PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

FACULTAD DE HÁBITAT, INFRAESTRUCTURA Y CREATIVIDAD

TEMA:

PREDICCIÓN DE LA DEMANDA EN GRUPO GLORIA ECUADOR BASADO EN ALGORITMOS DE ENSAMBLE UN ENFOQUE A MACHINE LEARNING PARA OPTIMIZAR LA GESTIÓN DE INVENTARIO Y MEJORAR LA PRECISIÓN PREDICTIVA DE VENTA.

AUTOR:

PAÚL QUEISSON BENALCÁZAR CISNEROS

DIRECTOR:

MGTR. EDUARDO JOSÉ MONTERO BERMUDEZ

TRABAJO PREVIO A LA OBTENCIÓN DEL TÍTULO DE MAGISTER EN SISTEMAS DE INFORMACIÓN MENCIÓN EN DATA SCIENCE

Quito

Índice

Índice de Figuras

Índice de Tablas

CAPÍTULO I: INTRODUCCIÓN

1.1. Generalidades

Grupo Gloria Ecuador es una empresa que se dedica a la industria de **consumo masivo** principalmente al lácteo y sus derivados. Forma parte del Grupo Gloria una empresa multinacional originaria de Perú y consolidada en países de Latinoamérica como Colombia, Uruguay, Chile, Puerto Rico y Ecuador. Grupo Gloria cumple un objetivo común para todas sus empresas es de satisfacer las necesidades de los consumidores en el sector de alimentos y bebidas.

La era digital que actualmente se vive a nivel mundial ha transformado el ambiente empresarial global, donde la industria láctea tampoco está exenta. El crecimiento exponencial en la recolección de los datos y el avance tecnológico ha causado que las empresas adopten nuevas soluciones innovadoras para optimizar las operaciones. En este marco la predicción de la demanda es una herramienta ideal para la eficaz gestión de la cadena de suministro y llegar a tomar decisiones **estratégicas** que ayuden a la empresa.

Actualmente Grupo Gloria Ecuador realiza **análisis descriptivo** de sus ventas, pero no aplica las nuevas tecnologías como aprendizaje automático. Estas herramientas permiten predecir la demanda futura ya que es crucial y garantiza la disponibilidad de los productos en el mercado con el fin de minimizar costo de inventario y maximizar la complacencia del cliente. La importancia de anticipar a la volatibilidad de la demanda permite a la empresa ajustar los niveles de producción y optimizar su distribución de producto terminado.

Los algoritmos de *Machine Learning* (ML) son una herramienta útil para las empresas permitiendo conseguir conocimiento de los datos ayudando a mejorar la predicción de sus ventas (Kelleher & Mac Namee, 2014). Cada uno de los algoritmos tiene la característica fundamental de manejar grandes volúmenes de datos permitiendo descubrir patrones muy complejos que para la estadística tradicional quizás sean imperceptibles (Athanasopoulos & Hyndman , 2021) .

La integración de *Machine Learning* enfocados en Algoritmos Ensamble a Grupo Gloria Ecuador no solo permite mejorar la demanda futura, sino también brinda a la empresa a ser flexibles a los cambios que el mercado presenta, esto es primordial en un entorno competitivo y dinámico, donde la característica principal es la precisión para la ayuda a la toma de decisiones. Dichos algoritmos permiten combinar múltiples modelos individuales para optimizar el desempeño de la predicción de los diferentes modelos reduciendo el error al utilizar modelos individuales (Cortez, 2022).

Un sector altamente competitivo como es el consumo masivo permite la integración de nuevas herramientas como *Machine Learning* el cual fortalecerá la capacidad de respuesta para Grupo Gloria y ayudara a mantener el liderazgo en el mercado ofreciendo productos en el instante adecuado y las cantidades precisas.

1.2. Planeamiento del problema

Grupo Gloria Ecuador es una empresa dedicada a la elaboración de productos lácteos como derivado de la leche, yogurts, bebidas lácteas, leche condensada y envasado de leche en sus diferentes presentaciones entera, semidescremada y descremada, enfrenta desafíos significativos para el pronóstico de productos afectado la cadena de suministro.

La **falta de precisión** en la predicción de la demanda provoca una serie de problemas operativos y financieros. Grupo Gloria Ecuador enfrenta el desafío constante de optimizar sus operaciones, la problemática que se ha identificado es la incapacidad para anticipar adecuadamente la demanda y el resultando ha sido visible, por otro lado, la sobreproducción puede llevar a pérdidas por obsolescencia o deterioro, mientras que la subproducción puede resultar en la pérdida de oportunidades de venta. En este contexto, la predicción **precisa** de la **demanda** se vuelve un factor crítico.

El **objetivo** de este proyecto es desarrollar un **modelo** de **predicción** de la **demanda** para Grupo Gloria, Ecuador, se propone una solución de *Machine Learning* basado en algoritmos de **ensamble** para abordar este problema. Los algoritmos de ensamble combinan múltiples modelos de *Machine Learning* para obtener mejores resultados que cualquier modelo individual.

Grupo Gloria Ecuador con el desarrollo de un modelo de predicción de la demanda a través de algoritmos ensamble le permitirá reducir el error entre la demanda real y la demanda pronosticada por otro lado ayudará a la toma decisiones más informadas sobre los niveles de inventario, reduciendo costos y evitando roturas de stock como también mejorar la planificación de la producción y distribución.

1.3. Justificación

La capacidad predictiva de la demanda es clave para la efectividad de las empresas dedicadas al consumo masivo en especial al sector lácteo. En el mercado actual existe alta variabilidad, estos **cambios** se presentan ya sea por el cliente o a su vez las variaciones de la estacionalidad en el consumo de productos lácteos, esto hace que la planificación y gestión de inventario de producto terminado sean tareas complejas. Una demanda inexacta deriva en consecuencias como costos adicionales, sean estos por sobreproducción, desperdicios de productos o por poco stock y pérdidas de ventas.

En el escenario actual, Grupo Gloria Ecuador está innovando al permitirse implementar un modelo de basado en algoritmos de ensamble en *Machine Learning* el cual le permitirá tener una solución efectiva. Dichos modelos al combinarse con múltiples predictores ayudan a minimizar errores mejorando la capacidad de generalización de las estimaciones.

Desde diferentes áreas de la empresa como el Financiero permitirá tener una minimización del error en la predicción de la demanda que conlleva al principal beneficio como empresa es la

disminución de los costos operativos ayudando a tener mejor rentabilidad de los productos. Otra área como la Logística tendrá la capacidad de prevenir y no tener perdidas por producto caducado y a su vez evitar quiebres de stock, todos esos beneficios serán reflejados a tener un servicio eficiente y mayor satisfacción del consumidor final.

1.4. Objetivos

1.4.1. Objetivo general

- Desarrollar e implementar un modelo de predicción de demanda basado en algoritmos de Ensamble para optimizar la gestión de inventario en Grupo Gloria Ecuador, mejorando la precisión predictiva de las ventas y contribuyendo a una toma de decisiones más eficiente en la cadena de suministro.

1.4.2. Objetivos específicos

- Identificar y recopilar los datos históricos relevantes sobre ventas, producción, eventos estacionales y otros factores que puedan influir en la demanda de los productos de Grupo Gloria.
- Realizar un análisis exploratorio de las ventas históricas para identificar patrones, tendencias, posibles fuentes de ruidos y determinar relaciones entre las variables.
- Diseñar y entrenar modelos de *machine learning* utilizando algoritmos de ensamblaje, como *Random Forests, Gradient Boosting Machine* y *XGBoost*, para predecir la demanda de productos en Grupo Gloria Ecuador.
- Evaluar el impacto del modelo en la gestión de inventario, midiendo indicadores como el nivel de servicio y la eficiencia de la cadena de suministro.

CAPÍTULO II: MARCO TEÓRICO

2.1 Predicción de la demanda

Para comenzar a entender el proceso de la predicción de la demanda es necesario comprender por separado cada una de las definiciones como: que es predicción y que es demanda para luego conjuntamente llegar a una sola definición sobre la predicción de la demanda.

La demanda se define como cuántos bienes y servicios desean comprar las personas a los precios actuales del mercado en un periodo determinado (Peiro Ucha, 2024). La demanda tiende a ser volátil y ser afectada por una diversidad características como:

a) **Precio:** es una de las principales características ya que la cantidad demanda comienza a disminuir cuando el precio aumenta y viceversa.
b) **Ingresos de las Personas:** el incremento de los ingresos de las personas tiene aumentar estacionalmente donde la demanda de los productos aumenta de igual manera muchas veces interpretada con una relación uno a uno.
c) **Gustos:** esta característica es primordial ya que depende de los cambios preferenciales y nuevas modas presentadas por las empresas esto tienen a influir significativamente en la demanda.
d) **Demografías:** crecimiento y variaciones en la población influye a los cambios de la demanda
e) **Medidas Gubernamentales:** por ejemplo, cambios en los Impuestos o subsidios que los gobiernos presenten influyen tanto en el precio y la demanda de los productos.

La predicción se define como una representación matemática o estadística que utiliza uno o varios conjuntos de datos históricos con el fin de identificar patrones y poder observar resultados futuros (Mera, 2022). Aplicación de las predicciones en diferentes áreas son:

a) **Finanzas:** permite anticiparte a las fluctuaciones del mercado con el fin de evaluar la inversión.
b) **Salud Publica:** se emplean modelos para realizar el pronóstico de enfermedades y evaluar la efectividad de los tratamientos.
c) **Marketing:** es una herramienta útil para poder identificar la tendencia de consumo con el fin de personalizar ofertas.
d) **Deportes:** dentro de este ámbito permite evaluar el rendimiento de los jugadores con el fin de poder pronosticar resultados en las competencias deportivas.

Dentro de la investigación actual se puede definir como predicción de la demanda se define como la estimación de la **cantidad de productos o servicios** que los clientes compraran en un determinado periodo de tiempo (Diaz Madero, 2024). Por lo tanto, la clave de realizar las predicciones de la demanda es para asegurar la cantidad suficiente del producto disponible para los consumidores.

Una de las características fundamental de realizar la predicción de la demanda es que el área correspondiente de la empresa puede ajustar las operaciones para la producción de productos, también otra característica es que el mantener un inventario optimo y a su vez evitar cambios bruscos de las ventas que no tengan afectación dentro de las operaciones (Diaz Madero, 2024). Un pronóstico sin errores permitirá a la empresa a optimizar sus recursos tanto en el área de operaciones, logísticas y ventas el cual permitirá visualizar información precisa, identificar tendencias del mercado, todas estas ventajas llevan a un objetivo común el cual es tomar decisiones estratégicas que beneficiaran a la empresa.

2.2 Big Data y su rol con la industria del consumo masivo

Cuando se habla del término de Big Data se hace referencia a la generación diaria de un gran volumen de datos, tanto estructurados como no estructurados que inundan los negocios cada día (Cataldo, 2025). En la actualidad el uso estratégico del Big Data en el sector del consumo masivo con el pasar de los años ha crecido exponencialmente con un fin común del cual se puede enriquecer la experiencia del cliente y a su vez potenciar las ventas de los productos.

En el sector del consumo masivo o también conocido como *retail*, cada vez que un cliente realiza la compra de un producto, está dejando un vestigio el cual brinda información sobre que compro, cuando, donde y como (Cataldo, 2025) . Toda esta información es analizada por las empresas llegando a tener conclusiones significativas como frecuencias de compra, tipos de pagos, segmentación de clientes entre otras, todo esto ayuda a mejorar la toma de decisiones.

Big Data es una nueva tecnología que ha revolucionado las empresas lo que ha permitido optimizar las operaciones, predecir y personalizar experiencias de usuario, algunos ejemplos donde está inmerso la terminología del Big Data.

a) **Inventarios y cadena de suministro:** Las empresas han logrado tener un stock adecuado, evitando tener quiebre de stock y una sobreproducción.
b) **Predicción de producto terminado:** permite facilitar las tendencias que se generar con las compras de los productos facilitando la realización de productos con ofertas
c) **Comportamiento del cliente:** grandes ventajas ya que realiza un análisis ya sea para las ventas en tiendas físicas como en línea el cual permite optimizar los productos en las tiendas físicas y mejorando la experiencia del usuario en las tiendas online (Cataldo, 2025).

El acceso que las empresas tienen a gran cantidad de datos actualizados está permitiendo a las empresas a identificar tendencias y problemas de inventario. La inmersión del Big Data está facilitando realizar ajustes vitales minimizando riesgos y aumentando la capacidad productiva. En el área Ventas (Comercial) de cada empresa está consiguiendo el objetivo de incrementar las ventas. En base a esta información, las empresas no solo modifican la estrategia de ventas, siendo está cada vez más personalizada a las preferencias de los clientes, sino que también realizan campañas de marketing más beneficiosas llegando al segmento más adecuado y potencial de compra (Muñoz, 2021).

2.3 *Machine Learning* en la predicción de la demanda.

El *Machine Learning* una rama de la Inteligencia Artificial que permite el desarrollo de algoritmos y utilización de modelos estadísticos, se enfoca en el procesamiento de grandes volúmenes de datos y entrenamiento de los algoritmos con la finalidad de encontrar patrones y correlaciones de grandes Data Set (Sap, 2020).

La presencia de los algoritmos de *Machine Learning* no son nuevos, estos han estado durante décadas permitiendo la evolución constante. Sin embargo, la aplicación era limitada por la falta de datos y/o a la falta de capacitad computacional. El avance tecnológico ha logrado aprovechar la gran cantidad de información que existe actualmente permitiendo la automatización de tareas complejas que llevan mucho tiempo para el ser humano (Småros & Kaleva, 2023).

La aplicación de *Machine Learning* en el pronóstico de la demanda de las empresas dedicadas al consumo masivo permite integrar una alta gama de factores y relaciones que influyen diariamente en la demanda. El uso de los datos con cada una de las características, algunas con un impacto potencial ayudan a la creación de modelos y generación de información valiosa como tendencias y patrones, esta información es consumida desde una o varias fuentes de información.

Descripción de patrones recurrente que ayudan al pronóstico de la demanda (Figura 1)

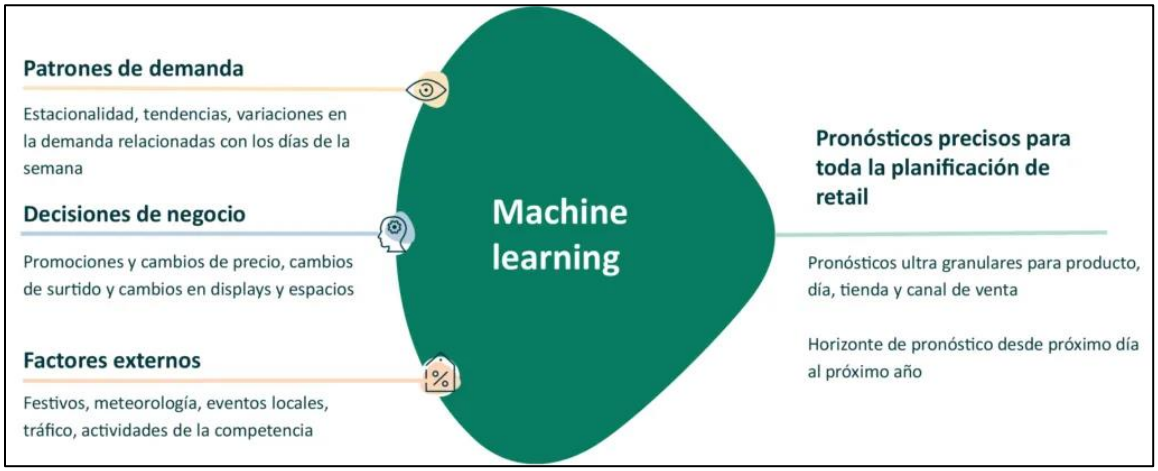

Figura 1: *Comprensión el impacto de los patrones recurrentes, sus estrategias comerciales internas y factores externo*

Finalmente, la implementación de algoritmos de *Machine Learning* cuando se utiliza de manera adecuada y enfocada en los objetivos de la empresa, puede ser una herramienta muy valiosa. Permite optimizar y acelerar los procesos internos. Las compañías que adopten e integren estas nuevas tecnologías en sus operaciones podrán obtener beneficios significativos y, como resultado, incrementar sus ganancias (Gonzales, 2024).

2.4 Modelos Ensamble

El nuevo concepto y utilización de aprendizaje automático cada día crece de manera exponencial, el objetivo primordial para la utilización de esta nueva herramienta es generar resultados predictivos de alta precisión. Un modelo muchas veces no suele ser robusto en relación con el descubrimiento de conocimiento de los datos lo que conlleva a tener un rendimiento bajo. Para optimizar el rendimiento en tareas de modelado predictivo, se han ideado estrategias denominadas *Modelos Ensamble*, estas técnicas consisten en la integración de múltiples modelos individuales, conocidos como modelos base, con el propósito de lograr una mejora sustancial en la precisión de las predicciones, así como en la estabilidad y la capacidad de generalización del modelo resultante (Corso, Maldonado, & Luque, 2018).

