UNIVERSIDAD ESAN

FACULTAD DE INGENIERÍA

INGENIERÍA INDUSTRIAL Y COMERCIAL
INGENIERÍA EN GESTIÓN AMBIENTAL
INGENIERÍA DE TECNOLOGÍAS DE INFORMACIÓN Y SISTEMAS

Diseño de un modelo de predicción de demanda online de paquete de huevos (15 unidades) para una empresa proveedora de productos avícolas en Lima mediante Machine Learning

Trabajo de Suficiencia Profesional presentado en satisfacción parcial de los requerimientos para:

Obtener el título profesional de Ingeniero Industrial y Comercial
Obtener el título profesional de Ingeniero en Gestión Ambiental
Obtener el título profesional de Ingeniero de Tecnologías de Información y Sistemas

AUTORES

Cabrera Reyes, Jairo
Camero Veneros, Mario
Castillon Medina, Densel Giomar
Garcia Condori, Guadalupe
Garcia Guzman, Rony Yeltsin

ASESOR

Calderón Niquin, Marks Arturo

ORCID N° 0000-0002-5440-3978

Noviembre, 2023

VERSION DEFINITIVA TSP_GRUPO 02

ORIGINALITY REPORT

14%	12%	2%	7%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Universidad ESAN -- Escuela de Administración de Negocios para Graduados Student Paper	3%
2	hdl.handle.net Internet Source	1%
3	www.coursehero.com Internet Source	<1%
4	Submitted to Universidad Internacional de la Rioja Student Paper	<1%
5	www.cienciadedatos.net Internet Source	<1%
6	repositorio.esan.edu.pe Internet Source	<1%
7	dokumen.pub Internet Source	<1%
8	qdoc.tips Internet Source	<1%
9	idoc.pub	

RESUMEN

Este estudio se enfoca en abordar los desafíos que enfrenta una empresa avícola en Lima (Perú), específicamente en su canal de ventas en línea, destacando la falta de herramientas de inteligencia artificial para prever la demanda de su producto estrella: paquetes de huevos de 15 unidades. La investigación adopta un enfoque experimental con base cuantitativa, entrenando 12 modelos que abarcan desde estadísticos tradicionales hasta avanzados de Machine Learning. La metodología se divide en cuatro pasos clave: extracción de datos, preprocesamiento, modelado y análisis de resultados. El Random Forest, con optimización de hiperparámetros y validación cruzada, se revela como el más eficaz, logrando un RMSE de 38.62 y un MAE de 28.94 que significan una reducción sustancial del 52.16% en MSE y 26.15% en MAE en comparación con un modelo estadístico base (SARIMAX). Además, se propone una optimización en el equipo de planificación, con reducciones significativas en personal (50%) y costos (62.5%). A pesar de los resultados positivos, se recomienda la exploración de modelos más complejos como redes neuronales artificiales y la consideración de la implementación en la nube de Google (GCP) para mejorar continuamente la eficiencia del modelo y adaptarse a las dinámicas cambiantes del mercado.

Palabras clave: predicción de demanda, paquetes de huevos, Inteligencia Artificial, Machine Learning, Random Forest.

ABSTRACT

This study focuses on addressing the challenges faced by a poultry company in Lima, Peru, particularly in its online sales channel, highlighting the lack of artificial intelligence tools to forecast the demand for its flagship product: 15-unit egg packages. The research adopts an experimental, quantitative approach, training 12 models ranging from traditional statistics to advanced machine learning. The methodology consists of four key steps: data extraction, preprocessing, modeling, and results analysis. The Random Forest, with hyperparameter optimization and cross-validation, emerges as the most effective, achieving an RMSE of 38.62 and a MAE of 28.94, signifying a substantial reduction of 52.16% in MSE and 26.15% in MAE compared to a baseline statistical model (SARIMAX). Additionally, a planning team optimization is proposed, with significant reductions in personnel (50%) and costs (62.5%). Despite the positive results, it is recommended to explore more complex models such as artificial neural networks and consider the implementation in the Google Cloud Platform (GCP) to continuously enhance the model's efficiency and adapt to the changing dynamics of the market.

Keywords: demand prediction, egg packages, Artificial Intelligence, Machine Learning, Random Forest.

ÍNDICE DE CONTENIDOS

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

ÍNDICE DE ECUACIONES

CAPÍTULO I: PLANTEAMIENTO DEL PROBLEMA

1.1 Descripción de la Realidad Problemática

1.1.1. Panorama Internacional

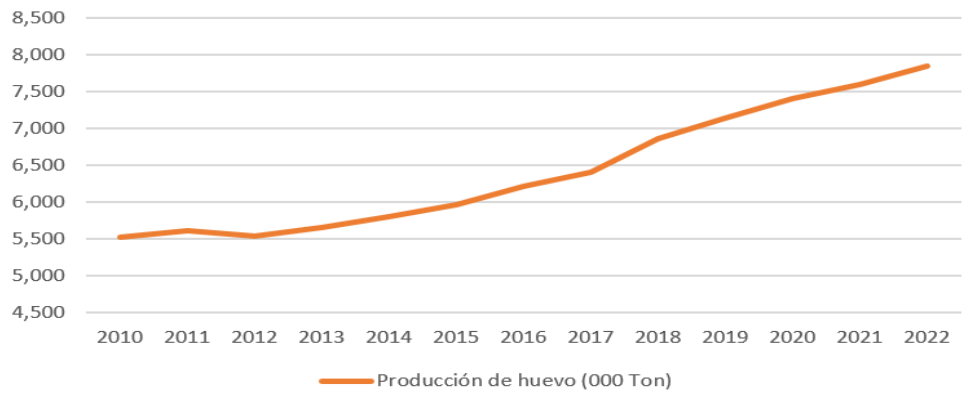
El consumo de huevo en Latinoamérica ha mantenido una tendencia al alza en los últimos años, lo cual se puede ver en la Figura 1, experimentando un crecimiento del 5% en 2022 en comparación con el año anterior y alcanzando un promedio de 230 unidades per cápita (Ruiz, 2023). Este aumento en el consumo refleja la importancia que el huevo tiene en la dieta de la región y su papel como una fuente accesible de proteínas y otros nutrientes esenciales.

México continúa siendo el líder en consumo de huevo en Latinoamérica, menciona Ruiz (2023), con un impresionante promedio de 392 unidades per cápita en 2022, lo que equivale a más de un huevo por día por persona en el país. Argentina (322 unidades) y Colombia (315 unidades) le siguen de cerca y sus niveles de consumo son un testimonio de las exitosas campañas realizadas para promover el consumo de huevo en estos países. Estas campañas han logrado aumentar la conciencia sobre los beneficios nutricionales del huevo y su versatilidad en la cocina. En contraste, en el resto de la región el consumo de huevo se sitúa por debajo de las 300 unidades per cápita, lo que indica que hay oportunidades para fomentar un mayor consumo en otros países latinoamericanos, ya que el huevo sigue siendo una opción económica y saludable para satisfacer las necesidades alimenticias de la población.

El aumento constante en el consumo de huevo en Latinoamérica sugiere que este alimento sigue siendo un pilar en la dieta de la región, con México liderando el camino como el país con el mayor consumo per cápita. Las campañas exitosas en Argentina y Colombia demuestran el potencial para fomentar un mayor consumo en otros países de la región, promoviendo así una alimentación balanceada y nutritiva para sus habitantes.

Figura 1

Evolución de la Producción Anual de huevo en Latinoamérica



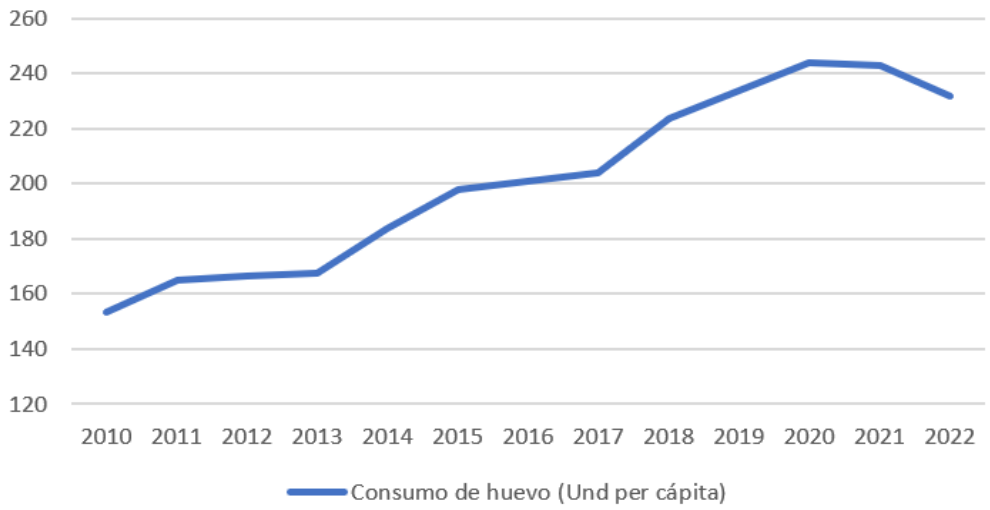
Nota. Elaboración propia a partir de la información extraída de Passport Euromonitor.

