Implementación de un modelo predictivo para la optimización de ventas empresariales a partir del procesamiento y análisis de datos no estructurados

Oscar Emilio Cubillos Torres

Naywa Dallys Mejia Escandon

Asesor

Felipe Alexander Pipicano Guzmán

Universidad Nacional Abierta y a Distancia UNAD

Escuela de Ciencias Básicas, Tecnología e Ingeniería ECBTI

Especialización en Ciencia de Datos y Analítica

2025

Resumen

El presente proyecto surgió de la necesidad crítica de optimizar las ventas de una empresa del sector comercial cuya principal dificultad era la gestión de su información. Inicialmente, los datos estaban dispersos y no estructurados en diversos formatos (imágenes, PDF, hojas de cálculo y registros manuales), lo que impedía cualquier análisis estratégico. Nuestra primera labor fue limpiar y unificar los datos utilizando herramientas como OpenRefine y Python, logrando estructurar la información para su procesamiento. Luego realizamos un análisis exploratorio que permitió identificar patrones de comportamiento comercial, canales de venta y distribución geográfica de clientes, comprendiendo así el funcionamiento diario de la empresa.

Con este conocimiento, diseñamos un modelo predictivo mediante regresión supervisada en Google Colab (Python) y Pycaret. Su eficacia se potenció con el enriquecimiento de datos a través de la técnica de la ventana corrediza, incorporando variables como cliente, mes, frecuencia de compras y totales acumulados. El modelo final (Linear Regression) obtuvo métricas adecuadas (MAE, RMSE y R^2 de 0,8272), con un 83% de precisión predictiva. Esto brinda a la empresa una herramienta para anticipar ventas, detectar oportunidades y respaldar decisiones estratégicas, demostrando el valor de la ciencia de datos para transformar información desordenada en acciones reales con impacto positivo.

Palabras clave: Datos no estructurados, Limpieza de datos, Análisis exploratorio, Modelo predictivo, Pycaret.

Abstract

This project arose from the critical need to optimize sales in a commercial sector company whose main difficulty was information management. Initially, the data were scattered and unstructured across various formats (images, PDFs, spreadsheets, and manual records), making any strategic analysis impossible. Our first task was to clean and unify the data using tools such as OpenRefine and Python, successfully structuring the information for processing. We then performed an exploratory analysis that allowed us to identify commercial behavior patterns, sales channels, and the geographical distribution of clients, gaining a clear understanding of the company's daily operations.

With this knowledge, we implemented a predictive model using regression in Google Colab (Python) and PyCaret. Its effectiveness was enhanced through data enrichment with the sliding window technique, incorporating variables such as customer, month, purchase frequency, and accumulated totals. The final model (Linear Regression) achieved suitable metrics (MAE, RMSE, and R^2 of 0.8272) with an 83% prediction accuracy. This provides the company with a tool to anticipate sales, identify opportunities, and support strategic decision-making, demonstrating the value of data science in transforming disorganized information into real actions with a positive impact.

Keywords: Unstructured data, Data cleaning, Exploratory analysis, Predictive model, PyCaret.

Tabla de Contenido

Introducción 8

Justificación 10

Objetivos 12

Objetivo General 12

Objetivos Específicos 12

Marco de Referencia 13

Marco Conceptual 13

Definiciones Conceptuales 13

Marco Teórico 14

Metodología 17

Enfoque Metodológico 17

Análisis Exploratorio de Datos 20

Punto 2 Estadísticas de la Columna Total 21

Punto 3 Selección de Variables 22

Punto 4 Enriquecimiento de la Base (Técnica de la Ventana Corrediza) 23

Punto 5 Nuevo Análisis Exploratorio 25

Punto 6 Modelado con PyCaret 26

Funcionamiento 26

Aplicación 27

Punto 7 Comparación de los Modelos 29

Resultado del Conjunto de Prueba 34

Punto 9 Análisis Grafico 35

Parte 10 Análisis del Desempeño del Modelo sobre el Conjunto de Prueba 41

Parte 11 Limitaciones del Modelo 43

Parte 12 Recomendaciones 44

Conclusiones 46

Recomendaciones 47

Referencias Bibliográficas 48

Lista de Figuras

Figura 1 *Base de Datos Estructurada* 21

Figura 2 *Código para Extraer las Medidas de Tendencia de la Columna Total* 22

Figura 3 *Transformación de Fecha, Extracción del Mes y Creación de Columna* 24

Figura 4 *Configuración y Creación de Columnas Nuevas* 24

Figura 5 *Base de Datos Enriquecida* 25

Figura 6 *Código del Desfase para Evitar Fuga Temporal* 28

Figura 7 *Validación Cruzada y Normalización* 29

Figura 8 *Resultados Obtenidos en la Comparación de Modelos* 30

Figura 9 *Mejores Modelos Según sus Métricas* 31

Figura 10 *Código para el Entrenamiento y Prueba del Modelo Linear Regression* 32

Figura 11 *Métricas con los 5 Pliegues en Entrenamiento y en el Modelo de Regresión Lineal* . 32

Figura 12 *Código para Obtener los Percentiles Junto a las Métricas* 34

Figura 13 *Percentiles del Error Absoluto* 34

Figura 14 *Dispersión (Real vs Predicción)* 35

Figura 15 *Residuales vs Predicho* 36

Figura 16 *Prueba Breusch-Pagan* 37

Figura 17 *Histograma (Residuales)* 38

Figura 18 *Importancia de las Variables Predictoras* 39

Figura 19 *Tabla de Coeficientes con sus Respectivos Signos* 40

Figura 20 *Top 10 Clientes que Más Comprarán* 41

Figura 21 *Top 10 Clientes que Menos Comprarán* 42

Lista de tablas

Tabla 1 *Herramientas y Recursos* 19

Introducción

Es un hecho que, en el mundo de hoy las empresas se ven obligadas a estar en constante mejora**.** Para cualquier negocio, la clave para seguir existiendo y crecer es saber anticiparse a los cambios del mercado, por esta razón el tomar decisiones equivocas basándose en datos, es un riesgo que no se puede correr, por el contrario, esta debe ser la base para sobresalir ante la competencia. El gran problema es que muchas organizaciones siguen almacenando su información de forma no estructurada, en una variedad de formatos distintos, y eso hace que sea casi imposible aprovecharla.

La idea del proyecto nació debido a la necesidad de la empresa del sector comercial, ya que su historial de ventas se encontraba en caos (facturas en fotos, tablas de Excel, documentos pdf, etc). Debido a esta falta de organización, era imposible para ellos ver qué estaban comprando sus clientes o qué tendencias podían aprovechar. El objetivo principal se convirtió entonces en la transformación de todos estos datos en una fuente de información confiable que se pudiera usar para predecir sus ventas futuras.

El enfoque fue paso a paso, primero, limpiar y estructurar los datos, después analizar todo para entender el negocio y al final, crear el modelo predictivo. Para esto, se apoyó en herramientas como Python, OpenRefine, SQLite y técnicas como la ventana corrediza y Pycaret para la comparación de distintos modelos. Con este proceso, se buscó construir un modelo que permitiera predecir el comportamiento de ventas de cada mes, brindando así una base sólida para su planificación.

Más allá de lo técnico, el trabajo demuestra que el valor de organizar información que no es muy útil hasta convertirla en una herramienta real para decidir es de vital importancia para el crecimiento empresarial. Los resultados que se obtuvieron no solo hacen que se entienda mejor

el funcionamiento del negocio, sino que también permite diseñar estrategias más inteligentes enfocadas en los clientes más potenciales. En este informe, se detalló cada fase del proceso, desde el problema hasta el diseño y la evaluación del modelo. El objetivo es dejar una solución clara y útil que otras empresas puedan aplicar.

Justificación

En el entorno empresarial, la información que se genera a diario como ventas, facturación y contacto con clientes, muchas veces representa una oportunidad perdida. Gran parte de estos datos se almacena de forma poco eficiente provocando que, aunque la empresa posea un gran volumen de datos, sea casi imposible convertirlos en conocimiento útil para la toma de decisiones.

Esta dificultad fue el problema central que motivó este proyecto. La empresa que se analizó disponía de un historial de ventas muy valioso, pero incompleto en muchos formatos. No existía una base de datos limpia y unificada que permitiera la lectura de comportamientos de compra en los clientes, ni la identificación de los ciclos de alta y baja demanda. Por esta razón cualquier intento de planificar una estrategia comercial no tenía un respaldo real.

El trabajo se centró, entonces, en transformar esta información en una herramienta estratégica.

El proceso tuvo dos fases clave:

Estructuración: Se organizaron y limpiaron los datos para convertirlos en una base confiable.

Análisis de Patrones: Se analizó el comportamiento histórico para identificar los factores clave (clientes recurrentes, meses de mayor venta y que canales son realmente útiles).

Al tener la base del historial clara, el proyecto avanzó con la construcción de un modelo predictivo. Al entregar esta herramienta permite anticipar el movimiento de las ventas, que le da a la empresa la capacidad de tomar decisiones inteligentes (como ajustar inventarios o enfocar campañas) que antes no tenía.

Este proyecto es muy valioso ya que demuestra que la ciencia de datos no es solo para las grandes empresas. Este trabajo prueba que cualquier organización, con las herramientas adecuadas, puede utilizar modelos predictivos para basar sus decisiones en datos. La metodología usada (limpieza, análisis exploratorio, modelado y evaluación) de hecho, es fácil de aplicar a otras empresas con el mismo problema.

En resumen, el proyecto contribuye a dar una respuesta real, convirtiendo los datos bien gestionados en el valor principal para generar ventajas reales y duraderas en el mercado.

Objetivos

Objetivo General

Diseñar e implementar un modelo predictivo fundamentado en inteligencia de negocios, con el fin de optimizar las ventas empresariales, mejorar la toma de decisiones estratégicas y aumentar la eficiencia comercial.

Objetivos Específicos

Estructurar una base de datos empresarial no estructurada, a través de técnicas de limpieza, transformación y normalización, para su uso en análisis predictivo.

Realizar un análisis exploratorio de los datos de ventas, identificando patrones, tendencias y relaciones útiles para la toma de decisiones.

Diseñar un modelo predictivo supervisado que permita anticipar el comportamiento de ventas y apoyar la planificación comercial.

Evaluar el modelo predictivo mediante métricas de rendimiento como MAE, RMSE, R2, y analizar su impacto potencial en las decisiones estratégicas de ventas.

Marco de Referencia

Marco Conceptual

El proyecto se fundamenta en las técnicas de ciencia de datos para la gestión en información no estructurada en el entorno empresarial, ya que el uso de los datos, cuando se encuentran dispersos en diferentes formatos, carece de utilidad para la toma de decisiones estratégicas, hasta que son transformados en información organizada, confiable y disponible para el análisis (Rahm & Do, 2000; Kandel et al., 2011).