Los modelos Ensamble se pueden aplicar en diferentes áreas y problemáticas dentro de una empresa como, predecir valores financieros, detección de fraudes, diagnósticos de enfermedades antes que aparezca. Como en toda la rama tecnología la mayoría siempre viene de la mano de una arquitectura, los modelos Ensamble tienen dos modelos de conjunto: la homogénea, que emplea un único algoritmo, y la híbrida, que combina múltiples algoritmos, como *Random Forests, Gradient Boosting Machines y XGBoost* (Corso, Maldonado, & Luque, 2018).

Los algoritmos Ensamble tienen la caracteriza de reducir errores individuales que pueden originarse por distintos motivos como por ejemplo la presencia de ruido en los datos o la falta de flexibilidad en un modelo específico. Hay tres razones fundamentales que justifican la eficiencia de los modelos Ensamble:

a) Reducción del Sesgo: Los modelos individuales, como los árboles de decisión simples, a menudo carecen de la capacidad para capturar patrones complejos en los datos, lo que resulta en un alto sesgo. Sin embargo, las técnicas de *Ensamble*, pueden mitigar este problema al combinar múltiples modelos y ajustar iterativamente los errores de los modelos anteriores

b) **Reducción de la Varianza:** La alta varianza puede ser un problema para modelos como los árboles de decisión, ya que son muy sensibles a ligeras alteraciones en los datos de entrenamiento. Los métodos de ensamble solucionan esto al promediar las predicciones de varios modelos entrenados en diferentes subconjuntos de datos.

c) Mejora de la Generalización: Mediante la unión de múltiples modelos, se consigue un balance entre sesgo y varianza, mejorando así la capacidad del modelo final para generalizar a datos no observados

2.5 *Overfitting* y *Underfitting* en Modelos Ensamble

El desempeño de los modelos predictivos en el aprendizaje automático depende en gran medida de evitar tanto el sobreajuste como el subajuste, lo que representa un desafío significativo.

a) ***Overfitting*** **(Sobreajuste):** Se produce cuando un modelo es demasiado complejo y se ajusta excesivamente a los datos de entrenamiento, capturando incluso el ruido. Como resultado, el modelo tiene un rendimiento excelente en los datos de entrenamiento, pero su capacidad de generalización es deficiente, lo que significa que su rendimiento disminuye considerablemente en nuevos datos no observados (Rosati G. , 2020). Los Modelos Ensamble pueden reducir el *overfitting* mediante varias técnicas, que promedia múltiples modelos para reducir la sensibilidad a datos específicos, y la regularización utilizada en métodos como *Gradient Boosting.*
b) ***Underfitting*** **(Subajuste):** Ocurre cuando un modelo es demasiado simple para captar las relaciones en los datos, lo que conduce a un bajo rendimiento tanto en el entrenamiento como en los datos de prueba. Este problema suele presentarse cuando el modelo tiene un alto sesgo y no logra aprender adecuadamente de los datos (Vera, 2020). Los Modelos *Ensamble* ayudan a mitigar el *underfitting* al permitir la combinación de modelos más diversos y flexibles, aumentando la capacidad del sistema para detectar patrones más complejos en los datos.

En conclusión, los Modelos Ensamble representan una estrategia clave para mejorar la precisión de las predicciones y minimizar los errores asociados con el *overfitting* y el *underfitting*, proporcionando modelos más robustos y generalizables para diversas aplicaciones del aprendizaje automático.

CAPÍTULO III: APLICACIÓN METODOLÓGICA

3.1 Recolección de Datos

Los datos seleccionados para el estudio provienen del ERP que maneja Gloria Ecuador el cual es SAP R3 en la versión más reciente. Gloria Ecuador realiza la recopilación de los datos de sus ventas a través de realizar un proceso de *ETL*, consiste en realizar proceso de extracción, transformación y carga de datos en diferentes fuentes de información destinado a un repositorio *on premise*.

Dentro de Gloria Ecuador existe procesos semiautomáticos que se encargan cargar cada uno de los datos desde su origen a su destino. La recopilación de los datos se realiza a través de la descarga manual de información a través de transacciones habilitadas tanto en el módulo de ventas como el módulo de contabilidad, toda esta información se recopila con una forma diaria brindando una visión completa de las ventas que se realiza en la empresa de todos sus clientes.

Para el presente estudio se utilizará una base de datos de ventas realizadas por la empresa el cual es extraída de diferentes transacciones del ERP empresarial que maneja la empresa que a través de *SQL Server Integration Services* el cual permite la creación de soluciones e integración de diferentes fuentes de datos permitiendo copiar, descargar, limpiar y realizar minería de las diferentes fuentes de datos. Ver figura 2

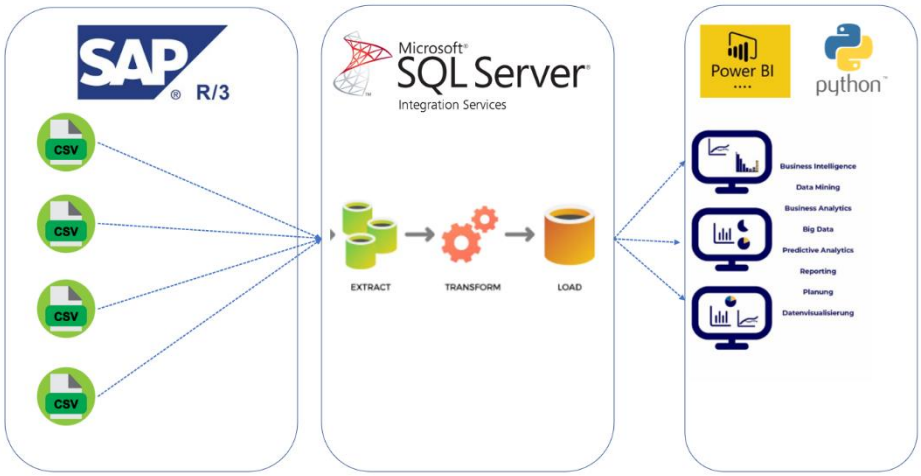

***Figura 2**: Recolección de Datos Gloria Ecuador*

Los datos de las ventas que se obtienen son de las ventas que realiza la empresa de manera diaria de todos sus puntos de emisión donde se facturas los productos Gloria. Sin embargo, la información descargada de las ventas mensuales tendrá las siguientes características como Fecha, Producto, Cliente, Cantidad, Total.

Los datos proporcionados comprenden de una recopilación desde el año 2022 hasta diciembre de 2024 de las ventas realizas de los productos fabricados solo son de la venta realizas de las marcas propias ya que las marcas que se maquilan a diferentes clientes son producciones y ventas hecho a la medida o también llamadas *made to order*.

3.2 Limpieza, preprocesamiento y transformación de datos.

Como se menciona en el apartado anterior el proceso de extracción, transformación y carga de datos se realiza mediante proceso semiautomáticos programados en el *SQL Server Integration Services* y ejecutados en la base de datos *on premise* y *Python*.

SQL Server Integration Services es un paquete desarrollado por *Microsoft SQL Server*, con su principal característica de utilizar para una variedad de tareas de migración de datos (Gorini, 2010). La plataforma está diseñada con la finalidad de resolver procesos afines a la integración y aplicación de *workflows*.

Python creado en el año de 1991, es un lenguaje de programación informático en la actualidad el más popular para realizar análisis de datos. Su alto nivel en la actualidad es fácil de usar debido a su simplicidad y eficiencia (Caballero, 2022). Esta herramienta es utilizada ya que proporciona una gran variedad de bibliotecas como *Pandas* el cual permite realizar transformaciones, ordenamiento, realizar subconjuntos de los datos.

En el proyecto actual se utilizó el lenguaje de programación de *Pyhton* para el procesamiento de los datos tanto para darles formato, realizar uniones de diferentes *Data Set*. Para poder realizar el análisis exploratorio de los datos se crea una *Jupyter Notebook* en *Visual Studio Code* una aplicación instalada en una maquina local además de instalar cada uno de sus complementos necesarios para realizar la limpieza, preprocesamiento y transformación de los datos.

3.3 Identificación y descripción de variables

Desde la base de datos *on premise* se genera un *script* donde permite exportar los diferentes archivos en formato *.csv* como los siguientes *MaestroProductos*, *MaestroClientes*, *DetalleVentas* todos estos han pasado por un proceso de limpieza y transformación.

Lectura de los archivos en formato *.csv* con la herramienta Python

```
dClientes=pd.read_csv('C:/Users/paulb/Documents/Tesis_Puce/DataVenta/dClientes/dC
lientes.csv',delimiter=';')
dProductos=pd.read_csv('C:/Users/paulb/Documents/Tesis_Puce/DataVenta/dProductos/
Productos.csv',delimiter=';')
dVentas_22=pd.read_csv('C:/Users/paulb/Documents/Tesis_Puce/DataVenta/dVentas/Det
alle de las Ventas22.csv',delimiter=',')
```

```
dVentas_23=pd.read_csv('C:/Users/paulb/Documents/Tesis_Puce/DataVenta/dVentas/Detalle de las Ventas23.csv',delimiter=',')
dVentas_24=pd.read_csv('C:/Users/paulb/Documents/Tesis_Puce/DataVenta/dVentas/Detalle de las Ventas24.csv',delimiter=',')
```

A continuación, se describe cada uno de los datos maestros para luego juntar en un solo datan *DataSet**:***

a) **dClientes:** Maestro de Clientes
b) **dProductos:** Maestro de los productos (SKU)
c) **dVentas_22:** Detalle de ventas del año 2022
d) **dVentas_23:** Detalle de ventas del año 2023
e) **dVentas_24:** Detalle de ventas del año 2024

Está fracción de código en Python está limpiando y transformando la columna *idCliente* en el *DataSet* Ventas

```
Ventas['idCliente'] = Ventas['idCliente'].apply(str)
Ventas['idCliente']=Ventas['idCliente'].astype('str').str.replace(r'\.0$', '', regex=True)
Ventas['idCliente']=Ventas['idCliente'].apply(lambda x: str(x).zfill(10) if len(str(x)) == 9 else str(x))
Ventas['idCliente']=Ventas['idCliente'].apply(lambda x: str(x).zfill(13) if len(str(x)) == 12 else str(x))
Ventas=Ventas.fillna(0)
Ventas
```

Este código está realizando dos uniones (*merge*) en pandas para enriquecer el *DataSet* Ventas con información adicional de otros *DataSet* (*dProductos* y *idClientes*)

```
VentasAux=pd.merge(Ventas,dProductos,how='left',left_on='idMaterial',
right_on='idMaterial')
Ventas=pd.merge(VentasAux,dClientes,how='left',left_on='idCliente',
right_on='idCliente')
```

Finalmente, un solo *DataSet*

```
Ventas
```

En el proceso de unión de los diferentes archivos *.csv,* se utiliza la herramienta *Visual Studio Code* juntamente con *Jupyter Notebook,* se llegó al siguiente archivo plano con la siguiente estructura:

Nombre	**Tipo Dato**	**Tipo Variable**	**Descripción**
Años	int	Numérica	Año de venta
Mes	String	Categórica	Mes de venta
IdMaterial	String	Categórica	Id del producto
idCliente	String	Categórica	Id del cliente
VentaBruta	float	Numérica	Dólares vendidos
Devol	float	Numérica	Dólares Devueltos
Desc	Float	Numérica	Dólares de descuento
VentaNeta	Int	Numérica	Cantidad vendida
LitrosBrutos	Float	Numérica	Cantidad de Litros vendidos
LitrosNetos	Int	Numérica	Cantidad Litros netos vendidos
CantBruta	Float	Numérica	Cantidad Bruta vendida
CantDev	Int	Numérica	Cantidad Devuelta
CantNeta	Int	Numérica	Cantidad neta vendida
PproB	Float	Numérica	Precio Promedio Bruto
PproN	Float	Numérica	Precio Promedio Neto
Categoria	String	Categórica	Categoría del Productos
Subcategoria	String	Categórica	SubCategoria del producto
Segmento	String	Categórica	Segmento que tiene el producto
TipoEnvase	String	Categórica	Tipo envase del producto
RazonSocial	String	Categórica	Nombre del cliente
Provincia	String	Categórica	Provincia de venta.

Tabla 1: *Data Set*

3.4 Creaciones de nuevos *features*

Para este proyecto se selecciona la variable objetivo que es *CantNeta* y a partir de ella se realiza la creación de nuevas *features* el cual permitirá tener más características importantes con el fin de evaluar la precisión y el rendimiento de un modelo predictivo.

Se tomó la decisión estratégica de aplicar realizar un análisis ABC de las ventas la cual permite visualizar de mejor manera su cadena de suministro.

El análisis ABC es una metodología para realizar la agrupación y separar elementos basándose en su valor monetario. Este análisis se basa en dividir los datos de las ventas en diferentes categorías por prioridades definiendo criterios (Phipps, 2020).

Esta técnica se basa en la regla del Pareto 80/20 por ejemplo "*categoría A contribuyen en un 70% al margen de la empresa, los de la categoría B contribuyen en un 20% y, por último, los de la categoría C contribuyen en un 10%"* (Scott, 2024) .

La venta se divide siempre en tres categorías principales A, B, C, considerando su valor estratégico:

- **Categoría A:** productos que requieren un control riguroso
- **Categoría B:** productos con menor prioridad, que, sin embargo, deben ser manejados con un control de nivel promedio
- **Categoría C:** Productos de importancia secundaria que requieren una gestión de control relajada.

Tener la visibilidad de los ***SKUs*** (Productos) tienen como parte fundamental saber que cantidad de productos contribuyen mi mayor venta, el cual me permitirá tener la visibilidad de tener stock de seguridad permitiendo al área de planificación enforcé la producción de dichos productos (Scott, 2024).

Función donde se realiza la categorización para el análisis ABC

```
#Funcion para la Categorizacion
def aplicar_abc_por_año(df):
    df = df.sort_values(by='CantNeta', ascending=False)
    df['Porcentaje'] = (df['CantNeta'] / df['CantNeta'].sum()) * 100
    df['PorcentajeAcumulado'] = df['Porcentaje'].cumsum()

    def clasificar(row):
        if row['PorcentajeAcumulado'] <= 80:
            return 'A'
        elif  row['PorcentajeAcumulado'] <= 95:
            #RunPerc >=0.6 and RunPerc < 0.85:
            return 'B'
        else:
            return 'C'

    df['Cat'] = df.apply(clasificar, axis=1)
    return df
```

Una vez entendido el principio del análisis ABC el nuevo data set queda de la siguiente manera:

Experimento 1

El presente código agrupa ventas por año, mes y segmento, aplicando una categorización ABC. Luego, ordena los meses cronológicamente y genera una tabla pivote para visualizar la distribución de ventas por categoría. Así, facilita el análisis de tendencias mensuales y la proporción de cada categoría en las ventas totales.

```
#categorizacion de calculos ABC por mes
VentasABC_Mes=Ventas.groupby(['Año','Mes','segmento'])['CantNeta'].sum().reset_in
dex()
VentasABC_Mes =
VentasABC_Mes.groupby('Mes').apply(aplicar_abc_por_año).reset_index(drop=True)
VentasABC_Mes = VentasABC_Mes.sort_values(['Año','Mes'])
#VentasABC_Mes
orden_meses = [
    "enero", "febrero", "marzo", "abril", "mayo", "junio",
    "julio", "agosto", "septiembre", "octubre", "noviembre", "diciembre"
]

# Convertir la columna 'Mes' en una categoría con el orden definido
VentasABC_Mes['Mes'] = pd.Categorical(VentasABC_Mes['Mes'],
categories=orden_meses, ordered=True)
# Crear una tabla pivote
tabla_pivote = VentasABC_Mes.pivot_table(
    index=['Año', 'Mes'],        # Columnas para las filas
    columns='Cat',          # Columna para las categorías
    values='CantNeta',           # Columna para los valores
    aggfunc='sum',               # Función de agregación
    fill_value=0,                 # Rellenar valores faltantes con 0
    observed=True
).reset_index()

# Mostrar la tabla ordenada

tabla_pivote
```

Nombre	Tipo Dato	Tipo Variable	Descripción
Año	Int	numérica	Año de venta
Mes	String	categórica	Mes de venta
A	Int	numérica	Cantidad A
B	Int	numérica	Cantidad B
C	Int	numérica	Cantidad C

Tabla 2: *Experimiento 1 ABC*

Para el experimento numero 1 el *dataset* presenta una dimensión 36 filas y 5 columnas, lo que indica que contiene información organizada en cinco variables diferentes a lo largo de 36 registros está estructurada con cinco variables el cual permite almacenar información detallada sobre diferentes variables categóricas y cuantitativas relacionadas con las ventas.