1.1.2. Panorama Nacional

En la década pasada, el consumo de huevo en nuestro país experimentó un crecimiento sostenido que tuvo un promedio de crecimiento 5% anual. Cabe señalar que en el contexto de pandemia (2020 y 2021) el consumo de este alimento alcanzó su pico con 244 de consumo anual per cápita, como se muestra en la Figura 2, para luego experimentar una ligera caída en el año 2022. La Asociación Peruana de Avicultura (2018) menciona que además en el periodo pandémico (2021) se alcanzó un importante hito al tener medio millón de toneladas de producción de huevos por primera vez en nuestro país.

Figura 2

Evolución del Consumo Anual per cápita de huevo en el Perú



Nota. Elaboración propia a partir de la información extraída de https://fenavi.org/estadisticas/consumo-per-capita-mundo-pollo/

1.1.3. Panorama de la empresa

La empresa productora y comercializadora de productos avícolas se enfrenta a una problemática crítica en su canal de ventas en línea, el cual se realiza a través de su E-commerce y ofrece envíos a domicilio. El producto estrella de este canal es el paquete de huevos de 15 unidades, el cual tiene la demanda más alta en el canal E-commerce, lo que muestra un gran acierto por parte de la empresa al lanzar este servicio. Sin embargo, la empresa carece de herramientas de inteligencia artificial para predecir con precisión la demanda de este producto, lo que conlleva a varios problemas operativos y comerciales.

En la actualidad, el equipo de planificación, compuesto por cinco personas, se reúne semanalmente los lunes para proyectar las ventas de los productos del canal E-commerce, incluyendo los paquetes de huevos quincena (15 huevos). Para realizar dicha planificación, dicho equipo ocupa el día laboral completo, es decir, ocho horas. Los datos del equipo de planificación se detallan en la Tabla 1.

Tabla 1
Datos del Equipo de Planificación

Cargo	Miembros	Remuneración x miembro	Costo x Día x miembro
Analista Sr. de Planificación	2	S/ 6 000	S/ 200
Analista de Planificación pleno	2	S/ 4 500	S/ 150
Analista Jr. de Planificación	1	S/ 3 000	S/ 100
Total	5	S/ 13 500	S/ 450

Nota. Elaboración propia con información proporcionada por la empresa.

De la Tabla 1 podemos calcular que debido a que el equipo de planificación se demora, por lo general un día en proyectar la demanda de la semana; entonces, se incurre en un gasto semanal de S/ 800. Esto quiere decir que al mes se invierte S/ 3200 solo en esta actividad.

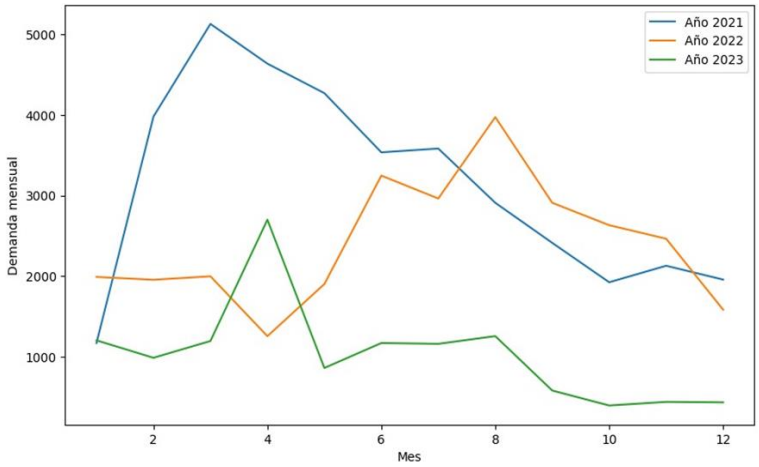
Su método de proyección carece de herramientas de Inteligencia Artificial y se basa en métodos tradicionales reforzados con medidas estadísticas que comparan las solicitudes de los últimos 30 días y de la semana anterior, de modo que no logran identificar tendencias que les ayude a agilizar este proceso. Esta metodología, aunque útil, no es suficientemente precisa para

anticipar las necesidades reales de paquetes de huevos, lo que lleva a dificultades pues actualmente se tiene más de 150 productos que son ofrecidos en el canal E-commerce e históricamente hasta 560 productos diferentes fueron solicitados alguna vez mediante este canal de venta.

Desde el lanzamiento del canal E-commerce, las órdenes de huevo han tenido una demanda bastante voluble mes a mes, como se puede ver en la Figura 3, incluso comparando los mismos periodos de cada año para encontrar posibles estacionalidades. Esta volatilidad durante el lanzamiento del canal se debe a la alta demanda de servicios de delivery en 2021 debido a la pandemia. Asimismo, este año la demanda se ha visto afectada por la gripe aviar y la escasez de soya, como menciona el diario Gestión (2023), que señala que a inicios de este año los costos se habían elevado entre 70% y 80% para la producción de huevo.

Figura 3

Demanda mensual de paquetes de huevos (15 huevos) por año



Nota. Elaboración propia.

Durante el año pasado, la empresa ha experimentado un alto número de órdenes que recibieron cantidades de paquetes de huevos (15 huevos) por debajo de las solicitadas por los clientes. Esto ha resultado en la insatisfacción de los clientes y en posibles pérdidas de ventas. A pesar de que las proyecciones del equipo de planificación han mejorado este año, aún existen muchas oportunidades de venta perdidas debido a la falta de stock.

Adicionalmente, otro desafío importante radica en la entrega de los productos. La empresa se compromete a entregar los productos dentro de un plazo de dos días, pero muchas órdenes son entregadas después de ese período, lo que afecta la satisfacción del cliente.

1.1.4. El producto más solicitado: paquetes de huevos (15 huevos)

El paquete de huevos ha sido el producto más solicitado por los clientes en el E-commerce desde que se lanzó dicho canal en febrero de 2021. En la Tabla 2 se puede ver que este producto ha estado presente en más de 30 mil pedidos realizados en dicho periodo, superando con diferencia considerable a la milanesa casera y el muslo de pollo (los otros dos productos más demandados).

Tabla 2

Top productos más demandados (Feb '21 - Set '23)

Producto	Pedidos (# órdenes)
Paquetes de huevos	32,452
Milanesa casera x 1Kg	24,044
Muslo de pollo fresco 7 und.	22,607
Pechuga especial fresca 2 und.	19,325
Pollo grande con menudencia fresco	19,256
Pollo con menudencia fresco	16,629
Filete de pechuga de pollo corte mariposa (Rango 1.00 a 1.10 Kg)	16,315
Carne molida de Pavita Bol X 500G	14,092
Pierna fresca 7 und.	12,286
Pack Duo Huevo	10,806
Pierna con encuentro fresca 3 und.	10,248

Nota: Elaboración propia.

1.1.5. Diferencia de Cantidades Ordenadas vs. Cantidades solicitadas

La falta de stock en el caso de la empresa representa una problemática crítica que ha afectado la satisfacción de los clientes y ha resultado en la pérdida de ventas a lo largo de los años. Esta carencia de producto disponible se debe a la falta de precisión en la proyección de demanda ha sido un desafío constante y se ha manifestado de diversas maneras a lo largo de los últimos tres años.

En el año 2021, por ejemplo, se registró un alarmante escenario en el cual el 77% de las órdenes recibieron una cantidad de paquetes de huevos (15 huevos) inferior a la cantidad requerida por los clientes. En el año 2022, si bien se evidenció una mejora, el problema persistió y el 28% de las órdenes aún recibieron una cantidad de huevos menor a la requerida, como se puede ver en la Tabla 3. Esta cifra, aunque menor en comparación con el año anterior, sigue siendo significativa y representa una fuente de preocupación para la empresa.

Para el presente año, hasta el mes de agosto, se ha logrado reducir el porcentaje de órdenes afectadas por falta de stock al 5%. Aunque este es un avance positivo, no se puede ignorar que todavía un número significativo de clientes sigue recibiendo menos paquetes de huevos (15 huevos) de los que habían solicitado. Además, se sabe que en el último mes se dejaron de entregar en promedio 5 unidades diarias por falta de stock, lo cual se puede ver en la Tabla 4, lo que tiene un impacto negativo en la satisfacción del cliente.

