Predicción de la fidelidad de clientes según los hábitos de compra

Luis Guillermo Portela Santos
Omar Darío Salazar Ruiz

Monografía presentada para optar al título de Especialista en Analítica y Ciencia de Datos

Tutor
Efraín Alberto Oviedo Carrascal, Magíster (MSc) en TICs

Universidad de Antioquia
Facultad de Ingeniería
Especialización en Analítica y Ciencia de Datos
Medellín, Antioquia, Colombia
2022

Cita	(Portela Santos & Salazar Ruiz, 2022)
Referencia **Estilo APA 7 (2020)**	Portela Santos, L., & Salazar Ruiz, O. (2022). *Predicción de la fidelidad de clientes según los hábitos de compra* [Trabajo de grado especialización]. Universidad de Antioquia, Medellín, Colombia.

Especialización en Analítica y Ciencia de Datos, Cohorte III.

UNIVERSIDAD DE ANTIOQUIA
Vicerrectoría de Docencia

Sistema de Bibliotecas

Centro de Documentación Ingeniería (CENDOI)

Repositorio Institucional: http://bibliotecadigital.udea.edu.co

Universidad de Antioquia - www.udea.edu.co

Rector: Jhon Jairo Arboleda Cespedes
Decano: Jesús Francisco Vargas Bonilla
Jefe departamento: Diego José Luis Botia Valderrama

TABLA DE CONTENIDOS

1. RESUMEN EJECUTIVO

ELO es una marca de pago de Brasil la cual ha establecido alianzas con comerciantes para ofrecer promociones y descuentos a los titulares de tarjetas de crédito. En tal sentido, la compañía requiere conocer la efectividad de las campañas de descuentos con los comercios aliados y evaluar si sus promociones funcionaron para sus clientes. Por eso, se apoya de los datos recopilados en los ciclos de vida de sus clientes para identificar a partir de características propias de estos, sus gustos y preferencias de compras, y de ese modo identificar y atender las oportunidades más relevantes para las personas, al descubrir señales en la lealtad del cliente. (Elo Merchant Category Recommendation | Kaggle, 2019)

Para esto, ELO, a través de la plataforma de kaggle, ha propuesto un concurso el cual busca predecir la lealtad de los clientes, la cual se define como un valor continúo normalizado. Los datos suministrados por el concurso consisten en cuatro bases de datos, las cuales contienen información agregada y desagregada relacionada con las transacciones realizadas por los clientes poseedores de una tarjeta de crédito.

Dado que la estimación de la fidelidad es una variable continua, se usarán inicialmente los modelos asociados la tarea de regresión, considerando desde los modelos sencillos, hasta los más complejos, garantizando de igual forma la optimización de los hiperparámetros principales de cada uno de los algoritmos evaluados. Paralelamente, con el fin de complementar la solución del problema analítico planteado, se propone la discretización de la variable “lealtad” en rangos, con el fin de hacer un acercamiento mucho real al problema, en el cual se buscaría identificar, de forma binaria, aquellos clientes que son fieles y aquellos que no lo son.

Sin embargo, los distintos acercamientos analíticos, ya sea para regresión o para clasificación, partieron de la premisa de la interpretación personal de las características analizadas. En este sentido, las distintas bases de datos suministradas en el concurso cuentan con la completa anonimización de las características suministradas y el escalamiento de la mayoría de los valores continuos presentes en las

tablas. Estos hechos, por definición, imposibilitan un acercamiento más profundo e interpretativo del problema y de los datos en cuestión.

La dirección del repositorio público de Github en donde se encuentra a disposición el conjunto de notebooks y documentos anexos para el preprocesamiento de datos es https://github.com/LuisPortela/ELO.

2. DESCRIPCIÓN DEL PROBLEMA

2.1 PROBLEMA DE NEGOCIO

La compañía de pagos brasileña ELO, busca principalmente conocer el comportamiento de compras de sus clientes. Por esta razón, la estimación de la fidelidad de los usuarios de las tarjetas de crédito es de gran utilidad desde un enfoque estratégico/comercial.

En este sentido, la estimación de la fidelidad de un cliente, y principalmente el hecho de identificar la receptividad que tiene un usuario de responder positiva o negativamente a las estrategias comerciales de los negocios asociados, permite orientar de forma más adecuada las campañas de marketing, focalizándose mayoritariamente a los clientes catalogados como fieles, y por ende, receptivos a estas campañas. De forma opuesta, se espera que los clientes menos fieles reciban con una menor frecuencia este tipo de campañas, permitiendo así la reducción de costos operacionales en el área de Mercadeo de la compañía.

Paralelamente, la predicción de la fidelidad de los clientes determinada a partir del comportamiento de compra y de la receptividad hacia las campañas publicitarias, permitiría que ELO realice alianzas más duraderas y rentables con los comercios asociados, al posibilitar un mejor direccionamiento de las campañas publicitarias a los clientes susceptibles de responder positivamente a las mismas.

2.2 APROXIMACIÓN DESDE LA ANALÍTICA DE DATOS

"Los sistemas de aprendizaje automático se pueden clasificar según la cantidad y el tipo de supervisión que reciben durante el entrenamiento. En el aprendizaje supervisado, los datos de entrenamiento que alimenta al algoritmo incluyen las soluciones deseadas, llamadas etiquetas. Dos tareas típicas de aprendizaje supervisado es la clasificación de etiquetas y predicción de un valor numérico objetivo". (Géron, 2017)

Considerando que la empresa de pagos brasileña, ELO, busca predecir la fidelidad de sus clientes con el objetivo de orientar de forma más adecuada las campañas publicitarias de la compañía, se propondrán dos acercamientos desde la analítica de datos

El primer acercamiento desde la analítica será la predicción del valor cuantitativo de la lealtad del cliente en función de sus hábitos de compra. Por esta razón, los algoritmos predictivos empleados serán asociados a una tarea de regresión.

En este sentido, la variable "lealtad" se define como una característica continua escalada, la cual puede tener tanto valores negativos como positivos y está concentrada principalmente alrededor del cero. Según la interpretación de la información suministrada por el concurso, se considera que el signo asociado a la lealtad define el grupo al cual se podría clasificar al cliente en cuestión. Es decir, un cliente con un score negativo se definiría como infiel mientras que aquellos con valores positivos serian considerados como usuarios fieles. Por otra parte, la magnitud de la variable podría interpretarse como la intensidad asociada al tipo de lealtad. Por esta razón, un cliente con un valor positivo grande es un cliente mucho más fiel que aquel que tenga también un valor positivo, pero con una magnitud más baja que el primero.

El segundo acercamiento desde la analítica se hará con el objetivo de realizar una clasificación de los clientes según su fidelidad. En este sentido, se sigue cumpliendo con el propósito de la identificación de la fidelidad de sus clientes a partir de su comportamiento de compra; pero desde la analítica de datos

se buscará predecir si un cliente es fiel o no (problema biclase). Por consiguiente, los modelos predictivos empleados serán los asociados a los de una tarea de clasificación.

A partir de las dos aproximaciones de la analítica, ya sea desde la tarea de regresión como la de clasificación, se estaría identificando el tipo de lealtad del cliente. Sin embargo, se considera que el objetivo inicial del concurso es el de poder clasificar a sus clientes para orientar de forma más adecuada las promociones de los comercios aliados. En este sentido, la predicción exacta del valor continuo de la lealtad no necesariamente aporta más información que la tarea de clasificación al problema de negocio planteado en el concurso. Indistintamente a la tarea de analítica planteada, ELO podrá determinar el tipo de fidelidad de sus nuevos clientes a partir de sus hábitos de compra, y de esta forma, proponerles en mayor o menor intensidad las ofertas asociadas a los comercios aliados.

2.3 ORIGEN DE LOS DATOS

Los datos han sido obtenidos a partir de un concurso *Elo Merchant Category Recommendation* propuesto en la Kaggle en noviembre del 2018 (Elo Merchant Category Recommendation | Kaggle, 2019). Los datos representan la información agregada y desagregada relacionadas no solo con la utilización de la tarjeta de crédito (Card_ID), sino que también abarca desde las características propias de la tarjeta de crédito, hasta información de los comercios aliados en los cuales se efectuaron las compras. En este marco, se obtienen como información “desagregada”, las transacciones comerciales efectuadas por los usuarios de la Card_ID, durante periodos de tiempo y criterios distintos. Se obtiene una primera base de datos “desagregada” llamada Historical Transactions la cual dispone de las transacciones realizadas por los usuarios durante tres meses en una cantidad de comercios dada. La segunda base de datos “desagregada” es New Merchants Transactions. Esta base de datos, como su nombre lo indica, abarca las transacciones de pagos efectuados en nuevos comercios que no fueron visitados por los usuarios

previamente en Historical Transactions. En este caso, se dispone de dos meses de información, la cual fue recopilada a posteriori de la información de Historical Transactions. Por definición, los comercios presentes tanto en Historical Transactions como en New Merchants Transactions son disjuntos.

Por otra parte, la información agregada se encuentra en la base de datos Train, la cual comprende características generales de tarjetas de crédito, junto con la fidelidad asociada a cada Card_ID. Finalmente, se cuenta con una tabla anexa con la información relativa a los distintos comercios que fueron visitados durante los dos periodos analizados tanto en Historical Transactions como en New Merchants Transactions.

2.4 MÉTRICAS DE DESEMPEÑO

2.4.1 Las métricas de Machine Learning

Como se planteó previamente, se harán dos acercamientos diferentes desde la analítica de datos para abordar la predicción de la fidelidad de los clientes. En el primer enfoque, se abordará la fidelización como una variable continua, lo cual responde a algoritmos asociados a las tareas de regresión.

Para esta tarea, se empleó principalmente el R^2 como métrica de error principal de los modelos. Los autores Juke J. Saunders, Richard A. Russell y David P. Crabb (2012) definen el estadístico R^2, o el coeficiente (múltiple) de determinación, como: “la proporción de variación en la variable de respuesta explicada por un modelo ajustado relativo a simplemente tomando la media de la respuesta. En otras palabras, describe qué tan bien se ajusta el modelo a los datos”. (Saunders, Russell, & Crabb,2012). Sin embargo, de forma paralela, se acompañará esta medida con el $RMSE$, métrica que es definida como: “el cálculo de la magnitud promedio del error entre el valor predicho y el valor actual. Por lo tanto, RMSE es la distancia promedio medida verticalmente desde el valor real hasta el valor pronosticado correspondiente en la línea de ajuste”. (Jierula, Wang, Oh & Wang, 2021). Es conveniente señalar que el

$RMSE$ fue empleado como métrica de desempeño, dado que esta es la métrica de error seleccionada para evaluar las predicciones en el concurso de Kaggle.

En el segundo enfoque, la discretización de la variable continua implica un cambio en la métrica de Machine Learning empleada, al considerarse de una tarea de clasificación. Para esto, la métrica seleccionada fue la curva ROC, ya que la curva ROC es de amplio uso para evaluar el desempeño de métodos clasificatorios, además provee una descripción de la separación entre las distribuciones de positivos y negativos sin requerir de hipótesis probabilísticas. (Bouza,2021).

2.4.2 Las métricas de negocio

A partir de la comprensión del concurso planteado por ELO, se logran identificar tres métricas de negocio que permitirían verificar la eficiencia del modelo predictivo de llegarse a implementar en producción. Las métricas son:

1. **% de campañas efectivas (Disfrutadas por los clientes):**

   Se espera que el indicador que mide la eficiencia de las campañas publicitarias aumente progresivamente. El propósito de este indicador es aproximarse al 100%

$$\%\ campañas\ efectivas = \left(\frac{Número\ de\ campañas\ efectivas\ efectuadas\ en\ periodo_i}{Número\ de\ campañas\ efectuadas\ en\ periodo_i}\right) * 100$$

2. **Variación de ingresos anuales:**

Esta métrica de negocio tomaría como punto de referencia el periodo $t = 0$ (implementación del algoritmo en producción). Se espera que la implementación del modelo de Machine Learning permita un aumento en la variación porcentual de los ingresos anuales generados por la compañía.

$$Variación\ de\ ingresos\ anuales = \left(\frac{ingresos\ anuales\ \ periodo_n}{ingresos\ anuales\ \ periodo_{n-i}} - 1\right) * 100$$

3. Variación del monto total de compras de los usuarios:

Un incremento en la eficacia de la campaña publicitaria implicaría un aumento de compras por parte de los usuarios repercutiendo así en un aumento en el monto total anual gastado.

$$Variación\ \ promedio\ del\ monto\ de\ compras\ de\ los\ usuario\ = \frac{\sum_{i=1}^{n}\left(\frac{monto\ total\ de\ compras\ usuario_i\ periodo_t}{monto\ total\ de\ compras\ usuario_i\ periodo_{t-i}} - 1\right)}{n} * 100$$

$$Donde\ n = \ numero\ de\ usuarios$$

2.4.3 Valor mínimo de las métricas

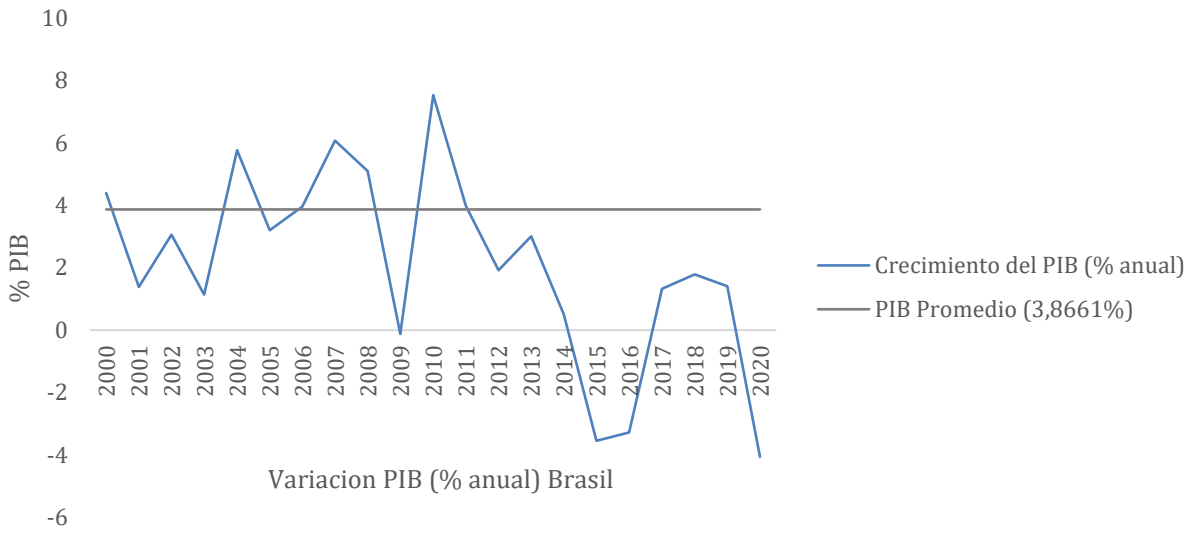
Tanto para la tarea de regresión como la de clasificación, se define que el valor mínimo de R^2 o de Curva ROC requerido para considerar el modelo de Machine Learning apto para implementarse en producción es del 80% en *test*.