Experimento 2

Para el nuevo experimento se crea un código donde realiza la **categorización ABC** de las ventas por mes y segmento, generando una tabla pivote para visualizar la distribución de cada categoría en las ventas mensuales.

```
#categorizacion de calculos ABC por mes
VentasABC_Mes=Ventas.groupby(['Año','Mes','segmento'])['CantNeta'].sum().reset_in
dex()
VentasABC_Mes =
VentasABC_Mes.groupby('Mes').apply(aplicar_abc_por_año).reset_index(drop=True)
VentasABC_Mes['Mes'] = pd.Categorical(VentasABC_Mes['Mes'].str.lower(),
categories=meses_orden, ordered=True)
VentasABC_Mes = VentasABC_Mes.sort_values(['Año','Mes'])
#VentasABC_Mes
orden_meses = [
    "enero", "febrero", "marzo", "abril", "mayo", "junio",
    "julio", "agosto", "septiembre", "octubre", "noviembre", "diciembre"
]

# Convertir la columna 'Mes' en una categoría con el orden definido
VentasABC_Mes['Mes'] = pd.Categorical(VentasABC_Mes['Mes'],
categories=orden_meses, ordered=True)
# Crear una tabla pivote
tabla_pivote = VentasABC_Mes.pivot_table(
    index=['Año', 'Mes','segmento'],        # Columnas para las filas
    columns='Cat',               # Columna para las categorías
    values='CantNeta',           # Columna para los valores
    aggfunc='sum',               # Función de agregación
    fill_value=0,                # Rellenar valores faltantes con 0
    observed=True
).reset_index()

# Mostrar la tabla ordenada

#
tabla_pivote
```

Nombre	Tipo Dato	Tipo Variable	Descripción
Año	Int		Año de venta
Mes	String	categórica	Mes de venta
Segmento	String	categórica	Segmento o categoría del producto
CantNeta	Int	numérica	Cantidad Vendida
A	Int	numérica	Cantidad A
B	Int	numérica	Cantidad B
C	Int	numérica	Cantidad C

***Tabla 3**: Experimento 2 ABC*

Para el experimento número 2 se realizan cambios como agregar el segmento (categoría) que el producto pertenece por ejemplo leche descremada, leche entera, leche condensada entre otras. El *dataset* quedo con una dimensión de 400 filas y 6 columnas lo que permite que esta información sea almacenada con variables cualitativas y cuantitativas.

Observando la dimensionalidad del *dataset* en cada uno de los experimentos se observó la existencia de muy pocas columnas, donde se vio la necesitas de realizar el análisis nuevamente el cual permitió crear otros *features* con el fin de mejorar el análisis y la toma de decisiones en varios aspectos como:

a) Permite identificar patrones estacionales o tendencias crecientes/decrecientes en las ventas.
b) Ayuda a detectar cambios en la demanda y anticipar ajustes en la producción o inventario.
c) Sirve como insumo para modelos de *Machine Learning* o métodos estadísticos que predicen ventas futuras basándose en datos histórico.

Lo que se realizo es crear tres columnas adicionales en el cual se agregó la cantidad del análisis ABC de los últimos 3 meses.

Experimento 1

El código desplaza (*shift*) los valores de las categorías A, B y C para incluir los datos de hasta 3 meses anteriores. Esto permite analizar tendencias y comparar ventas a lo largo del tiempo.

```
#Agregar los N meses anteriores dependiento la categoria
num = range(1, 4)
for n in num:
    tabla_pivote[f'A_M{n}'] =
tabla_pivote['A'].shift(n).fillna(0).astype(np.int64)
    tabla_pivote[f'B_M{n}'] =
tabla_pivote['B'].shift(n).fillna(0).astype(np.int64)
    tabla_pivote[f'C_M{n}'] =
tabla_pivote['C'].shift(n).fillna(0).astype(np.int64)
#union con la tabla Princial
VentasFinal=resultado = pd.merge(VentasM, tabla_pivote, on=['Año', 'Mes'],
how='left')
VentasFinal
```

Nombre	Tipo Dato	Tipo Variable	Descripción
Año	Int	numérica	Año de venta
Mes	String	categórica	Mes de venta
CantNeta	Int	numérica	Cantidad Vendida
A	Int	numérica	Cantidad A
B	Int	numérica	Cantidad B
C	Int	numérica	Cantidad C
Cant_M1	Int	numérica	Cantidad del mes 1 anterior
Cant_M2	Int	numérica	Cantidad del mes 2 anterior
Cant_M3	Int	numérica	Cantidad del mes 3 anterior
A_M1	Int	numérica	Cantidad A mes anterior
B_M1	Int	numérica	Cantidad B mes anterior
C_M1	Int	numérica	Cantidad C mes anterior
A_M2	Int	numérica	Cantidad A del segundo mes anterior
B_M2	Int	numérica	Cantidad B del segundo mes anterior
C_M2	Int	numérica	Cantidad C del segundo mes anterior
A_M3	Int	numérica	Cantidad A del tercer mes anterior
B_M3	Int	numérica	Cantidad B del tercer mes anterior
C_M3	Int	numérica	Cantidad C del tercer mes anterior

Tabla 4: *Agregación nuevas columnas Experimento 1*

El detalle que se muestra en la tabla numero 4 representan como queda el *dataset* para el experimento numero 1 donde se realiza el análisis ABC con el fin de crear nuevas *features* el cual permitió tener nuevos registros y el *dataset* quedo con 36 filas y 18 columnas.

Experimento 2

Para este experimento se presenta el código desplaza los valores de ventas de las categorías A, B y C dentro de cada segmento para obtener los datos de los 3 meses anteriores.

```
#Agregar los N meses anteriores dependiento la categoria
num = range(1, 4)
for n in num:
   #VentasMS[f'Cant.Mes M_{n}'] =
VentasMS.groupby(['segmento'])['CantNeta'].shift(n).fillna(0).astype(np.int64)
    tabla_pivote[f'A_M{n}'] =
tabla_pivote.groupby(['segmento'])['A'].shift(n).fillna(0).astype(np.int64)
    tabla_pivote[f'B_M{n}'] =
tabla_pivote.groupby(['segmento'])['B'].shift(n).fillna(0).astype(np.int64)
    tabla_pivote[f'C_M{n}'] =
tabla_pivote.groupby(['segmento'])['C'].shift(n).fillna(0).astype(np.int64)
```

```
VentasFinal=resultado = pd.merge(VentasMS, tabla_pivote, on=['Año',
'Mes','segmento'], how='left')
```

Nombre	Tipo Dato	Tipo Variable	Descripción
Año	Int	numérica	Año de venta
Mes	String	categórica	Mes de venta
Segmento	String	categórica	Segmento o categoría del producto
CantNeta	Int	numérica	Cantidad Vendida
A	Int	numérica	Cantidad A
B	Int	numérica	Cantidad B
C	Int	numérica	Cantidad C
A_M1	Int	numérica	Cantidad A mes anterior
B_M1	Int	numérica	Cantidad B mes anterior
C_M1	Int	numérica	Cantidad C mes anterior
A_M2	Int	numérica	Cantidad A del segundo mes anterior
B_M2	Int	numérica	Cantidad B del segundo mes anterior
C_M2	Int	numérica	Cantidad C del segundo mes anterior
A_M3	Int	numérica	Cantidad A del tercer mes anterior
B_M3	Int	numérica	Cantidad B del tercer mes anterior
C_M3	Int	numérica	Cantidad C del tercer mes anterior

***Tabla 5**: Agregación nuevas columnas Experimento 2*

Siguiendo la misma línea del experimento anterior para el experimento numero 2 el *dataset* realizado el análisis ABC queda de la siguiente manera con una dimensión de 400 filas y 16 columnas permitiendo que el *dataset* tener mayor cantidad de datos.

Tener las ventas de los meses anteriores ayuda a tomar decisiones más informadas, optimizar la gestión de inventario y mejorar la precisión de las predicciones de demanda.

3.5 Visualización de variable objetivo

Los datos que recopilan las empresas cada día son más valiosos para realizar un análisis, esto nos permite tener una visión grafica de las ventas de Gloria Ecuador. En la Figura 3 muestra una tendencia decreciente en la cantidad de ventas a lo largo de los tres años. El año 2022 tuvo la mayor cantidad de ventas, seguido de 2023, y 2024 tuvo la menor cantidad de ventas.

El código que a continuación se detalla genera un gráfico de barras que muestra la cantidad neta de ventas por año. Agrupa los datos, suma las ventas y las representa visualmente. Se convierten valores a millones y se ajustan detalles gráficos para mayor claridad y legibilidad.

```
dfVentasAño=Ventas.groupby('Año')['CantNeta'].sum().reset_index()
fig, ax = plt.subplots()
ax.bar(dfVentasAño['Año'],dfVentasAño['CantNeta'])
bars = ax.bar(dfVentasAño['Año'], dfVentasAño['CantNeta'])
for bar in bars:
    yval = bar.get_height()
    # Convertir el valor a millones
    yval_millones = yval / 1_000_000
    # Mostrar el valor formateado con "M"
    ax.text(
        bar.get_x() + bar.get_width() / 2,
        yval,
        f"{yval_millones:.2f}M",  # Redondear a 2 decimales
        ha='center',
        va='bottom',
        fontsize=10
    )
ax.set_xticks(dfVentasAño['Año'])
ax.set_xticklabels(dfVentasAño['Año'])
ax.set_xlabel('Año')
ax.set_ylabel('Canridad Venta')
ax.set_title('Cantidad Venta por Año')
plt.show()
```

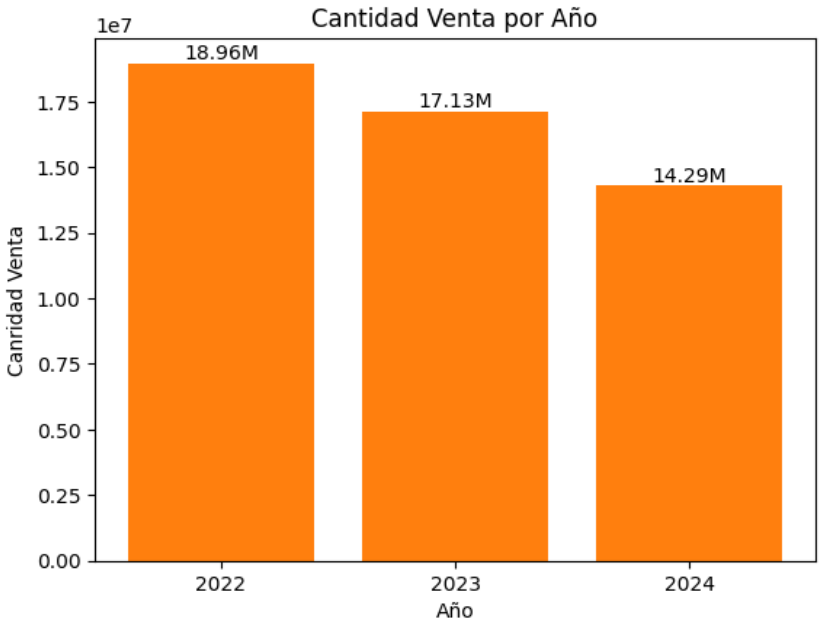

***Figura 3**: Venta Neta por Años*

Tomando en cuenta que Grupo Gloria Ecuador está inmersa en el consumo masivo, se realiza un análisis de un gráfico de barras donde se puede observar cómo está distribuida las ventas al largo de los años y dividías por su segmento que perteneces ver Figura 4.

A continuación, se presenta un nuevo código que genera un gráfico de barras apiladas porcentual para mostrar la distribución de ventas por segmento (o categoría) a lo largo de los años.

```
VentaT = VentaTop.pivot_table(index=['Año'], columns='segmento',
values='CantNeta', aggfunc='sum', fill_value=0)
# Calcular el porcentaje de ventas por categoría
porcentaje = VentaT.div(VentaT.sum(axis=1), axis=0) * 100

#VentaTop.plot(kind='bar', stacked=True, figsize=(20, 8))
ax = porcentaje.plot(kind='bar', stacked=True, figsize=(18, 7), colormap='tab10')
# Agregar porcentajes dentro de las barras
for i, bar_group in enumerate(ax.containers):
    for bar in bar_group:
        if bar.get_height() > 0:  # Solo mostrar valores no nulos
            height = bar.get_height()
            ax.text(
                bar.get_x() + bar.get_width() / 2,
                bar.get_y() + height / 2,
                f'{height:.1f}%',
                ha='center',
                va='center',
                fontsize=11,
                color='black'
            )
# Etiquetas y título
plt.title('Distribución Porcentual de Ventas por Categoría (Año)', fontsize=10)
plt.xlabel('Año y Mes', fontsize=10)
plt.ylabel('Porcentaje', fontsize=10)
# Mostrar el gráfico
#plt.tight_layout()
plt.show()
```

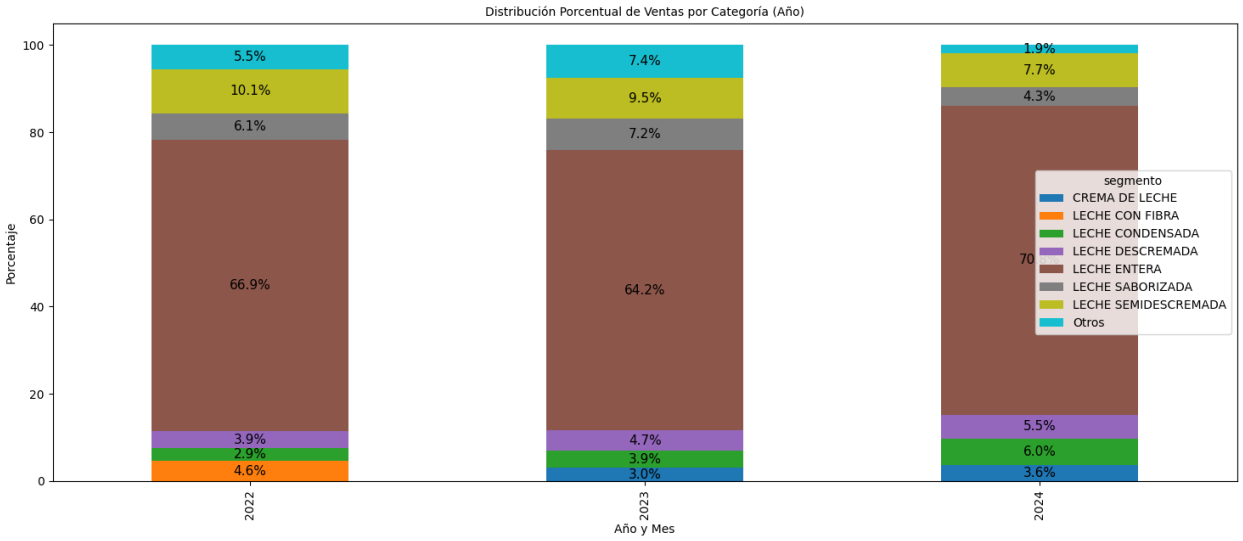

Figura 4: *Distribución de las Ventas*

Como se observa en la Figura 4, el mayor porcentaje de la venta en los últimos tres años está distribuida en su mayoría en la venta de productos lácteos especialmente en la Leche Entera, seguida de leche en su presentación de semidescremada, además de poder observar que para el año 2023 y 2024 la venta de leche con fibra desaparece.

El presente código genera un gráfico de líneas que muestra la evolución de las ventas a lo largo del tiempo (por año y mes).

```
meses_orden = ['enero', 'febrero', 'marzo', 'abril', 'mayo', 'junio', 'julio',
'agosto', 'septiembre', 'octubre', 'noviembre', 'diciembre']
dfVentasAño=Ventas.groupby(by=['Año','Mes'])['CantNeta'].sum().reset_index()
# Convertir la columna 'Mes' a tipo categórico con el orden deseado
dfVentasAño['Mes'] = pd.Categorical(dfVentasAño['Mes'], categories=meses_orden,
ordered=True)
dfVentasAño = dfVentasAño.sort_values('Mes')
df_pivot = dfVentasAño.pivot_table(index=['Año', 'Mes'], values='CantNeta',
aggfunc='sum', fill_value=0, observed=False)

#df_pivot.plot(kind='bar', stacked=True, figsize=(20, 6))
df_pivot.plot(kind='line', stacked=True, colormap='coolwarm',
marker='o',figsize=(15, 5),fontsize=8)
# Etiquetas y título

plt.title('Ventas por Año / Mes', fontsize=8)
plt.xlabel('Año y Mes', fontsize=8)
plt.ylabel('Ventas', fontsize=8)
plt.xticks(rotation=90)
plt.tight_layout()
#plt.minorticks_on()
```

```
plt.grid(True)
plt.show()

#grafico de lineas
```

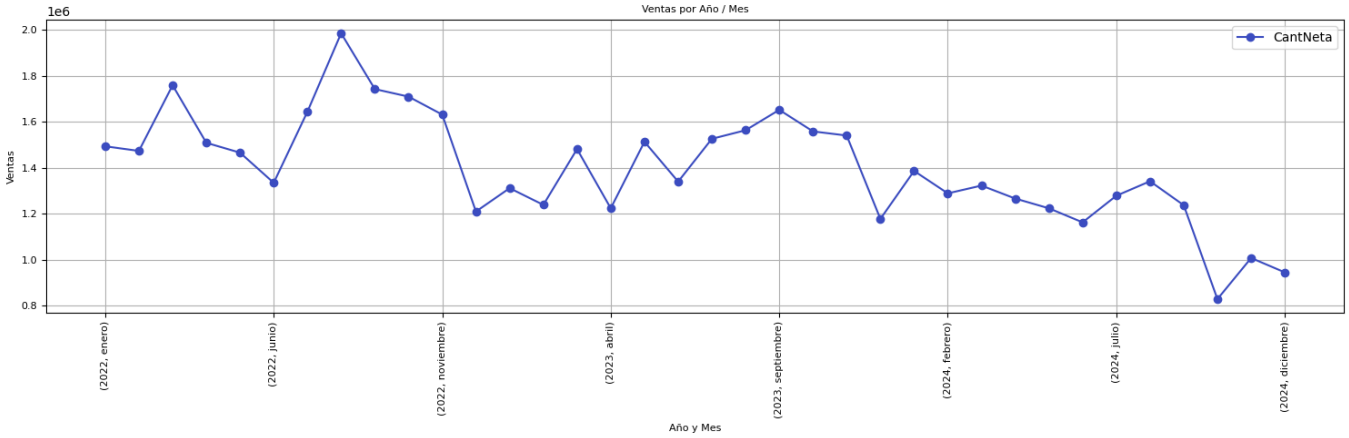

Figura 5: *Venta Distribuida por Año - Mes*

El presente grafico de la Figura 5 permite visualizar la evolución de las ventas que ha realizado Gloria Ecuador a lo largo del tiempo, en el eje X está representado los años y los meses en un periodo de enero 2022 a diciembre 2024 mientras que en el eje de las Y permite visualizar las cantidades de ventas expresadas en una escala de millones.