Los conceptos tales como la limpieza de datos, el análisis exploratorio, la ventana corrediza, los modelos predictivos supervisados y la regresión lineal constituyen los fundamentos conceptuales que permiten a las organizaciones anticipar el comportamiento de las ventas, optimizar sus recursos y respaldar las decisiones estratégicas. La literatura reciente respalda este enfoque , la limpieza de datos ha sido identificada como una etapa crítica para garantizar la calidad de la información antes del modelado (Kandel et al., 2011; Rahm & Do, 2000), el análisis exploratorio es indispensable para comprender relaciones y patrones (Tukey, 1977), y técnicas como la ventana corrediza en series temporales ha mostrado efectividad para capturar relaciones temporales y patrones recurrentes, mejorando así el desempeño predictivo de los modelos (Ensafi, Khalilzadeh, & Maleki, 2022; Li, 2022).

Definiciones Conceptuales

Datos no estructurados: Los datos no estructurados son información guardada en múltiples formatos que carecen de una estructura uniforme, lo cual constituye un reto significativo para su análisis directo mediante herramientas convencionales (Kandel et al., 2011).

Limpieza de datos: Este proceso consiste en la depuración de datos para corregir, estandarizar y validar la información, retirando cualquier elemento que comprometa su calidad, como valores nulos, duplicados o errores de digitación (Rahm & Do, 2000).

Análisis exploratorio de datos (EDA): Es el conjunto de técnicas estadísticas y gráficas utilizadas para comprender patrones, tendencias y relaciones entre variables (Tukey, 1977).

Ventana corrediza: Es una técnica de series temporales que se emplea para organizar el historial reciente, como en este caso el comportamiento de los clientes y sus compras en meses anteriores, transformando estos datos en nuevas variables predictoras (Ensafi et al., 2022; Li, 2022).

Modelo predictivo: Es una herramienta estadística de aprendizaje automático (machine learning) que permite anticipar un comportamiento futuro a partir de información histórica (Quispe et al., 2024).

Regresión lineal: Es un modelo estadístico supervisado que busca explicar la relación entre una variable dependiente y una o varias variables independientes, siendo ampliamente utilizado en la predicción de ventas y estudios económicos (Alves, Fonseca, & Maciel, 2021; Morales, Ramírez, & Rodríguez, 2019).

Marco Teórico

El análisis predictivo es una herramienta importante para cualquier empresa que busque ir un paso adelante, hoy en día es la manera de optimizar procesos y sobre todo de anticipar cómo se moverá el mercado, las compañías del sector comercial, en particular, están sacando mucho provecho de esto ya que convierten cantidades de datos que se encuentran dispersos en información clara y estructurada para tomar decisiones realmente estratégicas (Caro, Guardiola, & Ortiz, 2018; Zerpa, García, & Izquierdo, 2020).

En esta área la ciencia de datos aplicada a las ventas demuestra su enorme potencial, aplicando las herramientas necesarias como los modelos de regresión supervisada y las técnicas de machine learning, lo cual dejan de ser una simple tendencia para transformarse directamente en una ventaja competitiva concreta.

Hay evidencia clara de su impacto, Alves, Fonseca y Maciel (2021) demostraron que, en el comercio minorista, estos modelos son excelentes para estimar ventas, lo que facilita muchísimo la planificación de inventarios y el diseño de campañas que realmente funcionen, de igual forma en un entorno más local. Morales, Ramírez y Rodríguez (2019) mostraron cómo en el sector de alimentos en Colombia, usaron modelos estadísticos y redes neuronales para pronosticar ventas ayudar a reaccionar más rápido a cualquier cambio inesperado en la demanda.

Un detalle importante para que el modelado predictivo funcione bien es el enriquecimiento de las variables, en la literatura más reciente, se resalta la importancia del uso de las técnicas, como lo es la ventana corrediza, siendo esta herramienta la que organiza el historial de cada cliente en periodos recientes, permitiendo así un panorama más preciso de su comportamiento más actual.

Ensafi, Khalilzadeh y Maleki (2022) y Li (2022) confirman que esta estrategia es muy efectiva con series de tiempo de ventas, ya que permite a los modelos capturar patrones de comportamiento recurrentes y por temporadas (patrones que se repiten cada cierto tiempo, especialmente ventas altas en diciembre o bajas en enero), los cuales de otra manera se perderían si no se aplican. Por esto esta técnica ofrece a los modelos una mejor base para realizar predicciones más precisas.

En cuanto a las herramientas prácticas, el uso de las librerías automatizadas como PyCaret ha ganado mucha popularidad, Quispe, Quispe, Calvo y Toledo (2024),por ejemplo,

analizaron su aplicación para predecir el consumo eléctrico, y destacaron lo eficiente que es para comparar muchísimos modelos de machine learning rápidamente y con métricas estandarizadas, esto nos da un gran respaldo para usarla en un proyecto de predicción empresarial como el nuestro, ya que acelera la selección del mejor enfoque para el análisis.

Finalmente, no podemos hablar de modelos sin mencionar la evaluación de su desempeño, ya que es la única forma de saber si son aplicables en el mundo real, en este sentido, las métricas, como el Error Absoluto Medio (MAE), la Raíz del Error Cuadrático Medio (RMSE) y el Coeficiente de Determinación (R2) son las más usadas para medir qué tan precisas son las predicciones (Wedel & Kannan, 2016).

Para este proyecto, estas métricas fueron esenciales para validar un modelo de linear regression que, de hecho, logró explicar casi el 83% de la variabilidad de los datos, lo que nos brinda una base muy sólida para anticipar las tendencias de compras futuras.

Metodología

Enfoque Metodológico

Este proyecto corresponde a una investigación cuantitativa aplicada, dado que se busca medir, analizar y modelar datos numéricos con el fin de resolver un problema propio de la empresa. En estudios recientes, se reconoce que el enfoque cuantitativo permite obtener resultados útiles en diferentes escenarios mediante métricas objetivas (Ghanad, 2023). De igual forma la investigación es de tipo predictiva y exploratoria, pues no solo explora patrones presentes en los datos históricos, sino que también construye modelos para anticipar comportamientos futuros, de acuerdo con metodologías actuales que integran análisis de series temporales con aprendizaje automático (Research.com, 2025).

El desarrollo metodológico se estructuró en cuatro fases principales, las cuales fueron aplicadas directamente al proyecto:

1. Preparación de datos: se consolidaron y limpiaron las fuentes de información que se encontraban dispersas (facturas, PDFs y hojas de cálculo) utilizando las herramientas OpenRefine y Python, hasta lograr obtener una base estructurada en SQLite. Este paso de limpieza de datos es considerado crítico en estudios recientes que demuestran cómo el *data cleaning* impacta directamente en la calidad de los resultados analíticos (Gupta et al., 2023).

2. El análisis exploratorio: se revisaron los patrones de ventas, identificando los clientes más recurrentes, los meses de mayor compra y los canales más efectivos, lo que permitió comprender el comportamiento histórico de la empresa. Investigaciones actuales resaltan cómo el EDA ayuda a descubrir relaciones ocultas y patrones de comportamiento relevantes en distintos entornos de negocio (Naureen, 2023).

3. La construcción del modelo: se aplicó mediante la librería de PyCaret para la comparación de los algoritmos de regresión, en ellos seleccionando la linear regresión como el modelo final por su rendimiento y facilidad de interpretación.

4. La evaluación y validación: el modelo se probó con validación cruzada de cinco pliegues. Estudios recientes confirman que esta técnica sigue siendo una de las estrategias más confiables para la selección de modelos y la prevención del sobreajuste en *machine learning* (Teodorescu & Obreja Brașoveanu, 2025; Charilaou et al., 2022). De igual manera en se obtuvieron métricas de desempeño satisfactorias (MAE, RMSE y un R^2 cercano al 83%), lo que confirmó su capacidad predictiva para anticipar el comportamiento de las ventas.

Tabla 1

Herramientas y Recursos

Recurso	Descripción	Presupuesto
. Equipo Humano	1 analista de datos encargado de la planificación y supervisión general del proyecto y el análisis avanzado del proyecto.	$4’000.000 mensuales 4x4’000.000 = 16’000.000
	1 técnico en sistemas encargado del mantenimiento de computadores y actualizaciones de software.	$2’000.000 mensuales 4x2’000.000 = 8’000.000
	1 administrador de datos encargado de realizar la transformación y limpieza de datos	$2’500.000 mensuales 4x2’500.000 = 10’000.000
2. Equipos y Software	3 computadoras con procesadores core i 7, ram de 20gb y ssd de 500gb	$ 7’500.000
	Servidor local	$2’000.000 mensuales 4x2’000.000 = 8’000.000
	Red de internet	$160.000 mensual 4x160.000 =640.000
	Licencia office	$500.000
. Viajes y Salidas de Campo	Desplazamientos ibagué – bogotá y chía – bogotá 2 veces por mes	$700.000
	Alimentación para tres personas	$750.000
4. Materiales y suministros	Pizarra blanca y marcadores de tablero, papel y bolígrafos.	$280.000
	3 adaptadores de corriente	$120.000
	1 estabilizador de corriente	$45.000
	2 escritorios	$400.000
	1 impresora	$250.000
5. Bibliografía	No aplica	
Total		$53’185.000

Análisis Exploratorio de Datos

Después de realizar la limpieza de datos y tener una base de datos estructurada, con 5 columnas y mil setenta y siete filas, hemos encontrado los siguientes hallazgos:

1). información general

Las cinco columnas contienen nombre de cliente, el cual es una variable categórica ya que es tipo texto.

El total: el cual representa el costo total de la compra realizada por el cliente. (valor de la compra) y es de tipo numérico.

La fecha la cual representa el día, mes y año en que se realizó la compra del cliente (aún no se encuentra en un formato de fecha adecuado, por lo que se considera tipo categórico), se debe tener en cuenta que todas las compras se realizaron durante el año 2023.

La ciudad la cual nos indica el lugar exacto de la compra siendo estas ciudades del territorio colombiano (esta es de tipo categórico).

El vendedor: el cual se divide en dos categorías asesor PAP (puerta a puerta) y Marketing D (marketing digital).

Finalmente se puede observar que en las columnas no existen datos faltantes o nulos.

