CoderHouse

Modelo de predicción de Churn

Curso de Data Science

Abraham Ramírez Rodríguez
ING. EN MATEMÁTICAS

Tabla de contenido

1. Presentación del caso

1.1 Objetivo:

Construir un **modelo de Machine Learning** del tipo **"supervisado"** y de clasificación que permita definir aquellos clientes con mayor probabilidad de **"Churn"**, es decir, que estime cuales son los clientes con mayor probabilidad de **no realizar una compra** en un periodo particular de 3 meses.

1.2 Contexto comercial:

El **ecommerce** ha crecido a pasos agigantados dentro del territorio Mexicano a nivel nacional. Actualmente el negocio de comercio electrónico, de acuerdo con la asociación mexicana de venta online (AMVO) ha generado ingresos durante 2022 de hasta **$528 Billones de pesos (US $30.9 billones)**, el cual, representa un crecimiento contra **2021 de 23%**. También, México se posiciona entre el top 5 de países de Latinoamérica con mayor crecimiento de Ecommerce.

Se estimaron alrededor de 63 millones de personas que realizaron una compra a través de un sitio de **Ecommerce** durante **2022**. Lo comentado anteriormente refleja la importancia que el mercado del retail online empieza a tomar para los habitantes mexicanos, donde dicha propuesta de valor extiende la posibilidad de poder ofrecer productos a todos los habitantes del país a tan solo un clic, la posibilidad de comparar precios, tomar mejores decisiones de compra y mejorar la calidad de vida de los usuarios.

Dado este crecimiento de dicha industria, cada vez tenemos más competidores adoptando este modelo de negocio y así poder ganar participación de mercado.

La empresa con la cual trabajamos tuvo un **crecimiento** importante durante la pandemia, beneficiándose en mayor adquisición de ingresos y clientes nuevos. Este crecimiento se ha visto sostenible durante los años posteriores al 2020, donde ahora, la **estrategia comercial** a adoptar es la **retención de nuestros clientes nuevos adquiridos**, es decir, lograr su **fidelización** y mantenerlos en nuestro negocio en el **top of mind de engagement** de **compra** y el **path to profitability**. Con ello, adoptar **estrategias claves de retención de clientes** y mantener una mayor frecuencia de compra a través de clientes no nuevos es importante para identificar el **loyalty** que nuestros clientes tienen sobre la marca. Así, con el **modelo de predicción de "Churn"** se pretende identificar aquellos clientes con riesgo de **abandono de compra,**

establecer estrategias en conjunto con el equipo de CRM de retención de dichos clientes y mantener niveles óptimos del índice **CAC (Customer Acquisition Cost)** ya que atraer clientes nuevos suele tener un mayor costo que establecer estrategias de retención sobre aquellos clientes que ya nos habían realizado una compra en el pasado.

1.3 Contexto analítico:

En este caso trabajamos para una de las empresas de comercio electrónico más importantes del país enfocado principalmente en el negocio de **mercancías generales** (**e.g.: Televisores, Línea Blanca, Electrodomésticos, Muebles de cocina, celulares, cómputo, patio y jardín, etc.**). Los productos de índole **"Groceries"** o de **"Primera necesidad"** se venden a través de otra unidad/departamento de la empresa negocio.

Así, dicha empresa nos ha contratado solicitando un **modelo de predicción de "Churn"** sobre su base de clientes para establecer estrategias de retención / fidelización a través del equipo de CRM y así evitar el **abandono de compra** y lograr un aumento en el **loyalty** de los mismos.

1.4 Estructura del dataset de trabajo:

La empresa proporcionó un conjunto de información que contiene todos los pedidos realizar por los clientes a diferentes niveles de granularidad como el tipo de pago con el cual se realizó un pedido, el canal de compra mediante el cual se realizó la compra, los códigos identificadores de los artículos comprados, las piezas adquiridas, el código identificador el cliente, etc.

Cabe recordar que estos pedidos realizados corresponden solamente a la unidad de negocio de Ecommerce de productos de mercancías generales en México.

Dicho dataset tiene la siguiente estructura:

Característica	Valor
Columnas / Variables	98
Registros	16.6 M
Peso en memoria	+12.2 GB
Rango de información	2021 - 2023

