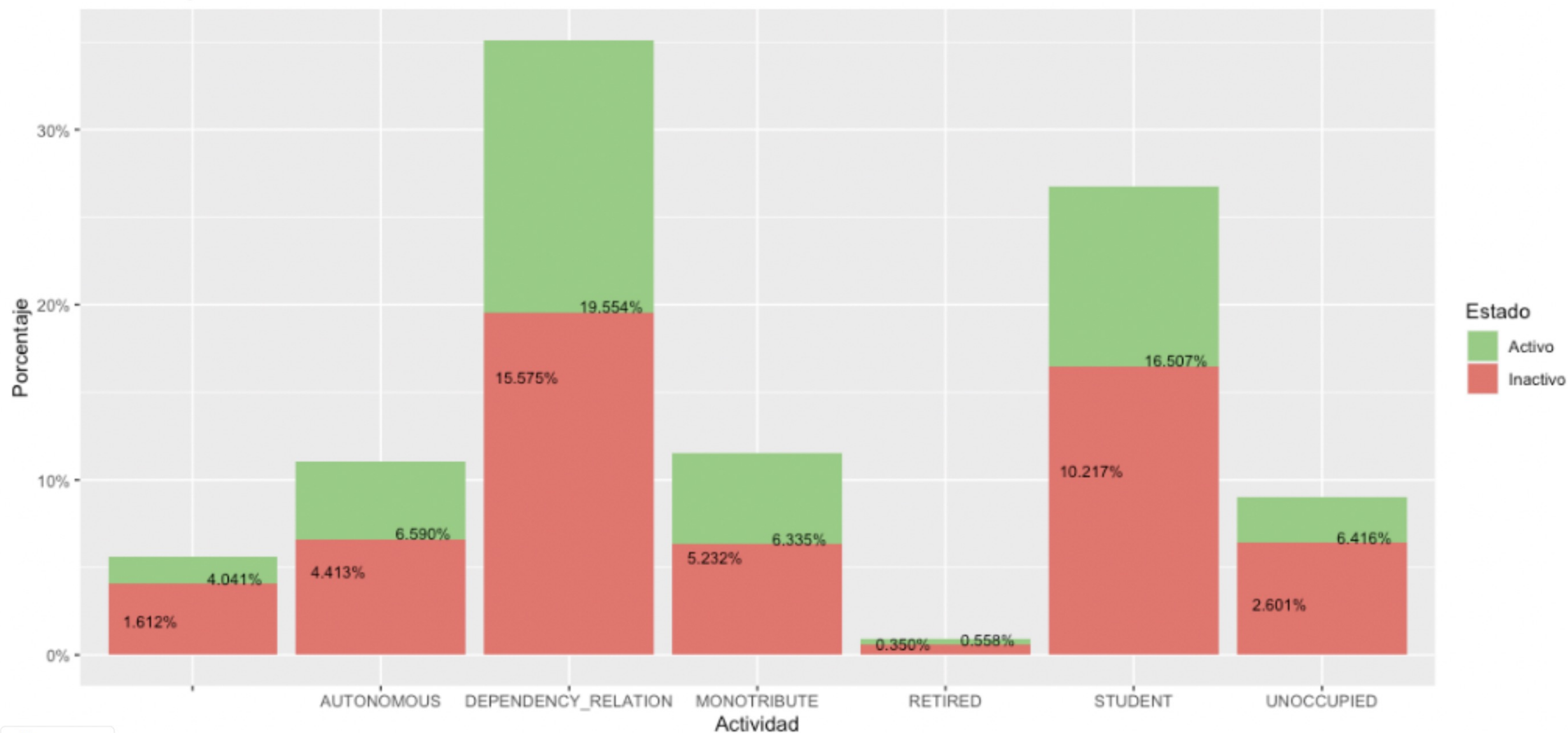
ANÁLISIS EXPLORATORIO



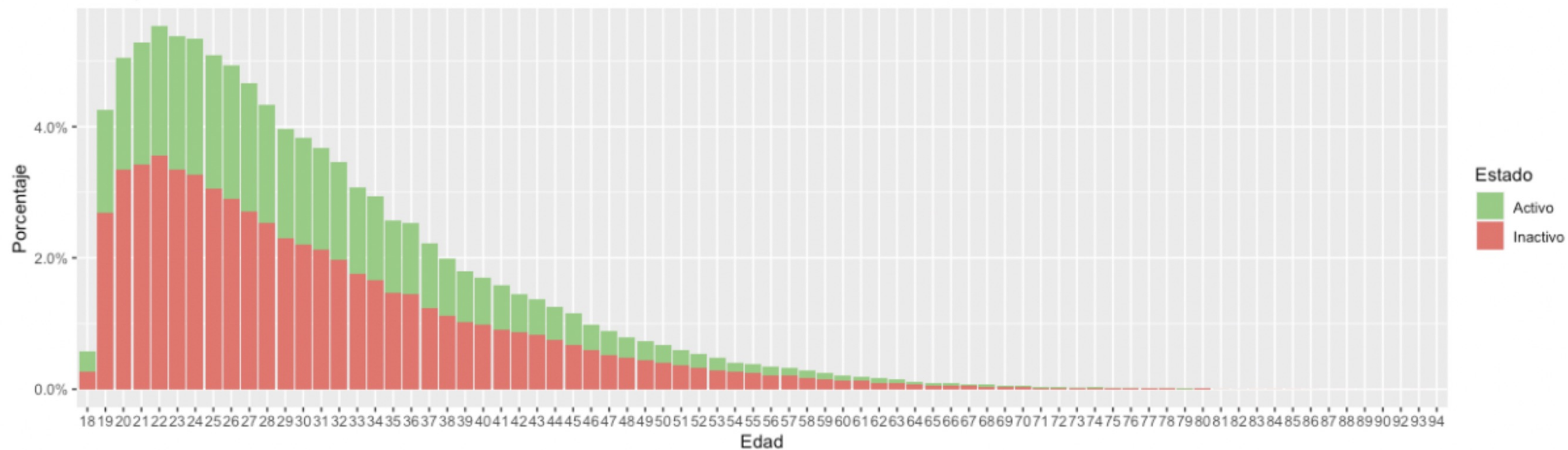
Genero por Estado