Figura 1

Base de Datos Estructurada

	A	B	C	D	E
1	Nombre cliente	Total	Fecha	Ciudad	Vendedor
2	MEGA REDES INGENIERIA SAS	1299000	02/01/2023	Madrid	Asesor PAP
3	Ferney Ortiz	70000	02/01/2023	Mosquera	Asesor PAP
4	MEGA REDES INGENIERIA SAS	586000	02/01/2023	Madrid	Asesor PAP
5	MEGA REDES INGENIERIA SAS	904100	16/01/2023	Madrid	Asesor PAP
6	Yerson Rico	740000	16/01/2023	Bogotá	Asesor PAP
7	Manuel Rodríguez	1743000	16/01/2023	Ubaté	Asesor PAP
8	MECCISS SAS	180000	17/01/2023	Mosquera	Asesor PAP
9	RM Gas Natural	1605000	17/01/2023	Mosquera	Asesor PAP
10	MILENA HASBLEIDY SIERRA PARADA	848000	17/01/2023	Cajica	Asesor PAP
11	MILENA HASBLEIDY SIERRA PARADA	691500	17/01/2023	Cajica	Asesor PAP
12	WG plomeria y gas	1315000	18/01/2023	Chía	Asesor PAP
13	Uriel Villamil	1075000	18/01/2023	Ubaté	Asesor PAP
14	Martha Pinzón	362000	18/01/2023	Mosquera	Asesor PAP
15	Mundial Gas y Resistencia	629000	19/01/2023	Bogotá	Asesor PAP
16	Carlos Parra	2916000	19/01/2023	Zipaquirá	Asesor PAP
17	WG plomeria y gas	3574000	19/01/2023	Chía	Asesor PAP
18	Victor Sierra	659000	19/01/2023	Mosquera	Asesor PAP
19	Ferney Ortiz	1038000	19/01/2023	Mosquera	Asesor PAP
20	RM Gas Natural	1388000	19/01/2023	Mosquera	Asesor PAP
21	Samuel Plaza	709000	20/01/2023	Cajica	Asesor PAP
22	Carlos Parra	600000	20/01/2023	Zipaquirá	Asesor PAP

Punto 2 Estadísticas de la Columna Total

Las ventas en promedio son de: 951,045.06 pesos.

Con una Desviación estándar de 857,131.07 pesos, podemos decir que hay bastante variabilidad entre los valores de las ventas.

Dado que la mediana es de 684,000.00 el (50%) podemos observar que la mitad de las ventas son menores a este valor y la otra mitad son mayores, y siendo que este valor es menor que este promedio, podemos decir que hay valores de ventas muy altas que llevan el promedio a elevarse.

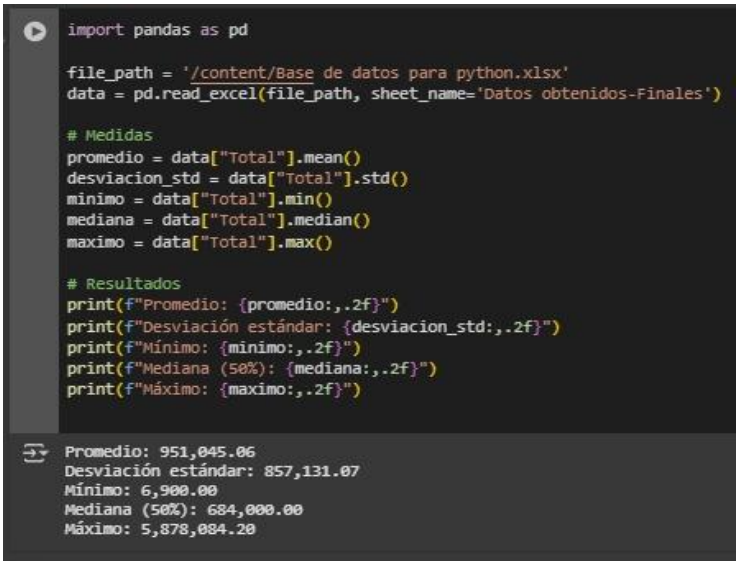
El valor mínimo observado es de 6.900.00 pesos indicando que hay ventas muy pequeñas comparadas con el promedio, esto quiere decir que existen clientes con aportes económicos poco significativos.

Tenemos un valor Máximo de 5,878,084.20 pesos que al ser tan grande puede ser una venta en particular y no aportaría mucho a la tendencia general de venta.

Después de obtener estos valores se podría decir que existen muchas ventas pequeñas y medianas en comparación de las ventas grandes, lo que podría traer implicaciones negativas a la empresa al depender de las ventas grandes, ya que, al perder estos clientes grandes la empresa podría llegar al cierre.

Figura 2

Código para Extraer las Medidas de Tendencia de la Columna Total



```
import pandas as pd

file_path = '/content/Base de datos para python.xlsx'
data = pd.read_excel(file_path, sheet_name='Datos obtenidos-Finales')

# Medidas
promedio = data["Total"].mean()
desviacion_std = data["Total"].std()
minimo = data["Total"].min()
mediana = data["Total"].median()
maximo = data["Total"].max()

# Resultados
print(f"Promedio: {promedio:,.2f}")
print(f"Desviación estándar: {desviacion_std:,.2f}")
print(f"Mínimo: {minimo:,.2f}")
print(f"Mediana (50%): {mediana:,.2f}")
print(f"Máximo: {maximo:,.2f}")
```

```
Promedio: 951,045.06
Desviación estándar: 857,131.07
Mínimo: 6,900.00
Mediana (50%): 684,000.00
Máximo: 5,878,084.20
```

Punto 3 Selección de Variables

Para la creación de nuestro modelo predictivo, ya que tenemos claro que el objetivo es predecir ventas futuras, nuestra variable objetivo será el total (valor de la venta total) mientras que para la selección de las variables predictoras diremos lo siguiente:

La variable objetivo será el total de ventas.

La variable nombre cliente se descartará, ya que es un identificador único y no aporta características que ayuden a describir el comportamiento de compra de un cliente, además de

esto puede provocar sobre ajustes haciendo que el modelo memorice los clientes que compran mucho y aprendan patrones que favorecen a los clientes que ya conocen (Sesgo).

De la variable fecha se puede extraer el año, el mes y el día de la compra lo cual nos serviría para identificar patrones o tendencias en las compras de los clientes.

La ciudad puede ayudar al modelo a identificar en que ciudades se encuentran los clientes más potenciales.

El canal de ventas (vendedor) lo escogemos como variable predictora, ya que presenta una correlación con el comportamiento de compra.

Punto 4 Enriquecimiento de la Base (Técnica de la Ventana Corrediza)

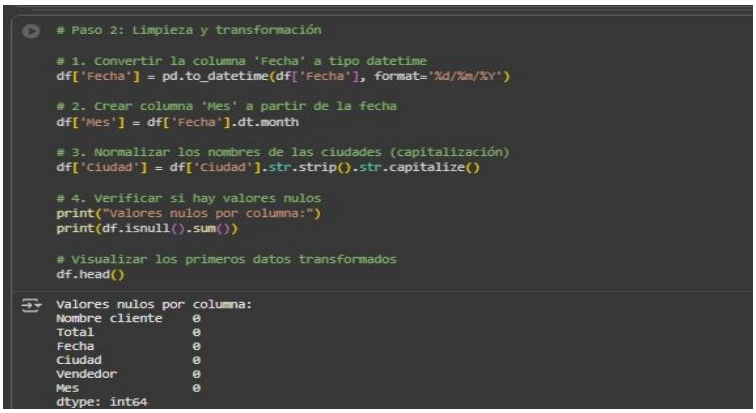
Para fortalecer la información de nuestra base de datos se utilizó la técnica conocida como la ventana corrediza (rolling window) ya que la base de datos tenía pocas variables que nos ayudaran a observar los patrones de comportamiento de los clientes, dicha técnica se aplicó sobre la columna de la fecha (después de llevar la fecha a un formato correcto y extraer el año y el mes) la cual nos ayudó a crear tres variables nuevas (número de compras en los últimos tres meses y el valor total de la compra en los últimos tres meses) que enriqueció la base de datos, esto ayudara al modelo a predecir de manera más precisa el comportamiento de los clientes.

Para llevar a cabo esta técnica primero se agruparon los datos por cliente, año, mes(año y mes se extrajeron de la fecha) gracias a esto se pudo conseguir el valor de la compra total del mes y el número de compras realizadas en dicho mes, basándonos en estas dos nuevas columnas aplicamos la técnica de la ventana corrediza la cual tuvo un tamaño de 3 meses (donde se tuvo en cuenta la observación de los tres meses anteriores incluyendo al actual de cada cliente, trabajando de tres en tres) finalmente tenemos una base de datos adecuada para suministrar más variables predictoras a nuestro modelo. Así alineamos nuestro enfoque con lo que ya muestran

estudios en ventas: una ventana deslizante que organiza el historial reciente de cada cliente para entrenar mejor el modelo (Li & Zhang, 2022; Gusmão, Moreira, & Tomé, 2021).

Figura 3

Transformación de Fecha, Extracción del Mes y Creación de Columna

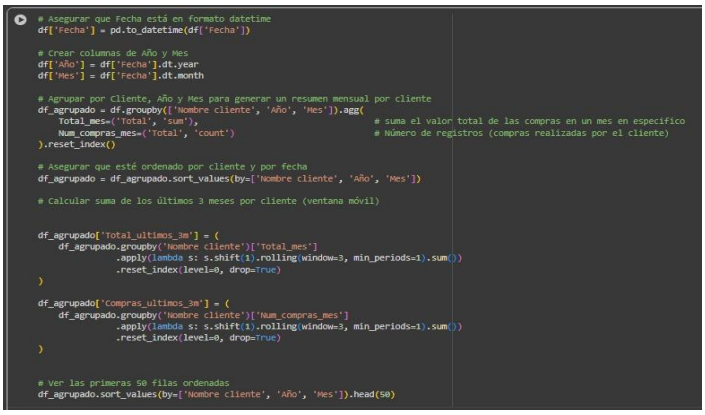


```
# Paso 2: Limpieza y transformación

# 1. Convertir la columna 'Fecha' a tipo datetime
df['Fecha'] = pd.to_datetime(df['Fecha'], format='%d/%m/%Y')

# 2. Crear columna 'Mes' a partir de la fecha
df['Mes'] = df['Fecha'].dt.month

# 3. Normalizar los nombres de las ciudades (capitalización)
df['Ciudad'] = df['Ciudad'].str.strip().str.capitalize()

# 4. Verificar si hay valores nulos
print("Valores nulos por columna:")
print(df.isnull().sum())

# Visualizar los primeros datos transformados
df.head()
```

```
Valores nulos por columna:
Nombre cliente    0
Total             0
Fecha             0
Ciudad            0
Vendedor          0
Mes               0
dtype: int64
```