Tabla 3

Porcentaje de órdenes que recibieron cantidad menor a la solicitada

	2021	**2022**	**2023**
Ene	---	45%	20%
Feb	89%	30%	6%
Mar	94%	34%	4%
Abr	89%	31%	3%
May	74%	35%	4%
Jun	75%	35%	4%
Jul	73%	28%	1%
Ago	75%	28%	3%
Set	61%	29%	---
Oct	58%	19%	---
Nov	55%	19%	---
Dic	48%	17%	---
Total Año	77%	28%	5%

Nota. Elaboración propia.

Tabla 4

Promedio diario - Unidades de paquetes de huevos faltantes por entrega

	2021	**2022**	**2023**
Ene	---	25	8
Feb	206	20	1
Mar	182	18	1
Abr	154	13	2
May	122	4	3
Jun	109	26	1
Jul	94	32	0
Ago	71	40	1
Set	48	34	5
Oct	31	20	---
Nov	35	12	---
Dic	25	11	---
Total Año	92	21	3

Nota. Elaboración propia.

Diferencia de Fecha de Orden y Fecha de entrega

Actualmente, como se indica en la Tabla 5, se registra una diferencia promedio de 2.2 días en el canal, lo que indica que los clientes deben esperar más tiempo del previsto para recibir sus pedidos, pues los términos del servicio de delivery señalan dos días de plazo para realizar la entrega. Además, es alarmante que, como se puede ver en la Tabla 6, que el 32% de las órdenes sean entregadas con una diferencia de 3 o más días con respecto a la fecha de solicitud.

Este retraso en las entregas tiene importantes implicaciones negativas para la experiencia del cliente. En primer lugar, un retraso de 2.2 días, en promedio, puede generar insatisfacción entre los clientes que esperan recibir sus productos en un plazo más corto. Esto puede llevar a que los clientes consideren otras opciones en el mercado o incluso a que cancelen sus pedidos, lo que resulta en pérdidas de ventas y daños a la reputación de la empresa.

Tabla 5

Días promedio de diferencia según mes - Fecha de entrega vs Fecha de solicitud

	2021	2022	2023
Ene	---	2,6	2,3
Feb	2,6	2,5	2,2
Mar	2,8	2,5	2,0
Abr	2,8	3,0	2,3
May	2,3	2,4	2,6
Jun	2,7	2,6	2,1
Jul	2,9	2,7	2,0
Ago	2,8	2,7	2,1
Set	2,8	2,7	2,1
Oct	2,8	2,4	---
Nov	2,9	2,2	---
Dic	2,8	2,5	---
Promedio	2,7	2,6	2,2

Nota. Elaboración propia.

Tabla 6

Share de órdenes según días de diferencia (fecha entrega vs fecha orden)

	Mismo día	1 día	2 días	3 días	4 días	5 días
Ene	1%	10%	46%	44%	0%	0%
Feb	1%	9%	57%	33%	0%	0%
Mar	2%	13%	65%	19%	0%	0%
Abr	1%	8%	63%	22%	3%	3%
May	1%	8%	45%	26%	20%	0%
Jun	1%	12%	58%	29%	0%	0%
Jul	7%	12%	53%	23%	2%	1%
Ago	7%	12%	52%	26%	1%	2%

Set	6%	10%	49%	35%	0%	0%
Total	3%	10%	55%	27%	4%	1%

Nota. Elaboración propia.

1.2. Justificación de la Investigación

1.2.1. Teórica

Existe la necesidad de aplicar herramientas de Machine Learning para abordar un problema crítico en la empresa productora y comercializadora de productos avícolas. La teoría de Machine Learning proporciona un marco sólido y efectivo para desarrollar modelos predictivos que pueden ayudar a anticipar con precisión la demanda de paquetes de huevos (15 huevos). La aplicación de Machine Learning permitirá aprovechar al máximo la gran cantidad de datos históricos de ventas y otros factores relevantes para construir un modelo que pueda predecir la demanda de manera más precisa y, por lo tanto, mejorar significativamente la planificación de inventario y la satisfacción del cliente.

1.2.2. Práctica

La aplicación de Machine Learning en este contexto se justifica debido a las numerosas dificultades operativas y comerciales que enfrenta la empresa. La falta de precisión en la proyección de la demanda de paquetes de huevos (15 huevos) ha llevado a problemas como la falta de stock, entregas retrasadas y pérdida de ventas. Estos problemas tienen un impacto directo en la experiencia del cliente y en la eficiencia operativa de la empresa. Al desarrollar un modelo de Machine Learning que pueda predecir la demanda con mayor precisión, la empresa puede optimizar su gestión de inventario, mejorar la entrega oportuna de productos y aumentar la satisfacción del cliente, lo que finalmente se traducirá en un mejor rendimiento comercial.

1.2.3. Metodológica

El estudio se justifica por la idoneidad de las herramientas de Machine Learning para abordar el problema planteado. El Machine Learning ofrece un enfoque sistemático y basado en datos para modelar y predecir la demanda, utilizando algoritmos y técnicas que pueden aprender patrones complejos a partir de datos históricos. La metodología de Machine Learning permitirá explorar y analizar conjuntos de datos complejos, identificar relaciones no lineales y capturar las variaciones estacionales y las tendencias en la demanda de paquetes de huevos (15

huevos). Además, brinda la capacidad de ajustar y mejorar continuamente el modelo a medida que se recopilan nuevos datos.

1.3. Delimitación de la Investigación

1.3.1. Espacial

El presente estudio se limitará a la venta de paquetes de huevos (15 huevos) a través del canal E-commerce de la empresa en cuestión, el cual tiene cobertura solo para los distritos de Lima Metropolitana, Asia y Cañete.

1.3.2. Temporal

Para el análisis de información y entrenamiento del modelo de Machine Learning, se considerará las órdenes de paquetes de huevos (15 huevos) recibidas por la empresa entre los meses de febrero de 2021 y septiembre de 2023.

1.3.4. Conceptual

El estudio se centrará en la predicción de la demanda de paquetes de huevos (15 huevos) del canal E-commerce de la empresa. Se elige este producto por ser el que recibe más órdenes en este canal de venta.

CAPÍTULO II: MARCO TEÓRICO

2.1 Antecedentes de la Investigación

Antecedente 1: Comparison of statistical and machine learning methods for daily SKU demand forecasting (Spiliotis et. al., 2020)

Resumen

Este estudio aborda el desafiante problema de la predicción diaria de la demanda de SKUs, que a menudo se caracteriza por series irregulares y erráticas. Se compara el rendimiento de una variedad de métodos de Machine Learning (ML) con métodos estadísticos tradicionales en la predicción de la demanda diaria de productos de consumo en Grecia. Los resultados revelan que algunos métodos de ML superan a los métodos estadísticos en términos de precisión y sesgo en la predicción de la demanda en 8.9%. Además, se explora el concepto de aprendizaje cruzado y se encuentra que puede mejorar la precisión de las redes neuronales en 1.8% cuando se entrenan de manera individual para cada serie de datos. Sin embargo, también se indica que este enfoque no es adecuado para todos los métodos de ML. En general, el estudio destaca la importancia de considerar cuidadosamente el equilibrio entre el rendimiento del pronóstico y los costos computacionales al diseñar procesos de pronóstico en entornos de gestión de inventario. Estas conclusiones ofrecen información valiosa para la toma de decisiones en la gestión de inventario y la planificación de la cadena de suministro.

Problema

El problema principal abordado en el artículo es la predicción de la demanda diaria de SKU (Stock Keeping Unit) que a menudo muestra series irregulares, intermitentes y erráticas. Esta problemática se vuelve aún más desafiante cuando se realiza la predicción a niveles transversales bajos, como el nivel de una tienda o almacén, o cuando se trata de productos de movimiento lento. La precisión en la predicción es esencial para tomar decisiones eficaces sobre la gestión de inventario y el reabastecimiento.