Tabla 1 Descripción básica del dataset

A continuación, mostramos la estructura básica del dataset a través del método “info” de pandas sobre el dataframe que contiene dicho conjunto de información:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 16692886 entries, 0 to 16692885
Data columns (total 98 columns):
 #   Column                                Dtype
---  ------                                -----
 0   Unnamed: 0                            int64
 1   Almacen de Origen                     object
 2   Almacen detalle                       object
 3   banco                                 object
 4   Bandera bundle final                  object
 5   Bandera CBT                           object
 6   Bandera pagos                         object
 7   canal                                 object
 8   Canal_formato_SO                      object
 9   cantidad                              int64
 10  Carrier                               object
 11  CATEGORIA                             object
 12  Categoria ecomm                       object
 13  CLICK AND COLLECT                     object
 14  codigo postal                         float64
 15  Comision MKP sin IVA                  float64
 16  Comision MSI                          float64
 17  COSTO CALCULADO                       float64
 18  cupon utilizado                       object
 19  Cupon_ID                              object
...
 96  New_canal                             object
 97  Canal_grouped                         object
dtypes: float64(43), int64(4), object(51)
memory usage: 12.2+ GB
```

Figura 1. Estructura expandida del dataset

A través del método ‘info’ de pandas, podemos tener un vistazo rápido sobre la estructura básica de nuestros datos, esto es, el número de columnas disponibles, el tipo de dato que tiene cada campo, el número de registros totales, el espacio en memoria, entre otros.

Como se observa, el número de registros con el cual se cuenta es considerable dado el volumen de transacciones que el negocio genera año tras año, el crecimiento de nuevos usuarios adquiridos, así como la participación en eventos de venta online como lo son el **“Buen Fin”** y **“Hot Sale”**. En la construcción de nuestro modelo, tenemos que encontrar la forma de poder agrupar estos datos para lograr tiempos de ejecución óptimos de los modelos.

A continuación, se muestran las definiciones de algunos de los campos más importantes del dataset, así como valores ejemplo que dichos campos contienen:

Campo	Descripción	Datatype	Valores Ejemplo
Almacen de origen	*Nodo de orígen del cual sale le mercancía final*	*String*	*FC, DSV, MKP*
Almacen detalle	*Nodo de orígen del cual sale le mercancía final (más granular)*	*String*	*FC_5870,FC_6505*
Banco	*Banco mediante el cual se realizó la compra*	*String*	*Banamex*
Canal	*Canal de venta mediante el cual se realiza la compra*	*String*	*Digital_A, Kiosco_A*
Cantidad	*Piezas vendidas para una compra particular*	*Int64*	*0,1,2,3*
Carrier	*Carrier de transporte de mercancía hacia ruta final*	*String*	*XHL, Fedex, FDD*
Categoria	*Categoría comercial a la que pertenece un producto*	*String*	*TV y Video, Celulares*
Tipo de pago	*Tipo de pago mediante el cual se comprobó una compra*	*String*	*Paypal, Tarjeta de crédito*
Codigo postal	*Código postal de envío final de la mercancía de un pedido*	*String*	*55400,05456*
Cupon_ID	*Código promocional utilizado en pedido*	*String*	*NULL, ASODESC2023*
Customer_ID	*Código identificador del cliente*	*String*	*C1000021,C1000023*
Estado de Republica	*Estado de la república a la cual se envía la mercancía*	*String*	*Mexico, Ciudad de Mexico, Chihuahua*
Fecha de creacion	*Fecha en la cual se generó la órden de compra*	*Date*	*11/23/2023,05/06/2023*
Fecha de cancelacion	*Fecha en la cual se generó la cancelación de la compra*	*Date*	*11/10/2023,04/07/2023*