Figura 4

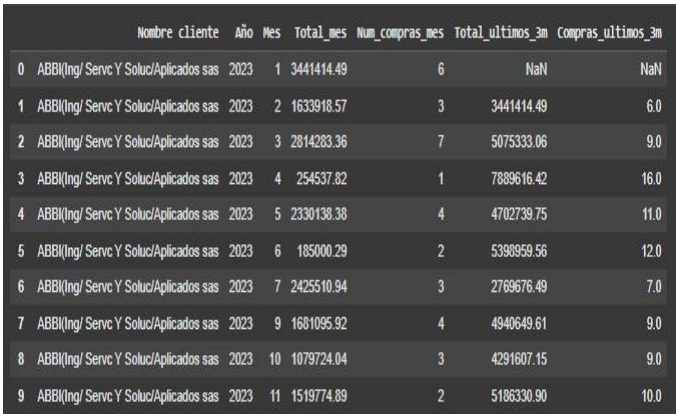
Configuración y Creación de Columnas Nuevas



```
# Asegurar que Fecha está en formato datetime
df['Fecha'] = pd.to_datetime(df['Fecha'])

# Crear columnas de Año y Mes
df['Año'] = df['Fecha'].dt.year
df['Mes'] = df['Fecha'].dt.month

# Agrupar por Cliente, Año y Mes para generar un resumen mensual por cliente
df_agrupado = df.groupby(['Nombre cliente', 'Año', 'Mes']).agg(
    Total_mes=('Total', 'sum'),                    # suma el valor total de las compras en un mes en especifico
    Num_compras_mes=('Total', 'count')             # Número de registros (compras realizadas por el cliente)
).reset_index()

# Asegurar que esté ordenado por cliente y por fecha
df_agrupado = df_agrupado.sort_values(by=['Nombre cliente', 'Año', 'Mes'])

# Calcular suma de los últimos 3 meses por cliente (ventana móvil)


df_agrupado['Total_ultimos_3m'] = (
    df_agrupado.groupby('Nombre cliente')['Total_mes']
               .apply(lambda s: s.shift(1).rolling(window=3, min_periods=1).sum())
               .reset_index(level=0, drop=True)
)

df_agrupado['Compras_ultimos_3m'] = (
    df_agrupado.groupby('Nombre cliente')['Num_compras_mes']
               .apply(lambda s: s.shift(1).rolling(window=3, min_periods=1).sum())
               .reset_index(level=0, drop=True)
)


# Ver las primeras 50 filas ordenadas
df_agrupado.sort_values(by=['Nombre cliente', 'Año', 'Mes']).head(50)
```

Figura 5

Base de Datos Enriquecida



	Nombre cliente	Año	Mes	Total_mes	Num_compras_mes	Total_ultimos_3m	Compras_ultimos_3m
0	ABBI(Ing/ Servc Y Soluc/Aplicados sas	2023	1	3441414.49	6	NaN	NaN
1	ABBI(Ing/ Servc Y Soluc/Aplicados sas	2023	2	1633918.57	3	3441414.49	6.0
2	ABBI(Ing/ Servc Y Soluc/Aplicados sas	2023	3	2814283.36	7	5075333.06	9.0
3	ABBI(Ing/ Servc Y Soluc/Aplicados sas	2023	4	254537.82	1	7889616.42	16.0
4	ABBI(Ing/ Servc Y Soluc/Aplicados sas	2023	5	2330138.38	4	4702739.75	11.0
5	ABBI(Ing/ Servc Y Soluc/Aplicados sas	2023	6	185000.29	2	5398959.56	12.0
6	ABBI(Ing/ Servc Y Soluc/Aplicados sas	2023	7	2425510.94	3	2769676.49	7.0
7	ABBI(Ing/ Servc Y Soluc/Aplicados sas	2023	9	1681095.92	4	4940649.61	9.0
8	ABBI(Ing/ Servc Y Soluc/Aplicados sas	2023	10	1079724.04	3	4291607.15	9.0
9	ABBI(Ing/ Servc Y Soluc/Aplicados sas	2023	11	1519774.89	2	5186330.90	10.0

Finalmente obtenemos las siguientes variables:

Total, de ventas (Variable objetivo)

Mes (cuantitativo)

Numero de compras mensuales (cuantitativo)

Compra total de los últimos tres meses (cuantitativo)

Numero de compras en los últimos tres meses (cuantitativo)

La ciudad (categórica)

Canal de ventas (categórica)

Punto 5 Nuevo Análisis Exploratorio

Este nuevo análisis exploratorio de datos se realiza con la base de datos enriquecida con nuevas variables, para descubrir patrones, relaciones en los datos y así encontrar variables predictoras útiles, detección de tendencias y comportamientos que puedan ayudar al modelo a aprender. Con esto obtenemos nuestras variables predictoras las cuales son:

Año

Mes

Numero de compras mensuales

Total, últimos tres meses

Compras últimos tres meses

Punto 6 Modelado con PyCaret

PyCaret es una biblioteca de código en Python que automatiza el proceso de comparación de modelos de machine learning con pocas líneas de código. Lo hace porque prepara los datos, compara modelos, ajusta sus hiperparámetros, evalúa el desempeño y guarda el modelo final. Todos estos pasos son posibles gracias a funciones integradas que reducen el código necesario y permiten obtener rápidamente la comparación de distintos modelos junto con sus métricas correspondientes. Esta es una gran herramienta dado que permite la reducción en la cantidad de pasos que generalmente se hace para la construcción de un modelo de principio a fin.

Funcionamiento

1. En el **setup()** se le dan indicaciones, indicando cuales son los datos, objetivo o variable de interés, así como las variables predictoras que deseas incluir, seguidamente Pycaret procede a realizar la limpieza y transformación de datos de ser necesaria para después realizar la separación de los datos para el entrenamiento y la prueba (80/20), dicha separación la realiza por defecto a menos que se cambien sus parámetros, finalmente fija las métricas y entrega un entorno configurado listo para que sus funciones inicien la siguiente etapa.

2. Ya listo su entorno procede a la prueba y comparación de modelos mediante la función compare models (), en la cual se realiza una validación cruzada consistente en el entrenamiento, validación y promedio, todo esto cuantas veces lo definamos, finalmente devuelve un ranking

con las métricas (MAE, RMSE, R^2, RMSLE, MAPE, TT (Sec)) y los mejores modelos respecto a la métrica de interés.

3. En esta fase toma el modelo que más se adecua a tus necesidades y por medio de la función **create_model('algoritmo')** lo entrena y muestra las métricas obtenidas, haciendo uso de la misma validación cruzada (cantidad de pliegues).

4. Ahora bien, por medio de la función **tune_model(modelo)** se encarga de los ajustes (de ser necesarios) a los parámetros para mejorar el desempeño del modelo seleccionado.

5. Después de los ajustes necesarios, se encarga de realiza la prueba del modelo con el 20% que guardó para la evaluación, haciendo uso de la función **predict_model ()**, finalmente por medio de la función **save_model (),** brinda la posibilidad se guardarlo para su posterior uso.

Aplicación

Después de ingresar las nuevas variables para enriquecer la base de datos, dado que la variable objetivo (Total_mes) es de tipo continua, se procede al modelamiento por medio de PyCaret, esta librería permitió entrenar y comparar distintos modelos de regresión supervisado, basándose en métricas tales como RMSE, MAE, R2 y MAPE, lo cual permitió la identificación y selección del algoritmo con una mayor capacidad predictiva para nuestro objetivo.

Para el uso y la configuración de PyCaret se realiza la construcción de un nuevo Dataframe llamado datos_nuevos (basándonos en df_agrupado) el cual contiene el nombre de las variables de interés. Para la configuración como primera medida y en busca de evitar un mal entrenamiento como lo es la fuga temporal, el conjunto de datos se dividió en un orden cronológico y se aplicó un desfase de un mes, evitando que se incluyera el mes que se deseaba predecir, con esto se aseguró que el modelo a la hora de predecir el comportamiento de un cliente se base solamente en la información de los meses anteriores.

Figura 6

Código del Desfase para Evitar Fuga Temporal

```
df_agrupado['Total_ultimos_3m'] = (
    df_agrupado.groupby('Nombre cliente')['Total_mes']
               .apply(lambda s: s.shift(1).rolling(window=3, min_periods=1).sum())
               .reset_index(level=0, drop=True)
)

df_agrupado['Compras_ultimos_3m'] = (
    df_agrupado.groupby('Nombre cliente')['Num_compras_mes']
               .apply(lambda s: s.shift(1).rolling(window=3, min_periods=1).sum())
               .reset_index(level=0, drop=True)
)
```

Después de esto se dividieron los datos en un 80% para entrenamiento y 20% para prueba, evitando que esta división se realizara sin mezclas aleatorias y manejando el orden cronológico, dicho orden también se mantuvo para evaluar y seleccionar el modelo por medio de la validación cruzada, ya que, para que fuese adecuada debía respetar su cronología. Por otra parte, dado que existen variables numéricas cuya magnitud es mayor a las otras (Total_ultimos_3m y Compras_ultimos_3m) se normalizaron con el objetivo de que todas las variables numéricas tengan escalas comparables y con esto evitar que las variables grandes dominen y desestabilicen el entrenamiento.

Figura 7

Validación Cruzada y Normalización

```
from sklearn.model_selection import TimeSeriesSplit
from pycaret.regression import setup, compare_models, pull

datos_nuevos = df_agrupado[['Año','Mes','Num_compras_mes',
                            'Total_ultimos_3m','Compras_ultimos_3m','Total_mes']].copy()
datos_nuevos = datos_nuevos.dropna()

VCT = TimeSeriesSplit(n_splits=5)  # 1) Validación cruzada temporal

reg_setup = setup(
    data=datos_nuevos,
    target='Total_mes',
    session_id=123,

    data_split_shuffle=False,   # 2) Split train/test SIN barajar (respeta el tiempo)
    fold_shuffle=False,         #    Pliegues SIN mezcla
    fold_strategy=VCT,          #    VC temporal (pasado -> futuro)

    normalize=True,             # 3) Normalización (opcional pero útil al comparar modelos)
    verbose=False
)

best_model = compare_models(sort='RMSE')
leaderboard = pull().copy()
leaderboard
```

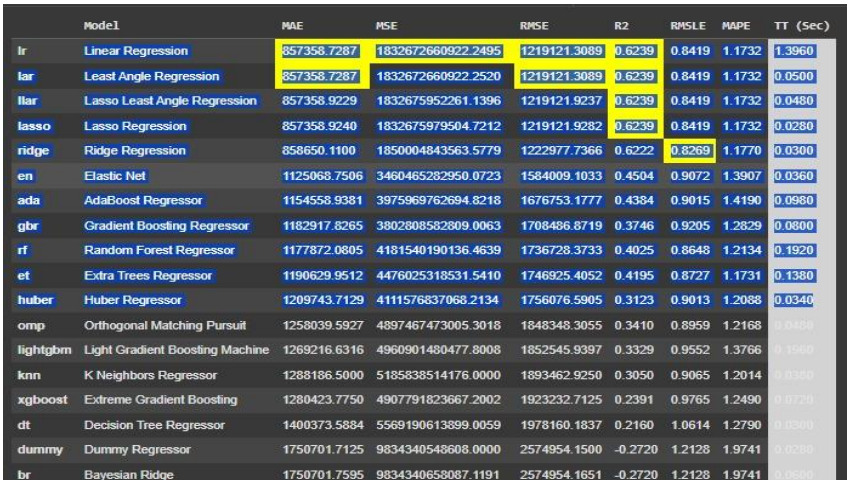
Punto 7 Comparación de los Modelos

Después de realizar la comparación de los modelos basándose en el RMSE (ya que penaliza de forma más fuerte los errores grandes, los cuales suelen ser los más costosos en un negocio) se obtuvieron los siguientes resultados:

Nota: También se tuvieron en cuenta las métricas MAE y R^2 pero como métrica fuerte RMSE.