Objetivo

El objetivo principal del artículo es comparar el rendimiento de varios métodos de Machine Learning (ML) con métodos estadísticos tradicionales en la predicción de la demanda diaria de SKU. Los métodos de ML se evalúan tanto en series individuales como en un enfoque

de "aprendizaje cruzado", y se busca determinar si los métodos de ML superan a los métodos estadísticos en términos de precisión y sesgo en la predicción de demanda. Los objetivos secundarios del estudio son los siguientes:

- Evaluar el rendimiento de una variedad de métodos de ML y no solo las Redes Neuronales (NN) en la predicción de la demanda de SKU.
- Investigar el impacto del "aprendizaje cruzado" en la precisión de la predicción de demanda diaria.
- Proporcionar una evaluación exhaustiva de diferentes métodos estadísticos en la predicción de demanda de SKU.
- Comparar los resultados de los métodos de predicción en términos de precisión y sesgo, y analizar su significancia estadística.
- Evaluar el equilibrio entre el rendimiento de la predicción y el costo computacional de los métodos utilizados.

Metodología

El estudio aplica una amplia gama de métodos de pronóstico estadísticos y de Machine Learning (ML) que son utilizados en la literatura para predecir la demanda en series de tiempo y los compara de manera detallada, considerando diferentes configuraciones y enfoques para el ML y la agregación temporal para saber cómo estos métodos funcionan en la práctica y cuáles podrían ser las mejores opciones en diferentes situaciones de pronóstico.

Los métodos aplicados en el estudio se dividen en dos categorías principales: 11 métodos estadísticos y 7 métodos de Machine Learning (ML). Los métodos estadísticos prescriben el proceso de generación de datos, mientras que los métodos ML permiten que el modelo aprenda las relaciones en los datos. Los hiperparámetros de los modelos de ML fueron optimizados mediante una búsqueda en cuadrícula en un conjunto de validación utilizando el Root Mean Squared Scaled Error (RMSSE) como criterio de optimización.

Además de entrenar los modelos de ML de manera individual para cada serie de tiempo, el estudio también explora modelos de aprendizaje cruzado. En este enfoque, los modelos se entrenan utilizando ventanas de datos de múltiples series de tiempo, lo que podría permitir una mejor generalización.

Se consideró la inclusión de características de series de tiempo, como el coeficiente de variación de las demandas no nulas (CV2) y el promedio de períodos entre dos demandas no nulas (ADI), como regresores adicionales para los modelos de aprendizaje automático.

Los métodos se evaluaron utilizando métricas de error de pronóstico en un conjunto de datos de prueba, y se compararon sus rendimientos en términos de precisión.

Resultados

Los resultados mostraron que cuatro de los siete métodos de ML (GPT, Random Forest - RF, Support Vector Regression - SVR y K-Nearest Neighbors Regression - KNNR) superaron a todos los métodos estadísticos en términos de precisión y sesgo, lo que sugiere que estos métodos tienen mayor potencial para mejorar pronósticos de la demanda diaria en comparación con métodos estadísticos tradicionales (8.9% mayor precisión en promedio al comparar los resultados de RMSSE).

Algunos métodos estadísticos superan a otros en términos de precisión y sesgo. Por ejemplo, al comparar los resultados de RMSSE, el método SBA superó al método Croston's en 1%, mientras que MA-opt superó al método MA (en 5%), al igual que el SBJA (en 2.1%). Esto demuestra que la elección del método de pronóstico es crucial.

El aprendizaje cruzado mejora la precisión de los métodos de Machine Learning en 1.8% cuando se usaba solo data histórica para el entrenamiento y en 2.5% cuando se incluyeron características de series de tiempo.

Los métodos de ML, cuando se entrenaban de manera individual para cada serie de tiempo, requerían significativamente más tiempo computacional en comparación con los métodos estadísticos. Sin embargo, cuando se utilizaba el aprendizaje cruzado, se reducía el tiempo computacional, lo que hacía que los métodos de ML fueran más eficientes.

Conclusiones

Las conclusiones del estudio son estadísticamente significativas y se basan en una evaluación exhaustiva de un gran conjunto de datos de series temporales de demanda diaria de productos de consumo en Grecia. Se concluye que los métodos de ML pueden proporcionar pronósticos significativamente menos sesgados y más precisos que los métodos estadísticos establecidos, como el método de Croston y sus variantes, en el contexto de la predicción de la demanda diaria de SKUs.

En cuanto al aprendizaje cruzado, se concluye que estas técnicas (cross-learning) pueden mejorar el rendimiento de las redes neuronales (NN) cuando se entrenan de manera individual para cada serie de datos, especialmente cuando se utilizan características de series temporales junto con datos históricos para entrenar las redes (2.5% de incremento en precisión). Sin embargo, el aprendizaje cruzado tuvo un impacto negativo en otros métodos de ML, lo que sugiere que diferentes enfoques de modelado deben aplicarse según las características particulares del método de pronóstico.

Finalmente, el estudio resalta la importancia de diseñar cuidadosamente los procesos de pronóstico en función de las necesidades y limitaciones de la organización, considerando la compensación entre el rendimiento del pronóstico y los costos computacionales.

Antecedente 2: Machine Learning Outperforms Classical Forecasting on Horticultural Sales Predictions (Haselbeck et.al., 2022)

Resumen

En el presente artículo, Haselbeck et al. (2022) comparan empíricamente el rendimiento de nueve algoritmos de Machine Learning de última generación y tres algoritmos de pronóstico clásicos para realizar predicciones de ventas hortícolas. Se demostró que los métodos de aprendizaje de Machine Learning fueron superiores en todos los experimentos, siendo el algoritmo XGBoost el que tuvo mejor desempeño en el 93% de las comparaciones realizadas (14 de 15 comparaciones), lo cual aumentó para conjuntos de datos con múltiples estaciones. Asimismo, se demostró que la inclusión de factores externos, como información meteorológica, vacaciones, feriados públicos, etc., resultó en el incremento del rendimiento predictivo, en especial feriados públicos, el cual obtuvo un desempeño del 62% en comparación con los otros factores. Adicionalmente, se investigó si los algoritmos podían predecir el aumento de la demanda de productos hortícolas durante la pandemia de SARS-CoV-2, resultando XGBoost superior.

Problema

Los productos hortícolas tales como plantas en macetas, flores cortadas y arbustos, se ven muy influenciados por la demanda. Pronosticar la demanda futura es muy importante para muchas empresas; puesto que influye en la toma de decisiones. Asimismo, la demanda se ve

influenciada por factores externos como días festivos, fenómenos meteorológicos, etc, y algunos de estos pueden ser inciertos o estacionales.

Predecir oportunamente la demanda puede resultar en mayores ingresos y más aún cuando se trata de productos que tienen una vida útil corta. Caso contrario, puede ocasionar situaciones de falta de existencias, ventas perdidas, exceso de existencias, eliminación de productos, pérdidas financieras y adicionalmente, provoca daños ambientales pues se genera desperdicio de recursos durante la producción y transporte.

Diversas organizaciones emplean metodologías para pronosticar la demanda, dentro de las cuales se tiene metodologías de pronóstico clásico y enfoques de aprendizaje automático como los algoritmos de Machine Learning. No obstante, aunque los algoritmos de Machine Learning se están volviendo más comunes en la literatura de pronóstico, no está del todo comprobado que sean superiores a los métodos de pronóstico clásico. En este contexto, determinar qué metodología es la más apropiada, contribuiría a realizar un pronóstico de demanda más preciso, en este caso en particular, para productos hortícolas.

Objetivo

Haselbeck et al. (2022) establecen los siguientes objetivos para su estudio:

- Determinar si las metodologías de aprendizaje automático son superiores en comparación con los enfoques clásicos de pronóstico para realizar predicciones de la venta de productos hortícolas.
- Determinar mejoras potenciales mediante conceptos multivariados haciendo uso de factores externos, como datos meteorológicos o de vacaciones, comparándolos con métodos univariados clásicos.
- Evaluar el consumo de recursos computacionales de los métodos aplicados, teniendo en cuenta la necesidad de reajustar los modelos debido a distribuciones de datos potencialmente cambiantes durante la operación en vivo de un sistema de pronóstico.
- Examinar si los modelos son capaces de determinar cambios repentinos en los datos, tal como los fuertes aumentos de demanda durante la pandemia de SARS-CoV-2.

Metodología

Con respecto a la metodología empleada en el presente estudio, el primer paso realizado por Haselbeck et al. (2022) fue la preparación de datos, para lo cual se analizó un conjunto de

datos típicos de ventas minoristas de productos hortícolas en Alemania. Se distinguió cinco tipos de conjuntos de datos. Se creó una fuente de datos manual mediante una documentación diaria de cifras de ventas de tulipanes “OwnDoc”, para un periodo corto y para un periodo largo. Los conjuntos de datos se caracterizaron por tener ciclos estacionales con una duración aproximada de tres meses y con cambios abruptos en su demanda. Con respecto a las predicciones, el estudio se enfocó en las ventas de tulipanes a clientes privados, denominadas “SoldTulips”. Además, se creó data en base al resumen de todas las ventas proporcionada por un sistema de cajero electrónico, denominada “CashierData”. Las variables objetivo fueron flores cortadas “CutFlowers” y plantas en macetas “PotTotal”, tanto para un periodo corto como para un periodo largo. Asimismo, se agregó información diaria sobre el clima y los días festivos para analizar si los factores externos respaldaban el pronóstico de las ventas hortícolas y se aplicó distintas estrategias para la imputación de datos para los valores faltantes.