Motivo cancelacion	*Motivo/reason por el cual un cliente realizó la cancelación de una c*	*String*	*LineBackOrder, No se aplicó el pago a I*
MSI	*Mes de parcialización en caso de ser una venta diferrida*	*String*	*0,1,3,6,9*
Producto	*Descripción del producto*	*String*	*Iphone 15, Laptop Asus 15A*
UPC	*Código identificador único del producto vendido*	*String*	*00019749716181*
Order_ID	*Número identificarlo de la órden*	*String*	*Order_22332628*
Venta bruta sin descuentos con flete	*Venta final registrada de la transacción*	*Double*	*1212.2,21554.2*
Venta cancelada	*Venta cancelada registrada de una transacción*	*Double*	*6565.1,659.5*
Piezas canceladas	*Piezas canceladas registrada de una transacción*	*Int64*	*0,1,2*
Fecha de confirmación	*Fecha en la cual se confirmó un pago realizado por el cliente*	*Date*	*1/11/2023,5/11/2023*

Tabla 2. Diccionario de datos del dataset de trabajo

Campo	Valores Ejemplo
Almacen de origen	*FC, DSV, MKP*
Almacen detalle	*FC_5870,FC_6505*
Banco	*Banamex*
Canal	*Digital_A, Kiosco_A*
Cantidad	*0,1,2,3*
Carrier	*XHL, Fedex, FDD*
Categoria	*TV y Video, Celulares*
Tipo de pago	*Paypal, Tarjeta de crédito*
Codigo postal	*55400,05456*
Cupon_ID	*NULL, ASODESC2023*
Customer_ID	*C1000021,C1000023*
Estado de Republica	*Mexico, Ciudad de Mexico, Chihuahua*
Fecha de creacion	*11/23/2023,05/06/2023*
Fecha de cancelacion	*11/10/2023,04/07/2023*
Motivo cancelacion	*LineBackOrder, No se aplicó el pago a MSI*
MSI	*0,1,3,6,9*
Producto	*Iphone 15, Laptop Asus 15A*
UPC	*00019749716181*
Order_ID	*Order_22332628*
Venta bruta sin descuentos con flete	*1212.2,21554.2*
Venta cancelada	*6565.1,659.5*
Piezas canceladas	*0,1,2*
Fecha de confirmación	*1/11/2023,5/11/2023*

Tabla 3. Valores ejemplo de cada uno de los campos del dataset

2. Exploratory Data Analysis (EDA):

Dentro del análisis exploratorio, se estudiaron las diferentes distribuciones y correlaciones de las variables independientes para ayudar a contestar las preguntas planteadas por negocio sobre el comportamiento de compra de nuestro cliente bajo el efecto de **"Churn"** y cómo este se ve afectado por las diferentes variables contenidas en el dataset.

A continuación, veremos la resolución de algunas de las preguntas planteadas por negocio a través de análisis univariados y multivariados, los cuales, permitirán identificar aquellos factores de influencia sobre el Churn e identificar cuales son las variables que deberán considerarse dentro de nuestro modelo de Churn.

2.1 ¿Cuántos de nuestros clientes presentan Churn respecto a la proporción total de clientes?

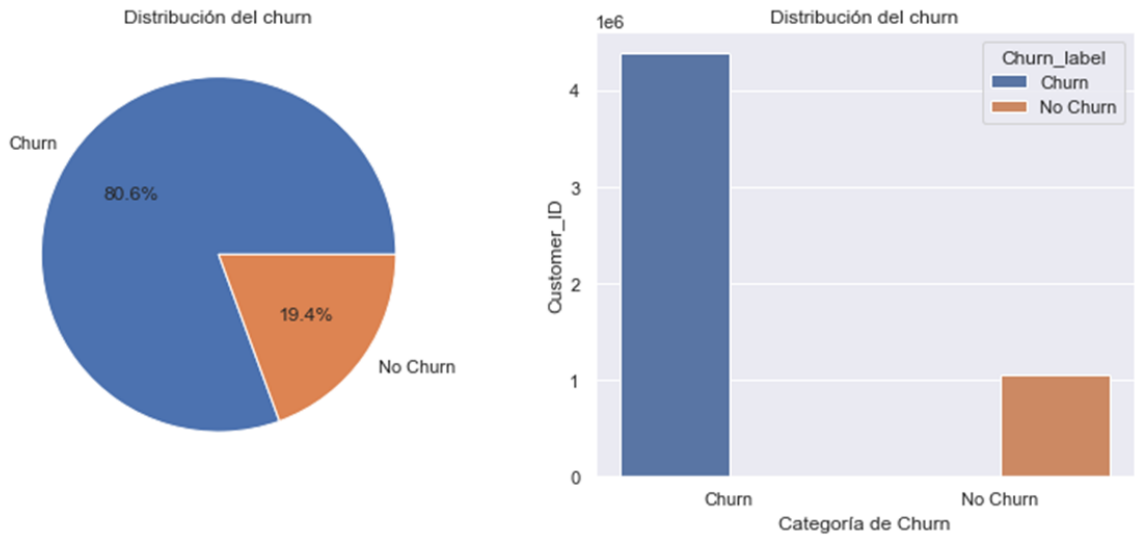

Figura 2 Proporción de clientes con Churn sobre la base total

Solo un 19% de la base total de la base **no presentan Churn**, es decir, han realizado una compra en los últimos tres meses. Esto resulta importante de remarcar ya que este hecho nos daría el insights de estar ante un problema de clases imbalanceadas, por lo que, para nuestro modelo de predicción, tendremos que tener en cuenta dicho efecto para que nuestro modelo pueda generalizar ambas clases, sin embargo, cabe remarcar que nuestro interés principal del modelo recae en identificar la clase **"Churn"**.

2.2 ¿Cómo luce el Churn en función del método de pago de nuestros clientes?

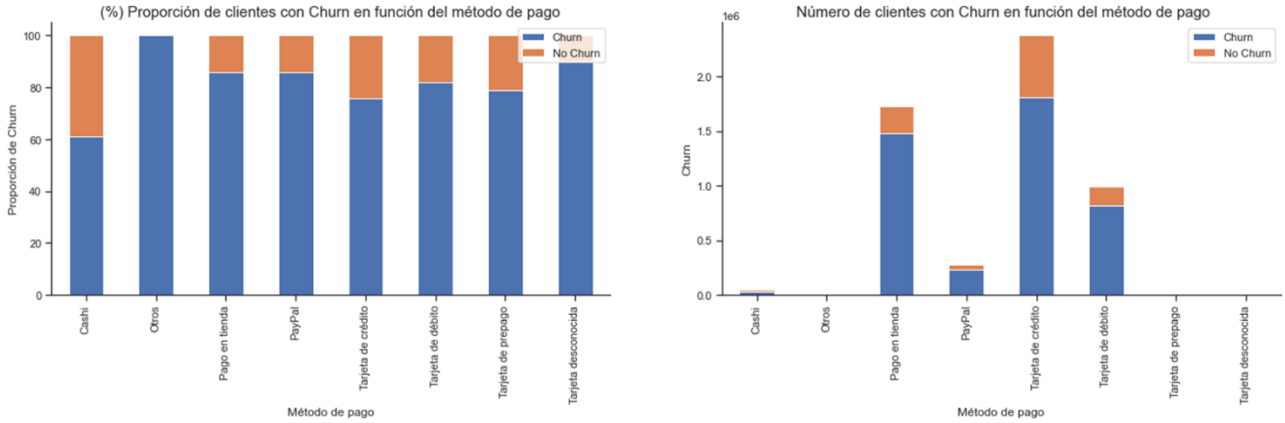

Figura 3 Descripción del churn en función del tipo de pago

Observamos que el método de pago de **tarjeta de Crédito** presenta el menor Churn: **78%**, mientras que el método de pago con mayor Churn corresponde a **"Pago en tienda" con 83%**. Cabe destacar que **Tarjeta de crédito** tendría el menor Churn dada las promociones financieras (MSI) que se ofrecen a los clientes para la compra de productos de mercancías generales. Así, el campo de **tipo de pago** sería **una variable potencial que considerar** dada la correlación del churn en función del tipo de pago.

2.3 ¿Cómo luce el Churn en función del método de pago de nuestros clientes?