Si se toma como referencia el PIB promedio de los últimos 20 años disponibles (2000, 2020) en el Banco Mundial en Brasil, el país ha tenido un PIB promedio anual del $\bar{x} = \ 3.8661\%$ (ver Figura 1) con una desviación $\sigma = 3{,}095\%$

Se espera que métrica de Machine Learning de llegar a ser superior a 80% en producción genere como mínimo un 10% adicional del valor $\bar{x} + \sigma$. Es decir, un aumento del 7.65% de **Variación de ingresos anuales** en el primer año

Figura 1

Variación PIB (% anual) en Brasil. Fuente: Banco Mundial



Nota. Datos tomados del Banco Mundial (2022)

A partir de la Figura 1, se concluye que el valor promedio del PIB de Brasil en los últimos 20 años oscila alrededor del 3.86%. Se puede observar que a partir del 2011 el valor del PIB anual se ha encontrado por debajo del valor promedio, alcanzando uno de sus valores más bajos en el año 2020 con un PIB cercano al -4%.

Se igual forma se observa que el PIB en Brasil ha presentado una tendencia bajista en la última década a excepción del repunte presentado durante los periodos comprendidos entre el 2017 y 2019, los cuales, sin embargo, se mantuvieron por debajo del valor promedio.

3. DATOS

3.1 DATOS ORIGINALES

El acceso a los datos se realizó directamente desde el API de Kaggle. Para esto, un token en formato *json* es generado desde Kaggle y almacenado en el repositorio de GitHub LuisPortela/ELO, el cual es clonado en los distintos notebooks de Google Colaboratory empleados para este proyecto. Esto permite que los datos sean importados directamente desde Kaggle al Notebook, evitando la manipulación de los archivos, y principalmente, minimizando los problemas de la lectura de datos generado por la importante volumetría de información suministrada en el concurso.

En la Tabla 1 se encuentra la descripción cuantitativa de las bases de datos suministradas por el concurso.

De igual forma, la descripción de las características que componen la bases de datos Historical Transactions y New Merchants Transactions se realizara en una misma tabla dado que las dos bases de datos contienen las mismas variables. La tabla descriptiva a asociada a Historical Transactions y New Merchants Transactions es la Tabla 2. Finalmente, las bases de datos Merchants y Train se describirán en la Tabla 3 y Tabla 4 respetivamente.

Tabla 1

Descripción cuantitativa de las bases de datos del concurso ELO de Kaggle.

Base de datos	Tamaño	Numero de registros	Numero de características
Historical Transactions	2,65 Go	29112361	14
New Merchant Transactions	181 Mo	1963031	14
Merchants	47,7 Mo	334696	22
Train	7,99 Mo	201917	5

Nota. Realización propria

Tabla 2

Descripción de las características de las bases de datos Historical Transactions y New Merchants Transactions.

Tipología de la característica	Característica	Descripción de la característica	Tipo de variable
Variables asociadas al comercio aliado	city_id	Es el identificador de la ciudad en la cual se asume la transacción con el comercio asociado fue realizada. Esta característica está anonimizada.	Categórica
	state_id	Es el identificador del estado en la cual se asume la transacción con el comercio asociado fue realizada. Esta característica está anonimizada.	Categórica
	subsector_id	Es el reagrupamiento de los comercios aliados en subsectores de actividad. De igual forma, el conjunto de subsectores ha sido anonimizado.	Categórica
	merchant_category_id	Es el identificador de la categoría del comercio aliado. Esta categoría ha sido anonimizada.	Categórica
Variables asociadas a la compra	month_lag	Es la diferencia temporal (en meses) que existe entre cada una de las transacciones efectuadas por el cliente con respecto a una fecha de referencia. Se asume que la fecha de referencia fue el momento en el cual el puntaje de la lealtad fue estimado.	Cuantitativa
	authorized_flag	Es la variable que representa si una transacción ha sido rechazada o aprobada.	Categórica
	purchase_date	Es la fecha en la cual cada una de las transacciones fue efectuada.	Temporal
	installments	Es el número de cuotas a las cuales fue diferido el pago de la transacción efectuada.	Cuantitativa
	Normalized purchase amount	Es el monto de la compra efectuada en cada una de las transacciones. El valor de la compra ha sido normalizado	Cuantitativa
	category_1	Es una categoría anónima asociada a cada transacción. Dos valores son posibles para esta variable: **N** o **Y**.	Categórica

Variables categóricas “anónimas”.	category_2	Es una categoría anónima asociada a cada transacción. Cinco valores son posibles para esta variable: **1**, **2**, **3 4**, o **5**.	Categórica
	category_3	Es una categoría anónima asociada a cada transacción. Tres valores son posibles para esta variable: **A**, **B** o **C**.	Categórica
Variable de identificación	card_id	Es el identificador único asociado a cada una de las tarjetas de crédito. Este identificador se encuentra represente en las bases de datos: Train, New Merchants Transactions et Historical Transactions	Categórica
	merchant_id	Es el identificador único asociado a cada uno de los comercios aliados en donde fueron efectuadas las compras de los usuarios. Este identificador se encuentra represente en las bases de datos: New Merchants Transactions, Historical Transactions y Merchants	Categórica

Nota: Información tomada de (Elo Merchant Category Recommendation | Kaggle, 2019)

Tabla 3

Descripción de las características de las base de datos Merchants

Tipología de la característica	Característica	Descripción de la característica	Tipo de variable
Variables de caracterización del comercio aliado.	merchant_group_id	Es el grupo al cual pertenece el comercio analizado. Esta característica ha sido anonimizada.	Categórica
	merchant_category_id	Es el identificador de la categoría del comercio. Esta categoría ha sido anonimizada.	Categórica
	subsector_id	Es el reagrupamiento de los comercios en subsectores de actividad. De igual forma, el conjunto de subsectores ha sido anonimizada.	Categórica

Variables geográficas	city_id	Es el identificador de la ciudad en la cual se asume que el comercio asociado se encuentra. Esta característica está anonimizada.	Categórica
	state_id	Es el identificador del estado en la cual se asume la transacción con el comercio asociado fue realizada. Esta característica está anonimizada.	Categórica
Variables categóricas "anónimas".	category_1	Es una categoría anónima asociada a cada comercio. Cinco valores son posibles para esta variable: **1**, **2**, **3 4**, o **5**.	Categórica
	category_2	Es una categoría anónima asociada a cada comercio. Cinco valores son posibles para esta variable: **1**, **2**, **3 4**, o **5**.	Categórica
	category_4	Es una categoría anónima asociada a cada comercio. Dos valores son posibles para esta variable: **N** o **Y**.	Categórica
Variables numéricas	numerical_1	Es una medida cuantitativa anonimizada y normalizada asociada a cada comercio.	Cuantitativa
	numerical_2	Es una medida cuantitativa anonimizada y normalizada asociada a cada comercio.	Cuantitativa
Variables relacionadas con las ventas	most_recent_sales_range	Es el rango de ingresos (en unidades monetarias) en el último mes activo. Cinco valores son posibles para esta variable **A**, **B**, **C**, **D** o **E**	Categórica
	avg_sales_lag3	Es el promedio mensual de los ingresos en los últimos tres meses lo cuales son divididos por los ingresos en el último mes activo	Cuantitativa
	avg_sales_lag6	Es el promedio mensual de los ingresos en los últimos seis meses lo cuales son divididos por los ingresos en el último mes activo	Cuantitativa
	avg_sales_lag12	Es el promedio mensual de los ingresos en los últimos doce meses lo cuales son divididos por los ingresos en el último mes activo	Cuantitativa
	most_recent_purchases_range	Es el rango de ingresos (en unidades monetarias) en el último mes activo. Cinco valores son posibles para esta variable **A**, **B**, **C**, **D** o **E**	Categórica

Variables relacionadas con las compras	avg_purchases_lag3	Es el promedio mensual de las transacciones en los últimos tres meses lo cuales son divididos por el total de transacciones en el último mes activo	Cuantitativa
	avg_purchases_lag6	Es el promedio mensual de las transacciones en los últimos seis meses lo cuales son divididos por el total de transacciones en el último mes activo	Cuantitativa
	avg_purchases_lag12	Es el promedio mensual de las transacciones en los últimos seis meses lo cuales son divididos por el total de transacciones en el último mes activo	Cuantitativa
Variables temporales preprocesadas	active_months_lag3	Es la cantidad de meses activos dentro de los últimos tres meses.	Cuantitativa
	active_months_lag6	Es la cantidad de meses activos dentro de los últimos seis meses.	Cuantitativa
	active_months_lag12	Es la cantidad de meses activos dentro de los últimos doce meses.	Cuantitativa
Variable de identificación	merchant_id	Es el identificador único asociado a cada uno de los comercios aliados en donde fueron efectuadas las compras de los usuarios. Este identificador se encuentra represente en las bases de datos: New Merchants Transactions, Historical Transactions y Merchants	Categórica

Nota: Información tomada de (Elo Merchant Category Recommendation | Kaggle, 2019)

Tabla 4

Descripción de las características de las base de datos Train.

Tipología de la característica	Característica	Descripción de la característica	Tipo de variable
Variables categóricas "anónimas".	feature_1	Es una categoría anónima asociada a cada tarjeta de crédito. Cinco valores son posibles para esta variable: **1**, **2**, **3**, **4** o **5**.	Categórica
	feature_2	Es una categoría anónima asociada a cada tarjeta de crédito. Tres valores son posibles para esta variable: **1**, **2** o **3**.	Categórica
	feature_3	Es una categoría anónima asociada a cada tarjeta de crédito. Dos valores son posibles para esta variable: **0** o **1**.	Categórica
Variables temporales	first_active_month	Es una variable temporal que indica la fecha de la primera compra realizada por una tarjeta de crédito	Temporal
Variables de identificación	card_id	Es el identificador único asociado a cada una de las tarjetas de crédito. Este identificador se encuentra represente en las bases de datos: Train, New Merchants Transactions et Historical Transactions	Categórica
Variable para predecir	target	Es el puntaje de la lealtad de los clientes	Cuantitativa

Nota: Información tomada de (Elo Merchant Category Recommendation | Kaggle, 2019)

3.2 DATASETS

Considerando que se dispone de múltiples bases de datos con niveles de agregación diferentes entre ellos, la etapa de construcción de la base de datos final (*features engineering*) se llevó a cabo en distintas etapas, no solo con el fin de constituir un único dataset etiquetado para el modelo supervisado, sino también para evaluar la pertinencia de las características en el algoritmo predictivo.

Para alcanzar este objetivo, se llevaron a cabo las siguientes etapas.

1. Filtrado de las bases de datos originales
2. Agrupamiento de las bases de datos desagregadas
3. Evaluación de la pertinencia de las bases de datos anexas

3.2.1 Filtrado de las bases de datos originales

En la sección *3.1 Base de datos originales* se evidencia que las bases de datos de las que se dispone presentan una volumetría importante, principalmente en el dataset Historical Transactions, la cual contiene aproximadamente 30.000.000 de registros sobre las transacciones de los clientes. Sin embargo, durante el análisis exploratorio de los datos (EDA), se encontró que las distintas bases de datos no contienen estrictamente la misma información para el conjunto de tarjetas de crédito, lo cual implica que alguno de los escenarios siguientes se hubiera presentado:

- **Base de datos no etiquetada:** Este escenario sería posible cuando se dispone de la información de una misma Card ID tanto en las bases de datos Historical Transactions como en New

Merchants Transactions, lo cual permitiría la creación de las características agregadas, sin embargo, no se contaría con la información en la base de datos Train, el cual es el dataset en donde se encuentra la variable "*lealtad"* a predecir.

- **Base de datos con características faltantes:** Este escenario se presentaría cuando una Card_ID esté presente tanto en Train como en una de las dos bases de datos desagregadas (Historical Transactions o en New Merchants Transactions), pero faltante en la otra.

Con el objetivo de eliminar estos posibles escenarios, se realizó un filtrado de las tarjetas de crédito (Card_ID) existentes en todas las bases de datos de donde se extraerán las características de la base de datos final (Historical Transaction, New Merchants Transactions y Train) a través de un inner join.

Este acercamiento en el procesamiento de los datos no sólo permitió la eliminación de los escenarios asociados a información faltante descritos anteriormente, sino también redujo la dimensionalidad de las bases de datos, conservando únicamente los registros explotables en los algoritmos supervisados que serán entrenados posteriormente, pero además permitió la disminución de los recursos computacionales necesarios para el desarrollo analítico del proyecto. La Tabla 5 muestra la dimensión resultante del proceso de filtrado.

Tabla 5

Dimensiones de las bases de datos luego de la etapa de filtrado

Base de datos	Número de registros antes	Número de registros después	Reducción porcentual de registros
Historical transactions	29112361	16782359	42,3%
New Merchant Transactions	1963031	1219685	37,8%
Train	201917	179986	10,8%

Nota: Realización propia

3.2.2 Agrupamiento de las bases de datos desagregadas

La siguiente etapa en el tratamiento de los datos brutos suministrados por el concurso, después del filtrado descrito en la etapa precedente, es el agrupamiento de los datos desagregados.