Realizada la preparación de datos, se procedió con el entrenamiento del modelo, para lo cual se separó un conjunto de entrenamiento, constituido por el 80% de todo el conjunto de datos. Se realizó un estudio comparativo de nueve métodos de aprendizaje automático de última generación y tres métodos de pronóstico clásicos. Dentro de las metodologías clásicas, se tiene: técnicas Univariadas de Suavizado Exponencial, Media Móvil Integrada Autorregresiva Estacional, Media Móvil Integrada Autorregresiva Estacional con factores externos. En cuanto a los métodos de aprendizaje automático, se emplean Modelos de Regresión Lineal Múltiple, Métodos de aprendizaje automático no lineales, como las redes neuronales artificiales y redes de memoria a corto plazo. Asimismo, se aplicó el aprendizaje conjunto Extreme Gradient Boosting (XGBoost) y un enfoque bayesiano no paramétrico, implementando la regresión del proceso gaussiano.

Adicionalmente, se implementó una configuración con reajuste regular del modelo de pronóstico con el fin de simular un escenario potencial para una operación productiva con una actualización continua de datos de ventas y una distribución de datos cambiante. Asimismo, se empleó la validación cruzada de series de tiempo. Para la evaluación, se utilizó la métrica dependiente de la escala del error cuadrático medio (RMSE), así como las medidas relativas del error porcentual absoluto medio (MAPE) y el error porcentual absoluto medio simétrico (SMAPE). Se experimentó con los cinco conjuntos de datos, considerando las tres métricas de evaluación, realizando en total 15 comparaciones para todas las líneas de base, algoritmos y

conjuntos de características. Las etapas del proceso metodológico del estudio se pueden apreciar a continuación, en la Figura 4.

Figura 4
Metodología de los modelos de predicción de demanda de tulipanes.

Nota. Elaboración propia en base a Haselbeck et al.

Resultados

Los resultados del estudio demostraron superioridad de los métodos basados en Machine Learning en comparación con los de pronóstico clásicos. El alumno del conjunto XGBoost obtuvo los mejores resultados en 14 de las 15 comparaciones (93%). No obstante, las redes de memoria a corto plazo (LSTM) en conjunto con el método bayesiano no paramétrico de regresión del proceso gaussiano (GPR) fueron más competitivas en comparación con XGBoost.

Asimismo, se analizó el consumo de recursos computacionales, que resulta importante especialmente cuando se tiene en cuenta un reajuste regular del modelo debido a una distribución de datos cambiante, siendo XGB el de mejor rendimiento, considerando la data creada manualmente, con un RMSE de 57.11, un SMAPE de 50.89 y un MAPE de 35.98 para el periodo corto, mientras que, para el periodo largo, obtuvo un RMSE de 43.52, un SMAPE de 48.65 y un MAPE de 54.22 para la variable de tulipanes vendidos. Asimismo, el modelo SARIMAX se ubica en segundo lugar con un RMSE de 58.06, un SMAPE de 52.58 y un MAPE de 85.75 para el periodo corto, mientras que, para el periodo largo, obtuvo un RMSE de 50.35, un SMAPE de 52.37 y un M APE de 231.52 para la variable de tulipanes vendidos.

Para la data proveniente del sistema de cajero electrónico, XGB obtuvo un RMSE de 390.96, un SMAPE de 20.03 y un MAPE de 24.52, para la variable de flores cortadas; un RMSE de 898.56, un SMAPE de 35.35 y un MAPE de 42.45, para la variable de plantas en maceta en periodo largo y un RMSE de 392.31, un SMAPE de 31.50 y un MAPE de 30.17 para la variable de plantas en macetero en periodo corto. No obstante, solo en este último caso XGB se vió superado por LSTM, con un RMSE de 390.98, un SMAPE de 28.58. Cabe destacar que, con respecto al análisis de la efectividad de los métodos presentados para determinar el aumento de ventas en contexto de pandemia, XGB obtuvo también el mejor resultado.

Por otro lado, se tomó en cuenta el número de veces que un conjunto de características, como factores meteorológicos, fechas festivas, etc., condujo al mejor resultado para cada algoritmo y conjuntos de datos. Los resultados indican que factores externos como la información meteorológica y las características derivadas como medidas estadísticas, contribuyó a un rendimiento de pronóstico mejorado ya que los métodos eran multivariados, lo cual es útil especialmente para los métodos basados en ML, en especial para el factor Calendric features (feriados públicos como el Día de la Madre y el Día de San Valentín), el cual obtuvo un desempeño del 62% en comparación con los otros factores. Asimismo, SARIMAX fue el enfoque clásico más competitivo, pero tiene poca escalabilidad en conjuntos de características y datos más grandes. Por el contrario, XGB fue el más eficiente. Adicionalmente, se obtuvo que la pandemia SARS-CoV-2 influyó en el incremento de la demanda de plantas en maceta, resultando XGBoost el algoritmo más eficiente para predecir el aumento de la demanda.

Conclusiones

Los autores del artículo presentan un estudio comparativo para realizar predicciones de ventas hortícolas aplicando nueve algoritmos de Machine Learning y tres métodos tradicionales. Se emplearon datos de horticultura con distintas características como tamaño y estacionalidad. Se configuró el pronóstico para simular una operación productiva de un sistema de pronóstico con una actualización continua de datos. Los resultados demostraron superioridad para los enfoques de Machine Learning, en especial del conjunto XGB. Asimismo, se experimentó un aumento en el rendimiento cuando se incluyeron funciones adicionales como datos meteorológicos.

Finalmente, es importante mencionar que la aplicación de enfoques combinados podría ser superior para pronosticar la demanda hortícola. Por lo tanto, la integración de métodos

combinados como, por ejemplo, técnicas de pronóstico clásicas basadas en ML, constituyen una oportunidad de investigación para el futuro.

Antecedente 3: Predictive Analytics for Demand Forecasting A comparison of SARIMA and LSTM in Retail SCM (Brandtner et. al., 2022)

Resumen

Como modelo paramétrico, se seleccionó SARIMA(X) y como modelo no paramétrico, LSTM, para proporcionar predicciones de venta de las hortalizas seleccionadas para el mes de enero de 2020 con base en los datos de los años 2017, 2018 y 2019. Los resultados muestran que el modelo LSTM superó al modelo SARIMA(X) en términos de precisión de pronóstico, con una mejora del 7,5% en la precisión de pronóstico. Además, el modelo LSTM también superó al modelo SARIMA(X) en términos de tiempo de ejecución, con una mejora del 98,5% en el tiempo de ejecución. En general, el modelo LSTM se considera más adecuado para la previsión de la demanda de frutas y verduras en el contexto de SCM minorista.

Problema

El problema que se aborda en la lectura es la predicción de la demanda en la gestión de la cadena de suministro minorista, específicamente en la predicción de la demanda de productos perecederos como frutas y verduras. El artículo destaca que la predicción precisa de la demanda es crucial para reducir el exceso o la falta de inventario, mejorar la eficiencia y la sostenibilidad, y evitar la pérdida de recursos naturales y la insatisfacción del cliente. El artículo también compara dos modelos de predicción de demanda, SARIMA y LSTM, para ayudar a los minoristas a hacer pronósticos más precisos.

Objetivo

- Presentar un estudio comparativo entre dos modelos de predicción de demanda, SARIMA y LSTM, para ayudar a los minoristas a hacer pronósticos más precisos.
- Destacar la importancia de la predicción de la demanda en la gestión de la cadena de suministro minorista y cómo la predicción precisa puede ayudar a reducir el exceso o la falta de inventario y mejorar la eficiencia y la sostenibilidad.
- Proporcionar una revisión de la literatura sobre los métodos de predicción de la demanda en la SCM minorista.

- Discutir las aplicaciones del mundo real de los modelos SARIMA y LSTM en el comercio minorista.
- Proporcionar una perspectiva sobre futuras investigaciones en el campo de la predicción de la demanda en la SCM minorista.