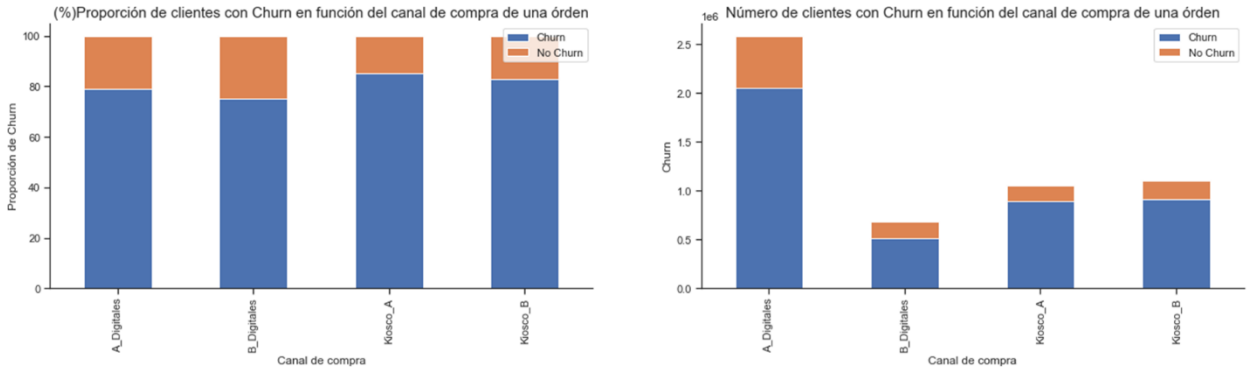

Figura 4 Descripción del churn en función del canal de compra

Vemos que **"Kiosco_A"**, el cual ofrece a nuestros clientes la posibilidad de realizar una compra de **ecommerce** a través de un módulo colocado en una tienda física, presenta el mayor Churn sobre todos los canales de compra. También implica el canal **"Kiosco_B"** bajo la misma modalidad.

Esto nos remarca que los canales digitales como el sitio Web o la App presentan menor Churn dado que son canales de compra más accesibles para nuestros clientes, mientras que en la tienda física la recurrencia de visita al

módulo de tienda tendería a ser menor. Por otra parte, vale la pena resaltar que no hay una gran variabilidad en el Churn para cada uno de los tipos de pago, por lo que la **adición de esta variable en el modelo de Churn no estaría considerado**.

2.4 ¿Cómo luce el ticket promedio en función de si un cliente presenta Churn o no?

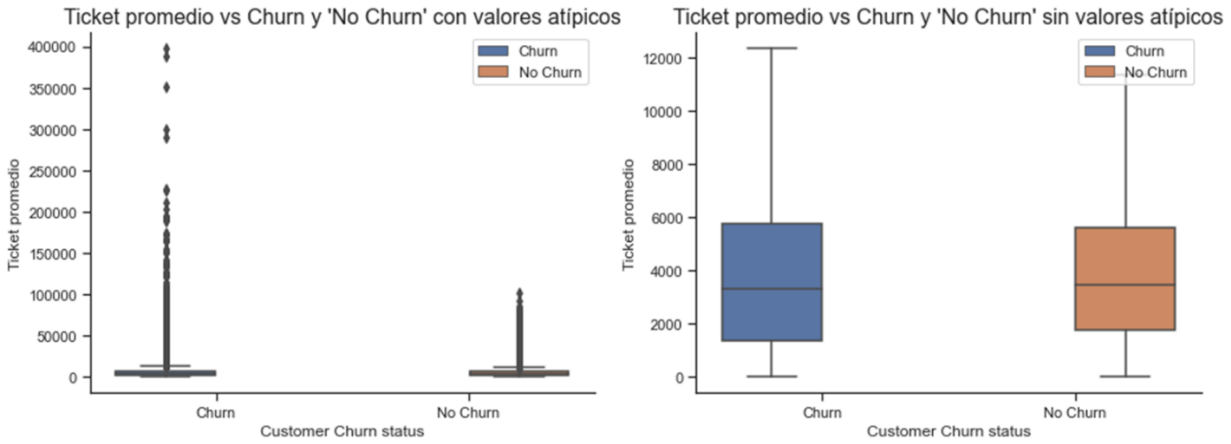

Figura 5 Distribución del churn en función del ticket promedio

Cabe señalar el gran número de **Tickets atípicos** para aquellos clientes que presentan Churn.

Sin embargo, graficando la distribución de nuestros tickets sin la consideración de valores atípicos, no se observa una variación importante entre los clientes que presentan Churn y los que no. Esto nos diría también que el mix de categorías o preferencia de compra de categorías entre los clientes que compran y que dejaron de comprar serían equivalentes o muy parecidas.

2.5 ¿Cómo luce la distribución del basket size en función de un cliente que presenta Churn o no?

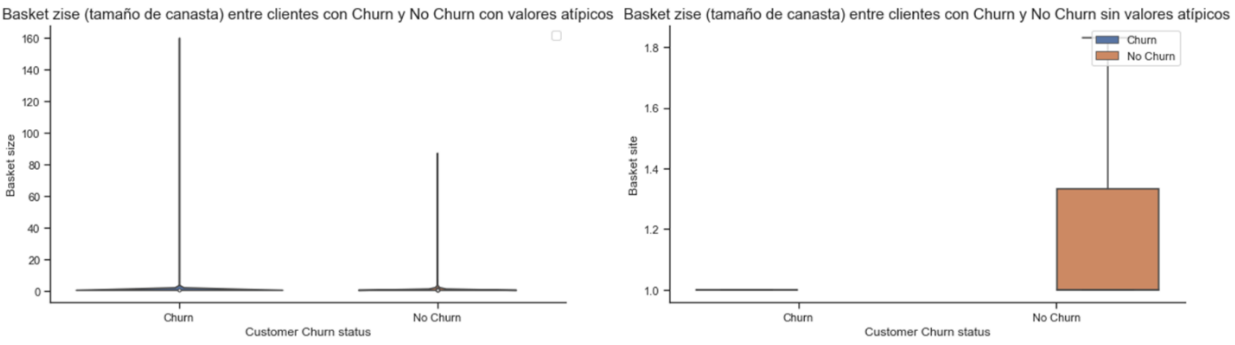

Figura 6 Distribución del churn en función del basket size

Observamos que los clientes con **Churn** se componen principalmente de datos atípicos con el modelo interquantil.

Por lo que parece que, de acuerdo con este modelo, **existe una gran variabilidad entre los productos que agregan a carrito para la transacción de su compra**. Mientras que los clientes con **"No Churn"**, también se presentan datos atípicos, sin embargo, un 75% de esos clientes agregan hasta **1.3 artículos por pedido**.

Esto haría sentido con la mercancía que el negocio desplaza, es decir, la mercancía que se desplaza solo son productos de Línea Blanca, Electrodomésticos, Videojuegos, TV y Video, Muebles, etc.

2.6 ¿Cómo luce la 'recencia' en función de si un cliente presenta Churn o no?

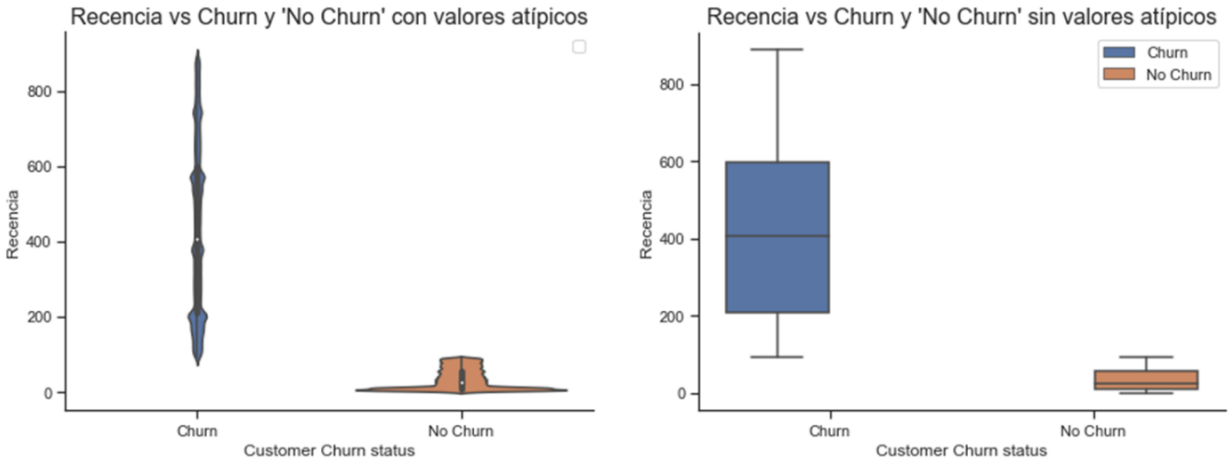

Figura 7 Distribución del churn en función de la recencia

La recencia de los clientes que **no presentan Churn** es esperadamente **menor**, esto, debido a que son aquellos clientes en los cuales siguen realizando compras de manera frecuente. Así, **el tiempo que pasa entre la compra más reciente será significativamente menor impulsando la frecuencia de compra**.

2.7 ¿Cómo lucen las ratios de cancelación en función de un cliente que presenta Churn o no?

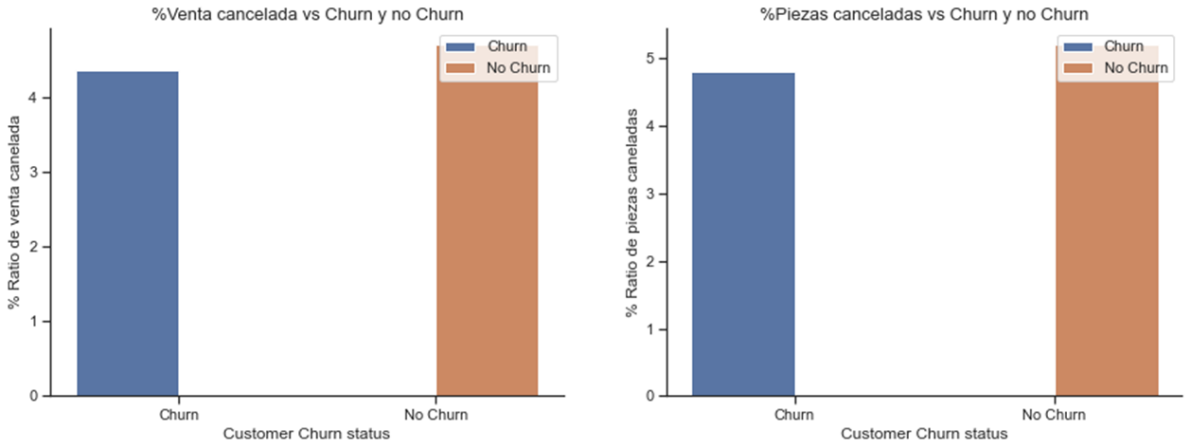

Figura 8 Proporción de venta y piezas canceladas de cliente con y sin Churn

Se observa que la **ratio de cancelaciones** entre los clientes que dejaron de comprar (Churn) **es menor** con aquellos que no tienen Churn. La ratio de los primeros es de **4.5%**, mientras que el segundo caso es un **5%**. Esto nos diría que **los clientes que dejan de comprar no tienen un alto índice de cancelaciones** lo cual, a primera instancia, **nos diría que las cancelaciones no necesariamente coadyuvan a que un cliente deje de comprar**.

Por otra parte, si vemos la perspectiva de **ratio de cancelaciones** en término de piezas, observamos que tenemos **las mismas ratios** en ambos casos.

A pesar de no ver una correlación importante o clara entre la variable Churn y no Churn, se considerará este factor dentro del modelo de predicción de **Machine Learning**.