Figura 8

Resultados Obtenidos en la Comparación de Modelos



	Model	MAE	MSE	RMSE	R2	RMSLE	MAPE	TT (Sec)
lr	Linear Regression	857358.7287	1832672660922.2495	1219121.3089	0.6239	0.8419	1.1732	1.3960
lar	Least Angle Regression	857358.7287	1832672660922.2520	1219121.3089	0.6239	0.8419	1.1732	0.0500
llar	Lasso Least Angle Regression	857358.9229	1832675952261.1396	1219121.9237	0.6239	0.8419	1.1732	0.0480
lasso	Lasso Regression	857358.9240	1832675979504.7212	1219121.9282	0.6239	0.8419	1.1732	0.0280
ridge	Ridge Regression	858650.1100	1850004843563.5779	1222977.7366	0.6222	0.8269	1.1770	0.0300
en	Elastic Net	1125068.7506	3460465282950.0723	1584009.1033	0.4504	0.9072	1.3907	0.0360
ada	AdaBoost Regressor	1154558.9381	3975969762694.8218	1676753.1777	0.4384	0.9015	1.4190	0.0980
gbr	Gradient Boosting Regressor	1182917.8265	3802808582809.0063	1708486.8719	0.3746	0.9205	1.2829	0.0800
rf	Random Forest Regressor	1177872.0805	4181540190136.4639	1736728.3733	0.4025	0.8648	1.2134	0.1920
et	Extra Trees Regressor	1190629.9512	4476025318531.5410	1746925.4052	0.4195	0.8727	1.1731	0.1380
huber	Huber Regressor	1209743.7129	4111576837068.2134	1756076.5905	0.3123	0.9013	1.2088	0.0340
omp	Orthogonal Matching Pursuit	1258039.5927	4897467473005.3018	1848348.3055	0.3410	0.8959	1.2168	0.0480
lightgbm	Light Gradient Boosting Machine	1269216.6316	4960901480477.8008	1852545.9397	0.3329	0.9552	1.3766	0.1560
knn	K Neighbors Regressor	1288186.5000	5185838514176.0000	1893462.9250	0.3050	0.9065	1.2014	0.0380
xgboost	Extreme Gradient Boosting	1280423.7750	4907791823667.2002	1923232.7125	0.2391	0.9765	1.2490	0.0720
dt	Decision Tree Regressor	1400373.5884	5569190613899.0059	1978160.1837	0.2160	1.0614	1.2790	0.0360
dummy	Dummy Regressor	1750701.7125	9834340548608.0000	2574954.1500	-0.2720	1.2128	1.9741	0.0280
br	Bayesian Ridge	1750701.7595	9834340658087.1191	2574954.1651	-0.2720	1.2128	1.9741	0.0600

Al analizar los resultados obtenidos podemos observar que se obtuvo una tabla con 19 modelos junto a sus métricas (MAE, MSE, RMSE, R2, RMSLE, MAPE y TT (sec)) distribuidas en 9 columnas, las cuales mostraron que los tres mejores modelos son linear regression, least angle regression y lasso least angle regression, ya que nos indican que tanto el MAE como el RMSE y el R2 definieron mejores valores para estos.

MAE (Mean Absolute Error) el cual entre más bajo es mucho mejor.

MSE (Mean Squared Error) el cual entre más bajo es mucho mejor.

RMSE (Root Mean Squared Error) trabaja en la misma unidad de objetivo y penaliza fuertemente los errores grandes, entre más bajo es mucho mejor.

R2 (Coeficiente de determinación) este nos indica que tanto porcentaje explican las variables predictoras el comportamiento de la variable objetivo, este entre más alto será mejor.

RMSLE (Root Mean Squared Log Error) este se utiliza cuando se le da más importancia al porcentaje del error, ya que trabaja comprimiendo grandes magnitudes por medio de

logaritmos, entre más bajo mejor. Debemos tener en cuenta que su uso no es ideal cuando se necesita dar interpretación al error en pesos.

MAPE (Mean Absolute Percentage Error) este nos indica en promedio que tanto se equivoca el modelo por lo tanto entre más bajo es mejor.

TT (sec) (Train Time) este nos indica que tanto tiempo en segundos tarda el modelo en entrenarse.

Figura 9

Mejores Modelos Según sus Métricas

	Model	MAE	MSE	RMSE	R2	RMSLE	MAPE	TT (Sec)
lr	Linear Regression	857358.7287	1832672660922.2495	1219121.3089	0.6239	0.8419	1.1732	1.3960
lar	Least Angle Regression	857358.7287	1832672660922.2520	1219121.3089	0.6239	0.8419	1.1732	0.0500
llar	Lasso Least Angle Regression	857358.9229	1832675952261.1396	1219121.9237	0.6239	0.8419	1.1732	0.0480

Como apreciación dado que linear regression y least angle regression tienen métricas similares podemos escoger linear regression por ser el modelo más adecuado, dado que tenemos pocas variables y no tiene hiperparámetros importantes entonces no requiere ajustar el modelo, lo que lo hace muy sencillo.

Después de seleccionar el modelo (linear regression) se procedió al entrenamiento de este, usando nuevamente la validación cruzada temporal, este reentrenamiento se realiza con el mismo conjunto que se usó para la selección del modelo con PyCaret es decir, 80% para entrenamiento con los meses antiguos y 20% para realizar la prueba con los más recientes, además de esto se tuvo en cuenta evitar mezclas aleatorias, se respetó la cronología de los datos y la normalización de las cantidades numéricas. Para la validación cruzada temporal dentro del 80% de los datos se realizó con cinco pliegues(repeticiones) ya que con esto es suficiente para tener un equilibrio entre la estabilidad y el tiempo de ejecución del modelo. Por otra parte, se

realizó la imputación de valores faltantes (llenar lugares vacíos) llenando con la media de la columna correspondiente, no hay que olvidar que se usaron las mismas variables predictoras, así como la misma variable objetivo.

Figura 10

Código para el Entrenamiento y Prueba del Modelo Linear Regression

```
# Preparación, entrenamiento Lineal Regression con el 80%, evaluación 20%  y almacenamiento
from sklearn.model_selection import TimeSeriesSplit
from pycaret.regression import setup, create_model, finalize_model, predict_model, save_model

# 1) Selección de variables
datos_nuevos = df_agrupado[['Año','Mes','Num_compras_mes',
                            'Total_ultimos_3m','Compras_ultimos_3m','Total_mes']].dropna().copy()

# 2) Validación cruzada temporal y pliegues 80/20 sin mezclar
VCT = TimeSeriesSplit(n_splits=5)
reg_setup = setup(
    data=datos_nuevos,
    target='Total_mes',
    session_id=123,
    train_size=0.8,            # 80% train / 20% prueba
    data_split_shuffle=False,  # Respetamos cronología
    fold_shuffle=False,        # Pliegues sin mezcla
    fold_strategy=VCT,         # Validación cruzada temporal (pasado -> futuro)
    normalize=True,            # escalado de variables predictoras (features) numéricas
    verbose=False
)

# 3) Entrenamiento de nuestro modelo elegido (Regresión Lineal)
base = create_model('lr')

# 4) Finalizamos el train y evaluar  (holdout)
final = finalize_model(base)
holdout_df = predict_model(final)   # imprimimos métricas finales y mostramos predicciones

# 5) Guardamos el modelo
save_model(final, 'modelo_ventas_lr')
```

Punto 8: Resultados Obtenidos

Figura 11

Métricas con los 5 Pliegues en Entrenamiento y en el Modelo de Regresión Lineal

Fold	MAE	MSE	RMSE	R2	RMSLE	MAPE
0	611116.7807	666682328198.6301	816506.1716	0.7794	0.8559	1.1547
1	596745.0078	647044607538.0748	804390.8301	-0.0042	1.0820	1.6370
2	1143984.6280	2815434919448.5776	1677925.7789	0.4267	0.8418	1.0896
3	1107713.0249	3383832604026.8101	1839519.6667	0.8565	0.7088	1.0822
4	686971.5411	826885087250.3846	909332.2205	0.7520	0.9228	0.8277
Mean	829306.1965	1667975909292.4956	1209534.9335	0.5621	0.8823	1.1582
Std	244329.0237	1184326788198.1028	452770.5311	0.3190	0.1216	0.2642

	Model	MAE	MSE	RMSE	R2	RMSLE	MAPE
0	Linear Regression	1032523.2031	1762456539712.2070	1327575.4365	0.8272	0.9792	0.8984

En la imagen anterior observamos la tabla de resultados compuesta por las métricas obtenidas en los cinco pliegues de la validación cruzada temporal que se realizó con el 80% de entrenamiento, se observa el resultado promedio y la desviación estándar (variabilidad entre pliegues) de dichas métricas, en la última fila se observa los resultados obtenidos de la prueba del modelo de linear regression.

Observamos que se muestra el rendimiento en promedio de los 5 pliegues observando métricas importantes como:

MAE con un valor promedio de 829306.1965 pesos indicándonos que el modelo se equivoca 829306.1965 pesos al predecir las ventas

MSE que es el promedio de los errores al cuadrado con un valor de 1667975909292.4956 es la base del RMSE al ser una medida tan grande es poco interpretable, es por eso que se usa esencialmente como la base del RMSE.

El RMSE al ser la raíz del error cuadrático medio indica que en promedio el modelo se equivoca en 1209534.9335 de pesos, pero se enfoca en los valores grandes, es decir penaliza más fuerte los errores grandes que los pequeños.

El R2 nos indica que el modelo sigue las variaciones de los gastos respecto al promedio, en un 83% lo cual nos sirve para detectar alzas y bajas en los consumos mensuales por clientes.

Finalmente, con las métricas del R^2 y el RMSE junto al MAE podemos definir que tanto el modelo capta o percibe el comportamiento del gasto mensual por cliente y el error de predicciones de estos gastos. En definitiva, con estas métricas podemos decir que el modelo es adecuado para poder predecir tendencias de compra y dar prioridad a clientes.

Resultado del Conjunto de Prueba

Para poder realizar un análisis más profundo, se observarán los percentiles de los errores del modelo para así tener una visión más general del desempeño del modelo.