Como se explicó en la sección *3.1 Base de datos originales*, las bases de datos Historical Transactions y New Merchants Transactions contienen información desagregada de las transacciones comerciales realizadas por los usuarios de las tarjetas de crédito en los comercios aliados de ELO. Por esta razón, se aplicaron funciones de agrupamiento sobre las características con el fin de obtener una medida de resumen estadística como valor agregado para cada variable. “El cálculo eficiente de un puñado de agregados agrupados sobre una unión de conjuntos de datos está bien respaldado por sistemas académicos y comerciales maduros y también ampliamente investigado” (Schleich, Olteanu, Abo, Ngo, & Nguyen,2019)

Considerando que las bases de datos desagregadas, Historical Transactions y New Merchants Transactions, contienen las mismas variables, se aplicaron las funciones de agregación de forma simétrica, es decir que las mismas medidas implementadas para la $característica_i$ de Historical Transactions fueron de igual forma implementadas para la $característica_i$ de New Merchants Transactions

Las medidas de agrupamiento consideradas dependen del tipo de variable que se debe agregar. En la Tabla 6 se presentarán las medidas de agrupamiento implementadas sobre las características según la naturaleza de la variable.

Tabla 6

Medidas de agrupamiento empleadas sobre los datasets desagregados

Tipo de variable	Funciones de agregación	Ejemplo de aplicación
Categóricas	Moda	$\widehat{Category}_2$
Continuas	Suma	$\sum_{i=1}^{n} Purchase\ amount$
	media	$\bar{X}$ $donde\ X = Purchase\ amount$
	desviación estándar	σ_X $donde\ X = Purchase\ amount$
	valor máximo	$\max\ (Purchase\ amount)$
Temporales	valor máximo	$\max\ (Purchase\ date)$
	valor mínimo	$\min\ (Purchase\ date)$

Nota: Realización propia

En la Tabla 6 se observan las distintas funciones de agregación implementadas con el fin de agrupar en una sola medida de resumen estadístico, la información desagregada de las cards_ID. En la columna *Ejemplo de aplicación,* se observa a título de ejemplificación algunos tipos de variables sobre los cuales se aplicó alguna de las funciones mencionadas.

En el caso de las variables de tipo categórico, la medida de agregación aplicada es la moda de la variable ($\hat{X}$). En este caso, la moda se obtuvo para las variables $category_2$ y $category_3$ tanto para las bases de datos Historical Transactions como New Merchants Transactions.

Para el caso de las variables continuas, las más numerosas dentro del conjunto de datos, múltiples medidas de agregación pudieron ser estimadas para una misma característica. En este caso, en el ejemplo ilustrado en la Tabla 6, la variable $purchase\ amount$ fue resumida a partir de los estadísticos siguientes: la media ($\bar{X}$), la desviación estándar (σ) y el valor máximo de la distribución (max).

Finalmente, para el tercer tipo de variable expuesta en la Tabla 6, las medida de agregación aplicadas fueron el valor mínimo y máximo de la serie temporal. Esto se realizó como etapa intermedia para la creación de nuevas características que determinaran la diferencia temporal entre los valores mínimos y máximos encontrados.

3.2.3 Evaluación de la pertinencia de las bases de datos anexas

Finalmente, la última etapa para la constitución de la base de datos final es la de evaluar la pertinencia de incluir el dataset anexo Merchants, el cual comprende la información sobre los comercios aliados que fueron visitados tanto en Historical transactions como en los nuevos comercios presentes en New Merchants Transactions.

Para determinar si era pertinente anexar la información de Merchants desde un punto de vista analítico, se tuvieron en cuenta dos criterios. El primer es si la inclusión de Merchants presentaba una fuerte correlación con la variable de respuesta a predecir. El segundo criterio considerado es si la presencia de esta información adicional generaba un incremento en la métrica de Machine Learning seleccionada para la tarea de regresión: R^2. Para esto se llevó a cabo la siguiente metodología:

1. Implementación de un $algoritmo_i$ no supervisado de clustering sobre los comercios aliados
2. Introducción y evaluación de los clusters generados por el $algoritmo_i$ en el modelo principal

3.2.3.1 Implementación de un $algoritmo_i$ de clustering.

La forma como se abordó el tratamiento de la base de datos anexa Merchants fue clusterizando los comercios aliados y verificando la pertinencia de los clusters generados. En este sentido, se puede entender la pertinencia de los clusters desde dos enfoques:

- Como la estimación del número de clusters óptimo que logre minimizar la métrica de error seleccionada. Con esto se busca seleccionar el mejor algoritmo no supervisado desde un enfoque analítico.
- Como la mejoría de la métrica de Machine Learning del modelo principal. Esto se hace con el fin de determinar si un $algoritmo_i$ es más adecuado desde un enfoque de la contribución del clúster en la base de datos final. Este ítem se ahondará en la sección 3.2.3.2

Para esto, se seleccionaron dos tipos de algoritmos de clustering: K Means y GMM-EM (Gaussian Mixture Models). “El método de k-medias es una técnica de agrupamiento ampliamente utilizada que busca minimizar el promedio distancia al cuadrado entre puntos en el mismo grupo”. (Arthur& Vassilvitskii, 2006). Por su parte “el agrupamiento del modelo de mezcla gaussiana (GMM) pertenece al algoritmo basado en la distribución. Utiliza la suma ponderada de varias funciones de distribución gaussiana para estimar la distribución de densidad de probabilidad de las muestras y el resultado de la agrupación es maximizar la densidad de probabilidad de las muestra. (Shi,He & Wands g 2019)

Considerando que cada algoritmo posee distintas métricas propias que permiten determinar la calidad del agrupamiento, se implementa la búsqueda del mejor hiperparámetro (número de clusters) según la combinación algoritmo-métrica de error seleccionada. La Tabla 7 muestra los resultados del número de clusters óptimo según la métrica y algoritmo implementado.

Tabla 7.

Número de clusters óptimos según la combinación algoritmo-métrica para el dataset Merchants

Algoritmo	Métrica	Número de clusters	Score
K Means	Inertia	8	450000
	Davies Bouldin	3	0.70
	Silhouette	2	0.95
GMM-EM	BIC AIC	22	-1.824790e+07 -1.847134e+07
	Davies Bouldin	18	2.10

Nota: Realización propia

Considerando que el score de las métricas listadas en la Tabla 7 es difícilmente comparable entre ellas, y además, debido a la variabilidad en el número de clusters óptimo de cada combinación algoritmo-métrica se realizó una evaluación exhaustiva de todos los resultados de los clusters óptimos de los algoritmos evaluados.

3.2.3.2 Introducción de los clusters generados en el modelo principal.

Como se esbozó previamente, una vez identificado el número de clusters óptimo para una combinación algoritmo-métrica dada, se emprendió un proceso iterativo de evaluación del performance de los clusters generados, comparándolos con un modelo base. En este proceso, inicialmente se realizó un modelo de regresión lineal, con las características resultantes de la etapa 3.2.2 Agrupamiento de las bases de datos desagregadas.

La elección del algoritmo de regresión lineal como base comparativa se hizo debido que para la primera iteración se consideró este modelo. Adicionalmente, durante la primera iteración no se consideró la base de datos Merchants, ni ningún algoritmo no supervisado que considerara la información relativa

a los comercios aliados. Sin embargo, es de esperarse que el resultado de la métrica de Machine Learning en el modelo base (sin clusters) no sea estrictamente igual al obtenido durante la primera iteración, dado que en esta se definieron nuevas características que no habían sido contemplabas previamente.

Una vez determinada la métrica de Machine Learning de la tarea de regresión (R^2) del modelo base con el conjunto de características obtenidas en la etapa 3.2.2 Agrupamiento de las bases de datos desagregadas, se incluyó iterativamente los clusters obtenidos a partir de la combinación algoritmo-métrica de la sección anterior.

Para cada iteración se obtuvo el R^2 del nuevo modelo considerando los clusters del $algoritmo_i$ y se estimó el coeficiente de correlación de cada uno de los clusters con respecto a la variable a predecir.

La Tabla 8 sintetiza la evaluación de los clusters bajo el criterio tanto del R^2 como el del valor promedio de la magnitud de la correlación de lo clusters $|\bar{\rho}|$:

Tabla 8

Resultados del performance de los clusters óptimos según los criterios definidos

Algoritmo	Métrica	Número de clusters	R^2 del modelo	$\|\bar{\rho}\|$ de los clusters respecto a target
Modelo base (sin clusters)		NA	8.185%	NA
K Means	Inertia	8	8.183%	0.0062
	Davies Bouldin	3	8.188%	0.0070
	Silhouette	2	8.185%	0.0090
GMM-EM	BIC AIC	22	8.204%	0.0054
	Davies Bouldin	18	8.197%	0.0049

Nota: Realización propia

A partir de los resultados obtenidos en la Tabla 8 se puede concluir que la introducción de los clusters obtenidos a partir de los $algoritmo_i$ descritos previamente no tendrían un aporte significativo en la base de datos final, dado que los resultados de R^2 no aumentan considerablemente con respecto al del modelo base. De igual forma, la magnitud promedio de la correlación con respecto a la variable target a predecir de los distintos clusters ($|\bar{\rho}|$) es casi nula.

Por esta razón, la base de datos final será aquel que se obtuvo al final de la etapa 3.2.2 Agrupamiento de las bases de datos desagregadas. Es decir, sin la introducción de los clusters asociados al dataset Merchants.

3.3 DESCRIPTIVA

La base de datos resultante de la etapa anterior está constituida finalmente por 25 características más las variables de respuesta Target, tanto en su forma continua para la tarea de regresión, como en su forma categórica producto de la discretización de la variable para la tarea de clasificación.

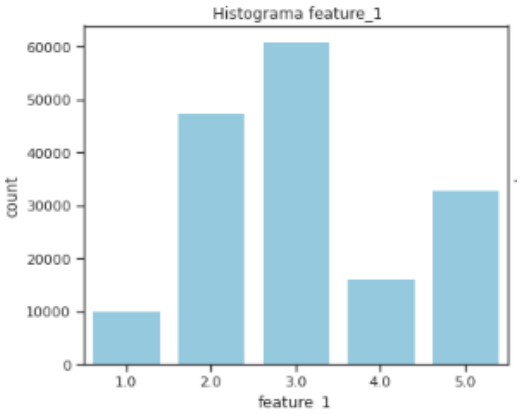
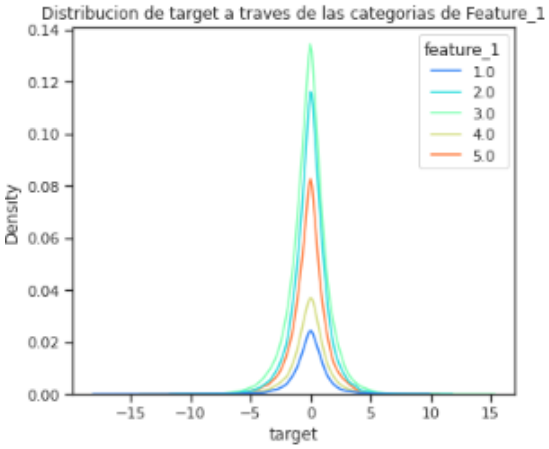
De las 25 variables explicativas presentes, 7 de ellas (es decir, el 28%) son categóricas. EL 72% restante, (18 características) son variables continuas.

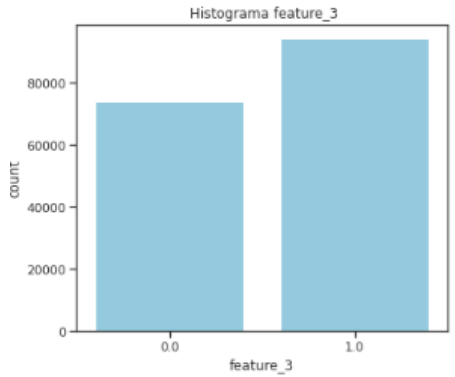
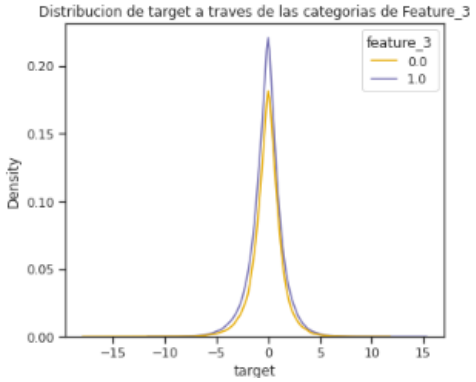
A continuación, se hará una breve descripción de algunas de las características que constituyen la base de datos según el tipo de variables de la siguiente manera:

1. Análisis descriptivo univariable y bivariable de ciertas de características categóricas. Este análisis se desarrollará en la Tabla 9
2. Análisis descriptivo univariable y bivariable de ciertas de características numéricas. Este análisis se desarrollará en la Tabla 10
3. Análisis descriptivo univariable de Target (tanto en forma continua como categórica). Este análisis se desarrollará en la Tabla 11

Tabla 9

Análisis descriptivo univariable y bivariable de ciertas de características categóricas

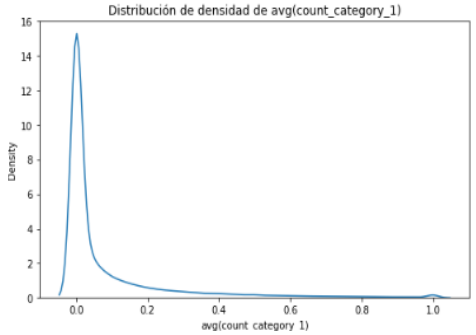
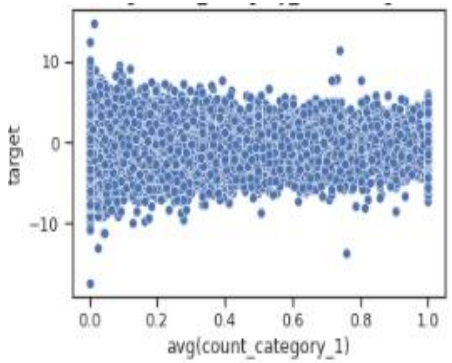
Variable	Distribución	Análisis
Feature 1		La variable categórica Feature 1, está compuesta por 5 niveles que van desde 1 hasta 5, representando así algunos atributos propios de la tarjeta de crédito. A partir del análisis univariable, se constata que el nivel 3 de esta característica, es el más frecuente dentro del conjunto de datos, representando un poco más de 60.000 tarjetas de crédito, de las 167758. Es decir, aproximadamente el 35% del total de registros. Inmediatamente después, le sigue el nivel "**2**", el cual representa alrededor del 30% de las muestras. Finalmente, se observa que los niveles menos frecuentes son los niveles 1 y 4, los cuales juntos representan tan solo el 15% de las tarjetas, contribuyendo cada uno con el 5% y 10% respectivamente.
Feature 1 a través de Target		El análisis bivariable se realiza comparando la densidad de los distintos niveles de la característica Feature 1 a través de la variable de respuesta Fidelidad (continua) con el objetivo de determinar si existe una variación en la variable respuesta según alguno de los niveles de esta característica. A partir de la visualización, se logra concluir que no existe una variación o cambio en el comportamiento de Target a partir de alguno de los niveles de Feature 1, dado que las distintas curvas de densidad asociadas a cada nivel se distribuyen aproximadamente en los mismos intervalos de Target (representada en el eje X), indistintamente del nivel. La sola variación se observa en la amplitud de la densidad, la cual varía según el número de muestras asociadas a cada nivel, en donde la densidad más alta se asocia al nivel 3, y la más baja al nivel 1, hecho que era previsible desde el análisis univariable siendo estos dos niveles, el más alto y bajo respectivamente. Se puede concluir gráficamente, que la característica Feature 1 no tendría una influencia en la predicción de la fidelidad de los clientes.