Metodología

EL autor compara dos metodologías para el pronóstico de la demanda: SARIMA (Seasonal Autoregressive Integrated Moving Average) y LSTM (Long Short-Term Memory), en el contexto de la gestión de la cadena de suministro minorista (Retail SCM). Para ello realiza los siguintes pasos:

- Selección de los modelos: La metodología incluye la selección de dos enfoques de pronóstico diferentes: SARIMA y LSTM.
- Preparación de los datos: Se recopilan y pre procesan los datos de demanda histórica, que son cruciales para el entrenamiento y la evaluación de los modelos de pronóstico.
- Implementación de SARIMA: Se aplica el modelo SARIMA, que es un método de series temporales clásico y ampliamente utilizado para el pronóstico. Este modelo tiene en cuenta la estacionalidad y los patrones temporales en los datos.
- Implementación de LSTM: Se aplica el modelo LSTM, que es un tipo de red neuronal recurrente (RNN) especialmente diseñada para el procesamiento de secuencias temporales. Las LSTM pueden capturar relaciones temporales más complejas en los datos.
- División de los datos: Se dividen los datos en conjuntos de entrenamiento y prueba para evaluar la capacidad de generalización de los modelos.
- Entrenamiento y validación de modelos: Se entrena tanto el modelo SARIMA como el LSTM utilizando los datos de entrenamiento. Luego, se valida su rendimiento utilizando los datos de prueba.
- Evaluación del rendimiento: Se comparan los resultados de los dos modelos para determinar cuál de ellos proporciona un pronóstico más preciso y confiable. Se utilizan métricas de evaluación de pronósticos, como la raíz del Error Cuadrático Medio (RMSE) y el Error de Porcentaje Medio Absoluto (MAPE).
- Análisis de resultados: Se analizan los resultados para comprender las fortalezas y debilidades de cada modelo en el contexto específico de la gestión de la cadena de suministro minorista. Se considera como mejor modelo a aquel que tiene un menor valor

de RMSE y/o MAPE. Asimismo, para este último se establece que el modelo es muy preciso cuando el MAPE es menor a 10, preciso cuando está entre 10 y 20, razonable cuando está entre 20 y 50 e inexacta cuando es mayor a 50

Los pasos de la metodología empleada se pueden observar en la Figura 5.

Figura 5

Pasos de la Metodología entre dos modelos de predicción de demanda

Nota. Elaboración propia.

Resultados

Para el producto ensalada, el mejor modelo fue el SARIMA con un MAPE de 13 y un RMSE de 949 seguido del LSTM con un MAPE de 14 y un RMSE de 1009. Para los tomates, el mejor desempeño lo tuvo el LSTM con un MAPE de 44 seguido del SARIMAX con 48 y un RMSE de 1170 y 1163 respectivamente. En el caso de las papas ganó el LSTM con un MAPE de 15 y un RMSE de 1184 seguido del SARIMA con 25 y 1626 respectivamente. Por último, para el pepino, ganó el SARIMA con 16 de MAPE y 2854 para RMSE, seguido del SARIMA con un 35 y 4502 para cada indicador.

Un segundo caso fue cuando se entrenó el modelo SARIMA y se incluyó la existencia de promociones como variable externa. A excepción del tomate, para los otros tres productos se obtuvo reducciones de entre 12.5% a 53% respecto al MAPE cuando se incluyó esta nueva variable.

Conclusiones

El artículo proporciona información valiosa sobre el uso de análisis predictivos para la previsión de la demanda en SCM minorista y sugiere enfoques híbridos para mejorar la calidad de la previsión a nivel de tienda y por ende aumentar las ventas y reducir el desperdicio de productos. Esto debido a que los resultados del estudio muestran que ambos modelos produjeron resultados de razonables a buenos. En general, LSTM tuvo mejores resultados para productos con demanda estable, mientras que SARIMA mostró mejores resultados para productos con comportamiento estacional. Además, el estudio comparó los resultados con SARIMAX agregando el factor externo de promociones y encontró que SARIMAX tuvo un desempeño significativamente mejor para los productos con promociones.

Se utilizó el método SARIMAX, agregando promociones como variable externa al modelo SARIMA, y se encontró una mejora significativa en los resultados. Sin embargo, se sugiere considerar otros factores externos influyentes para mejorar aún más la calidad del pronóstico. Se destacó la importancia de comprender las limitaciones de los datos, así como los conocimientos comerciales y temporales antes de realizar análisis.

Antecedente 4: Demand forecasting model for times-series pharmaceutical data using shallow and deep neural network model (Rathipriya et. al., 2022)

Resumen

El artículo presenta un modelo de red neuronal superficial y profunda para predecir la demanda de ocho grupos diferentes de productos farmacéuticos. Se utiliza el error cuadrático medio (RMSE) y el porcentaje de error (PE) para evaluar la precisión predictiva del modelo de previsión de la demanda (DFM). Además, se recomiendan estrategias de ventas y marketing basadas en los efectos de tendencia/estacional de los diferentes grupos de productos farmacéuticos. Los resultados muestran que el DFM basado en redes neuronales profundas tiene una precisión predictiva significativamente mejor que el DFM basado en redes neuronales superficiales, con un RMSE normalizado del 7,5% y un PE del 8,2%.

Problema

El problema que aborda el artículo es la predicción de la demanda de productos farmacéuticos utilizando modelos de pronóstico de demanda de series temporales basados en redes neuronales superficiales y profundas, y comparar su rendimiento con los modelos

estadísticos tradicionales como ARIMA y modelos de redes neuronales poco profundas. El objetivo es proporcionar recomendaciones potencialmente útiles a la industria farmacéutica para mejorar su planificación de la producción y la gestión de inventarios.

Objetivo

- Desarrollar una metodología para seleccionar los mejores modelos de pronóstico de demanda para datos farmacéuticos utilizando modelos estadísticos y de redes neuronales para una base de datos de ventas de series temporales.
- Investigar si los métodos de red neuronal superficial y profunda conducen a una predicción más precisa de la demanda de fármacos ATC que los modelos de redes no neuronales.
- Investigar el rendimiento del modelo ARIMA, Shal-Modelos de red neuronal baja y red neuronal profunda al tratar con conjuntos de datos de series temporales.
- Proporcionar recomendaciones potencialmente útiles a la industria farmacéutica para mejorar su planificación de la producción y la gestión de inventarios.

Metodología

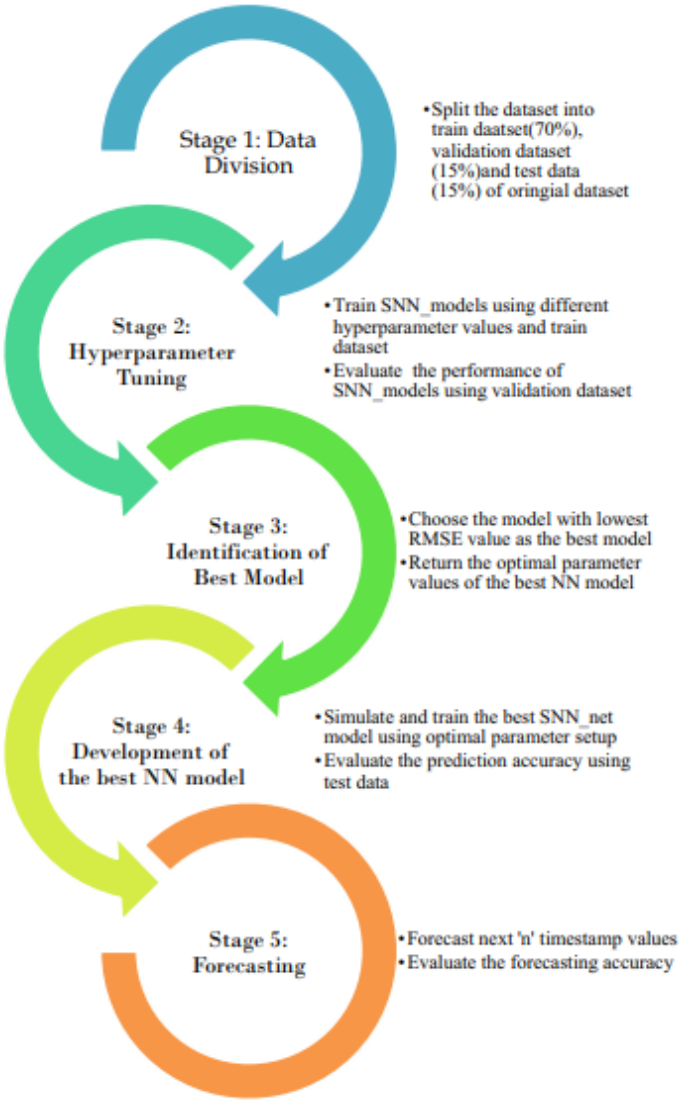
La metodología propuesta en el artículo consta de los siguientes pasos:

- Recopilación de datos: se recopilaron datos de ventas de 57 fármacos de una empresa farmacéutica durante un período de 5 años.
- Análisis temático de tratamiento químico anatómico (ATC): los 57 fármacos se clasificaron en ocho secciones de análisis temático de tratamiento químico anatómico (ATC).
- Selección de modelos de pronóstico de demanda: se seleccionaron varios modelos de pronóstico de demanda, incluidos modelos estadísticos como ARIMA y modelos de redes neuronales superficiales y profundas como RBF_NN, P_NN, GR_NN, LSTM y Stacked LSTM.
- Implementación de modelos de pronóstico de demanda: se implementaron los modelos seleccionados y se ajustaron los parámetros y hiperparámetros.
- Evaluación del rendimiento del modelo: se evaluó el rendimiento de cada modelo utilizando medidas de error de pronóstico como el error absoluto medio (MAE), el error cuadrático medio (MSE) y el coeficiente de determinación (R2).