Figura 12

Código para Obtener los Percentiles Junto a las Métricas

```
# Predicciones en la prueba (20%)
X_test, y_test = get_config('X_test'), get_config('y_test')
preds = predict_model(final, data=X_test, verbose=False)
col = 'prediction_label' if 'prediction_label' in preds.columns else 'Label'

y_true = np.array(y_test)
y_hat  = preds[col].to_numpy()
abs_err = np.abs(y_true - y_hat)

# Métricas + percentiles
r2   = r2_score(y_true, y_hat)
rmse = mean_squared_error(y_true, y_hat, squared=False)
mae  = mean_absolute_error(y_true, y_hat)
p50, p90, p95 = np.percentile(abs_err, [50, 90, 95])

M = lambda x: f"{x/1e6:.1f} M"

pd.DataFrame({
    "Métrica": ["R²", "RMSE", "MAE", "P50(|error|)", "P90(|error|)", "P95(|error|)"],
    "Valor":   [f"{r2:.3f}", M(rmse), M(mae), M(p50), M(p90), M(p95)]
})
```

Figura 13

Percentiles del Error Absoluto

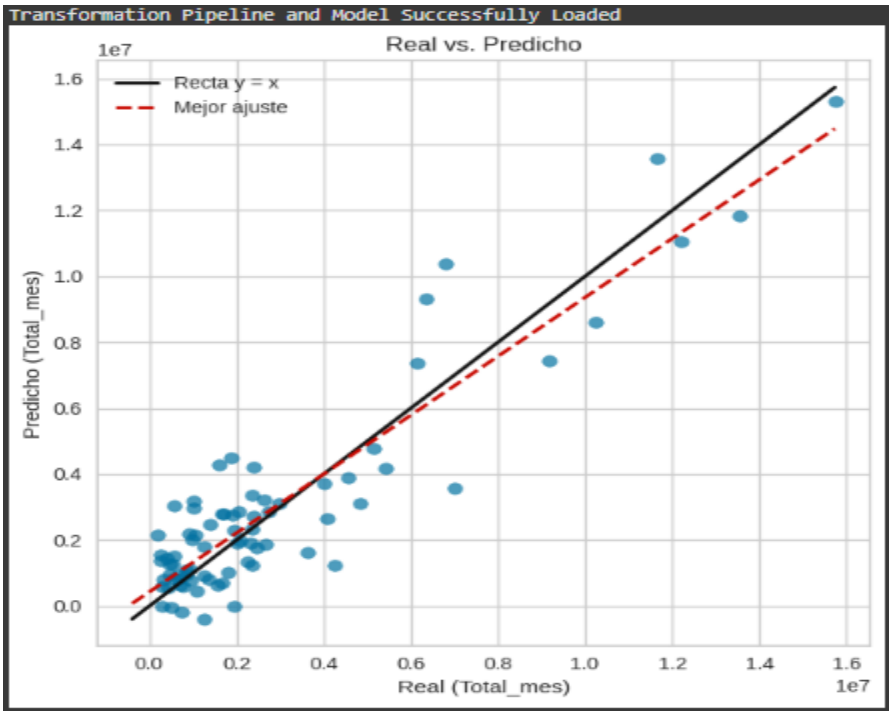
	Métrica	Valor
0	R²	0.831
1	RMSE	1.3 M
2	MAE	1.0 M
3	P50(|error|)	0.9 M
4	P90(|error|)	2.1 M
5	P95(|error|)	2.5 M

Observamos que el modelo nos muestra las métricas de las cuales se habló anteriormente y se dijo a grande rasgos que el R^2 explicaba la variabilidad del modelo en un 83% aproximadamente, que el RMSE penalizaba los errores grandes y se equivocaba un aproximado de 1.3 millones y que finalmente el MAE que penalizaba errores pequeños se equivocaba en 1 millón de pesos aproximadamente, ahora con los percentiles de los errores se puede observar que cuando hablamos de la mediana guiada por el P50 nos indica que el 50% de las predicciones se equivocan en menos de $1.000.000, que el 90% de las predicciones se equivocan por debajo de $2.100.000 y que el 95% de las predicciones se equivocan por debajo de $2.500.000, esto nos confirma que existe una cola con errores grandes pero en muy pocos casos.

Punto 9 Análisis Grafico

Figura 14

Dispersión (Real vs Predicción)



Respecto a la gráfica de dispersión (Real vs predicho) podemos observar lo siguiente:

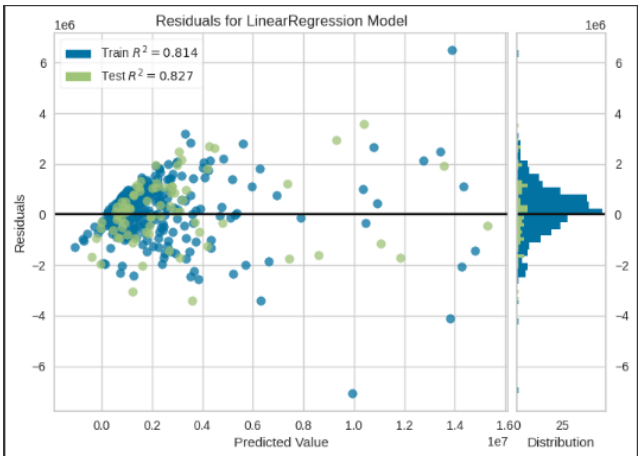
Los ejes están en notación científica y representan unidades de millones, el eje X (real) y Y (predicho).

Existen dos rectas, una de color negra (y = x) y otra roja punteada (indicando la tendencia de los puntos) esta recta roja es la que mejor se ajusta a las predicciones, indicándonos que en un principio en un rango de 0 a 5 millones aproximadamente, esta sobre prediciendo los valores bajos y después de su intersección está subprediciendo los valores altos, además de esto vemos que se aleja muy poco de la recta y=x esto quiere decir que las predicciones del modelo son muy cercanas a las reales.

En el rango bajo (0 a 5 millones aproximadamente) se observa una acumulación de puntos azules (observaciones)esto indica que el modelo recibe mucha información en este rango bajo que, en la parte superior, es decir, se observa que el modelo se entrena con muchos valores bajos y pocos altos por lo tanto con los valores altos el modelo tiende a generar más errores y puede presentarse la heterocedasticidad aumentando la variabilidad de error en los valores más altos.

Figura 15

Residuales vs Predicho



En el gráfico de los residuales podemos observar que se divide en dos partes, uno en lado izquierdo que nos muestra los residuales (eje Y) y los valores predichos (eje X) y otro a la derecha que nos muestra el histograma, en ambos se visualizan los resultados tanto para el entrenamiento como para la prueba, debemos tener en cuenta que el eje X nos muestras los valores predichos y el Y los valores residuales:

En el de la izquierda se observa que los valores van creciendo en forma de abanico a medida que avanzan de izquierda a derecha, estos nos indica que a medida que aumentan los valores la variabilidad del error aumenta a medida que aumenta el valor de las predicciones, esto indica la presencia de heterocedasticidad, dando a entender que el modelo es más preciso con predicciones bajas y menos estable con predicciones bajas.

En la siguiente línea se anexa un valor análisis por medio de la prueba de Breusch-Pagan para la heterocedasticidad.

Figura 16

Prueba Breusch-Pagan

Prueba de heterocedasticidad (holdout)

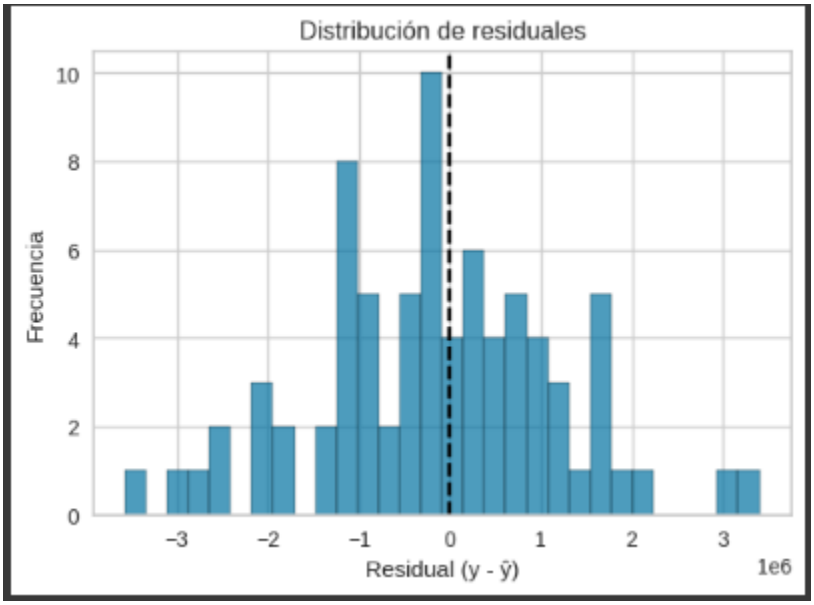
	Prueba	Chi²	p-valor	Conclusión
0	Breusch–Pagan	12.075	0.0005	Heterocedasticidad

Este valor nos indica que existe la variabilidad en los errores residuales, este resultado es coherente con en análisis grafico realizado en los residuales vs predicho, la dispersión crece al aumentar la predicción.

En el lado derecho se observa el histograma, el cual visualizaremos aparte.

Figura 17

Histograma (Residuales)



Aquí observamos que:

El eje X es el residual $(y - \hat{y})$ con esta idea y tomando como eje central el cero, partimos de que lo que está a la izquierda es la sobrepredicción y lo que está a la derecha la es la subpredicción, por otra parte, el eje Y nos muestra la frecuencia, es decir la cantidad de observaciones que cayeron en ese rango.

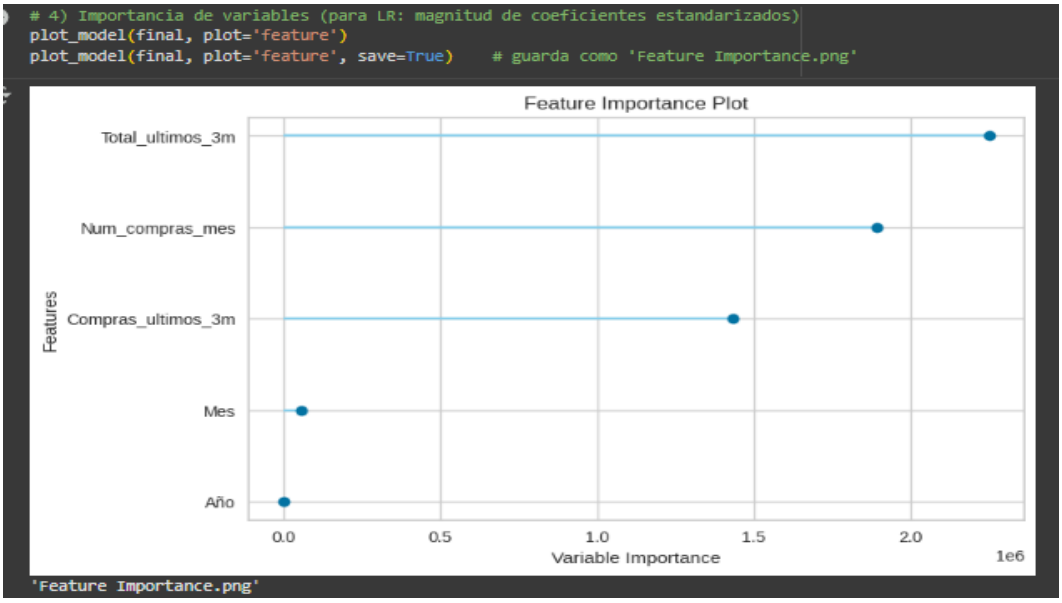
Se observa que la mayoría de los valores residuales están cerca del cero, esto quiere decir que la modelo no varía mucho, presenta poco sesgo por lo tanto el modelo predice mucho mejor con los valores bajos.

Observamos que en las colas algunos valores de los residuales cayeron cercanos a los $\pm$ 3.000.000de pesos.

El grafico no representa una campana simétrica ya que tiene puntos de subidas y bajadas lo cual es normal dado que son datos reales.