A lo largo de periodo analizado se visualiza tendencias tanto ascendentes y descendentes, en algunos meses concretos como a medio año del 2022 y del 2023 se observa una demanda alta de productos concluyendo que estos meses es donde la producción no debe caer para evitar la ruptura de stock. Por otro lado, así como existe meses de alta demanda, la Figura 5 permite observar meses con caídas notables en ciertos meses como finales del 2024, esto puede verse afectado con factores estacionales, días festivos e incluso en el mercado.

Realizar un análisis exploratorio de los datos permite deducir que Gloria Ecuador con la utilización de modelos de *Machine Learning* para la predicción de la demanda le permitirá anticipar a cambios estacionales, cambios del mercado con la finalizar de mejorar la estrategia de producción y distribución. Además, permitirá realizar un análisis útil para analizar factores externos que se presenten como tendencias económicas, comportamiento del cliente y promociones con la finalizar de anticiparse a lo que puede influir las variaciones de consumo de productos.

Como parte del análisis de la información se realiza un análisis exploratorio de la variable objetivo para ello se utilizó un diagrama de cajas. Este diagrama es una herramienta visual fundamental para analizar la distribución de la variable objetivo en un modelo predictivo (Churchil, 2005) . Este gráfico resume características clave como la mediana, los cuartiles y la presencia de valores atípicos, permitiendo una comprensión rápida de la dispersión y simetría de los datos.

El código crea un boxplot para analizar la distribución de la cantidad neta de ventas (*CantNeta*) por año. Esto ayuda a identificar la mediana, cuartiles y valores atípicos. Se habilita una cuadrícula (*plt.grid(True)*) para mejorar la legibilidad del gráfico. Es útil para detectar tendencias y variabilidad en los datos.

```
plt.grid(True)
sns.boxplot(x=Ventas['Año'],y=Ventas['CantNeta'],data=Ventas)
plt.show()
```

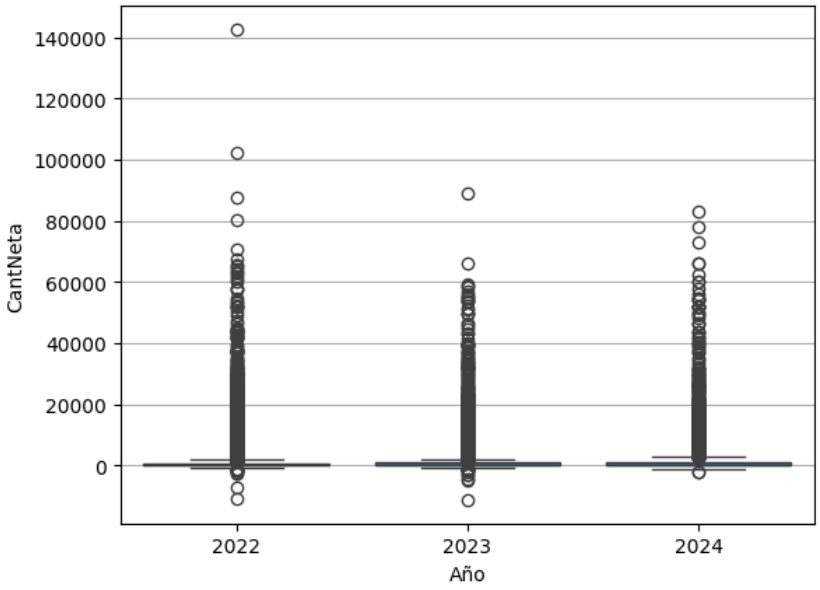

Figura 6:*Diagrama de Cajas - Venta Neta*

Como se muestra en la Figura 6, se observa una gran cantidad de valores atípicos. En negocios de ventas masivas a gran escala, la aparición de estos valores es común debido a factores como variaciones estacionales, promociones, cambios en la demanda y eventos imprevistos (Ablan, 2024). Lejos de ser anomalías descartables, estos valores pueden proporcionar información clave sobre tendencias, oportunidades de mercado y posibles riesgos, por lo que su análisis resulta fundamental para la toma de decisiones estratégicas.

Consideraciones que se presentan en las ventas para la visualización de valores atípicos, a continuación, se detallan los principales:

a) **Celebraciones especiales:** en las ventas son situaciones que pueden generar un aumento temporal en la demanda de productos o servicios. Estos eventos pueden influir en las predicciones de ventas y la gestión de inventarios (Danilova, 2024). Ejemplos de estos eventos Navidad, Black Driday.
b) **Calidad de los Datos:** Antes de decir que un valor es atípico primero es asegurarse que los datos no sean resultado de un error involuntario además que una baja calidad de los datos puede llevar a decisiones erróneas y afectar el rendimiento de los modelos predictivos (Aliyev, 2023).

c) **Decisiones del negocio:** Los valores que aparentemente se visualizan como atípicos estos valores pueden ser producto de una venta inusual el cual se debe realizar una investigación respectiva con el fin de tener la trazabilidad del o los productos que se marcan como atípicos ya que puede estar relacionado por campañas de marketing
d) **Cambios del mercado:** cambios en la estrategia de los competidores, como descuentos significativos, lanzamientos de productos innovadores o campañas agresivas, pueden alterar las dinámicas del mercado (Laskova, 2022). Esto puede generar cambios abruptos en la demanda de productos, creando valores atípicos que reflejan la alteración de las tendencias normales del mercado.

Adicionalmente en la Figura 5 se visualiza negativos en las ventas, en el contexto de la gestión de datos y decisiones de negocio, términos como devoluciones, descuentos, ajustes, movimientos internos y notas de crédito son fundamentales para el manejo eficiente de las operaciones. Cada uno de estos puede tener un impacto significativo en la predicción de la demanda, la gestión de inventarios, la optimización de precios y la toma de decisiones financieras. A continuación de describe cada una de ellas:

a) **Devoluciones:** Las devoluciones se producen cuando los clientes regresan productos previamente comprados, ya sea por defectos, insatisfacción o cambios en las preferencias. Un incremento inesperado en las devoluciones puede verse como un valor atípico, indicando un problema en la calidad del producto o en la percepción del cliente.
a) **Descuentos:** son promociones que reducen el precio de los productos o servicios para incentivar las compras de los productos. Estas ventas extraordinarias por un descuento pueden generar picos en los datos, que se pueden considerar atípicos si no se interpretan adecuadamente en el contexto de la campaña
b) **Ajustes** se refieren a las correcciones o modificaciones en los registros contables, financieros tanto en las ventas como los inventarios, esto conlleva a que un ajuste incorrecto o no previsto puede generar valores inusuales en los registros, afectando los análisis de tendencias y la toma de decisiones.
c) **Movimientos Internos:** el traslado de productos dentro de la empresa ya sea entre almacenes, entre departamentos o entre diferentes ubicaciones de venta, estos tipos de movimientos internos que no se registren correctamente o que no estén alineados con la demanda proyectada pueden generar valores atípicos en los inventarios.
d) **Notas de Crédito:** documentos emitidos por un proveedor o vendedor para registrar la devolución de dinero a un cliente, o bien para anular total o parcialmente una factura. Estos documentos emitidos por un proveedor o vendedor para registrar la devolución de dinero a un cliente, o bien para anular total o parcialmente una factura

En función al análisis los factores que pueden generar valores atípicos en los análisis de datos y deben ser manejados adecuadamente para tomar decisiones informadas y optimizar las operaciones del negocio por tal motivo se decide mantener los valores visualizados como atípicos con el fin de implementar los modelos para el proyecto.

3.6 Selección de Modelos

Para la selección de los algoritmos de aprendizaje es un proceso crucial ya que influye directamente en el rendimiento de un sistema de predicción o clasificación. Para elegir el modelo adecuado, se deben tener en cuenta diversos factores como el tipo de problema, las características del conjunto de datos, la precisión deseada, el tiempo de entrenamiento y la complejidad del modelo.

Los modelos predictivos han demostrado su valor en múltiples sectores, desde el análisis de tendencias de consumo hasta aplicaciones críticas en medicina (diagnóstico predictivo), economía (pronóstico de mercados) e ingeniería (simulación de diseños). Gracias a estas herramientas, las empresas pueden simular escenarios complejos, evaluar riesgos, optimizar recursos y mejorar significativamente su eficiencia operativa.

Los algoritmos de ensamble para la implementación en este proyecto como *Random Forests*, *Gradient Boosting Machine* (GBM) y *XGBoost*, son ampliamente utilizados en la predicción de la demanda debido a su capacidad para manejar grandes volúmenes de datos, identificar patrones complejos y mejorar la precisión predictiva (Steven, 2021) .

3.6.1 *Random Forests*

Es un modelo de ensamble basado en la técnica de *bagging* (*Bootstrap Aggregating*), en el que se utilizan múltiples árboles de decisión entrenados de manera independiente con subconjuntos aleatorios de los datos. La agregación de las predicciones de estos árboles ayuda a reducir el sobreajuste y mejora la capacidad de generalización del modelo (Fernández, 2023). Es una opción eficaz cuando se dispone de grandes cantidades de datos con características complejas y relaciones no lineales.

Una de las principales ventajas de *Random Forests* es que requiere poca sintonización de hiperparámetros, lo que lo hace fácil de usar. Además, puede manejar tanto problemas de clasificación como de regresión, y tiene una excelente capacidad para manejar datos faltantes

(Fernández, 2023). Sin embargo, su principal limitación es que puede ser menos preciso que otros algoritmos de ensamble más avanzados, como *Gradient Boosting* y *XGBoost*, especialmente en tareas donde la precisión es crítica.

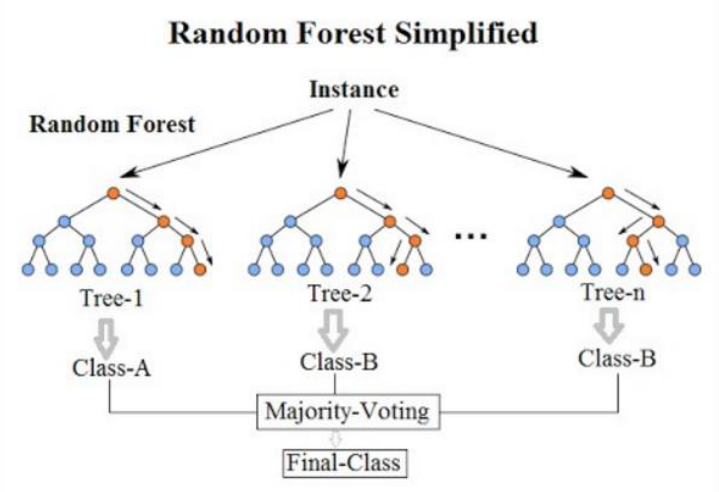

***Figura 7**: Random Forest*
Fuente: https://acortar.link/YdrjuM

3.6.2 Gradient Boosting Machine (GBM)

Es un algoritmo de *boosting* que construye árboles de decisión de manera secuencial. Cada nuevo árbol intenta corregir los errores cometidos por los árboles anteriores, lo que resulta en un modelo que se ajusta muy bien a los datos y produce predicciones precisas. En lugar de entrenar los árboles de forma independiente como en *Random Forests*, *GBM* los ajusta de manera que los errores anteriores son "corregidos" en el siguiente árbol (Rosati G. , 2019).

GBM es extremadamente poderoso y puede ser ajustado para obtener altos niveles de precisión en la mayoría de los problemas. Sin embargo, es más susceptible al sobreajuste si no se ajustan adecuadamente los hiperparámetros, y requiere más tiempo de entrenamiento debido a la naturaleza secuencial del algoritmo (Rosati G. , 2019). Es ideal cuando se tiene tiempo para realizar una sintonización cuidadosa de parámetros, y cuando se busca maximizar la precisión del modelo.

3.6.3 XGBoost

Es una implementación optimizada de *Gradient* Boosting, que incluye características adicionales como regularización, una mayor velocidad en el entrenamiento y el manejo eficiente de datos faltantes (Sanz, 2021) . Gracias a estas mejoras, *XGBoost* ha demostrado ser uno de los algoritmos más poderosos y eficientes en términos de rendimiento, ganando popularidad en competiciones de *machine learning* y aplicaciones comerciales.

Una de las mayores ventajas de *XGBoost* es su capacidad para manejar grandes volúmenes de datos con altísima precisión. La regularización también ayuda a prevenir el sobreajuste, lo que lo convierte en una opción robusta para tareas complejas (Sanz, 2021). Sin embargo, la sintonización de hiperparámetros en *XGBoost* puede ser más desafiante, y el modelo requiere más tiempo de entrenamiento que *Random Forest*s, especialmente cuando se trabaja con conjuntos de datos masivos.

3.6.4 Desventajas de los modelos implementados

***Random Forest*:** Aunque *Random Forests* es muy fuerte y genera predicciones sólidas, no siempre es tan exacto como otros modelos de ensamble más avanzados, presenta problemas especialmente en problemas complicados que requieren una modelización más fina de las relaciones no lineales entre las variables (Espinosa, 2020).

***Gradient Boosting Machines* (*GBM*):** el principal inconveniente de *GBM* para Grupo Gloria Ecuador es que, dado que es un algoritmo secuencial, el tiempo de entrenamiento puede ser considerablemente largo, especialmente cuando se trabaja con grandes volúmenes de datos históricos de demanda o cuando se ajustan múltiples parámetros para optimizar el rendimiento (Gutiérrez, 2023). Este aspecto puede ser problemático en entornos empresariales donde la rapidez de la predicción y la actualización de los modelos son esenciales para la toma de decisiones en tiempo real.

***XGBoost*:** el proceso de entrenamiento podría ser demorado cuando se utilizan datos históricos con muchas variables como promociones, precios, estacionalidad, etc., lo cual podría generar inconvenientes en situaciones donde la predicción rápida es crucial (Rodriguez, 2018). Por ejemplo, en la gestión de inventarios, si la demanda no se predice con suficiente antelación, podría haber problemas de **sobre stock** o desabastecimiento, afectando la eficiencia operativa de Grupo Gloria Ecuador.

CAPÍTULO IV: RESULTADOS Y PROPUESTA DE SOLUCIÓN

Se llevaron a cabo dos experimentos con diferentes dimensiones de datos para evaluar el rendimiento de los algoritmos ensamble en distintos escenarios.

El primer experimento consistió en un conjunto de datos con 36 filas y 18 columnas. Este conjunto de datos reducido permitió evaluar la capacidad de los algoritmos Ensamble para trabajar con datos limitados y detectar patrones en un espacio de características más compacto. Se aplicaron métodos como *Random Forest, XGBRegressor y GradientBoosting* para analizar la efectividad de cada uno en un entorno con menor cantidad de datos.

El segundo experimento se llevó a cabo con un conjunto de datos de mayor escala, compuesto por 400 filas y 16 columnas. Este conjunto de datos permitió evaluar la capacidad de los algoritmos Ensamble para manejar volúmenes de información más grandes y determinar su robustez en condiciones de mayor complejidad. Se aplicaron los mismos algoritmos utilizados en el primer experimento, permitiendo una comparación directa entre los resultados obtenidos con un *dataset* pequeño y uno de mayor tamaño.

Ambos experimentos tuvieron como objetivo analizar el impacto del tamaño del conjunto de datos en el desempeño de los algoritmos Ensamble, observando métricas MSE, RMSE, R^2, MAPE (%). Los resultados obtenidos permitirán determinar la idoneidad de cada enfoque en función de la cantidad de datos disponibles y la complejidad del problema a resolver.