- Comparación de modelos: se compararon los modelos de pronóstico de demanda seleccionados y se identificó el modelo más preciso para predecir la demanda de fármacos ATC.
- Análisis de resultados: se analizaron los resultados y se proporcionaron recomendaciones para mejorar la planificación de la producción y la gestión de inventarios en la industria farmacéutica.

Los pasos de la metodología empleada por el estudio se pueden observar seguidamente, en la Figura 6.

Figura 6
Flujo de trabajo de la metodología propuesta



Nota. Springer-Verlag London Ltd (06 de octubre del 2022). Computación neuronal y aplicaciones. https://doi.org/10.1007/s00521-022-07889-9(0123456789().,-volV)(0123

Resultados

El modelo LSTM multicapa tuvo un rendimiento de predicción más robusto que el modelo CNN y los modelos estadísticos tradicionales como ARIMA. Los modelos de redes neuronales superficiales y profundas, como RBF_NN, P_NN, GR_NN, LSTM y Stacked LSTM, tuvieron un rendimiento de predicción más preciso que los modelos de redes no neuronales. El modelo LSTM multicapa fue el modelo más preciso para predecir la demanda de fármacos ATC. Se proporcionaron recomendaciones para mejorar la planificación de la producción y la gestión de inventarios en la industria farmacéutica, como la implementación de modelos de pronóstico de demanda basados en redes neuronales superficiales y profundas y la segmentación de los datos de ventas por categorías de fármacos ATC.

Conclusiones

La conclusión del artículo es que los modelos de pronóstico de demanda basados en redes neuronales superficiales y profundas, como RBF_NN, P_NN, GR_NN, LSTM y Stacked LSTM, son más precisos que los modelos de redes no neuronales y los modelos estadísticos tradicionales como ARIMA para predecir la demanda de fármacos ATC. Además, se encontró que el modelo LSTM multicapa fue el modelo más preciso para predecir la demanda de fármacos ATC. Se proporcionaron recomendaciones para mejorar la planificación de la producción y la gestión de inventarios en la industria farmacéutica, como la implementación de modelos de pronóstico de demanda basados en redes neuronales superficiales y profundas y la segmentación de los datos de ventas por categorías de fármacos ATC. En general, el estudio demuestra que los modelos de redes neuronales pueden ser una herramienta valiosa para la industria farmacéutica en la planificación de la producción y la gestión de inventarios.

2.2 Bases Teóricas

2.2.1. Demanda

Según Elvira Arboleda, La demanda del mercado refleja las preferencias y requerimientos de un conjunto de individuos en un área específica, la cual está moldeada por sus intereses, requisitos y tendencias. (2021, p.4). De este modo surge la ley de la demanda que, en condiciones constantes, establece una relación inversa entre la cantidad necesitada por las personas y el precio de los productos ofertados. Así, cuando el precio de un artículo sube, los

consumidores muestran una menor disposición a pagar más por ese artículo en particular, lo que resulta en una reducción de la demanda del mismo.

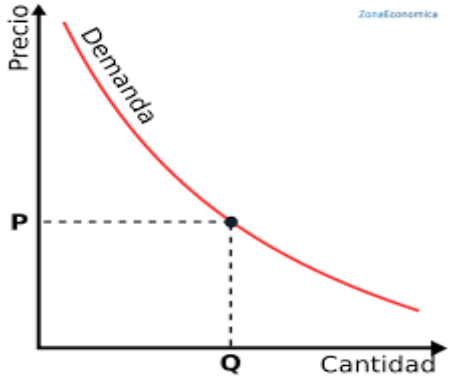
Desde el punto de vista de otros autores, el investigar las necesidades esenciales significa evaluar en términos de unidades físicas o valores monetarios, las variables que, en el contexto del mercado, microentorno y macroentorno, ejercen influencia sobre ello para proyectar las tendencias futuras anticipadas (Diez de Castro y Landa, 1994, p. 70).

Para Samuelson y Nordhaus (2010), la demanda es un principio fundamental en el campo de la economía y desempeña un rol significativo en la fijación de precios y la distribución eficiente de recursos en una economía basada en el mercado. Es importante destacar que la demanda no solo se trata de la cantidad que los consumidores desean, sino también de su capacidad y disposición para pagar por ello.

Esto significa que, si bien una persona puede desear algo, si no tiene los medios económicos para adquirirlo, no se considera parte de la demanda efectiva. Por ende, la comprensión de la demanda es esencial para comprender la dinámica entre consumidores y productores, así como para la determinación de precios y niveles de producción en un mercado. Adicionalmente, juega un papel crucial en las decisiones tomadas por empresas, gobiernos y otras entidades respecto a la producción y distribución de bienes y servicios.

La curva de demanda, por su parte, representa la correlación entre el precio de un bien o servicio específico y el nivel de demanda por parte de los consumidores en un momento y grupo de población determinados. Esta curva ilustra cuánto de un bien es solicitado en un período de tiempo específico por un grupo de personas específico, lo cual se puede apreciar en la Figura 7.

Figura 7
Curva de demanda



Nota. Extraído de zona económica. https://m.zonaeconomica.com/demanda

Tipos de demanda

Hay varios tipos de demanda y cada una describe situaciones particulares en las que los consumidores están dispuestos a comprar ciertos bienes o servicios. A continuación, se resumen las principales:

1. **Demanda individual:** Es la cantidad de un bien o servicio que un consumidor u organización está dispuesto y puede comprar a diferentes precios durante un período de tiempo específico.

2. **Demanda de mercado:** Es la sumatoria de las demandas individuales de todos los consumidores u organizaciones en un mercado particular para un bien o servicio.

3. **Demanda agregada:** Es la cantidad total de bienes y servicios demandados en una economía en un momento dado y a un precio determinado.

4. **Demanda efectiva:** Se refiere a la cantidad de bienes y servicios que realmente compran los consumidores, teniendo en cuenta sus capacidades financieras.

5. **Demanda potencial:** Es la cantidad de un bien o servicio que un consumidor está dispuesto a comprar si dispone de recursos suficientes. Podría ser mayor que la demanda real porque algunos consumidores pueden querer comprar más de lo que pueden pagar.

6. **Demanda inelástica y elástica:** Se refiere a la sensibilidad de la cantidad demandada a los cambios en el precio. Si la demanda es inelástica, los consumidores comprarán

aproximadamente la misma cantidad incluso si el precio cambia significativamente. Si es elástico, los consumidores reaccionan fuertemente a los cambios de precio.

7. **Demanda de bienes sustitutos y complementarios:** Los bienes sustitutos son bienes que pueden usarse uno en lugar de otro (por ejemplo, mantequilla y margarina), mientras que los bienes complementarios son bienes que se consumen juntos (por ejemplo, café y margarina).

8. **Demanda de bien normal e inferior:** Son bienes cuya demanda incrementa a medida que aumenta la renta del consumidor, mientras que los bienes inferiores son bienes cuya demanda disminuye a medida que aumenta la renta del consumidor.

9. **Demanda estacional:** Se refiere a hábitos de compra que están influenciados por factores estacionales, como las estaciones o los días festivos.

Factores que influyen

La demanda de un bien o servicio está influenciada por muchos factores diferentes y el precio es uno de los determinantes más importantes. Sin embargo, hay otros factores importantes que también desempeñan un papel importante a la hora de determinar la cantidad de un bien o servicio que un consumidor está dispuesto y es capaz de comprar. Aquí detallo algunos de los factores clave que influyen en la demanda:

Precio

En lo que respecta al valor monetario, el precio se puede definir como la suma de dinero desembolsada por un producto o servicio. (Kotler & Armstrong, 2008). Por consecuente, representa el monto de dinero que es equivalente a la valoración completa que los consumidores hacen a cambio de los beneficios que obtienen al poseer o utilizar un producto o servicio. Por otra parte, el valor de un producto está determinado por la percepción que el consumidor tiene de su imagen. (Bonta, P. & Faber, M., 2003).