2.8 Análisis multivariado: Identificando patrones de relación

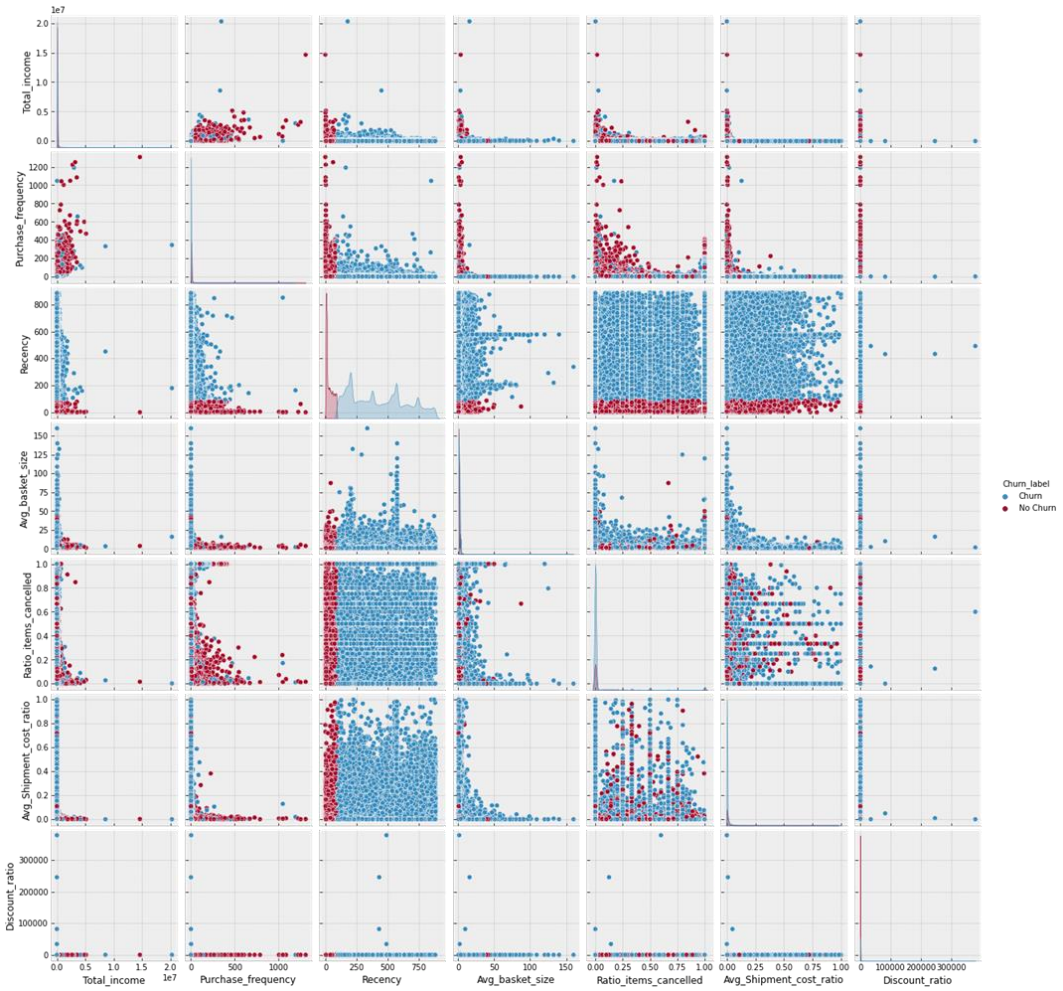

Figura 9 Análisis de correlación lineal entre las variables del modelo de Churn

Se realiza un gráfico del tipo **Pair Plot** el cual conjunta a través de scatterplots las diferentes relaciones lineales entre las variables que se utilizarán para el modelo de Churn.

Dentro de los histogramas de distribución, algunos gráficos no resultan claro del todo ya que vemos que las **distribuciones son de colas largas o de Curtosis elevada**, con lo cual, nos indica la **presencia de outliers** en nuestros datos. Identificar este comportamiento es importante ya que afectan de manera significativa como modelos de regresión lineal, logística, árboles de decisión, KNN, Support vector machines, entre otros.

También, no existen patrones claros de correlación lineal entre las diferentes variables del modelo de Churn

Si agregamos un código de color de los clientes con **“Churn” y “No Churn”**, observamos una diferenciación importante de dichas clases en los datos.

Dicho comportamiento nos daría un **"hint"** de que existe una **correlación** entre la variable dicotómica **"DidBuy"** y las **variables independientes** para la construcción del modelo.

2.9 Análisis de correlación bivariado: medida punto biserial

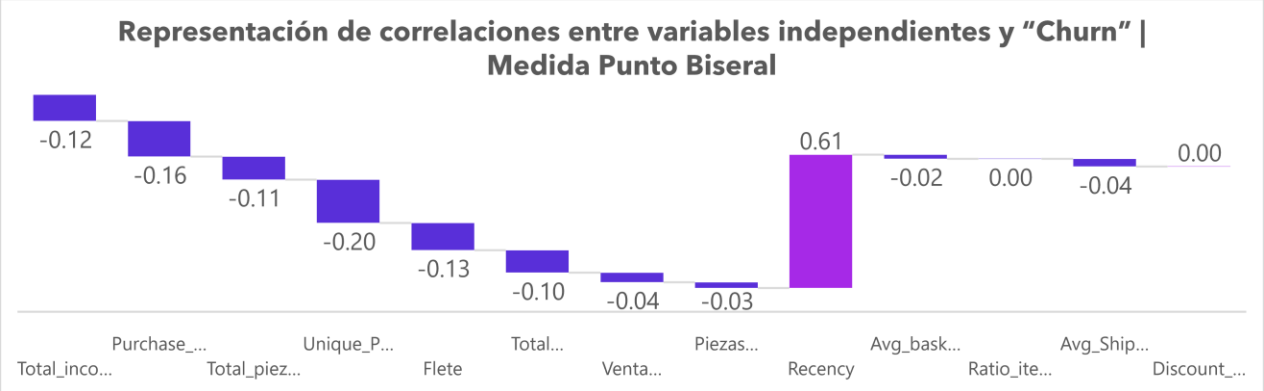

Figura 10 Correlación de la variable Churn a través de la medida Punto Biserial

Dentro de la naturaleza de nuestro problema, la variable objetivo de "Churn", denominada como "DidBuy", es una variable dicotómica que toma el valor "1" si el cliente ha comprado en los últimos tres meses y cero en otro caso. El interés recae en conocer cómo se correlaciona la variable objetivo de **"Churn"** contra las demás variables independientes del modelo de predicción. La correlación que nos puede dar esta visibilidad se denomina **"Punto Biserial"**.

En la **tabla 3**, se muestra el resumen de las correlaciones mostradas de la figura 10, donde se observan relaciones interesantes como en el caso de la variable **"Total_income"**, el cual, indicaría que cuando se toman valores igual a **"Churn"**, el **"Total_income"** tenderá a tomar valores a la baja o viceversa.

El **P value** de esta relación es cero, esto quiere decir que se **rechaza la hipótesis nula** de que la correlación sea igual a cero, es decir, la **correlación es estadísticamente significativa**

La misma correlación se puede observar para otras variables como **"Purchase_frequency"**, **"Total_piezas"**, **"Total_descuentos"**, entre otros.

Variable	Correlación con variable "Churn"	P Values
Total_income	-0.12	0.00
Purchase_frequency	-0.16	0.00
Total_piezas	-0.11	0.00
Unique_Products_purchased	-0.20	0.00
Flete	-0.13	0.00
Total descuentos	-0.10	0.00
Venta cancelada	-0.04	0.00
Piezas canceladas	-0.03	0.00
Recency	0.61	0.00
Avg_basket_size	-0.02	0.00
Ratio_items_cancelled	0.00	0.64
Avg_Shipment_cost_ratio	-0.04	0.00
Discount_ratio	0.00	0.43

Tabla 4 Resumen de correlaciones con punto biserial

3. Feature Engineering:

De acuerdo con nuestro análisis exploratorio realizado, se crearán variables adicionales basadas en un modelo de **segmentación de clientes** denominado **"RFM Model"**, el cual, proviene de sus siglas "Recency", "Frequency" y "Monetary value", así como aquellas que se estudiaron en el **EDA (Exploratory Data Analysis)** (Ratio de descuentos, ticket promedio, ratio de cancelaciones, entre otros).

También, se realiza un **reescalamiento** de los datos con el método de Scikitlearn denominado **"MinMaxScaler"**, esto, para que las variables independientes tengan valores entre 0 – 1 y así mejorar los tiempos de procesamiento

En la siguiente figura, se muestra la construcción con Python de algunas de las funciones que se utilizarán dentro de otra función la cual hará un proceso **recursivo** del cálculo de las variables de RFM.