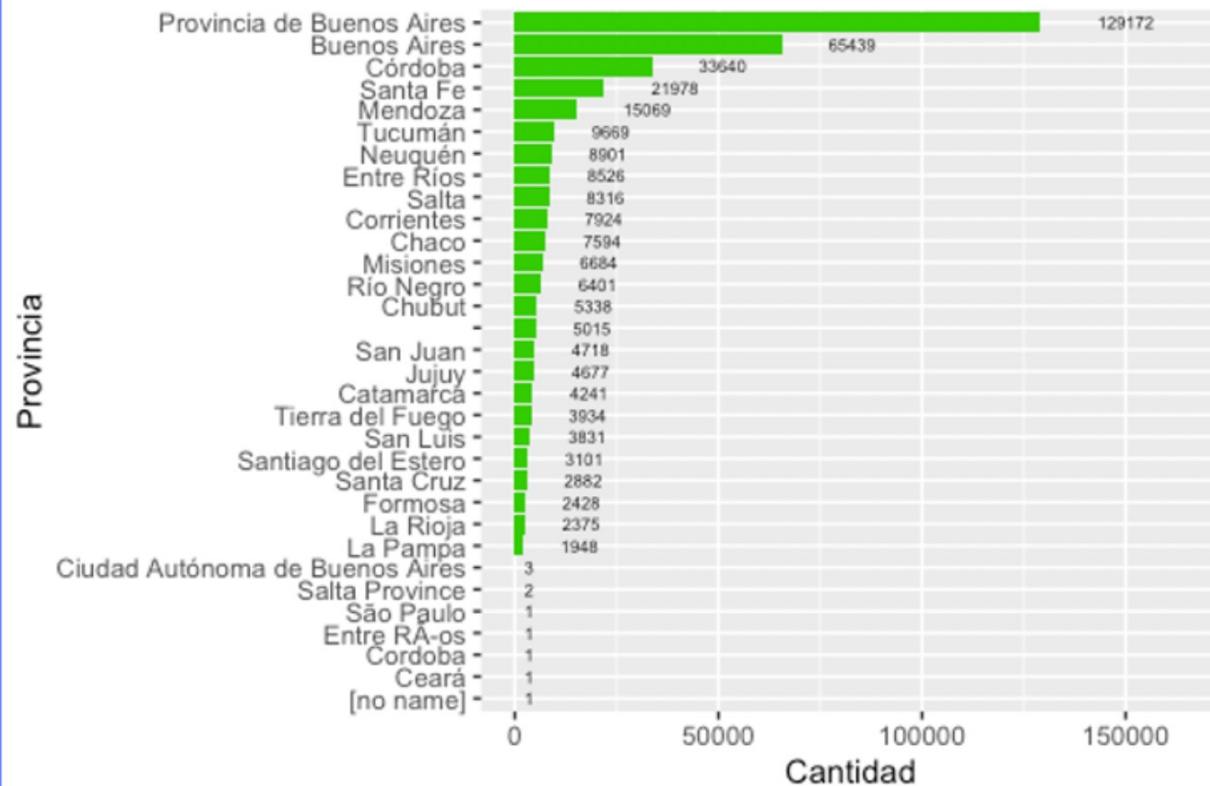
Actividad por Estado



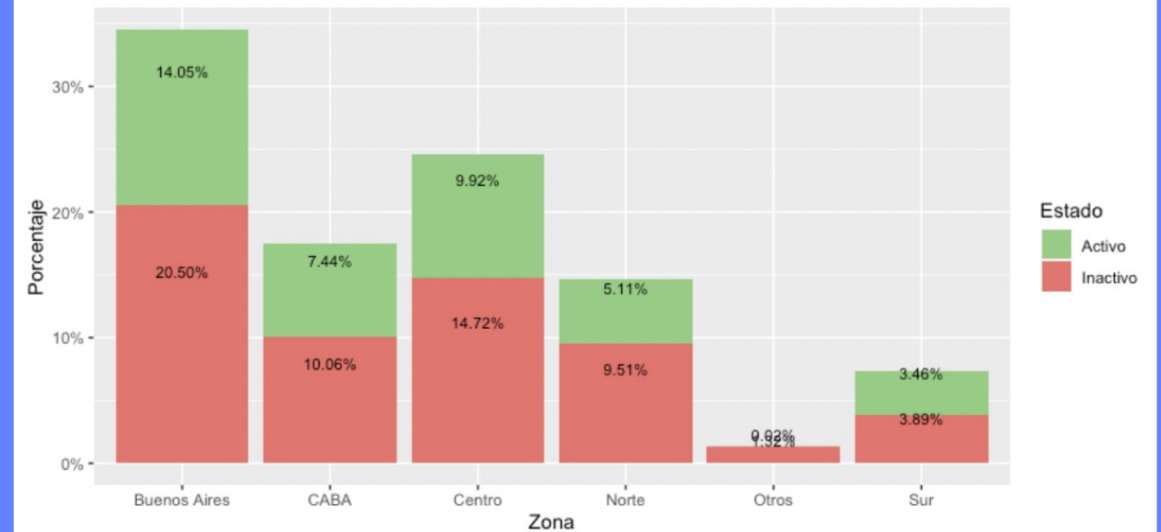
Edad por Estado