Figura 18

Importancia de las Variables Predictoras



El diagrama anterior nos muestra la importancia en forma ordenada de las variables usadas como predictoras, tomando como eje X la importancia de las variables en una escala de 0 a 2 millones (no es el valor en pesos) dándole un nivel de importancia máximo de dos millones debido a la magnitud que observó Pycaret, mientras que el eje Y representa las variables predictoras que se tomaron, en esta grafica se observa lo siguiente:

La variable más fuerte en cuanto al aporte en la predicción es la variable del costo total de las compras en los últimos tres meses, seguidas por variables de importancia como lo son el número de compras mensuales y el número de compras mensuales realizados en los últimos tres meses.

Las que menos aportan a un nivel casi nulo son el mes de la compra y el año de la compra, este último es dado a que todas las compras se realizaron el último año, dándonos a entender que es una variable predictora que no aporta mucho al modelo.

Para dar más peso a este análisis, se presenta a continuación la tabla de coeficientes junto con sus signos la cual nos ayudara a interpretar como cada variable aporta al aprendizaje del modelo.

Figura 19

Tabla de Coeficientes con sus Respectivos Signos

Coeficientes (con signo)

	Feature	Coeficiente	Signo
0	Total_ultimos_3m	2249361.490183	+
1	Num_compras_mes	1890214.664764	+
2	Compras_ultimos_3m	-1434596.960574	–
3	Mes	-57371.820548	–
4	Año	0.000000	+

Intercepto (β0): 2139420.9183

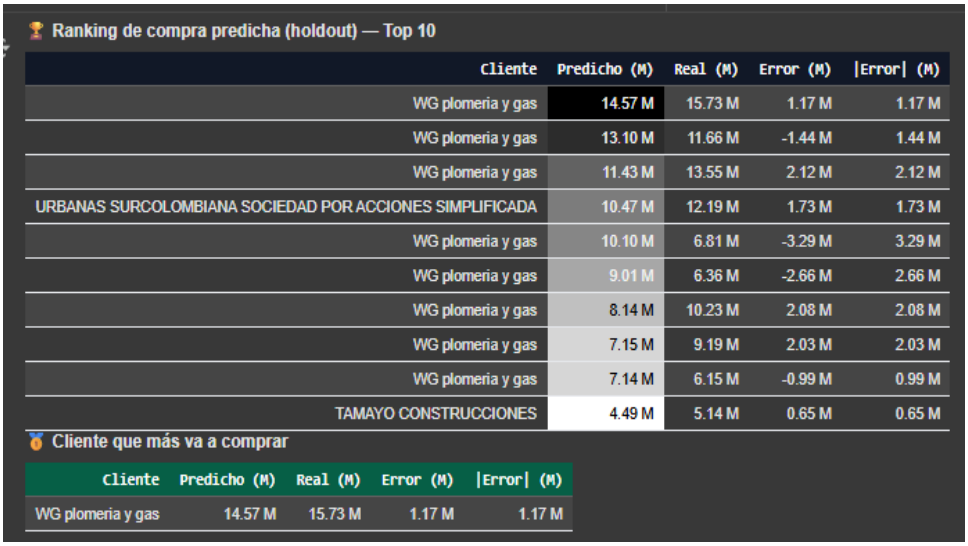
Como se observa en la tabla, los coeficientes que indican el mayor aporte a la predicción de Total_mes, está dado para las variables Total_ultimos_3m (con signo positivo) y Num_compras_mes (con signo positivo), es decir que estas dos las que mayor aporte dieron al aprendizaje del modelo, por otra parte tenemos Compras_ultimos_3m (con signo negativo) nos indica que pueden aumentar el número de compras pero el total a pagar es el mismo, haciendo que la predicción del modelo pueda bajar, por lo tanto se presenta la colinealidad (dos variables aportan la misma información) entre las compras de los últimos tres meses y el total de los últimos tres meses, haciendo difícil la credibilidad sobre si está variable si aporta constructivamente al modelo, algo similar sucede con la variable de mes que presenta signo negativo.

Parte 10 Análisis del Desempeño del Modelo sobre el Conjunto de Prueba

Después de realizar los análisis sobre los gráficos, la prueba Breusch-Pagan y los coeficientes (con sus signos), se procedió a realizar la prueba del modelo en el conjunto de datos (20%) y así reflejar el ranking obtenido, esto es de suma importancia ya que permite comprobar la capacidad predictiva del modelo en los datos que no se usaron para su entrenamiento, así podremos identificar a los clientes con más probabilidades de compra y buscar mejores decisiones estratégicas para las ventas.

Figura 20

Top 10 Clientes que Más Comprarán



Ranking de compra predicha (holdout) — Top 10

Cliente	Predicho (M)	Real (M)	Error (M)	|Error| (M)
WG plomeria y gas	14.57 M	15.73 M	1.17 M	1.17 M
WG plomeria y gas	13.10 M	11.66 M	-1.44 M	1.44 M
WG plomeria y gas	11.43 M	13.55 M	2.12 M	2.12 M
URBANAS SURCOLOMBIANA SOCIEDAD POR ACCIONES SIMPLIFICADA	10.47 M	12.19 M	1.73 M	1.73 M
WG plomeria y gas	10.10 M	6.81 M	-3.29 M	3.29 M
WG plomeria y gas	9.01 M	6.36 M	-2.66 M	2.66 M
WG plomeria y gas	8.14 M	10.23 M	2.08 M	2.08 M
WG plomeria y gas	7.15 M	9.19 M	2.03 M	2.03 M
WG plomeria y gas	7.14 M	6.15 M	-0.99 M	0.99 M
TAMAYO CONSTRUCCIONES	4.49 M	5.14 M	0.65 M	0.65 M

Cliente que más va a comprar

Cliente	Predicho (M)	Real (M)	Error (M)	|Error| (M)
WG plomeria y gas	14.57 M	15.73 M	1.17 M	1.17 M

En la tabla obtenida se puede observar a los 10 clientes con mayor predicción de compra, seguidos por el valor predicho frente al valor real de compra, finalmente se tiene el error que nos permite decir si el modelo sobrepredice o subpredice (Real - Predicho), finalmente se observa el valor absoluto del error el cual nos indica el monto de dinero en el que el modelo falló.

En primer lugar se ubica WG plomería y gas con un valor predicho de compra de $14.570.000 frente a un valor real de compra de $15.730.000, con una subpredicción de

$1.170.000 (predijo por debajo del valor real) este cliente ocupa varias de las posiciones lo que nos indica que se hicieron varias observaciones sobre este cliente en estos meses de prueba, la cuarta posición está dada para URBANAS SURCOLOMBIANA SOCIEDAD POR ACCIONES SIMPLIFICADA, con un valor predicho de compra $10.470.000 frente a valor real de compra de $12.190.000 con una subpredicción de $1.730.000, (predijo un valor inferior al valor real) finalmente se tiene al cliente TAMAYO CONSTRUCCIONES con un valor de compra predicho de $4.490.000 frente a un valor de compra real de $5.140.000 y un valor de subpredicción de $650.000.

Ahora realizaremos las observaciones respecto a los clientes con menos predicción de compra.

Figura 21

Top 10 Clientes que Menos Comprarán

Ranking de compra predicha (holdout) — Top 10 (menor → mayor)

Cliente	Predicho (M)	Real (M)	Error (M)	|Error| (M)
Servigas de la Sabana	0.00 M	1.27 M	1.27 M	1.27 M
TAMAYO CONSTRUCCIONES	0.00 M	0.75 M	0.75 M	0.75 M
TAMAYO CONSTRUCCIONES	0.00 M	0.50 M	0.50 M	0.50 M
SISTEMAS Y SERVICIOS GAS NATURAL SAS(Mos	0.00 M	0.30 M	0.30 M	0.30 M
TAMAYO CONSTRUCCIONES	0.00 M	1.94 M	1.94 M	1.94 M
Servigas de la Sabana	0.46 M	1.08 M	0.62 M	0.62 M
SISTEMAS Y SERVICIOS GAS NATURAL SAS(Mos	0.48 M	0.44 M	-0.04 M	0.04 M
SETIGAS -GASNATURAL	0.50 M	0.78 M	0.28 M	0.28 M
SETIGAS -GASNATURAL	0.52 M	0.30 M	-0.22 M	0.22 M
SISTEMAS Y SERVICIOS GAS NATURAL SAS(Mos	0.55 M	0.71 M	0.16 M	0.16 M

Cliente con menor compra predicha

Cliente	Predicho (M)	Real (M)	Error (M)	|Error| (M)
Servigas de la Sabana	0.00 M	1.27 M	1.27 M	1.27 M

En la tabla del top 10 de menor a mayor observamos las primeras cinco posiciones con un valor predicho de $0.0 pesos frente a sus valores reales, esto nos indica que de estos cinco primeros clientes (Servigas de la Sabana, TAMAYO CONSTRUCCIONES y SISTEMAS Y SERVICIOS DE GAS NATURAL S.A.S) no se espera la realización de la compra, sin embargo se observa que en la sexta posición y séptima posición de la tabla nos refleja que estos clientes realizaran compras, esto indica que el modelo subpredijo (predijo cero cuando realmente si hubo compras) ya que el modelo al tener valores bajos su predicción tiende a valores negativos o cero. Para este tipo de clientes se sugiere un seguimiento adecuado dado que, aunque sus compras son mínimas si realizan aportes económicos en el comercio de este sector.

Parte 11 Limitaciones del Modelo

Ya se ha hablado de la construcción del modelo y de su funcionamiento además de su utilidad principal la cual es predecir el monto de compra mensual por cliente y generar un ranking de clientes con estimación de compra mayor y estimación de compra menor, lo cual permite la planeación y ejecución de acciones comerciales ya que es posible el anticipar el comportamiento de ventas y apoyar la planificación comercial, sin embrago el modelo explica cerca del 83% (R2) del comportamiento de los datos, es decir, presenta algunas limitaciones que se describirán a continuación.

El modelo presenta mayor dispersión del error en los montos más grandes (heterocedasticidad) por lo tanto el modelo no explica de manera adecuada el comportamiento de los clientes con compras grandes, es decir la precisión del modelo disminuye con valores grandes

En cuanto a su comportamiento con valores pequeños, se observa la limitación de una tendencia a valores negativos o nulos ya que las compras han sido bajas el modelo no tiene restricciones y por ende toma las compras como nulas.

Las métricas del promedio (MAE y RMSE) muestran superficialmente los errores del modelo, es decir que tanto en promedio se equivoca el modelo, pero al ser promedios no reflejan los errores más grandes en algunas predicciones.

El modelo presenta una multicolinealidad ya que tenemos algunas variables (Total_ultimos_3m y Compras_ultimos_3m) que están correlacionadas, aportando la misma información al modelo, haciendo que los coeficientes sean inestables lo cual hace que sea difícil explicar el aporte individual de las variables al modelo.