```python
def customer_basket_size(df_customer, cut_off):
    #We only focused in the number of pieces cancelled, not the amount of monetary value.
    df_bs = df_customer.loc[df_customer.loc[:,"Fecha"] <= cut_off,:].copy()
    df_bs.drop(columns=["Año","Mes"], inplace=True)

    df_bs = df_bs.groupby(["Customer_ID"]).agg({'Piezas': np.sum , 'Order_ID': lambda x: x.nunique()}).reset_index().rename(columns = {'Order_ID':'Ordenes'})
    df_bs.loc[:,"Basket_size"] = df_bs.Piezas / df_bs.Ordenes

    return df_bs.loc[:,["Customer_ID","Basket_size"]]

def customer_shipping_ratio(df_customer, cut_off):
    #We only focused in the number of pieces cancelled, not the amount of monetary value.
    df_sr = df_customer.loc[df_customer.loc[:,"Fecha"] <= cut_off,:].copy()
    df_sr.drop(columns= ["Año","Mes"], inplace=True)

    df_sr = df_sr.groupby(["Customer_ID"]).agg({'Costo_envio': np.sum,'Venta_pagada': np.sum}).reset_index()
    df_sr.loc[:,"Shipping_ratio"] = df_sr.Costo_envio / df_sr.Venta_pagada

    return df_sr.loc[:,["Customer_ID","Shipping_ratio"]]

def customer_discount_ratio(df_customer, cut_off):
    #We only focused in the number of pieces cancelled, not the amount of monetary value.
    df_dr = df_customer.loc[df_customer.loc[:,"Fecha"] <= cut_off,:].copy()
    df_dr.drop(columns= ["Año","Mes"], inplace=True)

    df_dr = df_dr.groupby(["Customer_ID"]).agg({'Total_descuentos': np.sum,'Venta_pagada': np.sum}).reset_index()
    df_dr.loc[:,"Discount_ratio"] = df_dr.Total_descuentos / (df_dr.Venta_pagada + df_dr.Total_descuentos)