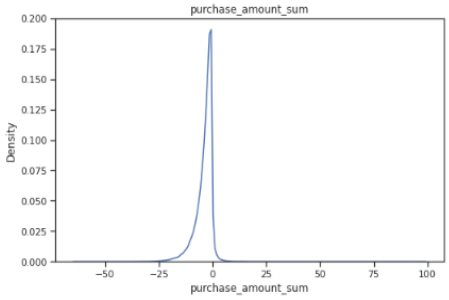
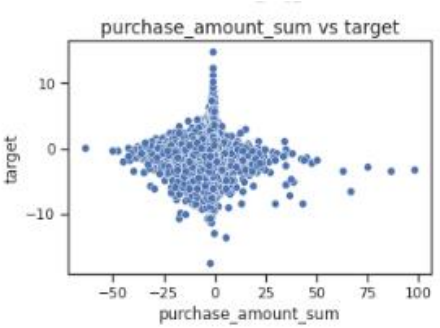
Feature 3		La característica Feature 3, la cual describe uno de los atributos de las tarjetas de crédito, a mismo título que Feature 1 analizada anteriormente, está compuesta únicamente por dos niveles. El nivel 0 y 1. A partir del diagrama de barras en el cual se visualiza el conteo de las tarjetas de crédito para los niveles de Feature 3, se constata que el nivel más frecuente es el 1, tendiendo aproximadamente 93957 registros, representando el 56% del total de registros. Por su parte, el nivel 0 representa el complemento, es decir el 44% de registros restantes. Se puede concluir entonces que no existe una diferencia abrupta en la repartición de la data entre los dos niveles de esta variable.
Feature 3 a través de Target		El análisis bivariable realizado a partir de la densidad de cada uno de los niveles respeto a la variable de respuesta Fidelidad (continua), permite concluir (de la misma manera que para la característica precedente “features 1”) que no existe una variación en el comportamiento de la variable Target, a la presencia de un nivel de la característica Feature 3. Esto se constata gráficamente al visualizar que las densidades de los niveles 0 y 1 tienen una distribución que prácticamente se traslapan entre ellas, a la única excepción de la amplitud de la curva, la cual es más pronunciada para el nivel 1, al tener mayor número de registros en la base de datos. Se puede concluir a partir del análisis exploratorio que no se espera que esta variable tenga una influencia en la predicción de la variable respuesta, dado que los cambios en sus niveles no generan variaciones en los valores de la distribución de Target (a través del eje X)

Nota: Realización propia

Tabla 10

Análisis descriptivo univariable y bivariable de ciertas de características numéricas

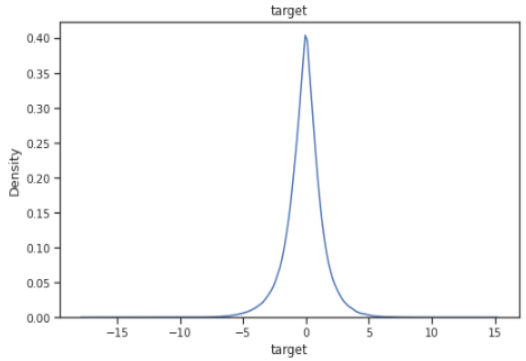
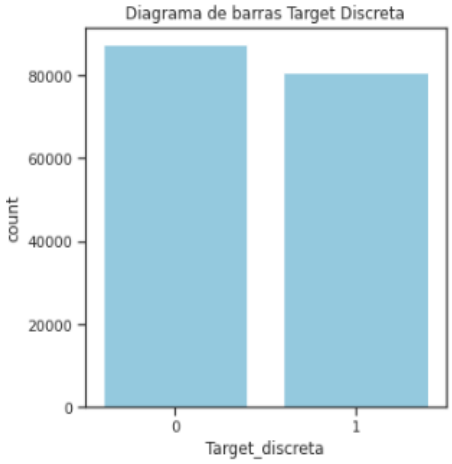
Variable	Distribución	Análisis
Avg (count_category_1)		La variable Avg (count_category_1) es una de las medidas de agregación que fueron creadas durante el agrupamiento de las bases de datos desagregadas. Esta característica, proveniente de la base de datos "Historical Transactions" se define como la proporción promedio de las tarjetas de crédito que se encuentran en la categoría "Y". Esta característica inicialmente tenía dos posibles niveles ("Y" y "N"). En este sentido, es el promedio de la proporción de las tarjetas que estaban asociadas al nivel "Y" únicamente. Se observa a partir de la densidad, que esta variable tiene una distribución asimétrica positiva y leptocúrtica, cuyos valores más frecuentes se encuentran en un intervalo de 0 y 0.2. Esto implica que la gran mayoría de las tarjetas de crédito de la base de datos Historical Transactions tenía el nivel "N" asociado a este atributo.
Feature 1 a través de Target		El análisis bivariable se realiza con respecto a la variable target, a través de un scatter plot, con el fin de determinar gráficamente la correlación existente entre esta característica y la variable de respuesta (continua). A partir del gráfico de dispersión se visualiza que los distintos pares ordenados de los valores de Avg (count_category_1) y target no forman algún tipo de tendencia lineal u otro tipo de comportamiento polinómico. En este caso, los pares ordenados se distribuyen en una nube de puntos a lo largo del gráfico. Esto implica que indistintamente a los valores de la variable Avg (count_category_1), la variable target puede tener el mismo rango posible de valores. Este hecho logra entrever exploratoriamente que la variable Avg (count_category_1) no está correlacionada con la variable respuesta, y por ende, es poco probable que sea una característica importante, analíticamente hablando, para las futuras predicciones.

Purchase amount sum		Purchase amount sum es otra variable inicialmente desagregada que fue construida durante el agrupamiento de los datasets. La variable "bruta" proviene de la base de datos "Historical Transactions" y representa la suma de las distintas transacciones efectuadas por una tarjeta de crédito durante el periodo de 3 meses. Considerando que la variable Purchase amount es una variable que había sido normalizada, se encuentra a partir de la distribución de su densidad, que la magnitud de los valores oscila alrededor de 0, dando la impresión de seguir una distribución normal estándar. Sin embargo, se logra observar que la media de esta distribución no se encuentra centrada en 0, sino en un valor negativo cercano a este, dejando interpretar que en promedio, fue más grande la magnitud de los montos de compra negativos que positivos. Adicionalmente, si bien hubo un tratamiento de valores atípicos antes del análisis exploratorio, se evidencia que aún existen ciertos outliers en el dominio negativo de la variable, sesgando ligeramente la distribución.
Purchase amount sum a traves de Target		El análisis bivariable de Purchase amount Sum (suma de los montos de compra) se realiza de igual forma con respecto a la variable Target para determinar la posible correlación entre este par de variables continuas. Visualmente, se evidencia que no existe algún tipo de correlación, ya sea lineal o polinómica, entre estas características. Los datos se reagrupan en una nube de puntos, encontrando su centro en el par ordenado cercano a (0,0) dado que ambas características están normalizadas

Nota: Realización propia

Tabla 11

Análisis descriptivo univariable de Target (tanto en forma continua como categórica)

variable	Distribución	análisis
Target Continua		A partir del gráfico univariable de "Fidelidad" como variable de respuesta continua, se evidencia que la distribución de esta característica es perfectamente simétrica, centrada en 0 y con colas robustas, lo cual indica la presencia de datos atípicos residuales del proceso de eliminación de outliers llevado a cabo en el preprocesamiento de los datos. Como se expuso previamente, la variable Target "Fidelidad", es un score normalizado cuyo signo define el tipo de lealtad. En este sentido, se puede deducir, dado que el concurso no lo precisa, que entre mayor sea la magnitud de esta variable, más fuerte será la lealtad a condición de que esta sea positiva.
Target Discreta		En la sección 2.2 APROXIMACIÓN DESDE LA ANALÍTICA DE DATOS se definió que desde la analítica se iba a abordar este problema de Machine Learning a partir dos perspectivas, siendo una de estas, el de una tarea de clasificación. Para lograrlo, se discretizó la distribución continua de la lealtad teniendo como punto de referencia el signo asociado a cada valor de la lealtad. Esto permitió la creación de dos clases siguiendo el criterio Si $lealdad_i < 0\ then\ clase_0\ otherwise\ clase_1$. En el diagrama de barras asociado a la variable Fidelidad en su forma categórica, se observa que la discretización de la variable continua a una variable biclase permite una repartición balanceada del dataset, atribuyendo 87246 registros para la clase 0 (clientes "infieles") y 80512 para la clase 1 (clientes fieles) representando el 52% y 48% respectivamente.

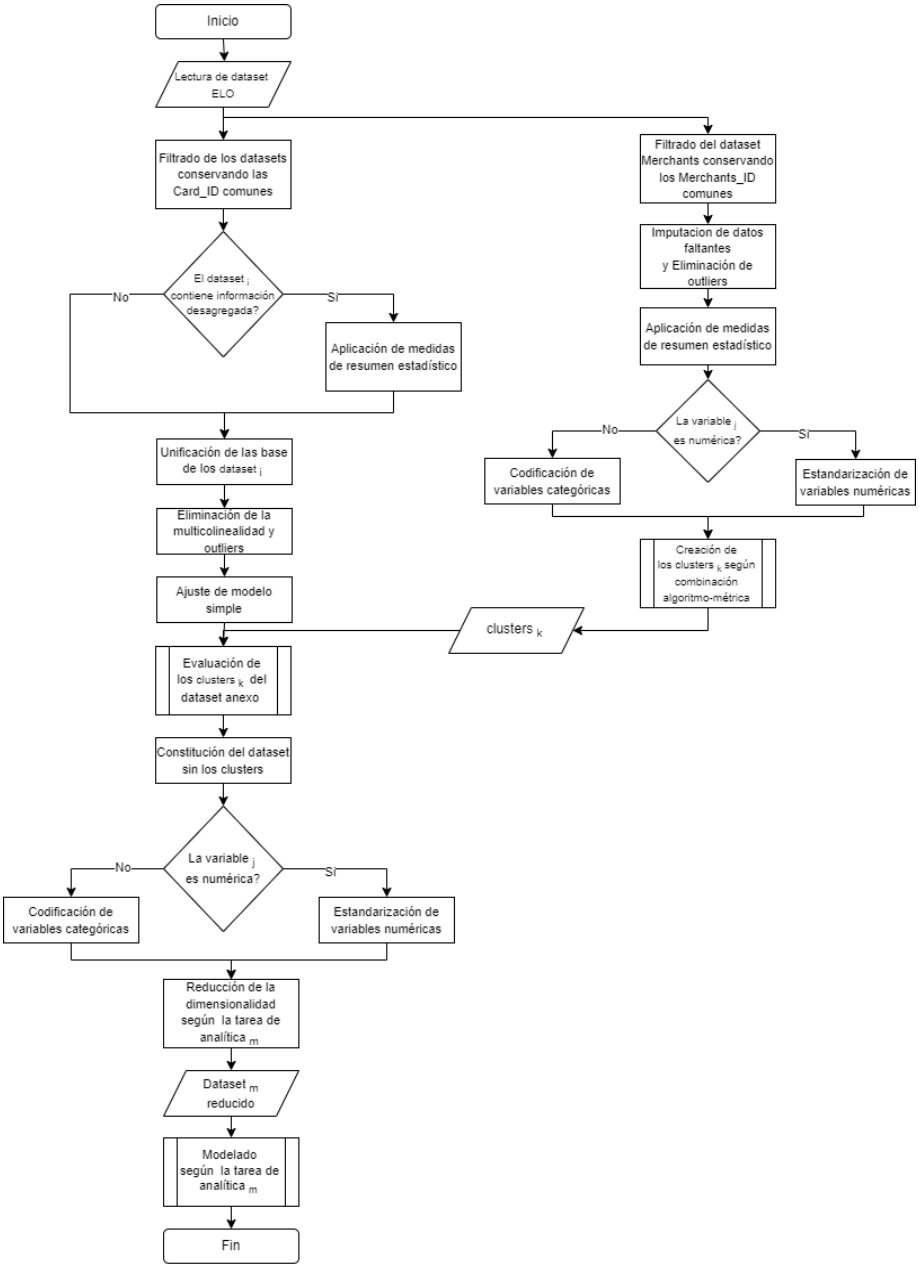
Nota: Realización propia

4. PROCESO DE ANALÍTICA

4.1 PIPELINE PRINCIPAL

Figura 2

Pipeline principal del proyecto



Nota. Realización propia

En la Figura 2 se describe la secuencia de las etapas llevadas a cabo en el proyecto. En esta figura se logran identificar visualmente dos estructuras secuenciales. La primera estructura, siendo esta el diagrama de flujo más extenso se encuentra a la izquierda de la figura y describe las etapas de preprocesamiento de datos y modelamiento para las tareas de analítica planteadas. Es decir, este es la descripción del pipeline principal.