4.1 Descripción de los Experimentos

Experimento número 1

El dataset que se tomó para el siguiente experimento consta de 36 filas y 18 columnas conformado por *Año, Mes, CantNeta, A, B, C, Cant_M1, Cant_M2, Cant_M3, A_M1, B_M1, C_M1, A_M2, B_M2, C_M2, A_M3, B_M3, C_M3*. Todo este dataset representan los registros de las ventas realizadas desde del año 2022 al 2024.

La dimensionalidad del dataset para este experimento es muy pequeño ya que solo consta de registros de ventas a nivel Año, Mes donde nos brinda registros de solo 36 filas el cual se experimenta el comportamiento de los algoritmos a ese nivel. Para entrenar cada uno de los modelos de predicción, se realizó un filtro de fecha lo que permitió fraccionar el conjunto de datos en entrenamiento y prueba donde las fechas anteriores al *31 de octubre de 2024* se tomaron como conjunto de entrenamiento mientras que las fechas posteriores como conjunto de prueba. Este procedimiento garantiza que el modelo aprenda de datos históricos y se evalúe con datos futuros, imitando una situación del mundo real.

Experimento número 2

Para el siguiente experimento se toma como base el dataset del experimento numero 1 agregando una característica el cual permite tener más registros que el primer dataset, la columna que se agrego fue el segmento (categoría) donde la dimensionalidad quedo en 400

filas y 16 columnas conformadas por Año, *Mes, CantNeta, A, B, C, Cant_M1, Cant_M2, Cant_M3, A_M1, B_M1, C_M1, A_M2, B_M2, C_M2, A_M3, B_M3, C_M3, Segmento.*

La división de los datos en este experimento sigue las mismas condiciones que en el experimento uno, las fechas anteriores al 31 de octubre de 2024 se consideran parte del conjunto de entrenamiento, mientras que las posteriores conforman el conjunto de prueba.

Las columnas eliminadas para ambos experimentos son *'CantNeta', 'Año', 'Mes','fecha_maxima','A','B','C',* en especial 'A','B','C' son columnas redundantes que no contribuyen a mejorar la predicción de CantNeta.

La variable objetivo que el modelo intentará predecir es **CantNeta**, representa la cantidad neta de ventas. Este enfoque permite alimentar un modelo de aprendizaje automático, como un *Random Forest, XGBRegressor y GradientBoosting*, para predecir futuras cantidades de ventas basadas en patrones históricos. El uso adecuado de filas y columnas es crucial para asegurar la calidad del modelo y la validez de las predicciones.

4.2 Random Forest

Las siguientes tablas se observó los principales datos obtenidos con los modelos propuestos en los diferentes experimentos

4.2.1 Experimento 1

Dataset	MSE	RMSE	R^2	MAPE (%)
Entrenamiento	5.936974e+09	77051.76	0.8764	4.23
Prueba	8.721137e+10	295315.71	-88.1840	30.28

***Tabla 6**: Random Forest - Experimento 1*

En base a los resultados obtenidos de las métricas del modelo, se observa un rendimiento significativamente mejor en el conjunto de entrenamiento en comparación con el conjunto de prueba. En el conjunto de entrenamiento, el modelo presenta un MSE (Error Cuadrático Medio) de 5.94e+09, un RMSE de 77,051.76, un R^2 de 0.8764 y un MAPE del 4.23%, lo que indica que el modelo se ajusta adecuadamente a los datos de entrenamiento y tiene un buen desempeño predictivo.

Sin embargo, al analizar el conjunto de prueba, el modelo presenta un MSE de 8.72e+10, un RMSE de 295,315.71, un R^2 negativo de -88.184 y un MAPE del 30.28%, lo que sugiere que el modelo generaliza mal en datos no vistos.

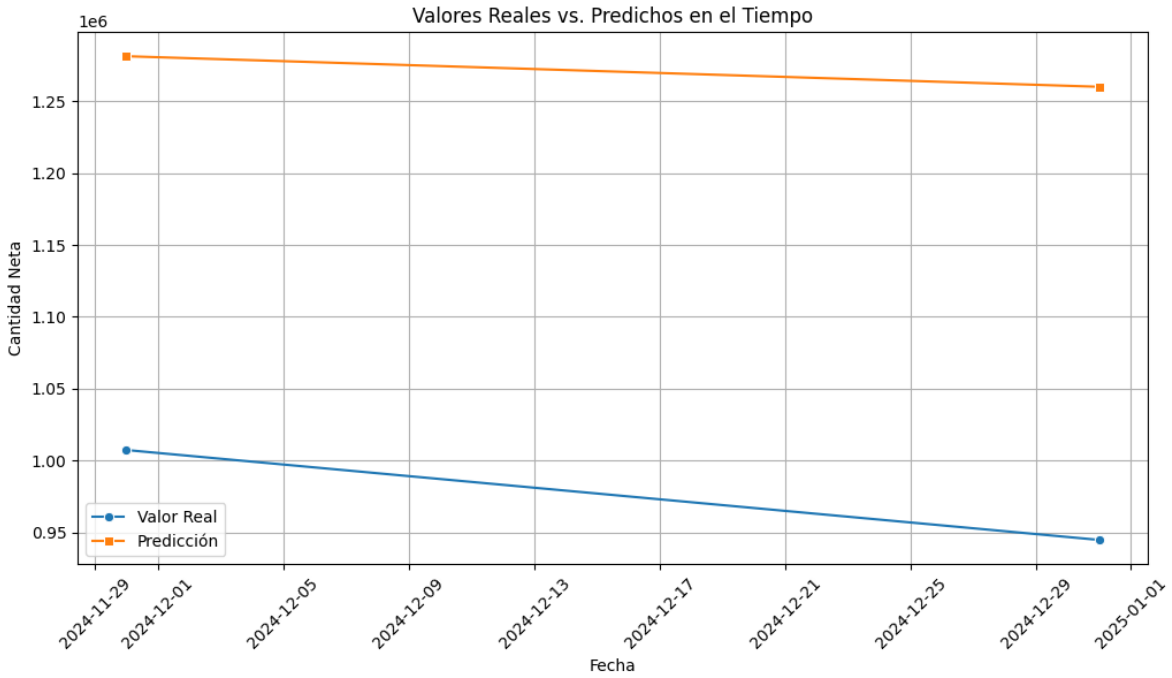

***Figura 8**:Random Forest - Experimento 1*

Se observa que tanto los valores reales como los predichos muestran una tendencia a la baja, aunque los valores predichos se mantienen consistentemente por encima de los valores reales. Esto sugiere que el modelo de predicción utilizado tiende a sobreestimar los valores reales

4.2.2 Experimento 2

Dataset	MSE	RMSE	R^2	MAPE (%)
Entrenamiento	2.902434e+09	53874.245	0.958	
Prueba	6.613170e+09	81321.396	0.865	27.449

***Tabla 7:** Random Forest - Experimento 2*

El análisis del modelo muestra un buen desempeño durante el entrenamiento, con un R^2 de 0.958, lo que indica que el modelo puede explicar un alto porcentaje de la variabilidad de los datos. Además, el MSE de 2.902434e+09 y el RMSE de 53874.245 durante el entrenamiento son relativamente bajos, lo que sugiere que el modelo se ajusta bien a los datos de entrenamiento. Sin embargo, al evaluar el rendimiento en los datos de prueba, se observa una disminución significativa en el R^2, que baja a 0.865, lo que indica una menor capacidad de generalización del modelo. Además, el MAPE de 27.449% en la prueba señala un margen considerable de error, lo que sugiere que el modelo podría mejorar en su capacidad predictiva al trabajar con datos no vistos.

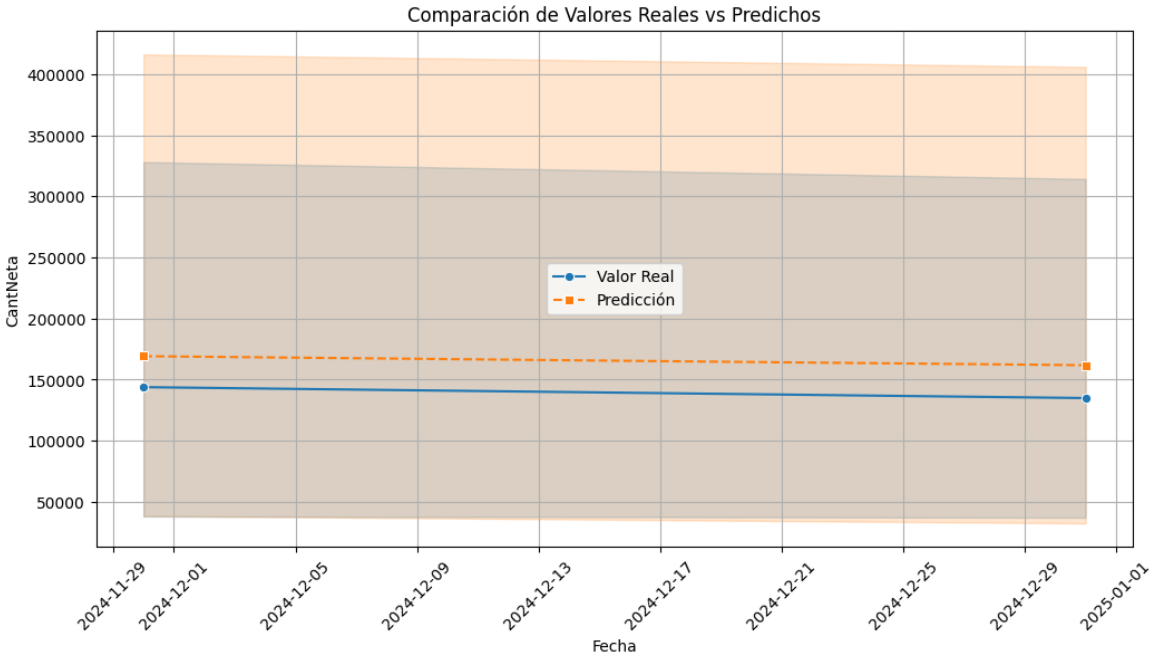

***Figura 9**:Random Forest - Experimento 2*

La gráfica muestra que el modelo de predicción tiene un buen desempeño al seguir la tendencia general de los datos, pero podría beneficiarse de ajustes para mejorar la precisión en los puntos de datos con valores extremos. La diferencia en la magnitud de los valores, sugiere que el modelo podría necesitar ser ajustado para reducir la sobreestimación en estos puntos.

4.3 XGBRegressor

4.3.1 Experimento 1

Dataset	MSE	RMSE	R^2	MAPE (%)
Entrenamiento	1.011792e+06	1005.87	0.99	0.041
Prueba	8.252645e+10	287274.17	-83.39	28.761

***Tabla 8:** XGBRegressor - Experimento 1*

Los resultados obtenidos del análisis muestran un rendimiento sobresaliente en el conjunto de entrenamiento, con un R^2 de 0.999, lo que indica que el modelo ajusta muy bien los datos, explicando prácticamente toda la variabilidad. Además, el MSE de 1.011792e+06 y el RMSE de 1005.87 son valores relativamente bajos, lo que resalta la precisión del modelo en esta fase. Sin embargo, los resultados del conjunto de prueba indican un desempeño mucho más deficiente. El R^2 de -83.39 es muy bajo, lo que sugiere que el modelo no tiene capacidad para generalizar correctamente a nuevos datos. El MAPE de 28.761% también indica un margen de error significativo.

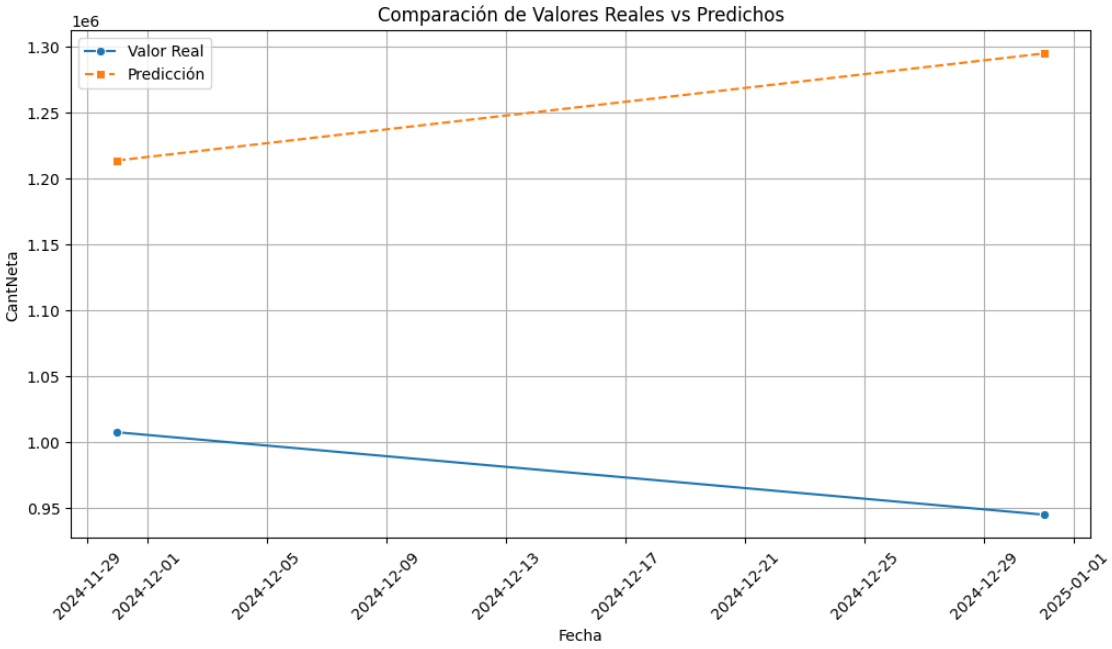

***Figura 10**:XGBRegressor - Experimento 1*

Se observa que los valores reales muestran una tendencia a la baja, mientras que los valores predichos muestran una tendencia al alza. Esto indica que el modelo de predicción no está capturando correctamente la tendencia de los datos reales.

4.3.2 Experimento 2

Dataset	MSE	RMSE	R^2	MAPE (%)
Entrenamiento	2.559832e+09	50594.785	0.963	
Prueba	3.587030e+10	189394.56	0.271	34.055

***Tabla 9**:XGBRegressor - Experimento 2*

El modelo presenta una diferencia significativa en su desempeño entre el conjunto de entrenamiento y el conjunto de prueba, lo que indica un posible sobreajuste. En el conjunto de entrenamiento, el modelo tiene un R^2 de 0.963, lo que sugiere que es capaz de explicar el 96.3% de la variabilidad de los datos, mientras que en el conjunto de prueba este valor baja drásticamente a 0.271, lo que refleja una capacidad limitada para generalizar a nuevos datos. Además, el RMSE y MSE en el conjunto de prueba son considerablemente más altos, lo que indica errores más grandes en las predicciones de los datos no vistos. El MAPE del 34.055% en el conjunto de prueba resalta una tasa de error considerable en las predicciones.

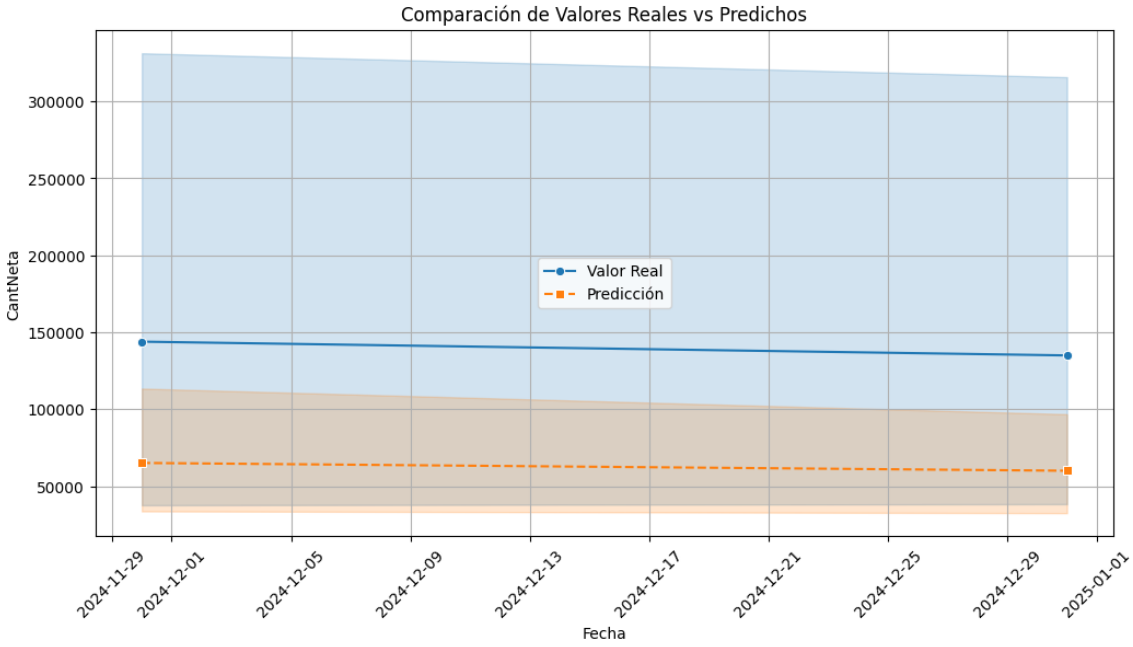

Figura 11: *XGBRegressor - Experimento 2*

Se observa que el modelo logra capturar la tendencia general de los datos, reflejada en la similitud entre las líneas de valores reales y predichos. Sin embargo, el modelo muestra limitaciones en la predicción donde está por debajo de los valores reales donde la diferencia entre los valores reales y predichos es más pronunciada.