    return df_dr.loc[:,["Customer_ID","Discount_ratio"]]
```

Figura 11 Construcción de funciones para el cálculo de nuevas variables

EL propósito de realizar el **proceso recursivo** es **iterar** sobre múltiples **periodos** para capturar los efectos de **temporalidad y estacionalidad** que impactan o se reflejan dentro del comportamiento del cliente.

Con esto, **realizaríamos un análisis de Churn con múltiples periodos**. La frecuencia de análisis se haría de forma **trimestral** y el **dataset final** es un consolidado con los atributos del RFM y Labels de “Churn” para cada periodo de análisis.

El periodo de iteración comprende desde Junio 2022 hasta el 2023, por lo que, de manera trimestral, iremos calculando las Etiquetas de Churn y las características del enfoque del RFM.

A continuación, se muestra el resultado final de la base de información a utilizar dentro de nuestro modelo de predicción de Churn:

	Customer_ID	Recency	Frequency	Average_sales	Total_sales	Age	Ratio_cancellations	Basket_size	Shipping_ratio	Discount_ratio	DidBuy
0	C1000000	87.0	1.0	1033.610000	1033.610000	87.0	0.0	1.0	0.000000e+00	0.000010	1.0
1	C1000007	80.0	1.0	11874.120000	11874.120000	80.0	0.0	2.0	1.408441e-02	0.153621	0.0
2	C1000020	34.0	1.0	6801.710000	6801.710000	34.0	0.0	1.0	0.000000e+00	0.122360	0.0
3	C1000021	36.0	1.0	4783.623793	4783.623793	36.0	0.0	1.0	7.929351e-07	0.000000	0.0
4	C1000024	53.0	1.0	654.311034	654.311034	53.0	0.0	1.0	1.581026e-06	0.000000	1.0

Figura 12 Resultado final del dataset para entrenamiento de modelos

Las variables finales para utilizar dentro del modelo son los siguientes:

Variables finales	Definición
Recency	Tiempo en días de la última compra
Frequency	Número de pedidos
Average_sales	Venta generada promedio del cliente
Age	Tiempo en días de la primer compra
Ratio_cancellations	Ratio de cancelaciones de items
Basket_size	#Items promedio por pedido
Shipping_ratio	Ratio de costo promedio de envío
Discount_ratio	Ratio de descuentos promocionales
DidBuy	Variable de Churn o no

Tabla 5: Variables finales para construcción de modelo de Churn

4. Selección y entrenamiento de modelos de Machine Learning

En esta sección, procederemos con el entrenamiento y selección de nuestros modelos de Machine Learning a través de las librerías "ScikitLearn" y "xgboost" para modelar la probabilidad de "Churn" en función de las variables finales resultado del EDA.

Los modelos utilizados son los siguientes:

a) Classification decision trees
   Variante A: criterion = 'gini'; max_depth = 8; min_simples_split = 8; max_simples_leaf = 1; splitter = 'best'
b) Random Forests | Bagging methods
   Variante A: n_estimators = 100, criterion = 'entropy', min_simples_split = 3, min_simples_leaf = 2, max_depth = 10
   Variante B: n_estimators = 200, criterion = 'gini', min_simples_split = 3, min_simples_leaf = 2, max_depth = 7
c) XGboost | Boosting Methods
   Variante A: n_estimators = 100, learning_rate= 0.1, max_depth = 6
   Variante B: n_estimators = 200, learning_rate= 0.5, max_depth = 8
d) Logistic Regression
   Variante A: solver = 'lbfgs'

Se decide utilizar métodos de árboles al ser de rápida implementación y efectivos en cuanto al manejo de outliers para el caso de **"Random Forests"** y **"XGboost"**. Cómo se observó en el EDA, los histogramas de frecuencia presentaban **colas largas** o alta **curtosis** por lo que también se recomienda tomar modelos que puedan tener un buen manejo de estos factores.

5. Resultados de performance de los modelos de Machine Learning

A continuación, se presenta un resumen de los resultados obtenidos de performance de los modelos de machine learning entrenados (la validación utilizada para la medición del performance es del tipo **"Simple"**):

Resultados:

	Decision Tree		Random Forest A		Random Forest B		XGboost A		XGboost B		Logistic Regression	
Métrica	Churn	No Churn	Churn	No Churn	Churn	No Churn	Churn	No Churn	Churn	No Churn	Churn	No Churn
Precision	0.93	0.24	0.89	0.66	0.89	0.68	0.91	0.27	0.90	0.30	0.89	0.62
Recall	0.79	0.51	1	0.05	1	0.04	0.90	0.28	0.95	0.18	1	0.04
F1 Score	0.86	0.33	0.94	0.10	0.94	0.08	0.91	0.28	0.92	0.22	0.94	0.08
Accuracy	0.76		0.89		0.89		0.83		0.86		0.89	
AUC (ROC)	0.71		0.73		0.72		0.67		0.66		0.71	

Figura 13 Resultados de entrenamiento con validación simple

Observamos que los modelos de Boosting de árboles como los modelos **XGBoost** son los más consistentes en relación con las métricas de **Precision**, **Recall** y **F1 Score**. Cabe destacar que en todos los modelos tenemos bajos índices en las métricas mencionadas sobre la **clase positiva** o **"No Churn"**, con lo cual, el modelo presentaría una oportunidad para identificar y generalizar dicha clase. El indicador **AUC** de la curva **ROC** de todos los modelos son muy similares bajo un promedio de **0.7**, por lo cual, nuestros modelos tienen buen performance en cuanto a diferenciar las dos clases: "Churn" y "No Churn".

Se muestran algunas de las curvas ROC y matrices de confusión resultado del entrenamiento de los modelos de machine learning,

XGboost model, variante B:

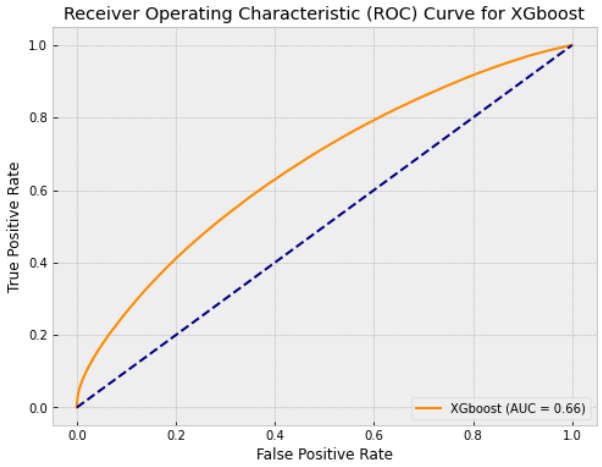

Figura 14 Curva ROC de la variante B de XGboost

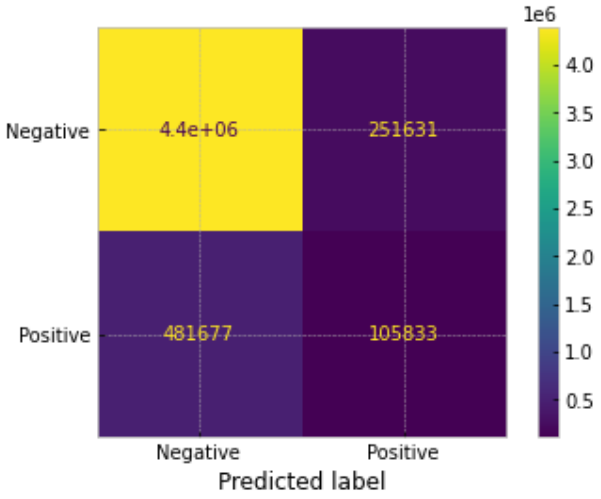

Figura 15 Matriz de confusión Xgboost model, variante B

Random Forest, variante A:

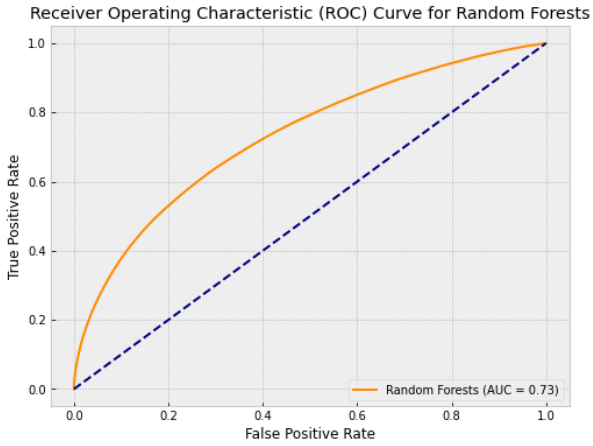

Figura 16 Curva ROC de la variante A de Random Forests

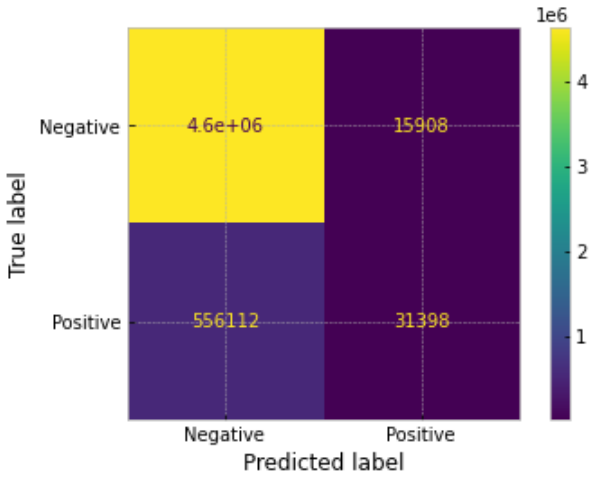

Figura 17 Matriz de confusión Random forests, variante A

6. Atacando el problema de desbalanceo de clases: Oversampling

Como se observó en el análisis exploratorio, de todas las instancias de clientes que tenemos identificadas, el **20% de los clientes no presentan "Churn"**, mientras que el **80%** son al caso contrario.

Para atacar este problema y que nuestro modelo permita generalizar ambas clases, utilizaremos un método de **balanceo de clases denominado SMOTE**, el cual, realiza un **'oversampling'** de la clase **minoría "No churn"**, manteniendo una proporción **50% - 50%**.

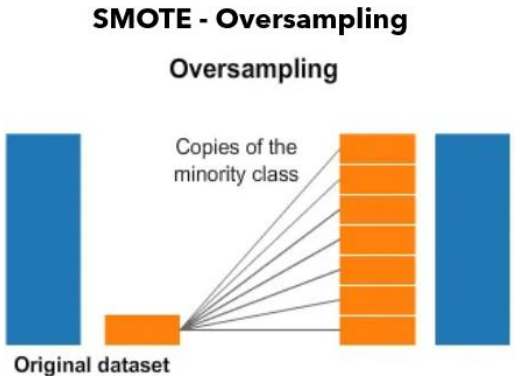

Figura 18: Técnica de oversampling sintético: SMOTE

7. Comparación de performance de modelos de ML con oversampling