Por otra parte, el diagrama de flujo ubicado a la derecha de la figura describe las grandes etapas llevadas a cabo en el tratamiento de la base de datos anexa Merchants, a la cual se ajustaron múltiples algoritmos no supervisados de clustering con el fin de introducirlos dentro del pipeline principal. El proceso descrito dentro de la Figura 2 como la *Evaluación de los* $clusters_k$ *del dataset anexo* presente en el pipeline principal (flujograma a la izquierda de la figura), es el punto de intersección de estos procesos de analítica llevados a cabo paralelamente. Durante este proceso se evaluó la pertinencia de los $clusters_k$ según los criterios expuestos en la sección 3.2.3 Evaluación de la pertinencia de las bases de datos anexas.

A partir de esta etapa, se evidencia gráficamente que los procesos de analítica se unifican en un solo pipeline. Las etapas posteriores a la intersección describen la continuación del preprocesamiento sobre el conjunto de datos y por último, el modelamiento de los algoritmos de Machine Learning sobre el dataset final.

4.2 PREPROCESAMIENTO

El preprocesamiento implementado se puede resumir en las etapas siguientes:

1. Reducción de la multicolinealidad
2. Eliminación de outliers
3. Imputación de datos faltantes
4. Estandarización y codificación de variables

5. Reducción de la dimensionalidad del dataset

4.2.1 Reducción de la multicolinealidad

La primera etapa del preprocesamiento de los datos tiene como objetivo el de reducir la multicolinealidad (o correlación entre las variables explicativas) de la base de datos. Considerando que durante la realización de la etapa 3.2.2 Agrupamiento de las bases de datos desagregadas se efectuaron múltiples medidas de resumen estadístico sobre una misma variable, principalmente en el caso de las variables continuas (ver Tabla 6) era de esperarse que algunas de estas nuevas características agrupadas estuviesen correlacionadas entre sí.

Con el fin de evitar este problema, la estrategia implementada fue la de identificar y eliminar las características que más estuviesen provocando la multicolinealidad en el set de datos. (Vega & Guzmán, 2011). El criterio de identificación de multicolinealidad entre las variables independientes se definió como $|\rho| \geq 0.8$ dado que en la literatura, los pares de variables con coeficiente de correlación $|\rho| \geq 0.8$ son definidos como una correlación lineal "fuerte" ya sea negativa como positivamente. (Lahura,2003).

La verificación de la multicolinealidad se realizó en un proceso de dos iteraciones. La primera iteración se llevó a cabo con el fin de identificar aquellas variables que más estaban correlacionadas con los otros pares de variables. La segunda etapa se realizó con el fin de controlar si después de la eliminación de las variables encontradas durante la primera iteración, las características restantes seguían presentado multicolinealidad. En la primera iteración se encontraron 9 características entre las cuales se encontraba la variable $sum(purchase\_amount)$. La variable $sum(purchase\_amount)$ está correlacionada con las características $max(purchase\_amount)$, $stddev\_samp(purchase\_amount)$ y $avg(purchase\_amount)$ provenientes de la base de datos Historical Transactions. En este sentido, la

eliminación de la característica $sum(purchase\_amount)$ permitió resolver el problema de multicolinealidad existente con las otras tres variables permitiendo conservarlas en el set de datos.

La segunda iteración de verificación de multicolinealidad se llevó a cabo bajo los mismos criterios de la primera iteración, es decir una frontera de decisión equivalente a $|\rho| \geq 0.8$. Contrario a lo que se esperaba, el número de características clasificadas con multicolinealidad es superior en la segunda etapa que en la primera, presentando 12 características.

Finalmente, al eliminar las variables encontradas durante la segunda etapa, una última verificación se llevó a cabo. En esta verificación final se evidencia que la correlación máxima entre los pares de variables independientes del set de datos es inferior al criterio definido para determinar la multicolinealidad ($|\rho| < 0.8$) permitiendo considerar como finalizada esta etapa.

4.2.2 Eliminación de outliers

> "La detección de valores atípicos es la identificación de objetos, eventos u observaciones que no se ajustan a un patrón esperado u otros elementos en un conjunto de datos. Como una de las tareas importantes de la minería de datos, la detección de valores atípicos se usa ampliamente. El método de detección de valores atípicos locales basado en la densidad puede resolver eficazmente los problemas anteriores al describir el grado de valores atípicos de los puntos de datos cuantificados por la densidad local. Local Outlier Factor calcula el valor atípico de la medida de densidad relativa de cada punto de datos relativo a sus puntos circundantes, llamado valor lof, que se utiliza para describir el grado de valor atípico en los datos." (Cheng, Zou, & Dong, 2019)

La eliminación de outliers se llevó a cabo tanto en la base de datos resultante de la etapa 3.2.2 Agrupamiento de las bases de datos desagregadas como en el dataset anexo Merchants para asegurar que la conformación de los clusters no estuviese penalizada por la presencia de datos atípicos.

La estrategia de eliminación de outliers se realizó por medio de un algoritmo no supervisado LOF (Local Outlier Factor). Este algoritmo, tiene como hiperparámetro el número de vecinos (n_neighbors) el cual debe ser definido. Para esto, inicialmente se identificó el valor óptimo de este hiperparámetro.

El número de vecinos óptimo se determinó por medio de un proceso iterativo, en el cual se evaluaron distintos valores impares definidos en un rango inferior a 12. Paralelamente, considerando que durante el análisis exploratorio de los datos se encontró que los distintos datasets contenían un número importante de registros atípicos, se definió como criterio de selección del número de vecinos óptimo aquel que identificara el mayor número de outliers.

En la Tabla 12 se muestra el número de muestras atípicas detectadas por el n_neighbors óptimo según la base de datos analizada.

Tabla 12

Registros atípicos de las bases de datos

Base de datos	n_neighbors	Muestras atípicas detectadas	Porcentaje del dataset
Merchants	3	2294	3,67%
Base de datos resultante de la etapa 3.2.2 Agrupamiento de las bases de datos desagregadas	3	11413	6,34%

Nota: Realización propia

Algunas de las muestras del conjunto de datos consideradas como atípicas fueron las muestras superiores al valor 1828 de la variable $sum(installements)$. En este sentido, antes de la utilización del algoritmo no supervisado LOF, el valor máximo de esta variable es 2413. Después de aplicar el algoritmo se observa que el valor máximo de la variable $sum(installements)$ disminuye, llegando a tener como valor máximo 1828, presentando una disminución del 24% en el valor máximo de la variable. En este

sentido, las muestras existentes dentro del dataset cuyo valor en la variable $sum(installements)$ fuera superior o igual a 1828 fueron considerados como outliers potenciales dentro del conjunto de datos y por consiguiente eliminadas del dataset.

Por otra parte, otras muestras consideradas como atípicas por el algoritmo LOF fueron aquellos registros cuyo valor en la variable $ratio\_purchase\_amount$ eran inferiores a -239. En este sentido, el valor mínimo de la variable antes aplicar el algoritmo LOF era de -757.64. La presencia de estos datos extremos generaba una distribución con colas robustas debido a la existencia registros tan alejados del valor central de la distribución, cuya mediana se posiciona alrededor del 0. En este sentido, las muestras cuyos valores en la variable $ratio\_purchase\_amount$ eran inferiores a -239.4, fueron considerados como atípicas.

Considerando los resultados de la Tabla 12, se logra identificar que la etapa de identificación y posterior eliminación de datos atípicos generó una reducción de la dimensionalidad de los dataset tanto para la base de datos Merchants como para base de datos resultante de la etapa 3.2.2 Agrupamiento de las bases de datos desagregadas, disminuyendo en un 3,67% y 6,34% el número de registros, respectivamente.

La eliminación de estos registros atípicos es principalmente atribuida a un proceso de depuración de ciertos datos erráticos presentes dentro de las base de datos suministradas por el concurso. Este hecho permite que los análisis exploratorios realizados sobre los datos y los algoritmos de Machine Learning que se ajustarán posteriormente, sean realizados con información estadísticamente confiable y por ende, los resultados obtenidos puedan tener el mismo nivel de confiabilidad.

4.2.3 Imputación de datos faltantes

"La existencia de datos faltantes afecta a la mayoría de los algoritmos de aprendizaje automático; ya que muchos algoritmos de aprendizaje automático se basan en la suposición de que los datos están completos. Cuando faltan grandes cantidades de observaciones en un conjunto de datos, la generalización del estudio y las conclusiones estadísticas se ven afectadas debido al tamaño reducido de la muestra, lo que podría reflejar estimaciones de parámetros sesgadas y conclusiones engañosas. Por lo tanto, manejar los datos faltantes es una tarea clave." (Hammad & Kimura, 2020)

La imputación de los datos faltantes se realizó de igual forma, tanto para la base de datos resultante de la etapa 3.2.2 Agrupamiento de las bases de datos desagregadas como para el dataset anexo Merchants.

Para esto, un pipeline fue definido según el tipo de variable.

- **Imputación de datos faltantes para variables numéricas:** La estrategia adoptada para la imputación de los de datos faltantes en los distintos datasets para las variables numéricas fue el uso de la *mediana* de la distribución. Como se explicó previamente, muchas de las características que componen los conjuntos de datos poseen una gran cantidad de outliers, que si bien el algoritmo LOF previamente permitió la eliminación de cierta cantidad de muestras del dataset al ser consideradas como atípicas, esto no descarta la existencia de outliers residuales que no hayan sido detectados por el algoritmo, y por ende, sigan presentes en las distribuciones.

  Con el fin de evitar una imputación de datos sesgada por los datos atípicos, se seleccionó como estrategia el valor de la mediana de la distribución, siendo esta medida de tendencia central una de las menos sensibles a la presencia de outliers.

Un ejemplo de la imputación de los valores faltantes de las variables numéricas es la que se realizó a los datos nulos presentes en la variable $month\_diff\_std$. Esta variable tenía 153199 registros no nulos antes del proceso de imputación, considerando que la base de datos consta de 179986 muestras. Esto implica que existen alrededor de 26787 registros nulos los cuales deberán ser imputados con el valor de la mediana de esta variable ($\tilde{X} = 0.478713$) en el pipeline.

- **Imputación de datos faltantes para variables categóricas**: Por otra parte, la estrategia llevada a cabo para imputar los datos faltantes de las variables categóricas es por medio del valor de la moda de la distribución.

  En este caso, un ejemplo de la imputación realizada a las variables categóricas es la que se llevó a cabo sobre a la variable $Category_2$ presente en la base de datos anexa Merchants. La etapa de imputación se realizó antes de la generación de los clusters que fueron evaluados en la etapa 3.2.3 Evaluación de la pertinencia de las bases de datos anexas como se muestra en la Figura 2. La variable $Category_2$ dispone de 322809 registros no nulos mientras que la base de datos anexa Merchants contiene 334696 registros. Es decir, que existen aproximadamente 11887 valores nulos (alrededor del 3.6% del total de datos) que deberán ser imputados con el valor de la moda de esta variable ($\hat{X}$=1)

4.2.4 Estandarización y codificación de variables

Con el objetivo de normalizar los valores de las distribuciones numéricas, se empleó la estrategia de Standard Scaler, ya que menudo es necesario normalizar los valores de los atributos, especialmente en aquellos casos donde los valores son muy diferentes en escala. (Zaki & Meira, 2014). Por otra parte, “el preprocesamiento de variables categóricas se vuelve importante ya que la mayoría de los modelos de aprendizaje automático solo consideran variables numéricas; por lo tanto, se debe transformar estas

variables categóricas a números para que el modelo pueda comprender y recuperar la información útil. Existen muchas formas de codificar variables categóricas para el modelado, y una de las técnicas de codificación más utilizadas One hot encoded" (Dahouda & Joe 2021).

Considerando lo anterior, la codificación de las variables categóricas se llevó a cabo por medio del One Hot Encoder. Por lo cual, se obtuvieron tantas nuevas variables binarias como niveles en las variables categóricas codificadas. A partir de la etapa de codificación, el número de columnas aumentó en un 28% pasando de 25 a 32 variables.

4.2.5 Reducción de la dimensionalidad del dataset

"Los datos de alta dimensionalidad pueden causar problemas para la extracción y el análisis de datos. No obstante, es importante comprobar si la dimensionalidad se puede reducir conservando las propiedades esenciales de la matriz de datos completa. Esto puede ayudar a la visualización de datos, así como a la extracción de información". (Zaki & Meira, 2014).

Debido a la alta dimensionalidad de la base de datos, lo cual generaba problemas de RAM durante las iteraciones, se decidió implementar una estrategia que permitiera reducir la volumetría del dataset sin penalizar el rendimiento de los futuros modelos.

Para esto, se consideraron dos modalidades de reducción de la dimensionalidad (Reducción por Filas y Reducción por columnas) que fueron evaluadas tanto desde una perspectiva del rendimiento de los futuros modelos como en el tiempo de ejecución del modelo en cuestión. Estas modalidades fueron comparadas con la base de datos completa, la cual fue denominada durante las iteraciones como "full dataset".

Como se ha explicado previamente, el problema planteado será abordado desde dos enfoques de la analítica: una tarea de regresión y una tarea de clasificación. Partiendo del principio que no necesariamente las mismas características imprescindibles para la tarea de regresión son las mismas que

para la tarea de clasificación, se llevó a cabo paralelamente el mismo proceso de reducción de dimensionalidad de la misma base de datos inicial para cada una de las tareas. En este sentido, si bien la metodología empleada es similar indistintamente al tipo de tarea, los algoritmos que permitieron llevarla a cabo difieren, adaptándose así a la tarea en cuestión. La metodología adoptada puede resumirse en las siguientes etapas:

4.2.5.1 Elección de la modalidad de reducción de dimensionalidad.

Reducción por filas: En la reducción por filas se conservó el 50% de los registros de la base de datos. Las muestras resultantes fueron seleccionadas de forma aleatoria con el fin de no sesgar los resultados con la elección de un periodo concreto.