4.4 GradientBoosting

4.4.1 Experimento 1

Dataset	MSE	RMSE	R^2	MAPE (%)
Entrenamiento	9.532130e+05	976.326	0.999	0.056
Prueba	5.596927e+10	236578.25	-56.23	24.16

Tabla 10: *GradientBoosting - Experimento 1*

En conclusión, los resultados obtenidos para los dos conjuntos de datos, entrenamiento y prueba, muestran una gran diferencia en el desempeño del modelo. Para el conjunto de entrenamiento, el MSE (Error Cuadrático Medio) es de 953,213, lo que implica que el modelo tiene un ajuste muy cercano a los datos reales. Además, el RMSE (Raíz del Error Cuadrático Medio) es de 976.33, lo que también refleja una alta precisión. El valor de R^2 de 0.999 sugiere que el modelo explica prácticamente toda la variabilidad de los datos.

Sin embargo, para el conjunto de prueba, los resultados son preocupantes. El MSE se incrementa drásticamente a 55,969,270,000, lo que indica un error mucho mayor.

El RMSE es 236,578, y el R^2 es negativo (-56.23), lo que indica que el modelo no se ajusta bien a los datos de prueba.

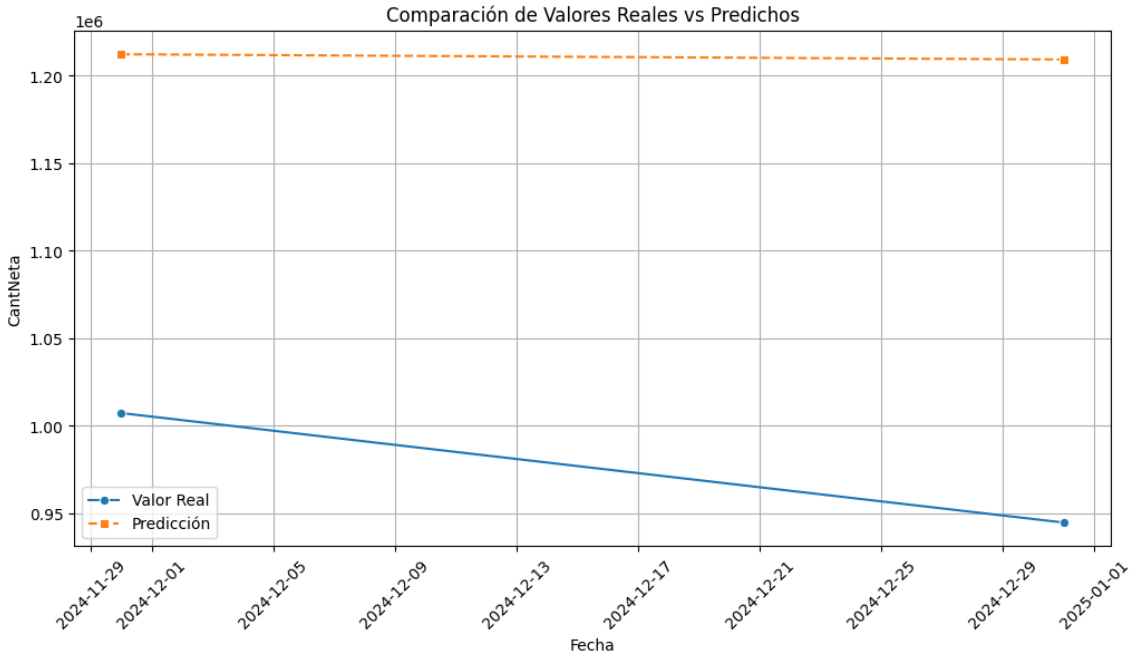

Figura 12:*GradientBoosting - Experimento 1*

Se observa que el modelo no captura la tendencia de los valores reales, ya que muestra una tendencia opuesta. Además, los valores predichos son consistentemente más altos que los valores reales, lo que indica un sesgo en el modelo.

4.4.2 Experimento 2

Dataset	MSE	RMSE	R^2	MAPE (%)
Entrenamiento	2.733496e+09	52282.844	0.960	
Prueba	5.661258e+09	75241.33	0.885	22.85

Tabla 11:*GradientBoosting - Experimento 2*

El modelo tiene un buen rendimiento en el conjunto de entrenamiento, pero presenta ciertos problemas cuando se evalúa con los datos de prueba. Para el conjunto de entrenamiento, el MSE (Error Cuadrático Medio) es de 2.73e+09 y el RMSE (Raíz del Error Cuadrático Medio) es de 52,282.84, lo que indica un modelo relativamente preciso. Además, el valor de R^2 es de 0.960, lo que sugiere que el modelo explica el 96% de la variabilidad de los datos en el conjunto de entrenamiento.

Sin embargo, en el conjunto de prueba, el MSE aumenta a 5.66e+09 y el RMSE se incrementa a 75,241.33, lo que refleja un mayor error. Además, el valor de R^2

disminuye a 0.885, lo que indica que el modelo no generaliza tan bien como en el conjunto de entrenamiento. El MAPE del 22.85% muestra una mayor desviación en las predicciones, lo que podría sugerir que el modelo está sufriendo de sobreajuste

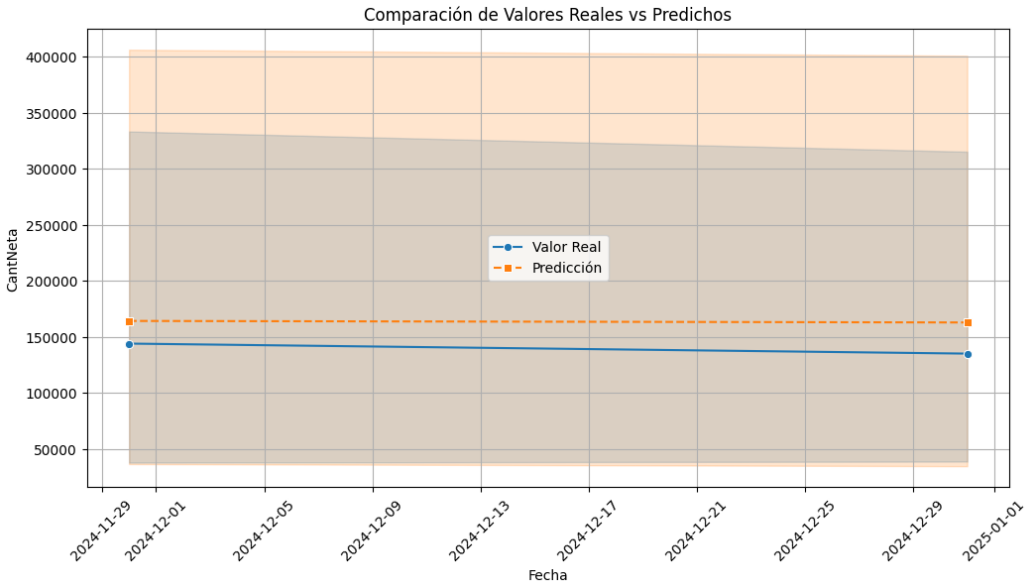

Figura 13:*GradientBoosting - Experimento 2*

La línea de valores predichos presenta una mayor suavidad en las fluctuaciones en comparación con la línea de valores reales, lo que sugiere que el modelo tiende a subestimar la volatilidad de los datos. Esta discrepancia podría deberse a la complejidad inherente de los datos o a la necesidad de ajustar los parámetros del modelo para una mayor precisión.

En general, el gráfico proporciona una visión clara del rendimiento del modelo, destacando tanto sus fortalezas como sus áreas de mejora. La evaluación visual de la precisión del modelo es crucial para tomar decisiones informadas sobre su aplicabilidad y para identificar posibles ajustes que mejoren su rendimiento.

Características importantes del modelo

Este código genera un gráfico de barras horizontales para visualizar las características más importantes

El código realiza lo siguiente:

```
import matplotlib.pyplot as plt

# Obtener las importancias de las características
importancia_features = gbm.feature_importances_

# Crear un DataFrame para asociar las características con sus
importancias
importancia_df = pd.DataFrame({
    'Característica': X_train.columns,
    'Importancia': importancia_features
})

# Ordenar por la importancia de mayor a menor
importancia_df = importancia_df.sort_values(by='Importancia',
ascending=False)

# Mostrar las 10 características más importantes
print(importancia_df.head(20))

# Graficar las importancias de las características
plt.figure(figsize=(10, 6))
plt.barh(importancia_df['Característica'],
importancia_df['Importancia'], color='skyblue')
plt.xlabel('Importancia')
plt.ylabel('Características')
plt.title('Importancia de las Características en el Modelo
GradientBoostingRegressor')
plt.gca().invert_yaxis()  # Para que la característica más
importante esté arriba
plt.show()
```

Pasos y explicación:

Obtener las importancias de las características:

El modelo GBM (gbm.feature_importances_) calcula la importancia de cada característica, indicando cuánto contribuye cada una al rendimiento del modelo.

Crear un DataFrame con las características y sus importancias:

Se crea un DataFrame (importancia_df) donde se asocian las características (columnas de X_train) con sus respectivas importancias.

Ordenar las importancias de mayor a menor:

Se ordena el DataFrame por la columna Importancia para visualizar las características más relevantes primero.

Mostrar las 20 características más importantes:

Imprime las primeras 20 características con mayor importancia.

Generar el gráfico:

Se crea un gráfico de barras horizontales usando matplotlib para mostrar las importancias. Se invierte el eje Y para que la característica más importante aparezca en la parte superior.

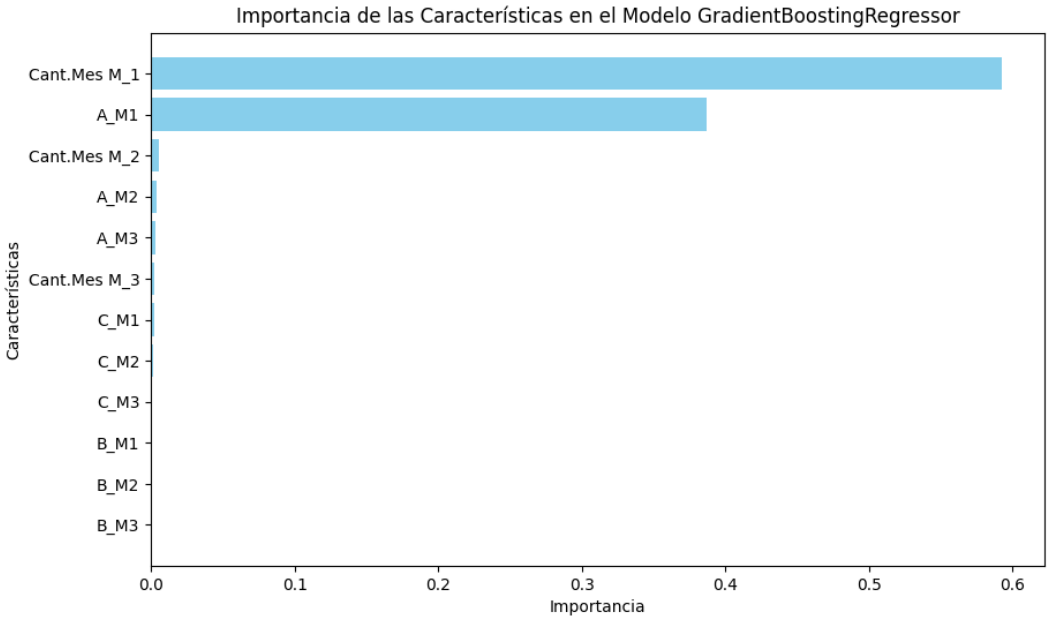

Figura 14: *Características importantes*

El grafico de la Figura 14 presenta la importancia que brinda el algoritmo a las características más relevantes donde destaca Cant.Mes M_1 seguida de A_M1 ambas con una importancia significativa. Esto apunta a que estas dos variables son los importantes promotores de las predicciones del modelo.

Las demás características tienen una importancia muy baja en relación a las anteriormente descritas lo que se concluye que el impacto en las predicciones del modelo es limitado. Toda esta información descrita es crucial para la selección de características, permitiendo a los modelados seleccionar las variables más

significativas y potencialmente descartar aquellas que favorecen poco al rendimiento del modelo.

Esta grafica también ayuda a proponer la creación de nuevas variables basadas en las más importante con el fin de mejorar la precisión predictiva del modelo.

4.5 Validación aplicando Rolling Horizon Validation

Realizar la validación cruzada en modelos de *Machine Learning* es importante ya que evalúa la calidad y permite identifica limitaciones con el fin de garantizar generalización a datos no vistos. Aplicar este proceso en la construcción de modelos permite la construcción de modelos confiables adaptables y robustos. Utilizar validación cruzada como Rolling Horizon Validation es una técnica fundamental donde permite evaluar modelos de *Machine Learning* principalmente en series de temporales (Lemaire & Penz, 2021). La particularidad de realizar este tipo de validación donde asume autonomía entre las observaciones, esta herramienta respeta la distribución temporal de los datos mitigando fugas de información del futuro al pasado con su particularidad donde los datos se dividen en conjunto de entrenamiento y pruebas de manera secuencia, donde cada iteración expande el conjunto de entrenamiento incorporando datos recientes mientras mantiene la coherencia temporal.

La principal ventaja de este método que sirve para resolución de problemas financieros, meteorológicos y demanda de productos, donde las tendencias y patrones tienen a cambiar con el tiempo, otra particularidad de este método es permite analizar la estabilidad del modelo a lo largo del tiempo, identificando períodos donde su desempeño puede degradarse.

4.5.1 Random Forest

4.5.1.1 Experimineto 1

Dataset	MSE	RMSE	R^2	MAPE (%)
Entrenamiento (Promedio)	5.857070e+09	7.530566e+04	8.106734e-01	4.084733e+00
Prueba (Promedio)	6.548736e+10	2.400400e+05	-2.516240e+00	1.610632e+01

Tabla 12 *- Validación Experimento 1*

Los resultados del modelo indican un alto grado de sobreajuste, ya que el error cuadrático medio (MSE) y la raíz del error cuadrático medio (RMSE) en prueba son significativamente mayores que en entrenamiento, mientras que el coeficiente de determinación (R^2) en prueba es negativo, lo que sugiere que el modelo no generaliza bien y es menos efectivo que una simple media. Además, el error absoluto medio porcentual (MAPE) en prueba es mucho mayor, lo que confirma que las predicciones son imprecisas. Esto puede deberse a una distribución de datos

distinta entre entrenamiento y prueba, un modelo demasiado complejo que memoriza los datos sin aprender patrones generales o una falta de regularización.

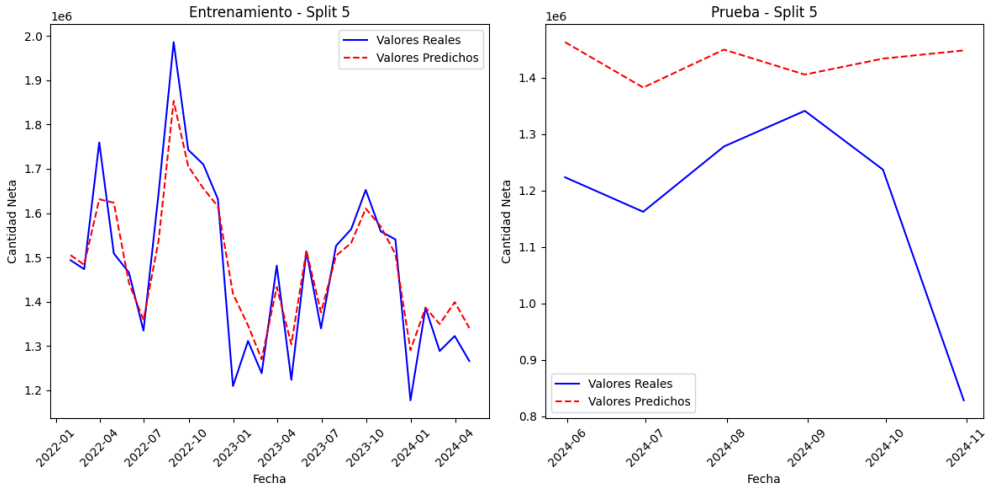

***Figura 15** - Validación Random Forest*

En la figura 15 los dos gráficos de series temporales que comparan valores reales y predichos en entrenamiento y prueba (Split 5). En el gráfico de entrenamiento (izquierda), la línea azul (valores reales) y la roja punteada (valores predichos) siguen patrones similares, indicando un buen ajuste. En el gráfico de prueba (derecha), hay una discrepancia mayor: los valores reales disminuyen, pero los predichos se mantienen estables. Esto sugiere que el modelo se ajustó bien a los datos de entrenamiento, pero no generaliza correctamente en prueba, posiblemente debido a sobreajuste o falta de captura de tendencias cambiantes en los datos nuevos

4.5.1.2 Experimento 2

Dataset	**MSE**	**RMSE**	**R^2**	**MAPE (%)**
Entrenamiento (Promedio)	7.350779e+09	8.236184e+04	8.995379e-01	3.717621e+12
Prueba (Promedio)	2.990824e+09	5.202625e+04	9.567073e-01	6.770672e+13

***Tabla 13** - Validación Experimento 2*

La validación Rolling Horizon aplicada al modelo de predicción muestra resultados prometedores, con un MSE promedio de 7.35×10^9 en entrenamiento y 2.99×10^9 en prueba, lo que indica que el modelo logra minimizar los errores. Además, el RMSE es menor en prueba (52,026.25) que en entrenamiento (82,361.84), sugiriendo una buena capacidad de generalización. El coeficiente de determinación (R^2) es alto en ambos conjuntos, con 0.8995 en entrenamiento y 0.9567 en prueba, lo que implica que el modelo explica bien la varianza de los datos. Sin embargo, el MAPE es extremadamente alto (3.71×10^{12} % en entrenamiento y 6.77×10^{13} % en

prueba), lo que podría deberse a valores cercanos a cero en el denominador de la fórmula, afectando la interpretación de la métrica.