Resultados sin oversampling (SMOTE)

	Decision Tree		Random Forest A		Random Forest B		XGboost A		XGboost B		Logistic Regression	
Métrica	Churn	No Churn	Churn	No Churn	Churn	No Churn	Churn	No Churn	Churn	No Churn	Churn	No Churn
Precision	0.93	0.24	0.89	0.66	0.89	0.68	0.91	0.27	0.90	0.30	0.89	0.62
Recall	0.79	0.51	1	0.05	1	0.04	0.90	0.28	0.95	0.18	1	0.04
F1 Score	0.86	0.33	0.94	0.10	0.94	0.08	0.91	0.28	0.92	0.22	0.94	0.08
Accuracy	0.76		0.89		0.89		0.83		0.86		0.89	
AUC (ROC)	0.71		0.73		0.72		0.67		0.66		0.71	

Resultados con oversampling (SMOTE)

	Decision Tree		Random Forest A		Random Forest B		XGboost A		XGboost B		Logistic Regression	
Métrica/Clase	Churn	No Churn	Churn	No Churn	Churn	No Churn	Churn	No Churn	Churn	No Churn	Churn	No Churn
Precision	0.93	0.24	0.93	0.24	0.93	0.22	0.91	0.27	0.90	0.30	0.94	0.21
Recall	0.79	0.51	0.77	0.56	0.73	0.59	0.90	0.28	0.95	0.18	0.69	0.63
F1 Score	0.86	0.33	0.84	0.33	0.82	0.32	0.91	0.28	0.92	0.22	0.80	0.31
Accuracy	0.76		0.75		0.72		0.83		0.86		0.68	
AUC (ROC)	0.71		0.72		0.72		0.67		0.66		0.71	

Figura 19 Comparación de resultados vs Oversampling (SMOTE)

No se observa una mejora clara en la aplicación de SMOTE sobre nuestros modelos de ML. También se observa una caída de las métricas en términos de precisión y recall para algunos casos cómo en los modelos de Random Forests.

8. Optimización de modelos de Machine Learning

Para realizar la optimización correspondiente de los modelos de Machine Learning, se utilizó un **método de validación cruzada**, particularmente, **Stratified K Fold Cross validation** para el diagnóstico y prevención de "**Overvitting**" en nuestro modelo, así como el **ajuste de los hiperparámetros** a través de **"Grid Search CV"**. Los resultados fueron los siguientes:

Resultados iniciales de modelos

	Decision Tree		Random Forest A		Random Forest B		XGboost A		XGboost B		Logistic Regression	
Métrica	Churn	No Churn	Churn	No Churn	Churn	No Churn	Churn	No Churn	Churn	No Churn	Churn	No Churn
Precision	0.93	0.24	0.89	0.66	0.89	0.68	0.91	0.27	0.90	0.30	0.89	0.62
Recall	0.79	0.51	1	0.05	1	0.04	0.90	0.28	0.95	0.18	1	0.04
F1 Score	0.86	0.33	0.94	0.10	0.94	0.08	0.91	0.28	0.92	0.22	0.94	0.08
Accuracy	0.76		0.89		0.89		0.83		0.86		0.89	
AUC (ROC)	0.71		0.73		0.72		0.67		0.66		0.71	

Resultados con SMOTE

	Decision Tree		Random Forest A		Random Forest B		XGboost A		XGboost B		Logistic Regression	
Métrica/Clase	Churn	No Churn	Churn	No Churn	Churn	No Churn	Churn	No Churn	Churn	No Churn	Churn	No Churn
Precision	0.93	0.24	0.93	0.24	0.93	0.22	0.91	0.27	0.90	0.30	0.94	0.21
Recall	0.79	0.51	0.77	0.56	0.73	0.59	0.90	0.28	0.95	0.18	0.69	0.63
F1 Score	0.86	0.33	0.84	0.33	0.82	0.32	0.91	0.28	0.92	0.22	0.80	0.31
Accuracy	0.76		0.75		0.72		0.83		0.86		0.68	
AUC (ROC)	0.71		0.72		0.72		0.67		0.66		0.71	

Resultados con Grid Search CV y Stratified K fold cross validation

	Decision Tree		Xgboost		Logistic Regression	
Métrica/Clase	Churn	No Churn	Churn	No Churn	Churn	No Churn
Precision	0.9	0.58	0.89	0.63	0.94	0.21
Recall	0.99	0.08	0.99	0.08	0.70	0.63
F1 Score	0.94	0.14	0.94	0.14	0.80	0.31
Accuracy	0.89		0.89		0.69	
AUC (ROC)	0.72		0.73		0.72	

Figura 20 Comparación de resultados vs Oversampling (SMOTE) y Grid Search CV

En este caso, solo se **entrenaron tres modelos** bajo las configuraciones de **Stratified K Fold Cross validation** y **Grid Search CV**.

Los mejores parámetros de cada uno de los modelos obtenidos por **Grid Seach CV** son los siguientes:

a) Classification decision trees

Hiperparémetros:

{'criterion': 'gini', 'max_depth': 4, 'min_samples_leaf': 1, 'min_samples_split': 2, 'splitter': 'best'}

b) Logistic Regression

Hiperparámetros:

{'C': 1, 'class_weight': 'balanced', 'penalty': 'none'}

c) XGboost

Hiperparámetros:

{'learning_rate': 0.2, 'max_depth': 8, 'n_estimators': 200}

9. Conclusiones

A través del análisis exploratorio, podemos conocer la estructura de nuestros datos e identificar aquellas variables que puedan tener correlación con nuestra variable objetivo, en este caso, con la variable Churn. Con ésto, pudimos conocer como el **Churn** se relacionaba con los **métodos de pago**, los **canales de compra**, los **carriers de entrega** de la mercancía y cuáles son los principales **motivos de cancelación**.

Una vez realizado el análisis, a través del proceso de **"Feature engineering",** creamos las variables adicionales que den pie a la construcción del modelo de predicción. El método de **segmentación de clientes recursivo RFM**, así como la adición de variables como cancelaciones, ratio de descuentos, costo de envío, etc., permiten capturar la mayoría de los factores que ayuden a explicar el efecto de **"Churn"** sobre nuestros clientes.

A través del entrenamiento de diferentes modelos de Machine Learning, podemos tomar aquel con el mejor performance en términos de predicción de la variable objetivo. Los modelos **de árboles fueron preferentemente seleccionados dado su rápida implementación y flexibilidad** para el manejo de datos atípicos como **XGboost** o **Random Forests**, sin embargo, esto no sucede en modelos de árboles simples o en **logistic regression**. También, la implementación de métodos de validación cómo **cross validation** nos permite reducir el efecto de **"Overfitting"** y tener una mejor visibilidad del performance real de nuestro modelo sobre diferentes subconjuntos del dataset.

Respecto a cuál modelo debemos elegir, dado que nuestro interés es predecir cuales tienen mayor **probabilidad de Churn, tomaremos aquel** con mayor **precision** y **recall** sobre la clase "Churn". **Así, tomamos el modelo "XGBoost variante A sin SMOTE (validación simple)**. Tiene buen **precision**, **recall** y **f1 score,** así como buen performance de estos mismos indicadores sobre la clase "No Churn" Mantiene buen **AUC de 0.67** y un **accuracy de 83%**. Debemos tener cuidado sobre este último por el problema de desbalanceo, sin embargo, la aplicación de oversampling no contribuyó a la mejora de dicho modelo. Adicionalmente, recomendámos realizar pruebas sobre otra malla de parámetros de Grid Search CV para el algoritmo de **XGBoosting**.