Reducción por columnas: La reducción por columnas se realizó considerando la importancia de las características del dataset. En este sentido, al encontrar cuales son las variables más importantes para los futuros modelos predictivos, se puede igualmente encontrar cuales son las características que menos contribuyen en la predicción y por ende, su presencia tanto en el modelo como en la base de datos seria prescindible. De esta forma, las variables consideradas como menos importantes son aquellas que se eliminarían del dataset. El algoritmo empleado para estimar la importancia de las características fue a partir del modelo de XGBoost (XGBoostRegressor y XGBoostClassifier para la tarea de regresión y clasificación respectivamente). El XGBoots implementa algoritmos de aprendizaje automático bajo el marco Gradient Boosting y proporciona un impulso de árbol paralelo (también conocido como GBDT, GBM) que resuelve muchos problemas de ciencia de datos de una manera rápida y precisa.” (XGBoost Documentation, n.d.) . Para llevar a cabo la reducción por columnas, se consideraron tres formas distintas para estimar la importancia de las características usando como base el algoritmo XGBoost:

1. La primera forma es a partir de la utilización del atributo feature_importances disponible en el algoritmo XGBoost después de haber entrenado el modelo.

2. La segunda forma es a partir del uso del método permutaton_importance. Este método de permutación consiste en mezclar aleatoriamente cada una de las características y posteriormente calcular el cambio en el rendimiento del modelo. En este sentido, las variables que mayor impacto tienen en el rendimiento son consideradas como las más importantes en el modelo y por consiguiente, en la base de datos.
3. La tercera forma es a partir del paquete SHAP. Este paquete usa los valores de Shaley provenientes de la teoría de juegos, el cual busca determinar cómo contribuye cada una de las características en la predicción del modelo.

La forma para determinar si una variable es importante o no es comparando entre sí la magnitud de los valores cuantitativos resultantes de cada técnica empleada. Durante el análisis exploratorio de los resultados, se encontró que si bien había múltiples variables cuya importancia relativa dentro de modelo era cercana a cero, esta no llegaba a ser necesariamente nula.

Por consiguiente, se empleó el siguiente criterio para definir la frontera de decisión de la importancia de las características:

$$Si\ Feature\ Importance\ Variable_i < \frac{1}{100}\ Max(Feature\ Importance) \ \rightarrow Variable_i\ no\ es\ importante$$

En otros términos, si la importancia de una de una característica es 100 veces inferior al valor de la importancia más alta obtenida para alguna de las características, entonces esta variable es considerada como “no importante” y por ende eliminada. Por el contrario, si la importancia de una característica es superior o igual a dicho valor, la característica se conserva, y es entonces considerada como importante.

La Tabla 13 sintetiza los resultados obtenidos a partir de cada modalidad empleada sobre la dimensionalidad de la base de datos para la tarea de regresión, empleando XGBoost Regressor.

Tabla 13

Síntesis del procesos de reducción de dimensionalidad para la tarea de regresión

Modalidad	Tipo de modalidad	Modificación	Dimensión Dataset final
Full Dataset	NA	NA	(167758, 32)
Row Reduction	NA	Eliminación 50% de registros	(83879, 32)
Feature Reduction	Feature Importance	Eliminación de 7 features	(167758, 25)
	Permutation Importance	Eliminación de 16 features	(167758, 16)
	Shap	Eliminación de 11 features	(167758, 21)

Nota: Realización propia

A partir de los resultados de la Tabla 13, se puede concluir que el tipo de modalidad empleado para reducir el número de variables permutation Importance, es aquel que identifica el mayor número de características “no importantes” dentro del dataset, con un total de 16 características sobre las 32 existentes. Por otra parte, el atributo Feature_Importance del modelo XGBoost es aquel que detecta el menor número de características irrelevantes.

De forma paralela, la Tabla 14 muestra los resultados obtenidos a partir de la implementación de la misma metodología utilizada previamente para la tarea de regresión, pero en este caso, para la tarea de clasificación. De igual forma, el algoritmo empleado es XGBoost para clasificación (XGBoost Classifier), por consiguiente, los mismos tipos de modalidades asociados a la Reducción por Columnas (Feature Reduction) pudieron ser efectuados.

Tabla 14

Síntesis del procesos de reducción de dimensionalidad para la tarea de clasificación

Modalidad	Tipo de modalidad	Modificación	Dimensión Dataset final
Full Dataset	NA	NA	(167758, 32)
Row Reduction	NA	Eliminación 50% de registros	(83879, 32)
Feature Reduction	Feature Importance	Eliminación de 6 features	(167758, 26)
	Permutation Importance	Eliminación de 19 features	(167758, 13)
	Shap	Eliminación de 12 features	(167758, 20)

Nota: Realización propia

Se pueden afirmar a partir de los resultados obtenidos anteriormente que:

- El comportamiento de cada una de las modalidades se conservó entre las tareas (regresión y clasificación). En este sentido, tanto para la tarea de regresión como para la de clasificación, fue la modalidad de reducción por columnas *permutation importance,* la cual identifica el mayor número de características irrelevantes para el modelo.
- También se evidencia que, si bien el comportamiento de los resultados entre las dos tareas es relativamente similar, no lo son las magnitudes de los valores obtenidos. En este sentido, se observa que para la tarea de clasificación, tanto por *permutation importance* como por el *método Shap*, se obtuvo un número superior de características irrelevantes al compararlos con los resultados obtenidos por estos mismos métodos para la tarea de regresión.
- Finalmente, con base a la conclusión inmediatamente anterior, se confirma la hipótesis que no necesariamente una característica tendrá la misma importancia al modificarse el tipo de problema de Machine Learning para la cual se está empleando, a pesar de que las variables a

predecir estén estrechamente relacionadas, como lo son en este caso: Fidelidad (variable continua) y Fidelidad discretizada (variable categórica).

A continuación, en la etapa 4.2.5.2 Ajuste de modelos. se ajustarán distintos algoritmos de Machine Learning con el fin de verificar si alguna de las modalidades de reducción de dimensionalidad impacta ya sea de forma positiva o negativa el rendimiento de los modelos.

4.2.5.2 Ajuste de modelos.

Como se expuso anteriormente, la etapa descrita a continuación no hace parte de modelamiento que se realizará para resolver las tareas de analítica planteadas para la regresión o la clasificación. Por el contrario, este ajuste de modelos se realiza como etapa del preprocesamiento de los datos, en donde se busca determinar si las bases de datos resultantes a partir de cierto tipo de modalidad de reducción responden en mayor o menor medida a los futuros modelos que se entrenaran posteriormente.

El algoritmo de Machine Learning a emplear en cada escenario dependerá de la tarea en cuestión. Para el caso de la **tarea de regresión** se seleccionaron dos modelos: El modelo base *(Linear Regression)* y un modelo optimizado a partir de la búsqueda de hiper parámetros. La elección del modelo base se hizo debido a la elección efectuada durante la primera iteración, en la cual fue empleada un modelo de regresión lineal. Por otra parte, el segundo modelo seleccionado al cual se le realizó la búsqueda de mejores hiper parámetros, fue un *Decisión Tree Regressor.*

En este caso, no solo se buscaba evaluar la forma como impacta la modalidad de reducción de la dimensionalidad al performance de los modelos seleccionados, sino además el de poder verificar si las bases de datos resultantes de las modalidades de reducción responden en mayor o menor medida al entrenamiento de modelos más complejos. Tanto la base de datos inicial (Full dataset) como cada una de las bases de datos resultantes de las modalidades de reducción de dimensionalidad explicadas anteriormente fueron evaluadas a través de estos dos modelos. Esto implica que 10 iteraciones fueron

realizadas con el objetivo de llevar a cabo esta reducción. La Tabla 15 reagrupa los resultados de la métrica de Machine Learning para la tarea de regresión (R^2) durante la fase de test obtenidos a partir de las distintas combinaciones de modalidad de reducción-algoritmo:

Tabla 15

Resultados del modelado para la selección de la modalidad de reducción de dimensionalidad en la tarea de regresión

Modalidad	Tipo de modalidad	R^2 (LR)	R^2 (DTR)	Time Grid Search
Full Dataset	NA	0.077795	0.094892	1304.385317
Row Reduction	NA	0.081030	0.095496	635.025165
Feature Reduction	Feature Importance	0.072151	0.094892	1258.879223
	Permutation Importance	0.070400	0.092192	1073.929934
	Shap	0.072161	0.094832	1155.950313

Nota: Realización propia

Según los resultados obtenidos en la Tabla 15 se puede constatar lo siguiente:

- Se logró uno de los objetivos planteados en esta etapa, el cual consistía en verificar la forma como respondía los datasets resultantes de las modalidades de reducción de dimensionalidad en un modelo de Machine Learning más complejo. En el caso de la tarea de regresión, se observa que si bien los valores de las métricas de Machine Learning (R^2) son bajos tanto para la regresión lineal como para el *Decision Tree Regressor* optimizado, este último obtuvo un valor superior en el R^2, la cual es una métrica que se busca maximizar.
- Si bien los resultados obtenidos por medio del *Decision Tree Regressor* son más elevados al compararse con los de la regresión lineal, no existe una gran variabilidad entre los resultados obtenidos por el primero. Esto implica que el rendimiento del modelo no será necesariamente el único criterio de elección para determinar si se selecciona una técnica de reducción o si se

conserva el full dataset. En este caso se considerará igualmente los resultados del tiempo de ejecución.

- El hecho de implementar una de las modalidades de reducción definidas (reducción por filas o reducción por columnas), no penaliza necesariamente el rendimiento del modelo, dado que al compararse los resultados del rendimiento obtenido para cada una de estas modalidades de reducción con el rendimiento de la base de datos completa (Full Dataset), no existe una disminución significativa en las métricas de Machine Learning (R^2).
- Por las razones explicadas previamente, se decide implementar una técnica de reducción de dimensionalidad al considerarse que la supresión de la información en la base de datos no repercute negativamente en el rendimiento del modelo ajustado. En este sentido, a R^2 en fase de test constante, se selecciona como mejor técnica de reducción de dimensionalidad la reducción por columnas considerando las características irrelevantes encontradas por el método **permutation importance**. Si bien, no es necesariamente la técnica que minimiza los tiempos de entrenamiento, si es la que logra el mejor compromiso entre la maximización de la cantidad de información relevante dentro de la base de datos y la minimización de los tiempos de ejecución, siendo esta, la segunda más baja dentro de las iteraciones realizadas.

Por otra parte, para la tarea **de clasificación** únicamente se seleccionó el algoritmo *Decision Tree Classifier* al cual se le realizó, de igual forma, la búsqueda de sus mejores hiperparámetros.

En este caso, únicamente se cuenta con este algoritmo dado que ningún otro modelo de clasificación había sido entrenado previamente. En este sentido, durante la primera iteración, únicamente se abordó este problema desde un enfoque de regresión dado que es la tarea asociada al concurso propuesto por ELO. Por esta razón, no se cuenta necesariamente con un resultado previo con el cual poder comparar el progreso de la tarea de clasificación.

Por consiguiente, para la tarea de clasificación, se implementa un único modelo complejo utilizado únicamente con el fin de verificar el rendimiento del modelo a partir de las distintas bases de datos obtenidas por alguna de las modalidades de reducción de la dimensionalidad.

De igual forma, cada una de las modalidades de reducción de la dimensionalidad fueron evaluadas en el modelo. Esto implica que 5 iteraciones fueron realizadas con el objetivo de llevar a cabo esta reducción para el problema de clasificación.

La Tabla 16 reagrupa los resultados de la métrica de Machine Learning para la tarea de clasificación ($ROC\ Score$) durante la fase de test obtenidos a partir de las distintas combinaciones de modalidad de reducción-algoritmo:

Tabla 16

Resultados del modelado para la selección de la modalidad de reducción de dimensionalidad en la tarea de clasificación

Modalidad	Tipo de modalidad	ROC Score (DTC)	Time Grid Search
Full Dataset	NA	0.609777	1675.161698
Row Reduction	NA	0.609777	1671.819072
Feature Reduction	Feature Importance	0.609777	1687.338954
	Permutation Importance	0.608574	1360.149719
	Shap	0.607555	1534.036532

Nota: Realización propia

A partir de los resultados obtenidos por la Tabla 16 se puede concluir que:

- De forma similar a los resultados obtenidos en la Tabla 15 asociada a la tarea de regresión explicada anteriormente, no existe mucha variabilidad entre los resultados obtenidos por el modelo Decision Tree Classifier en las distintas modalidades de reducción de la dimensionalidad.

Esto permite concluir que si bien el rendimiento de los distintos modelos es relativamente bajo, no existe una penalización del performance del modelo de llegarse a seleccionar un dataset cuya dimensionalidad haya sido reducida, ya sea por filas (Row reduction) o columnas (Feature reduction).

- Por esta razón, y considerando como criterio el de seleccionar la base de datos que logre minimizar los tiempos de ejecución en la búsqueda de hiperparámetros del Decision Tree Classifier, se realizará una reducción de dimensionalidad por columnas considerando como tipo de modalidad las características prescindibles encontradas por el tipo **Permutation Importance**.

Como conclusión general de esta etapa asociada de reducción de dimensionalidad, se decide reducir la base de datos tanto para la tarea de regresión como la de clasificación, eliminando las características menos importantes encontradas por el tipo de reducción asociada a **Permutation Importance**. Esta decisión se toma al considerar que la base de datos resultante a partir de esta reducción es una de la que garantiza en mayor medida la preservación de la información relevante dentro del set de datos logrando minimizar, de igual forma, los tiempos de ejecución computacional.

Las bases de datos resultantes de la reducción de la dimensionalidad son aquellos que serán empleados para ajustar los algoritmos de Machine Learning, tanto para la tarea de regresión como para la de clasificación.

4.3 MODELOS

Los algoritmos empleados dependen de la tarea a resolver. Por consiguiente, se procederá a explicar de forma independiente los distintos modelos que fueron ajustados tanto para la regresión como para la clasificación.

Sin embargo, la estrategia adoptada para implementar estos algoritmos es común para las dos tareas. En este sentido, cada uno de los modelos de Machine Learning empleados fueron sometidos a dos fases sucesivas para determinar los hiperparámetros óptimos, y así poder maximizar las métricas de Machine Learning definidas para cada una de las tareas.

La metodología común puede ser resumida en las siguientes etapas:

1. Búsqueda aleatoria de hiperparámetros en un rango amplio de valores
2. Búsqueda exhaustiva de hiperparámetros acotados

4.3.1 Metodología

Búsqueda aleatoria de hiperparámetros en un rango amplio de valores

"En general, construir un modelo efectivo de aprendizaje automático es un proceso complejo y lento que implica determinar el algoritmo apropiado y obtener una arquitectura de modelo óptima ajustando sus hiperparámetros. Para construir un modelo de machine Learning óptimo, se debe explorar una gama de posibilidades. El proceso de diseñar la arquitectura del modelo ideal con una configuración óptima de hiperparámetros se denomina ajuste de hiperparámetros. (Yang & Shami, 2020).