Para mejorar el desempeño, se recomienda revisar la escala de los datos, aplicar transformaciones como logaritmos o evaluar métricas alternativas como SMAPE o MAE. También es importante analizar posibles outliers que afecten el error en la prueba. En general, el modelo parece ser sólido en términos de varianza explicada, pero requiere ajustes para mejorar la estabilidad de los errores porcentuales y lograr predicciones más precisas en datos futuros.

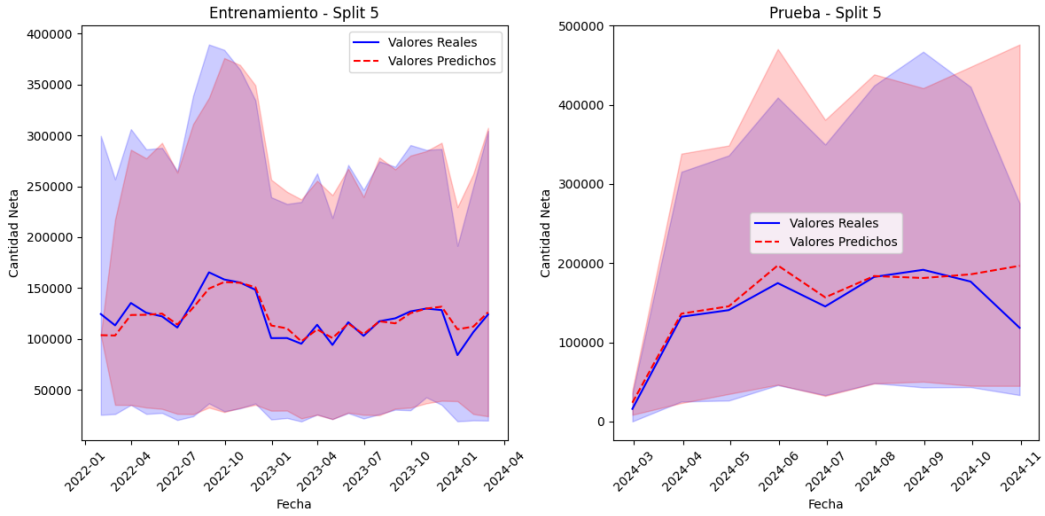

***Figura 16** -Validación Random Forest*

El gráfico de la izquierda muestra el rendimiento del modelo en el conjunto de datos de entrenamiento, abarcando desde enero de 2022 hasta abril de 2024. Se observa que el modelo se ajusta bastante bien a los datos de entrenamiento, ya que las líneas que representan los valores reales y los valores predichos están muy cerca entre sí.

El gráfico de la derecha muestra el rendimiento del modelo en el conjunto de datos de prueba, abarcando desde marzo de 2024 hasta noviembre de 2024. En este caso, se observa que el modelo no se ajusta tan bien a los datos de prueba como lo hizo con los datos de entrenamiento, ya que hay una mayor discrepancia entre los valores reales y los valores predichos, especialmente hacia el final del período de prueba.

4.5.2 XGBRegressor

4.5.2.1 Experimento 1

Dataset	MSE	RMSE	R^2	MAPE (%)
Entrenamiento (Promedio)	571687.637	748.6847	0.999	0.023
Prueba (Promedio)	6.315906e+10	2.369046e+05	-2.631105e+00	1.429440e+01

***Tabla 14** - Validación Experimento 1 – XGBRegressor*

El modelo muestra un sobreajuste extremo. En entrenamiento, tiene un MSE bajo (571,687.637) y un R^2 casi perfecto (0.999), pero en prueba, el MSE es muy alto (6.32×10^{10}) y el R^2 es negativo (-2.63), indicando que no generaliza bien. Además, el MAPE en prueba (14.29%) es mucho mayor que en entrenamiento (0.023%), lo que sugiere errores significativos en predicciones reales.

Este comportamiento puede deberse a una complejidad excesiva del modelo, diferencias en la distribución de los datos de entrenamiento y prueba, o una posible fuga de información.

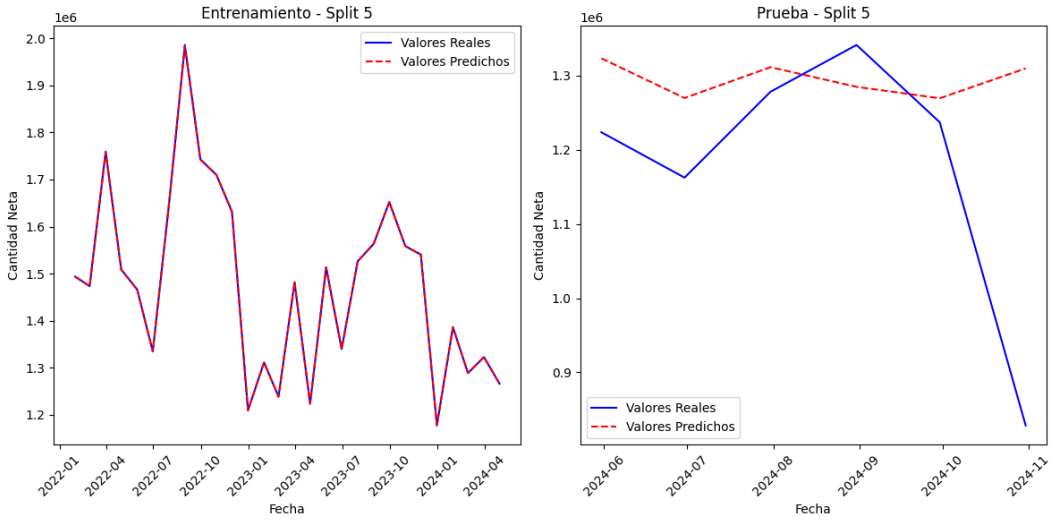

***Figura 17** - Validación XGBRegressor*

El gráfico confirma el sobreajuste detectado en las métricas. En el conjunto de entrenamiento (izquierda), las predicciones (línea roja) siguen casi exactamente los valores reales (línea azul), lo que indica que el modelo se ajusta demasiado a los datos de entrenamiento.

En el conjunto de prueba (derecha), la discrepancia es evidente: los valores reales fluctúan, mientras que las predicciones son más planas y no capturan la variabilidad. Esto sugiere que el modelo ha aprendido patrones específicos del entrenamiento en lugar de generalizar correctamente.

4.5.2.2 Experimento 2

Dataset	MSE	RMSE	R^2	MAPE (%)
Entrenamiento (Promedio)	6.810637e+09	7.902408e+04	9.069964e-01	1.668361e+12
Prueba (Promedio)	1.375409e+10	1.115490e+05	7.765033e-01	4.433811e+13

***Tabla 15** - Validación Experimento 2 – XGBRegressor*

En el Experimento 2, el modelo muestra una mejor generalización, con un R^2 de 0.91 en entrenamiento y 0.78 en prueba, lo que indica que ahora captura mejor la variabilidad de los datos de prueba. Sin embargo, los errores absolutos siguen siendo alarmantes.

El MAPE es extremadamente alto (1.67×10^{12}% en entrenamiento y 4.43×10^{13}% en prueba), lo que sugiere un problema en la escala de los datos o en el cálculo del error. Además, el MSE y RMSE siguen siendo altos, lo que indica predicciones con grandes desviaciones.

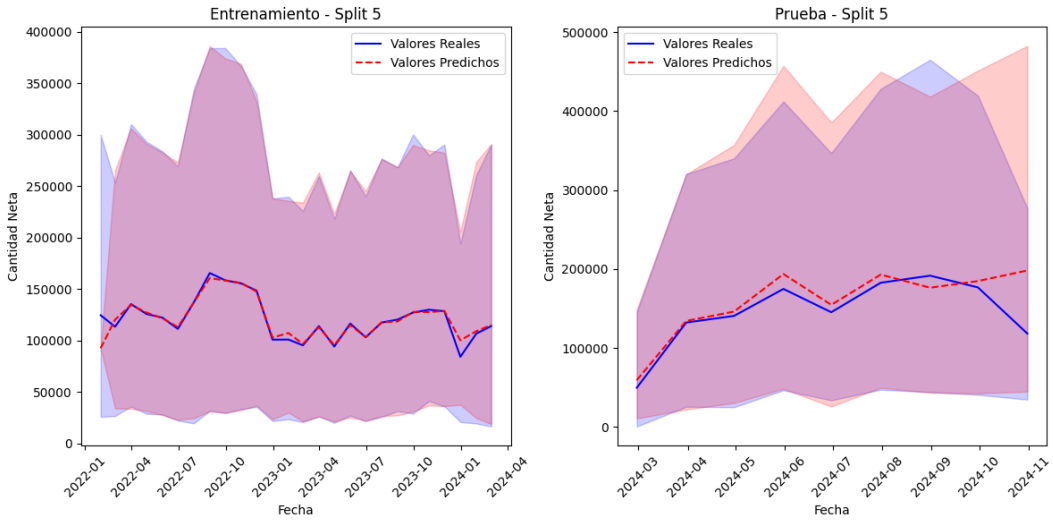

***Figura 18** - Validación Experimento 2 – XGBRegressor*

Las gráficas en la Figura 18 presentadas muestran el rendimiento de un modelo de predicción en dos conjuntos de datos: entrenamiento y prueba, específicamente para el "Split 5". En el conjunto de entrenamiento, el modelo muestra un ajuste notablemente bueno, con las predicciones (línea roja discontinua) siguiendo de cerca los valores reales (línea azul sólida). Esto sugiere que el modelo ha aprendido bien los patrones presentes en los datos de entrenamiento.

Sin embargo, en el conjunto de prueba, la precisión del modelo disminuye. Aunque la tendencia general se mantiene, se observan desviaciones significativas entre las predicciones y los valores reales, especialmente en los picos y valles. La mayor amplitud del área sombreada en la gráfica de prueba indica una mayor incertidumbre en las predicciones.

Este comportamiento sugiere un posible sobreajuste del modelo, lo que significa que se ha adaptado demasiado a los datos de entrenamiento y no generaliza bien a

nuevos datos. Para mejorar el rendimiento, se recomienda revisar el modelo, ajustar sus hiperparámetros, considerar otros modelos de regresión y realizar un análisis de errores detallado.

4.5.3 GradientBoosting

4.5.3.1 Experimento 1

Dataset	MSE	RMSE	R^2	MAPE (%)
Entrenamiento (Promedio)	164048.530	218.839	0.999	0.0123
Prueba (Promedio)	7.050106e+10	2.483397e+05	-2.833436e+00	1.576381e+01

***Tabla 16** - Validación Experimento 1 – GradientBoosting*

Los resultados del Experimento 1 revelan un grave problema de sobreajuste en el modelo de regresión. Las métricas de entrenamiento, como el MSE bajo (164048.530), el RMSE (218.839), el R^2 cercano a 1 (0.999) y el MAPE mínimo (0.0123%), indican un ajuste casi perfecto a los datos de entrenamiento.

Sin embargo, el rendimiento en los datos de prueba es desastroso. El MSE se dispara a 7.050106e+10, el RMSE a 2.483397e+05, el R^2 se vuelve negativo (-2.833436e+00) y el MAPE asciende a 15.76381%. Esto significa que el modelo no logra generalizar a nuevos datos, prediciendo peor que el promedio.

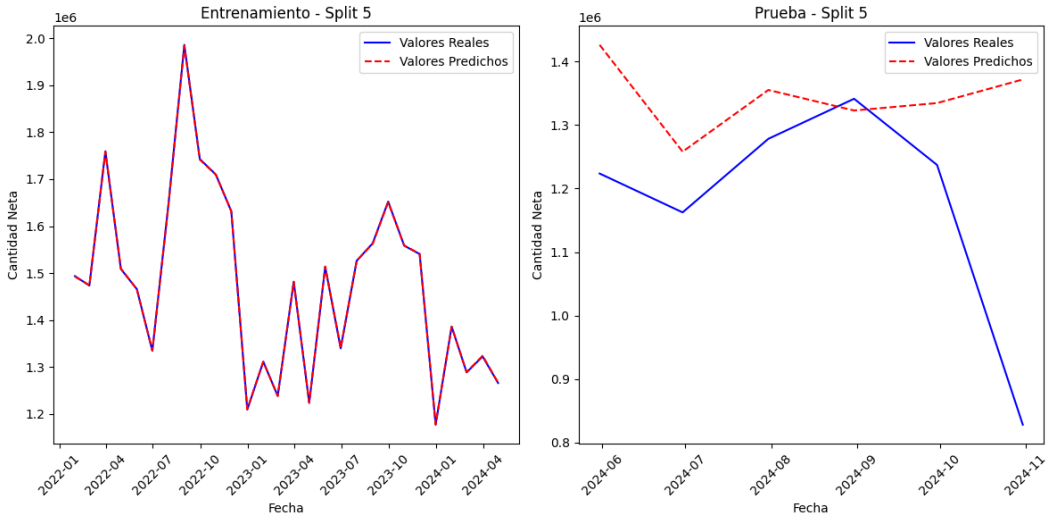

***Figura 19** - Validación Experimento 1 – GradientBoosting*

El análisis de las gráficas de la Figura 19 revela un claro problema de sobreajuste en el modelo de predicción. La gráfica de "Entrenamiento - Split 5" muestra un ajuste casi perfecto, donde los valores predichos siguen muy de cerca a los valores reales. Esto indica que el modelo ha aprendido los datos de entrenamiento con gran precisión, capturando incluso las fluctuaciones más sutiles.

Sin embargo, la gráfica de "Prueba - Split 5" cuenta una historia diferente. Aquí, el modelo muestra un rendimiento significativamente peor. Los valores predichos se desvían notablemente de los valores reales, lo que indica que el modelo no logra generalizar bien a nuevos datos. Esta discrepancia entre el rendimiento en el entrenamiento y la prueba es un signo clásico de sobreajuste.

El modelo parece haber memorizado los detalles específicos del conjunto de entrenamiento, incluido el ruido, lo que dificulta su capacidad para hacer predicciones precisas en datos no vistos. Además, se observa un comportamiento anómalo hacia el final de la gráfica de prueba, donde el modelo no logra seguir una caída abrupta en los valores reales.

4.5.3.2 Experimento 2

Dataset	MSE	RMSE	R^2	MAPE (%)
Entrenamiento (Promedio)	6.877340e+09	7.955271e+04	9.113492e-01	6.341029e+12
Prueba (Promedio)	3.310818e+09	5.538057e+04	9.489484e-01	3.792932e+13

***Tabla 17** - Validación Experimento 2 – GradientBoosting*

En el conjunto de entrenamiento, el modelo exhibe un MSE de 6.877340×10^{9}, un RMSE de 7.955271×10^{4}, un R^2 de 0.9113492, y un MAPE de 6.341029×10^{12}. En el conjunto de prueba, el modelo muestra un MSE de 3.310818×10^{9}, un RMSE de 5.538057×10^{4}, un R^2 de 0.9489484, y un MAPE de 3.792932×10^{13}.

El modelo muestra un buen ajuste en ambos conjuntos, evidenciado por los altos valores de R^2. Sin embargo, los valores extremadamente altos de MAPE sugieren posibles problemas con la escala de los datos o la presencia de valores atípicos que afectan significativamente el error porcentual.