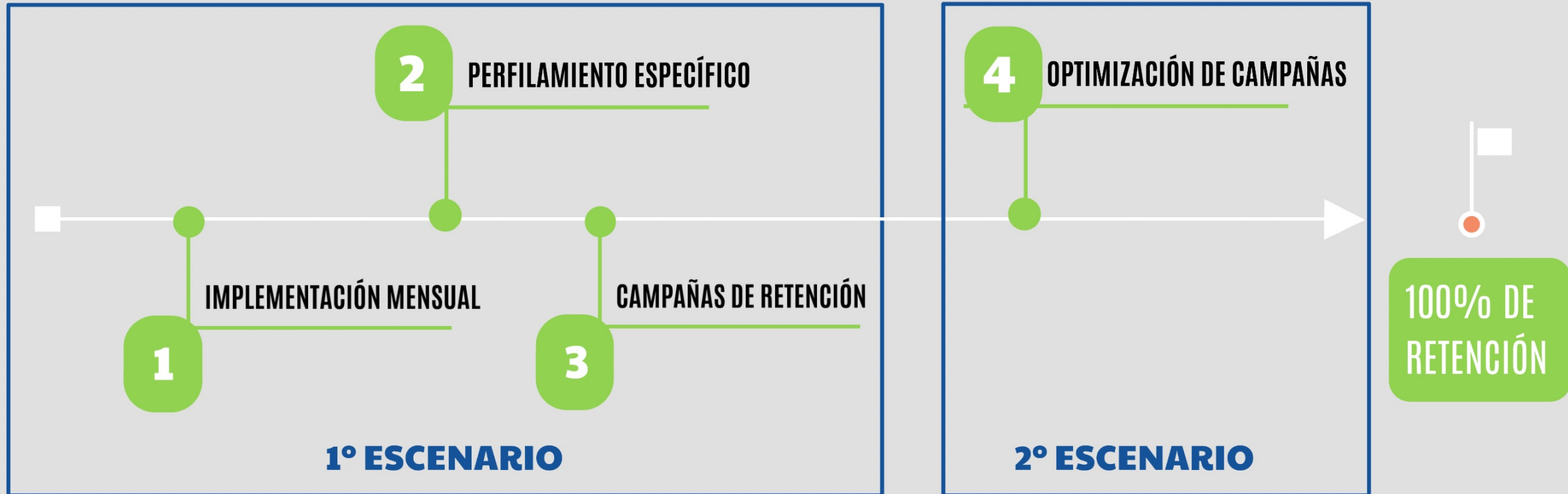
Número de usuarios por Provincia



Zona por Estado



PRÓXIMOS PASOS



¡Muchas gracias!

¿Preguntas?