Dado que los valores de los hiperparámetros pueden estar definidos dentro de dominios demasiado amplios de valores, es computacionalmente imposible realizar una búsqueda de hiperparámetros considerando cada uno de estos, y sobre todo, las distintas combinaciones resultantes

por los distintos hiperparámetro del modelo. Por esta razón, se decide acotar los valores máximos posibles para cada hiperparámetro a partir de la revisión de la literatura, en la cual sugieren un rango de valores normales en los cuales suelen oscilar los hiperparámetros para ciertos modelos. Por otra parte, se empleó el **RandomizedSearchCV**, el cual permite definir rangos amplios en los valores de los hiperparámetros a evaluar, y que sea el **RandomizedSearchCV** quien seleccione aleatoriamente el valor en cada ajuste, con el fin de determinar dentro de las búsquedas efectuadas, la combinación de hiperparámetros que generó el mejor rendimiento del modelo en términos de la métrica de Machine Learning seleccionada para la tarea.

Con el objetivo que los criterios de evaluación fueran similares con los de la búsqueda exhaustiva de hiperparámetros que se explicara en la siguiente sección, la estrategia de validación cruzada fue la misma tanto en el **RandomizedSearchCV** como en el **GridSearchCV**.

Búsqueda exhaustiva de hiperparámetros acotados

Los hiperparámetros encontrados en la búsqueda aleatoria realizada previamente, si bien son los que ofrecen el mejor rendimiento dentro de las búsquedas efectuadas, no tienen que ser necesariamente los valores óptimos para considerar en los algoritmos finales. La búsqueda realizada previamente permite acotar el rango de posibles valores óptimos para los hiperparámetros y de esta forma realizar la búsqueda exhaustiva en estos valores cercanos al hiperparámetro óptimo encontrado en la búsqueda aleatoria precedente.

4.3.2 Partición del dataset

La estrategia de validación empleada durante el proyecto fue un Kfolds, con un número de particiones (n_splits) igual a 3. Esta técnica fue empleada tanto para la tarea de regresión como para la tarea de clasificación. En este sentido, es necesario precisar que las clases obtenidas a partir de la discretización de la variable Target para llevar a cabo la tarea de clasificación, fueron dos clases

balanceadas, cada una con el 52% y 48% del total de registros. Por esta razón, ninguna técnica de balanceo fue necesaria durante la selección de la estrategia de validación para el problema de clasificación. Del 100% de los registros existentes en el dataset sobre el cual se ajustaron los modelos, se realizó una partición del 80%-20% entre el dataset de train y de test respectivamente. Es decir, que el proceso de validación Kfolds, se realizó un split de tres particiones del 80% asociado a dataset de train.

4.3.3 Modelos para la tarea de regresión

"La regresión es un tipo de método de aprendizaje supervisado que utiliza un algoritmo para comprender la relación entre las variables dependientes e independientes. Los modelos de regresión son útiles para predecir valores numéricos basados en diferentes puntos de datos". (IBM ,2021)

En los modelos para la tarea de regresión, se pueden identificar dos tipos de estrategias: la primera es el ajuste de los modelos simples optimizados a partir de la búsqueda de hiperparámetros, y la segunda estrategia es la utilización de métodos de ensemble considerando como estimadores base a los algoritmos simples que mejores resultados obtuvieron, yendo de esta manera, desde los métodos más sencillos hasta los más complejos.

4.3.3.1 Modelos simples para regresión.

La Tabla 17 reagrupa la información asociada a los distintos modelos simples ajustados para la tarea de regresión:

Tabla 17

Ajuste de los modelos de regresión

Modelo	Mejor hiperparámetro aleatorio	Mejor hiperparámetro exhaustivo	R^2Train	R^2 Validation	R^2 Test
Linear Regression	NA	NA	0.0757	NA	0.0704
Ridge Regression	$alpha$: 139	$alpha$: 144	0.0758	0.0751	0.0704
Lasso Regression	$alpha$: 521	$alpha$: 500	0	-2.8581	0.0704
ElasticNet Regression	$alpha$: 41 $l1\_ratio$: 0	$alpha$: 35, $l1\_ratio$: 0	0.0046	0.0046	0.0044
Quadratic Regression	NA	NA	0.0999	NA	0.1070
Decision Tree Regressor	max_depth: 10 $max\_features$: $None$ max_leaf_nodes: 100 $min\_samples\_leaf$: 17 $min\_samples\_split$: 24 $splitter$: $best$	max_depth: 8 $max\_features$: $None$ max_leaf_nodes: 90 $min\_samples\_leaf$: 17 $min\_samples\_split$: 20 $splitter$: $best$	0.1138	0.0942	0.9245
Random Forest Regressor	max_depth: 17 $max\_features$: $log2$ max_leaf_nodes: 75 $min\_samples\_leaf$: 6 $min\_samples\_split$: 28 $n\_estimators$: 169	max_depth: 15 $max\_features$: $log2$ max_leaf_nodes: 76 $min\_samples\_leaf$: 5 $min\_samples\_split$: 29 $n\_estimators$: 170	0.1208	0.1071	0.1044

Nota: Realización propia

A partir de los resultados obtenidos de los distintos modelos de regresión ajustados, se pueden hacer las siguientes conclusiones:

- Existe una evolución de aproximadamente 5 puntos porcentuales en la Métrica de Machine Learning en test entre el modelo de regresión más sencillo ajustado (Linear Regressor) y el modelo simple más complejo, el cual corresponde a un Random Forest Regressor.
- Los distintos modelos de regresión lineal (desde el más básico hasta los regularizados con hiperparámetros optimizados), fueron lo que obtuvieron los rendimientos más bajos con valores de R^2 inferiores al 8% en test.
- Se observa que, si bien existe una leve mejora en el rendimiento de los modelos a medida que estos se van haciendo más complejos, claramente los modelos son incapaces de predecir correctamente la variable target (continua). Este hecho era previsible en cierta medida desde el análisis exploratorio de los datos, en donde se encontró que las características de las cuales se disponían en el set de datos no presentaban una correlación fuerte con la variable respuesta.
- Al constatarse que la mejoría en el rendimiento del modelo está asociada a la complejidad de este, se procederá a emplear métodos de ensamble para la regresión considerando como estimadores base a los mejores modelos simples obtenidos.

4.3.3.2 Métodos de ensemble para regresión.

"Los métodos de ensamble entrenan un conjunto de varios modelos simples de machine learning y asumen que sus predicciones colectivas pueden superar la precisión de las individuales, o incluso mejorar propiedades como la robustez o generalizabilidad. Los métodos de ensamble siguen un paradigma de aprendizaje. que se resume con la expresión "juntos mejor", y en muchos casos, son

capaces de superar las predicciones de otros métodos de Machine Learning más complejos. (Castillo-Botón, et al, 2022).

Se realizó un método de ensamble con los mejores modelos simples obtenidos anteriormente conservando los valores de sus hiperparámetros optimizados. El método de ensemble empleado es el Bagging Regressor, considerando como estimadores base: Quadratic Regression y Decision Tree Regressor.

El metamodelo Bagging Regressor tiene como hiperparámetro principal el **número de estimadores** *(n_estimators)*. Por consiguiente, con el fin de determinar el valor óptimo del número de estimadores, un proceso iterativo a través de un ciclo *for* se realizó considerando diferentes valores enteros comprendidos entre 10 y 100 para *n_estimators*.

El criterio de selección del número de estimadores óptimo era aquel que maximizará el R^2 en test.

En la Tabla 18 se encuentran los resultados de R^2con los datos de Train y Test para los modelos de Bagging considerando como estimador de base tanto Quadratic Regression como Decision Tree Regressor.

Tabla 18

Hiperparámetros óptimos para el método de ensamble en regresión según estimador base

Estimador Base	Hiperparámetro óptimo	R^2 Train	R^2 Test
Decision Tree Regressor	$n\_estimators = 100$	0.1112	0.1081
Quadratic Regression	$n\_estimators = 40$	0.0977	0.1069

Nota: Realización propia

A partir de los resultados obtenidos por el Bagging regressor, se logra evidenciar que la Métrica de Machine Learning (R^2) en test no mejora drásticamente si se compara con los resultados de algoritmos simples que sirvieron como estimadores base.

Esto implica que los métodos de ensamble debido tanto al bajo rendimiento proporcionado como a los recursos computacionales adicionales que estos necesitan respecto a algoritmos más simples no justifican su utilización para la tarea de regresión con el set de datos disponible.

Por consiguiente, de ser necesario seleccionar el modelo de regresión que mejor rendimiento haya obtenido en la fase de test sería el algoritmo **Random Forest Regressor**, el cual genera un performance con los datos de test cercano al 12%.

4.3.4 Modelos para la tarea de clasificación.

De forma similar a la tarea de regresión, los modelos empleados para la tarea de clasificación van desde los algoritmos más sencillos hasta los más complejos, considerando de igual forma los métodos de ensamble.

4.3.4.1 Modelos simples para clasificación.

La Tabla 19 sintetiza los resultados obtenidos por los distintos modelos simples ajustados para la tarea de clasificación, junto con los valores de los hiperparámetros utilizados para obtener los resultados de ROC Score en Train, Validation y Test

Tabla 19

Ajuste de los modelos de clasificación

Modelo	Mejor hiperparámetro aleatorio	Mejor hiperparámetro exhaustivo	ROC ScoreTrain	ROC Score Validation	ROC Score Test
Simple Logistic Regression	NA	NA	0.6437	NA	0.6027
Logistic Regression $Solver$: $newton-cg$, $lbfgs$, sag	C: 2 tol: 1	C: 1 $penalty$: $none$ $solver$: $lbfgs$ tol: 1	0.6439	0.6437	0.6027
Logistic Regression $Solver$: $liblinear$	C: 2 tol: 1	C: 5 $penalty$: $l1$ $solver$: $liblinear$ tol: 1	0.6356	0.6354	0.5920
Logistic Regression $Solver$: $saga$	C: 2 tol: 1	C: 1 $penalty$: $l1$ $solver$: $saga$ tol: 0	0.6439	0.6437	0.6027
Decision Tree Classifier	$criterion$: $entropy$ max_depth: 18 $max\_features$: $None$ max_leaf_nodes: 93 $min\_samples\_leaf$: 10 $min\_samples\_split$: 34 $splitter$: $best$	$criterion$: $entropy$ max_depth: 15 $max\_features$: $None$ max_leaf_nodes: 93 $min\_samples\_leaf$: 10 $min\_samples\_split$: 32 $splitter$: $best$	0.6584	0.6421	0.6077
Random Forest Classifier	$criterion$: $gini$ max_depth: 18 $max\_features$: $sqrt$ max_leaf_nodes: 87 $min\_samples\_leaf$: 6 $min\_samples\_split$: 25 $n\_estimators$: 194	$criterion$: $gini$ max_depth: 18 $max\_features$: $sqrt$ max_leaf_nodes: 88 $min\_samples\_leaf$: 26 $min\_samples\_split$: 24 $n\_estimators$: 195	0.6699	0.6538	0.6065

Nota. Realización propia

A partir de la Tabla 19 se puede afirmar que el comportamiento de los modelos para la tarea de clasificación difiere del observado para la tarea de regresión, en donde en este último se evidenció una leve mejoría de la Métrica de Machine Learning en función de la complejidad del modelo ajustado. Sin embargo, los resultados obtenidos para la tarea de clasificación no siguen este mismo comportamiento, dado que la métrica no mejora según la complejidad del algoritmo. En este sentido, indistintamente al algoritmo evaluado, el ROC score asociado a la fase de test alcanza un valor máximo cercano al 60%.

A pesar de estos resultados, se decide de igual manera utilizar los métodos de ensemble con el fin de verificar si estos permiten aumentar el performance para esta tarea.

4.3.4.2 Métodos de ensemble para clasificación.

Para la tarea de clasificación se emplearon los métodos de ensamble VotingClassifier, BaggingClassifier y AdaBoostClassifier considerando como estimadores bases los modelos simples que presentaron los mejores resultados. Si bien los rendimientos obtenidos a partir de los modelos simples varían muy poco entre ellos, se seleccionaron como estimadores aquellos que presentaron los resultados más altos.

Del mismo modo que para la tarea de regresión explicada previamente, dos de los tres métodos de ensamble seleccionados para la clasificación necesitan de una metodología para determinar sus respectivos hiperparámetros. A continuación, se explicarán las estrategias llevadas a cabo para encontrar los hiperparámetros según el metamodelo:

- **VotingClassifier**: ninguna estrategia de búsqueda de hiperparámetros fue necesaria para emplear este metamodelo. Por el contrario, se consideraron tres modelos base para llevar a cabo la votación, los cuales corresponden a los dos mejores modelos simples obtenidos en la Logistic Regression asociados a los solvers lbfgs y saga y al Decision Tree Classifier.

- **BaggingClassifier:** La búsqueda del número de estimadores óptimo para el Bagging Classifer siguió la misma estrategia utilizada para la regresión, pero adaptada a la tarea de clasificación. En este sentido, un proceso iterativo a través de un ciclo *for* se llevó a cabo con múltiples valores posibles para el hiperparámetro n_estimators. Posteriormente, se procedió a estimar el valor de ROC score con los datos de test. Finalmente, el número de estimadores seleccionado corresponde al valor que maximiza el ROC score dentro del conjunto de valores evaluados.
- **AdaBoostClassifier:** Dado que este metamodelo tiene dos hiperparámetros para optimizar, la estrategia empleada para encontrar los mejores valores para estos no fue un proceso iterativo a través de un ciclo *for*, como en el **BaggingClassifier,** sino la metodología empleada para los modelos simples. Es decir, una búsqueda inicial de hiperparámetros aleatorios para acotar el rango de valores y posteriormente, una búsqueda exhaustiva cercana a los valores encontrados previamente. Los resultados obtenidos por los métodos de ensamble para la clasificación son presentados en la Tabla 20.