Existen clientes que al no comprar con frecuencia (meses con compras y meses sin compras) no brindan suficiente información para que el modelo prediga su comportamiento, además de eso existen meses con valores de compra muy altos o muy bajos (outliers) respecto a las compras habituales del cliente, lo que hace que el error de ese mes aumente en gran medida y el RMSE sea mayor.

Finalmente podemos hablar del alcance del modelo (lo que no puede predecir), dado que, aunque el modelo estima el total de compra por mes de cada cliente, no nos puede decir cuál es la probabilidad de compra ni cual es el producto que probablemente compre, ni en qué fecha aproximadamente realice la compra, ya que su uso es indicar clientes por monto esperado mas no describir los detalles de la compra que finalmente pueden ayudar al crecimiento del negocio.

Parte 12 Recomendaciones

Las recomendaciones están sujetas a las limitaciones encontradas.

Dado que existen clientes con compras grandes y otros clientes con compras pequeñas, se da la heterocedasticidad, es decir, el modelo se puede equivocar en compras grandes, por lo tanto, se recomienda incluir los percentiles P50, P90 y P95 para tener mayor claridad en cuanto se equivoca el modelo en la mitad de los casos (P_{50}), y hasta donde pueden llegar los posibles

errores (P_{90} y P_{95}), esto servirá para observar riesgos y tener más clara la toma de decisiones acertadas.

Dado que el modelo arrojó predicciones de compras negativas o en cero en los clientes con muy poca frecuencia de compra, se tomó la decisión de truncar el modelo para evitar predicciones negativas, no obstante, se recomienda realizar un seguimiento del ranking de clientes de menor a mayor pronóstico de compra, teniendo claro que los clientes cuya predicción sea igual a cero (o nula), se tomen como clientes de los que no se espera compra por falta de información.

Las métricas del MAE y el RMSE muestran los errores promedios, pero es pro esto que no permiten ver los errores realmente altos, llevando al uso de los percentiles P50, P90 y P95, es por eso que se recomienda toma el P50 como el error típico que se puede esperar, es decir el desempeño en general, el P90 se puede tomar como un margen adecuado para tomar decisiones con predicciones grandes y el P95 se puede tomar como como casos inusualmente grandes, a los que se les debe hacer un seguimiento.

Existen variables que aportan información similar del cliente al modelo (multicolinealidad), esto lo observamos al analizar los coeficientes y sus signos, es por eso que se recomienda, seleccionar las variables más adecuadas, teniendo como criterio cuál de estas aporta más a la predicción, o combinarlas de ser posible, además de esto no se debe confiar en la exactitud del coeficiente (valor numérico y signo), ya que son inestables.

Conclusiones

El proyecto demostró que el verdadero punto de partida para cualquier análisis de negocios es la organización exhaustiva de los datos, ya que al obtener la información dispersa debemos recolectarla para poder analizarla a través de herramientas como OpenRefine, Python y SQLite, de esta forma logramos construir una base de datos sólida lo que nos permitió comprender de una manera más clara el comportamiento de los datos.

El análisis exploratorio permitió identificar los patrones más importantes de comportamiento en los clientes, como la recurrencia de compras y la frecuencia de las ventas, estos hallazgos fueron esenciales para comprender el comportamiento del negocio y definir las variables más relevantes para el modelado.

La implementación de la librería PyCaret permitió comparar de manera eficiente distintos modelos de regresión, seleccionando la linear regression como el modelo más adecuado, alcanzando un R^2 cercano al 83%, lo que nos confirma que el modelo tiene una capacidad predictiva suficiente para anticipar las ventas y proporcionar a la empresa una base sólida para la planificación estratégica.

Al utilizar técnicas de ventana corrediza para mejorar los datos, el modelo funcionó mejor, esto demuestra que el análisis de datos no solo hace que las predicciones sean más exactas, sino que también da a las empresas una ventaja competitiva al ayudarles a tomar mejores decisiones.

Aunque el modelo presenta una capacidad de predicción favorable, presento heterocedasticidad, valores negativos en la predicción de compras pequeñas o poco frecuentes, y multicolinealidad debido al uso de la técnica de la ventana corrediza, esto nos lleva a realizar ajustes y recomendaciones lo cual en nuestra labor como analista de datos.

Recomendaciones

La empresa debe establecer políticas continuas de recolección y almacenamiento de información en formatos estructurados, evitando así la dispersión en archivos, lo que asegurará la actualización permanente de la base de datos para futuros análisis.

Una vez obtenido la validación del modelo, la empresa deberá considerar la implementación de este enfoque predictivo en otras líneas de negocio (segmentar mejor a los clientes según su historial de compras, usar el modelo en nuevas ciudades o regiones para anticipar la demanda antes de invertir en abrir un punto de venta o identificar diferencias de consumo entre localidades, Bogotá comparado con Chía y también en comparación con Cajicá), aprovechando el aprendizaje obtenido para mejorar su competitividad en el mercado.

Es fundamental establecer un sistema periódico de revisión y actualización de la base de datos, así garantizando que la información corresponda a los cambios reales de la empresa, esto permite asegurar que los análisis y proyecciones realizadas siempre se basen en datos vigentes y confiables.

De igual forma también se recomienda a la empresa que fortalezca los procesos de formación y sensibilización en el uso de los datos para la toma de decisiones, ya que esto permitirá que diferentes áreas (comercial, ventas, logística, contabilidad y servicio al cliente) comprendan la importancia de mantener la información precisa y actualizada, fortaleciendo una cultura dentro de la empresa basada en el buen uso de los datos para tomar decisiones.

Referencias Bibliográficas

Alves, G. O., Fonsêca, J. C. B., & Maciel, A. M. A. (2021). Evaluation of machine learning models for estimating sales in physical retail. En Anais do IX Symposium on Knowledge Discovery, Mining and Learning (KDMiLe 2021). Sociedade Brasileira de Computação. https://sol.sbc.org.br/index.php/kdmile/article/view/17459

Caro, M., Guardiola, J., & Ortiz, M. (2018). Classification trees as a tool to predict financial difficulties in Latin American companies through their accounting ratios. Contaduría y Administración, 63(1), 25–26. https://www.cya.unam.mx/index.php/cya/article/view/1148/1221

Caro, N. P., Guardiola, M., & Ortiz, P. (2018). Árboles de clasificación como herramienta para predecir dificultades financieras en empresas latinoamericanas a través de sus razones contables. Contaduría y Administración, 63(1), 1–22. https://doi.org/10.22201/fca.24488410e.2018.1148

Charilaou, P., Andreou, A., & Constantinou, A. (2022). Machine learning models and overfitting considerations: An empirical study. Journal of Biomedical Informatics, 127, 104017. https://doi.org/10.1016/j.jbi.2022.104017

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. En Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 785–794). ACM. https://doi.org/10.1145/2939672.2939785

Ensafi, Y., Khalilzadeh, O., & Maleki, M. (2022). Time-series forecasting of seasonal items sales using machine learning: A comparative analysis. Journal of King Saud University - Computer and Information Sciences, 34(9), 6649–6658. https://www.sciencedirect.com/science/article/pii/S2667096822000027

Fontecha Ballestas, J. A. (2021). Sistematización del diseño del área de proyectos para la empresa CI Tecnología Alimentaria S.A.S [Tesis de especialización, Corporación Universitaria Minuto de Dios – UNIMINUTO]. Repositorio Institucional UNIMINUTO. https://repository.uniminuto.edu/bitstream/10656/12822/1/T.EGP_FontechaJesus_2021.pdf

Ghanad, A. (2023). What is quantitative research? An overview and guidelines. SAGE Journals. https://journals.sagepub.com/doi/full/10.1177/14413582241264622

Gupta, A., Singh, R., & Bansal, S. (2023). Normal workflow and key strategies for data cleaning: Improving data quality for analytics. International Journal of Medical Research, 12(1), e44310. https://www.i-jmr.org/2023/1/e44310

Ensafi, Y., Khalilzadeh, O., & Maleki, M. (2022). Time-series forecasting of seasonal items sales using machine learning: A comparative analysis. *Journal of King Saud University - Computer and* Li, Z. (2022). Short-term demand forecast of e-commerce platform. Computational Intelligence and Neuroscience, 2022, Artículo ID 5227829, 9 páginas. https://onlinelibrary.wiley.com/doi/10.1155/2022/5227829

Marketing Team. (2024). Título del recurso del Marketing Team (indicar nombre del sitio si aplica). Recuperado de https://www.spanishdict.com/translate/%28si%20aplica%29

Morales Castro, A., Ramírez Reyes, E., & Rodríguez Albor, G. (2019). Pronóstico de ventas de las empresas del sector alimentos: una aplicación de redes neuronales. Semestre Económico, 22(52), 161–177. https://doi.org/10.22395/seec.v22n52a7

Murillo, C., & Restrepo, M. (2019). Caracterización de las pymes colombianas y de sus fundadores: un análisis desde dos regiones del país. Estudios Gerenciales, 35(150), 81–91. https://doi.org/10.18046/j.estger.2019.150.2968

Naureen, A. (2023). Exploratory data analysis (EDA) for Amazon Alexa product data. SSRN Electronic Journal. https://doi.org/10.2139/ssrn.4742251

Piatetsky-Shapiro, G. (Ed.). (1991). Knowledge discovery in databases. AAAI/MIT Press. https://doi.org/10.5555/1972514

Quispe, J. O. Q., Quispe, A. C. F., Calvo, N. C. L., & Toledo, O. C. (2024). Analysis and selection of multiple machine learning methodologies in PyCaret for monthly electricity consumption demand forecasting. Materials Proceedings, 18(1), 5. https://doi.org/10.3390/materproc2024018005

Rahm, E., & Do, H. H. (2000). Data cleaning: Problems and current approaches. IEEE Data Engineering Bulletin, 23(4), 3–13. http://dbs.uni-leipzig.de/file/rahm2000data.pdf

Ruiz Silva, C. E., & Gómez Salazar, E. E. (2020). Estrategia para la implementación de la inteligencia de negocios en empresas pequeñas [Tesis de especialización, Universidad Distrital “Francisco José de Caldas”]. Repositorio Institucional. https://repository.udistrital.edu.co/bitstream/handle/11349/27976/GomezSalazarElioEdwin2020.pdf

Santos, J. (2023). Título del recurso de Santos (indicar nombre del sitio si aplica). Recuperado de https://www.spanishdict.com/translate/%28si%20aplica%29

Teodorescu, V., & Obreja Brașoveanu, L. (2025). Assessing the validity of k-fold cross-validation for model selection. Computation, 13(5), 127. https://doi.org/10.3390/computation13050127

Tukey, J. W. (1977). Exploratory data analysis. Addison-Wesley. ISBN 978-0201076165

Wedel, M., & Kannan, P. K. (2016). Marketing analytics for data-rich environments. Journal of Marketing, 80(6), 97–121. https://doi.org/10.1509/jm.15.0413

Zerpa, H., García, R., & Izquierdo, H. (2020). Datamart basado en el modelo estrella para la implementación de indicadores clave de desempeño como salida del Big Data. Universidad Ciencia y Tecnología, 24(102), 47–54. https://doi.org/10.47460/uct.v24i102.342