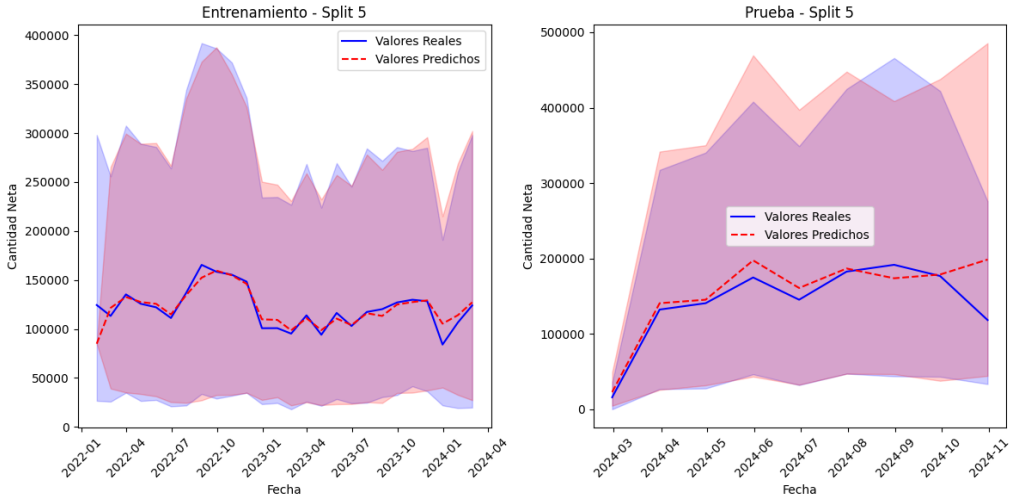

***Figura 20** - Validación Experimento 2 – GradientBoosting*

El gráfico de la izquierda muestra el conjunto de entrenamiento, abarcando desde enero de 2022 hasta abril de 2024. Se observa que el modelo (línea roja discontinua) sigue la tendencia general de los valores reales (línea azul), pero con cierta suavidad en las fluctuaciones. La banda sombreada alrededor de las líneas indica la variabilidad o incertidumbre, siendo relativamente estrecha.

El gráfico de la derecha muestra el conjunto de prueba, desde marzo hasta noviembre de 2024. Aquí, el modelo también sigue la tendencia general, pero con mayor dificultad para capturar los picos y valles. La banda sombreada es más ancha, indicando mayor incertidumbre en las predicciones.

Concluyendo que el modelo muestra un buen ajuste en el conjunto de entrenamiento, pero su rendimiento disminuye en el conjunto de prueba, sugiriendo posibles problemas de generalización o sobreajuste. Se recomienda analizar métricas de error y realizar ajustes en el modelo para mejorar su capacidad predictiva en datos no vistos.

Basándonos en los resultados de los experimentos 2 por su cantidad mayor de datos en base al experimento 1 donde *Gradient Boosting* tiende a ser el mejor modelo. Obtuvo el menor MSE y RMSE en el conjunto de prueba, lo que revela una mayor exactitud en la predicción. Además, su R^2 en el conjunto de prueba es el más alto, sugiere que explica una mayor proporción de la variabilidad en los datos no vistos. Aunque todos los modelos tienen valores de MAPE considerablemente altos, se propone realizar una investigación adicional, *Gradient Boosting* muestra un rendimiento superior en las métricas clave de MSE, RMSE y R^2. *Random Forest* le sigue muy de cerca en cuanto a rendimiento. *XGBRegressor*, aunque funciona bien en el conjunto de entrenamiento, muestra un rendimiento significativamente peor en el conjunto de prueba, lo que indica un posible sobreajuste. Por lo tanto, *Gradient Boosting*, con su capacidad para generalizar bien a los datos no vistos, se destaca como el modelo más sólido entre los tres.

4.6 ARIMA

El objetivo principal de este tipo de modelo es buscar entender cómo los valores de una serie temporal se influyen entre sí a lo largo del tiempo, representando la serie mediante combinaciones de sus propios valores pasados o de variables aleatorias (González , 2020).

El modelo ARIMA(1,1,1) brinda un punto de partida para el análisis para esto es necesario realizar ajustes adicionales para mejorar la predicción en series con tendencias complejas o estacionales.

Para pronosticar la variable objetivo CantNeta en una serie temporal se tomaron 399 observaciones donde permitió tener los siguientes resultados, un AIC de 11084.639 y un BIC de 11096.598, valores que ayudan a evaluar el ajuste y comparar con otros modelos. El coeficiente AR(1) fue de -0.1455, lo que apunta a una débil correlación negativa entre las observaciones, pero con un valor p de 0.535, revelando que no es estadísticamente significativo. En cambio, el coeficiente MA(1)

de -0.9972 tiene un valor p cercano a 0, lo que indica que es altamente significativo y representa una fuerte dependencia de los residuos pasados.

El análisis de residuos mostró que el modelo tiene ciertos problemas mientras que la prueba de Jarque-Bera reveló que los residuos no siguen una distribución normal (p-valor = 0.00), esto brinda información donde indica que el modelo no está capturando todas las estructuras subyacentes. Además, la prueba de Ljung-Box no halló autocorrelación significativa en los residuos, lo que indica que el modelo ha hecho un buen trabajo de ajuste.

El pronóstico para los próximos periodos (desde el 399 al 410) muestran una estabilidad en torno al valor de 127,200. Esto podría indicar que el modelo no está capturando la variabilidad futura de manera efectiva, lo que sugiere que podría ser necesario ajustar el modelo. Es recomendable explorar modelos estacionales como *SARIMA* si hay patrones estacionales, o bien utilizar enfoques más complejos para mejorar el ajuste del modelo.

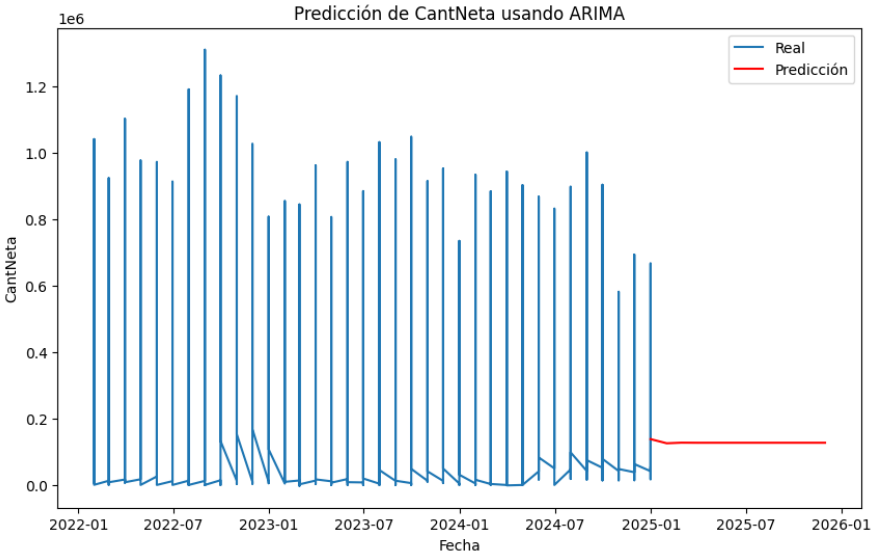

Figura 21 *- Modelo Arima*

CAPÍTULO VI: CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

- La investigación realizada ha logrado avances importantes en el análisis de datos, lo que permite a Gloria Ecuador comprender mejor tanto la situación pasada como el comportamiento actual de las ventas. Este entendimiento profundo de la información facilita la toma de decisiones informadas, especialmente al considerar futuras inversiones en tecnología. A través del análisis de los datos, la empresa puede identificar patrones y tendencias clave que ayudarán a optimizar su estrategia comercial y mejorar su rendimiento a largo plazo.
- Aplicar *Machine Learning* en la empresa no solo es una estrategia para mantenerse a la vanguardia tecnológica, sino que también permite solucionar problemas de manera inmediata. Esto impulsa el crecimiento, optimiza procesos y fortalece la competitividad en el mercado.
- Los resultados obtenidos indican que el modelo *Gradient Boosting* presenta la mejor precisión en ambos experimentos realizados, con un coeficiente de determinación (R^2) de 0.885. En comparación con otros algoritmos, este modelo demuestra un mejor ajuste a los datos del conjunto de Gloria Ecuador, lo que sugiere que es el más adecuado para predecir las ventas ya que el *dataset* cuenta con nuevos *features* creados a partir de un análisis ABC el cual permitió tener una mayor comprensión de los datos.
- La presentación de resultados para Gloria Ecuador proporciona una visión clara sobre la evolución de su marca propia, lo cual permite a todas las áreas de la empresa tener una percepción unificada sobre su desempeño. Esta información genera una cultura organizacional orientada a la mejora continua, favoreciendo la colaboración entre los diferentes departamentos para fomentar un crecimiento conjunto. Al comprender cómo ha evolucionado la marca y las áreas clave de oportunidad, todos los equipos pueden trabajar de manera alineada hacia objetivos comunes, contribuyendo al fortalecimiento y expansión de la marca.

5.2 Recomendaciones

- Investigar los beneficios de los algoritmos de ensamblaje en la base de datos de ventas, con el fin de que puedan ser implementados en otras áreas de la empresa. Esto permitirá mejorar la precisión, crear robustez frente a ruidos y reducir el sobreajuste, lo que a su vez permitirá que los diferentes algoritmos de ensamblaje se adapten mejor a los cambios.
- Gloria Ecuador piense en implementar herramientas avanzadas de análisis de datos y tecnologías predictivas para seguir aprovechando las tendencias identificadas. Además, invertir en software de inteligencia de negocios podría mejorar aún más la toma de decisiones, permitiendo a la empresa ajustar su estrategia comercial de manera más ágil y eficiente en el futuro.
- Realizar un seguimiento continuo de su rendimiento y considerar la actualización del modelo conforme se disponga de nuevos datos, asegurando que las predicciones se mantengan lo más precisas y actualizadas posible.
- Establecer un sistema de seguimiento continuo y dinámico para monitorear el desempeño de la marca propia de Gloria Ecuador. Este sistema debe ser capaz de proporcionar información actualizada y relevante sobre los indicadores clave de rendimiento (KPIs) de la marca, permitiendo una evaluación constante y la toma de decisiones ágiles.

Referencias Bibliográficas

Ablan, M. (25 de 08 de 2024). *Ixpantia: Ciencia de datos* . Obtenido de Ixpantia: Ciencia de datos : https://www.ixpantia.com/es/blog/ciencia-de-datos-para-gestionar-la-canasta-de-productos-de-consumo-masivo

Aliyev, E. (15 de 01 de 2023). *Linkedin*. Obtenido de Linkedin: https://www.linkedin.com/advice/3/how-can-you-measure-data-quality-ml-skills-machine-learning-bsjbc?lang=es&lang=es&originalSubdomain=es

Athanasopoulos, G., & Hyndman , R. (2021). *Forecasting: Principles and Practice.* 2021.

Caballero, J. (12 de 11 de 2022). *DataScientest*. Obtenido de DataScientest: https://datascientest.com/es/pandas-python

Cataldo, A. (25 de 02 de 2025). *Transformación Digital*. Obtenido de Transformación Digital: https://www.sydle.com/es/blog/big-data-en-retail-6728c305e1cb634ba7741ca3

Churchil, W. (12 de 10 de 2005). *Estadística para todos*. Obtenido de Estadística para todos: https://www.estadisticaparatodos.es/taller/graficas/cajas.html

Corso, C., Maldonado, C., & Luque, C. (2018). *Diseño de Método de Ensamble Homogéneo para Clasificadores Débiles .* Argentina: Universidad Tecnológica Nacional .

Cortez, S. (02 de 08 de 2022). *Medium:Introducción a los Métodos de Ensamble* . Obtenido de Medium: Introducción a los Métodos de Ensamble : https://medium.com/@oscars.cortezmo/introducci%C3%B3n-a-los-m%C3%A9todos-de-ensamble-y-al-algoritmo-de-xgboost-caso-pr%C3%A1ctico-e8cb0d58394b

Danilova, E. (2 de 04 de 2024). *Foodeo:Eventos y Promociones*. Obtenido de Foodeo:Eventos y Promociones: https://foodeo.es/blog/eventos-para-impulsar-ventas-restaurante/

Diaz Madero, C. (01 de 12 de 2024). *NetLogistk*. Obtenido de NetLogistk: https://www.netlogistik.com/es/blog/como-llevar-a-cabo-un-pronostico-de-la-demanda

Espinosa, Z. (2020). *Aplicación de algoritmos Random Forest y XGBoost en una base de solicitudes de tarjetas de crédito.* Distrito Federal - Mexico.

Fernández, A. (14 de 07 de 2023). *Blog: Guía completa sobre Random Forest*. Obtenido de Blog: Guía completa sobre Random Forest: https://anderfernandez.com/blog/guia-completa-random-forest/

Gonzales, A. (01 de 10 de 2024). *Strategic Platform*. Obtenido de Strategic Platform: https://strategicplatform.com/articulos/como-puede-machine-learning-ayudar-a-la-prevision-de-la-demanda

González , M. (09 de 04 de 2020). Sarrik - Análisis de series temporales. Euskadi, España.

Gorini, M. (12 de 08 de 2010). *Bi Smart*. Obtenido de Bi Smart: https://blog.bismart.com/que-diferencia-etl-y-ssis

Gutiérrez, J. (07 de 07 de 2023). *Data Science - Entendiendo el Gradient Boosting*. Obtenido de Data Science - Entendiendo el Gradient Boosting: https://www.linkedin.com/pulse/entendiendo-el-gradient-boosting-probabil%C3%ADstico-un-pompas-guti%C3%A9rrez/

Kelleher, J., & Mac Namee, B. (2014). Fundamentals of Machine Learning for Predictive Data Analytics: Algorithms. En *Fundamentals of Machine Learning for Predictive Data Analytics: Algorithms.* London .

Laskova, D. (20 de 09 de 2022). *FuturamoBlog*. Obtenido de FuturamoBlog: https://futuramo.com/blog/how-machine-learning-is-redefining-demand-forecasting-6-key-methods/

Lemaire, P., & Penz, B. (19 de 11 de 2021). New rolling horizon optimization approaches to balance short-term and long-term decisions: An application to energy planning. Obtenido de New rolling horizon optimization approaches to balance short-term and long-term decisions: An application to energy planning.

Mera, J. (12 de 11 de 2022). *Inesem: Businnes School*. Obtenido de Inesem: Businnes School: https://www.inesem.es/revistadigital/informatica-y-tics/modelos-de-prediccion/

Muñoz, C. (17 de 02 de 2021). *HIBERUS BLOG*. Obtenido de HIBERUS BLOG: https://www.hiberus.com/crecemos-contigo/big-data-impulsa-ventas-en-empresa/

Peiro Ucha, A. (26 de 01 de 2024). *Economipedia: Demanda* . Obtenido de Economipedia: Demanda : https://economipedia.com/definiciones/demanda.html

Phipps, S. (1 de 10 de 2020). *MBS:Análisis ABC*. Obtenido de MBS:Análisis ABC: https://www.munich-business-school.de/es/l/diccionario-de-estudios-empresariales/analisis-abc

Rodriguez, Y. (31 de 10 de 2018). *Diego Calvo*. Obtenido de Diego Calvo: https://www.diegocalvo.es/xgboost/

Rosati, G. (21 de 05 de 2019). *Entendiendo Gradient Boosting Machines*. Obtenido de Entendiendo Gradient Boosting Machines: https://gefero.github.io/flacso_ml/clase_3/notebook/boosting_intuicion_notebook.nb.html

Rosati, G. (02 de 10 de 2020). *¿Qué es Machine Learning?* Obtenido de ¿Qué es Machine Learning?: https://gefero.github.io/flacso_ml/clase_1/slides/Clase1b.pdf

Sanz, F. (01 de 12 de 2021). *The Machine Learners: Descubriendo la IA*. Obtenido de The Machine Learners: Descubriendo la IA: https://www.themachinelearners.com/xgboost-python/

Sap, S. (10 de 05 de 2020). *Sap: Machine Learning*. Obtenido de Sap: Machine Learning: https://www.sap.com/latinamerica/products/artificial-intelligence/what-is-machine-learning.html

Scott, P. (2 de 04 de 2024). *MRPeasy*. Obtenido de MRPeasy: https://www.mrpeasy.com/blog/es/analisis-abc/

Småros, J., & Kaleva, H. (15 de 10 de 2023). *Relax*. Obtenido de a Guía Completa sobre Machine Learning en la Previsión de la Demanda en Retail: https://www.relexsolutions.com/es/publicaciones/la-guia-completa-sobre-machine-learning-en-la-prevision-de-la-demanda-en-retail/

Steven, J. (02 de 06 de 2021). *Huawei: Algoritmos de aprendizaje automático: Ensemble Learning*. Obtenido de Huawei: Algoritmos de aprendizaje automático: Ensemble Learning: https://forum.huawei.com/enterprise/intl/es/thread/blog/667225164137512960?blogId=667225164137512960

Vera, A. (20 de 05 de 2020). *Plazi: Métodos de regularización: overfitting y underfitting.* Obtenido de Plazi: Métodos de regularización: overfitting y underfitting: https://platzi.com/clases/2565-redes-neuronales-tensorflow/42848-metodos-de-regularizacion-overfitting-y-underfitti/