Tabla 20

Hiperparámetros óptimos para los métodos de ensamble en clasificación según estimador base

Método de ensemble	Estimador Base	Hiperparámetro óptimo	ROC Score Train	ROC Score Test
Voting Classifier	Logistic Regression $Solver$: $lbfgs$ Logistic Regression $Solver$: $saga$ Decision Tree Classifier	NA	0.6033	0.6027
Bagging Classifier	Decision Tree Classifier	$n\_estimator = 60$	0.6532	0.6114
AdaBoosting	Decision Tree Classifier	$learning\_rate$: 2 $n\_estimators$: 31	0.6952	0.5967

Nota: Realización propia

A partir de la Tabla 20 se logra evidenciar que los resultados obtenidos en ROC score para los valores de test con los métodos de ensamble utilizados no son necesariamente mejores a las métricas ya obtenidas por los modelos simples.

Por esta razón, de llegarse a seleccionar el mejor modelo para la tarea de clasificación, considerando tanto el rendimiento predictivo del algoritmo como la eficiencia computacional, este sería el modelo simple **Decision Tree Classifier**.

4.4 MÉTRICAS

Métricas de Machine Learning

Las métricas de Machine Learning empleadas tanto para la tarea de regresión como de clasificación fueron importadas a partir de las funciones disponibles de sklearn.

- **Tarea de regresión**: Como se expuso previamente, la métrica de machine Learning principal para esta tarea es el R^2. La utilización de esta métrica dentro del proyecto se hizo de las siguientes formas:
    - como función: r2_score()
    - como parámetro *scoring*: 'r2'

- **Tarea de clasificación:** La métrica empleada para la tarea de clasificación es el ROC Score. De igual forma, el ROC score puede ser utilizado tanto como función, la cual se empleó para obtener la métrica con los datos de test, como parámetro *scoring* en el RandomizedSearchCV o GridSearchCV**.** La utilización de esta métrica en sklearn es la siguiente:
    - como función: roc_auc_score()
    - como parámetro *scoring*: 'roc_auc'

Métricas de negocio:

Considerando que el set de datos proviene de un concurso propuesto en Kaggle, las métricas de negocio expuestas en la sección 2.4.2 Las métricas de negocio, fueron planteadas partiendo del principio que estas métricas permitirían medir la utilidad de los modelos predictivos en producción. En este caso, dado que los mejores modelos obtenidos tanto para la tarea de regresión como la clasificación no superan el criterio del 80% de performance en el conjunto de datos de test, no sería pertinente implementar los modelos en producción y, por ende, las métricas de negocio no podrían ser estimadas.

5. METODOLOGÍA

5.1 BASELINE

La primera iteración realizada sobre el conjunto de datos no es necesariamente comparable con los resultados expuestos anteriormente. La razón principal se debe a la integración de la base de datos New Merchants Transactions, la cual fue obviada durante la primera iteración.

Este hecho genera una variación en la cantidad de características empleadas para realizar la predicción. De igual forma, el número de registros varía dado que durante la primera iteración, al no considerarse la base de datos New Merchants Transactions, no fue necesario llevar a cabo completamente la etapa expuesta en la sección 3.2.1 Filtrado de las bases de datos originales.

Finalmente, la primera iteración se llevó a cabo únicamente para la tarea de analítica propuesta por el concurso de Kaggle, es decir, una tarea de regresión. El resultado obtenido en la tarea de regresión

durante la primera iteración considerando las condiciones expuestas anteriormente, es de $R^2 = 0.71\%$ en test.

Si se compara el resultado de la métrica R^2 obtenida durante la primera iteración con el resultado del mejor modelo de regresión (Random Forest Regressor) obtenido durante esta iteración (con un $R^2 = 11.19\%$ en test), se puede concluir que la Métrica aumentó en un 1400% aproximadamente.

5.2 ITERACIONES y EVOLUCIÓN

Las iteraciones llevadas a cabo fueron enfocadas principalmente en el preprocesamiento de los datos, y más precisamente, en la selección del dataset final para entrenar los algoritmos. En este sentido, dos grandes iteraciones fueron empleadas con este fin, las cuales fueron explicadas previamente.

La primera iteración llevada a cabo se realizó sobre los clusters resultantes de la base de datos anexa Merchants. En este caso, cinco clusters resultantes a partir de distintos modelos y métricas de error fueron evaluados con el dataset obtenido después de la etapa de agrupamiento de las bases de datos desagregadas. Para esto, de forma iterativa, cada uno de los resultados obtenidos de los cinco procesos de clustering fueron introducidos individualmente para evaluar el impacto sobre el rendimiento del modelo.

Al final de este proceso iterativo para la evaluación de pertinencia de los clusters, se concluyó que la introducción de los clusters al dataset final no mejoraría el performance de los futuros modelos predictivos, al ser evaluados bajo el mismo algoritmo base (regresión lineal simple).

El segundo proceso iterativo llevado a cabo durante el proyecto fue en la reducción de la dimensionalidad. En este caso, distintas estrategias fueron diseñadas para reducir el tamaño del dataset, ya sea minimizando el número de registros de forma aleatoria, o eliminando las características

irrelevantes del conjunto de datos. Posteriormente, los dataset resultantes de cada una de las estrategias definidas fueron evaluados con el mismo algoritmo, con el objetivo de tener resultados comparables de los desempeños obtenidos. A partir de este proceso iterativo, se logró reducir el tamaño del dataset, considerando únicamente las características relevantes para el proceso predictivo, tanto para la tarea de regresión como para la de clasificación.

5.3 HERRAMIENTAS

Las herramientas necesarias para llevar a cabo el proyecto son principalmente las clases, métodos y atributos existentes en sklearn para realizar el análisis descriptivo de variables, los métodos de reducción y el proceso de modelamiento. De igual forma, los métodos de Pyspark fueron empleados inicialmente para el preprocesamiento y agregación de los datos de los datasets desagregados considerados como Big Data, tal como Historical Transactions.

6. RESULTADOS

6.1 MÉTRICAS

Los resultados de los distintos algoritmos ajustados tanto para la tarea de regresión como para la de clasificación fueron abordados de forma exhaustiva en la sección 4.3.3 Modelos para la tarea de regresión y 4.3.4 Modelos para la tarea de clasificación.

Sin embargo, en la Tabla 21 se sintetizan los resultados obtenidos de los modelos seleccionados para los dos problemas de analítica planteados.

Tabla 21

Mejores modelos obtenidos según la tarea de analítica

Tarea de analítica	Algoritmo	Hiperparámetros	Métrica de ML Train	Métrica de ML Validation	Métrica de ML Test
Regresión	Random Forest Regressor	max_depth: 15 $max\_features$: $log2$ max_leaf_nodes: 76 $min\_samples\_leaf$: 5 $min\_samples\_split$: 29 $n\_estimators$: 170	R^2 =0.1208	R^2 =0.1071	R^2 =0.1044
Clasificación	Decision Tree Classifier	$criterion$: $entropy$ max_depth: 15 $max\_features$: $None$ max_leaf_nodes: 93 $min\_samples\_leaf$: 10 $min\_samples\_split$: 32 $splitter$: $best$	$ROC\ Score =$ 0.6584	$ROC\ Score =$ 0.6421	$ROC\ Score =$ 0.6077

Nota: Realización propia

6.2 EVALUACIÓN CUALITATIVA

A partir de los resultados obtenidos para la Métrica de ML en Train, Métrica de ML en Validation y Métrica de ML en Test de la Tabla 21, se logra evidenciar que los algoritmos no presentan overfiting para ninguna de las dos tareas. Sin embargo, a partir de estos resultados, se evidencia el underfitting de los modelos.

El underfitting en este sentido, no se debería necesariamente a un límite del algoritmo para modelar la complejidad de los datos, dado que estos fueron ajustados con los hiperparámetros óptimos. En este caso, la razón principal de underfitting se debe a la ausencia de correlación entre las características suministradas en el concurso (y sus posteriores medidas de agrupamiento realizadas sobre estas), y las variables que se buscan predecir para cada una de las tareas.

7. CONCLUSIONES

Las distintas etapas llevadas a cabo en este proyecto necesitan una gran variedad de conceptos provenientes de la analítica, permitiendo reforzar las temáticas abordadas a lo largo de la especialización. Desde el análisis exploratorio de las bases de datos desagregadas procesadas como bigdata, pasando por modelos no supervisados de clustering sobre las bases de datos anexas y sobre todo, el abordar el problema inicial como dos tareas de analítica (regresión y clasificación), hacen de esta monografía un proyecto amplio desde una perspectiva de ciencia de datos.

Una de las principales conclusiones del proyecto realizado para el concurso de kaggle ELO, es la ausencia de relación lineal o polinómica entre las características utilizadas para entrenar los algoritmos de Machine Learning y las variables Target en su versión continua o discreta. Por esta razón, se considera que los algoritmos de Machine Learning entrenados para las tareas de regresión o clasificación no logran representar la complejidad de los datos a pesar de que estos modelos fueron entrenados con los hiperparámetros óptimos. Eso conlleva como consecuencia al underfitting de los algoritmos, el cual se evidencia en las métricas de error seleccionadas para cada una de las tareas.

Como consecuencia del underfitting de los modelos ajustados sería inviable implementar estos algoritmos en un entorno de producción, dado que no cumplen con el criterio mínimo establecido, el cual fue definido en un valor mínimo de 80% en la métrica de error con los valores de test.

Sin embargo, es importante resaltar que a pesar de la falta de correlación entre las características y la variable Target asociada a la tarea de regresión, el procesamiento de los datos realizado durante la

última iteración permitió aumentar la métrica de machine Learning para la tarea de regresión en un 1400% al compararse con el resultado obtenido durante la primera iteración. Esto implica que, si bien el modelo presenta underfitting, este ofrece mucho mejor rendimiento que el algoritmo base de referencia empleado en la primera iteración.

La anonimización de las características suministradas limitan la capacidad de obtener valor y conocimiento de ellos. El concurso suministra diferentes características que podrían relacionarse potencialmente con el puntaje de lealtad si no estuvieran anonimizadas, sin embargo, en el desarrollo del proyecto se pudo constatar que muchas de estas características pierden su utilidad.

Abordar el proyecto desde dos tipos de aprendizaje supervisado otorga un panorama amplio en la implementación de técnicas aplicadas en el proyecto. Del mismo modo, puede ofrecer una nueva oportunidad de mejora en el planteamiento del problema por parte del negocio, ya que podría implementar procesos que se ajusten más hacia sus necesidades y así lograr aumentar la rentabilidad de sus estrategias de fidelización.

8. REFERENCIAS

1. Arthur, D., & Vassilvitskii, S. (2006). *k-means++: The advantages of careful seeding*. Stanford.
2. Bouza, Carlos. (2021). Las curvas roc teoría y herramientas para su uso.
3. Castillo-Botón, C., Casillas-Pérez, D., Casanova-Mateo, C., Ghimire, S., Cerro-Prada, E., Gutierrez, P. A., ... & Salcedo-Sanz, S. (2022). Machine learning regression and classification methods for fog events prediction. *Atmospheric Research*, *272*, 106157.
4. Cheng, Z., Zou, C., & Dong, J. (2019, September). Outlier detection using isolation forest and local outlier factor. In *Proceedings of the conference on research in adaptive and convergent systems* (pp. 161-168).
5. Dahouda, M. K., & Joe, I. (2021). A Deep-Learned Embedding Technique for Categorical Features Encoding. *IEEE Access*, *9*, 114381-114391.
6. Géron, A. (2017). Hands-On Machine Learning with Scikit-Learn and TensorFlow. O'Reilly Media, Inc.
7. Hammad Alharbi, H., & Kimura, M. (2020, August). Missing Data Imputation Using Data Generated By GAN. In *2020 the 3rd International Conference on Computing and Big Data* (pp. 73-77).
8. Jierula, A., Wang, S., Oh, T.-M., & Wang, P. (2021). Study on accuracy metrics for evaluating the predictions of damage locations in deep piles using artificial neural networks with acoustic emission data. Applied Sciences (Basel, Switzerland), 11(5), 2314. https://doi.org/10.3390/app11052314
9. Kaggle.com. 2019. Elo Merchant Category Recommendation | Kaggle. [online] Available at: <https://www.kaggle.com/c/elo-merchant-category-recommendation> [Accessed 29 May 2022].

10. Lahura, E. (2003). *El coeficiente de correlación y correlaciones espúreas* (Vol. 218). Pontificia Universidad Católica del Perú, Departmento de Economía.
11. Saunders, L. J., Russell, R. A., & Crabb, D. P. (2012). The coefficient of determination: what determines a useful R2 statistic? Investigative Ophthalmology & Visual Science, 53(11), 6830–6832. https://doi.org/10.1167/iovs.12-10598
12. Schleich, M., Olteanu, D., Abo Khamis, M., Ngo, H. Q., & Nguyen, X. (2019, June). A layered aggregate engine for analytics workloads. In *Proceedings of the 2019 International Conference on Management of Data* (pp. 1642-1659).
13. Shi, J., He, Q., & Wang, Z. (2019). GMM clustering-based decision trees considering fault rate and cluster validity for analog circuit fault diagnosis. *IEEE Access*, *7*, 140637-140650.
14. *Supervised vs. Unsupervised learning: What's the difference?* (2021, marzo 12). Ibm.Com. https://www.ibm.com/cloud/blog/supervised-vs-unsupervised-learning
15. Vega-Vilca, J. C., & Guzmán, J. (2011). Regresión PLS y PCA como solución al problema de multicolinealidad en regresión múltiple. *Revista de Matemática Teoría y Aplicaciones*, *18*(1), 09-20.
16. XGBoost Documentation — xgboost 2.0.0-dev documentation. (n.d.). Readthedocs.Io. Retrieved May 30, 2022, from https://xgboost.readthedocs.io/en/latest/index.html
17. Yang, L., & Shami, A. (2020). On hyperparameter optimization of machine learning algorithms: Theory and practice. *Neurocomputing*, *415*, 295-316.
18. Zaki, M. J., Meira Jr, W. (2014). *Data mining and analysis: fundamental concepts and algorithms*. Cambridge University Press.
19. Bancomundial.org. 2019. Crecimiento del PIB (% anual) - Brazil. (s/f). [online] Available at: <https://datos.bancomundial.org/indicator/NY.GDP.MKTP.KD.ZG?end=2020&locations=BR&stat=2000&view=chart> [Accessed 22 February 2022].