Precios de bienes relacionados:

- **Sustitutos:** Un incremento en el costo del artículo podría ocasionar un aumento en la demanda del sustituto, y a su vez, una reducción en la demanda del producto original (y viceversa).

- **Complementos:** Un incremento en el precio de uno puede disminuir la demanda del otro. (y viceversa).

Oferta

Se refiere a la cantidad de un bien o servicio que un productor está dispuesto a ofrecer en el mercado a diversos precios durante un intervalo de tiempo específico. (Mankiw, 2018). Esta cantidad aumentará a medida que su precio aumente y disminuirá a medida que su precio disminuya. (Samuelson y Nordhaus, 2010). Esta relación directa se debe a que a medida que el precio aumenta, los productores encuentran mayor incentivo para poner a disposición más unidades del bien o servicio en el mercado ya que la producción se vuelve más rentable para ellos.

Una oferta elástica indica que la cantidad ofrecida es altamente sensible a cambios en el precio, mientras que una oferta inelástica indica que la sensibilidad es menor. (Varian, 2014). No obstante; además del precio, la oferta está influenciada por diversos factores como los costos de producción, los precios de los insumos, la tecnología, la capacidad de producción de las empresas y las expectativas del mercado.

Esta relación entre el precio y la cantidad ofrecida se representa gráficamente en lo que se conoce como la curva de oferta. Esta ilustra la cantidad de un bien o servicio que los productores están dispuestos a ofrecer en el mercado a diferentes niveles de precios, manteniendo constantes otros factores que pueden influir en la oferta. Cabe señalar que la oferta no es estática y puede cambiar con el tiempo debido a modificaciones en los factores que afectan la producción.

Calidad del producto

La calidad implica un proceso de mejora constante en el que todos los departamentos de la empresa trabajan para satisfacer o anticipar las necesidades del cliente, participando de manera activa en el desarrollo del producto o en la provisión de servicios. (Álvarez, 2006).

La calidad tiene muchas vertientes, pero todas contienen palabras que son claves para una definición común: satisfacción de necesidades, expectativas, percepciones, cualidades o características. Con base en esas palabras, podemos definirlo como “La satisfacción proporcionada por las características de un producto o servicio se alinea con las expectativas del consumidor específico de un producto o servicio de alto nivel de calidad.”. (López, 2005).

Competencia

Villa y Poblete (2004, p.8) afirman que la competencia implica "rendir eficazmente en situaciones auténticas y desafiantes" por lo que requiere la combinación y aplicación de conocimientos, destrezas, actitudes y principios. Al respecto, Jaume Sarramona nos indica que la competencia implica:

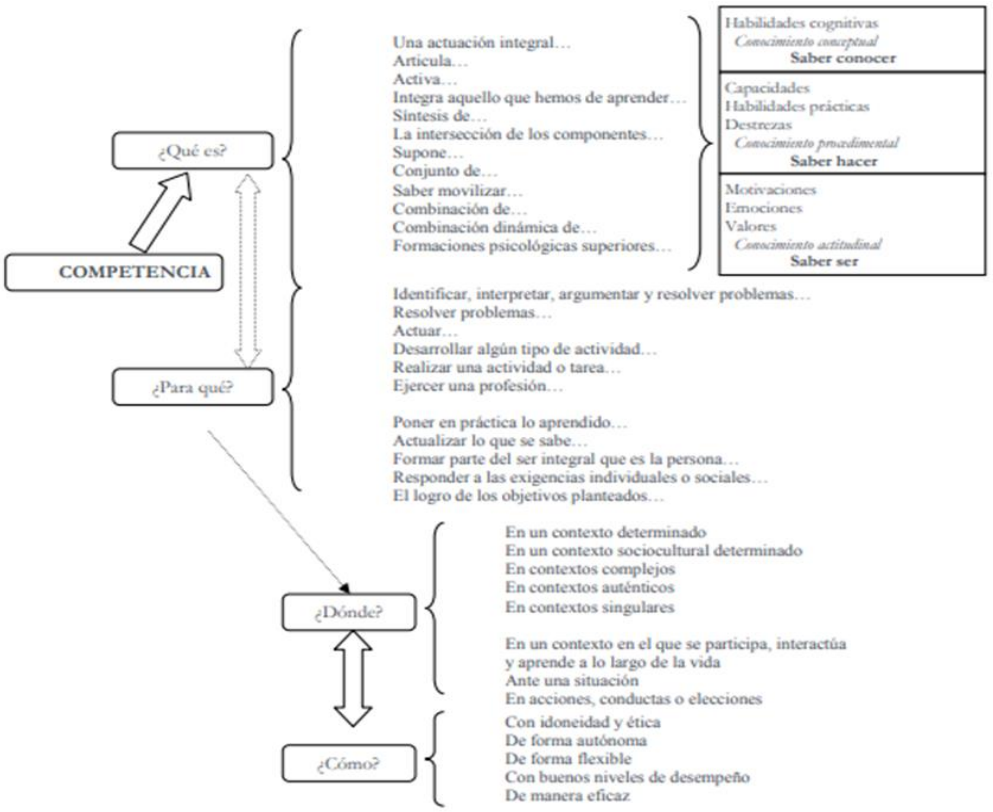
> La integración de conocimientos, habilidades y actitudes que capacitan a una persona para desempeñarse de manera efectiva en una situación. Por lo tanto, las habilidades tienen un componente práctico definido, aunque su utilidad no se limite únicamente a una perspectiva práctica. (Sarramona, 2007, p.32).

Medina (2009, p.13) indica que el crecimiento de habilidades debe abarcar "lo que necesitamos adquirir de conocimiento, cómo debemos implementar y poner en marcha lo aprendido, además de nuestras actitudes, sentimientos, emociones y valores que respaldan el proceso de enseñanza y aprendizaje". En la misma línea, Sevillano (2009, p.7) menciona que las competencias abarcan "valores, actitudes y motivaciones, así como conocimientos, capacidades, habilidades y destrezas, todos los cuales forman parte de la naturaleza humana" y se aplican en un contexto específico, donde la persona se involucra y colabora, identificándose como alguien que está en constante proceso de aprendizaje y mejora a lo largo de su vida.

Zabalza (2003, p.70) afirma que las competencias se refieren al "conjunto de conocimientos y destrezas que las personas requieren para llevar a cabo una actividad específica". Asimismo, López, (2016), en su investigación brinda un mayor alcance sobre la definición de competencia, el cual se puede apreciar con mayor detalle en la Figura 8.

Figura 8

Examinando el contenido de las definiciones de competencia.



Nota. Extraída de López (2016)

Gustos y preferencias

"La preferencia es principalmente un fenómeno conductual basado en las emociones. Para los individuos, este enfoque puede llevar a diversas acciones como dar comentarios favorables, comprar productos, cambiar de religión, etc." (Zajonc y Markus, 1982; Holbrook y Hirschman, 1982; Holbrok y Corfman, 1984).

Los gustos del consumidor se refieren a preferencias personales y subjetivas que determinan las elecciones de consumo de un individuo. Estos gustos están influenciados por factores culturales, sociales y personales, y juegan un papel importante a la hora de determinar qué bienes y servicios se consideran deseables. (Samuelson y Nordhaus, 2010).

Las preferencias de los consumidores cambian con el tiempo debido a factores como la moda, las tendencias y los cambios culturales. Estas preferencias pueden influir en los hábitos de compra y la demanda de determinados bienes y servicios. (Krugman y Wells, 2009).

Métodos de planificación tradicionales

- **Planificación a corto plazo:** Se centra en actividades y decisiones que deben tomarse en el corto plazo, generalmente dentro de un año. Implica asignar recursos y realizar tareas específicas para lograr objetivos inmediatos. (Koontz y Weihrich, 2006).
- **Planificación a mediano plazo:** Centrada en el período de mediano plazo, la planificación a mediano plazo se centra en identificar estrategias y tácticas para lograr las metas establecidas a corto y largo plazo. Puede durar de uno a cinco años. (Koontz y O'Donnell, 1976).
- **Planificación a largo plazo:** Este método implica identificar metas y objetivos a largo plazo, que pueden abarcar varios años o incluso décadas. Se basa en predicciones y pronósticos sobre el futuro y busca establecer una dirección clara para la organización. (Drucker, 1954).

2.2.2. Series de tiempo

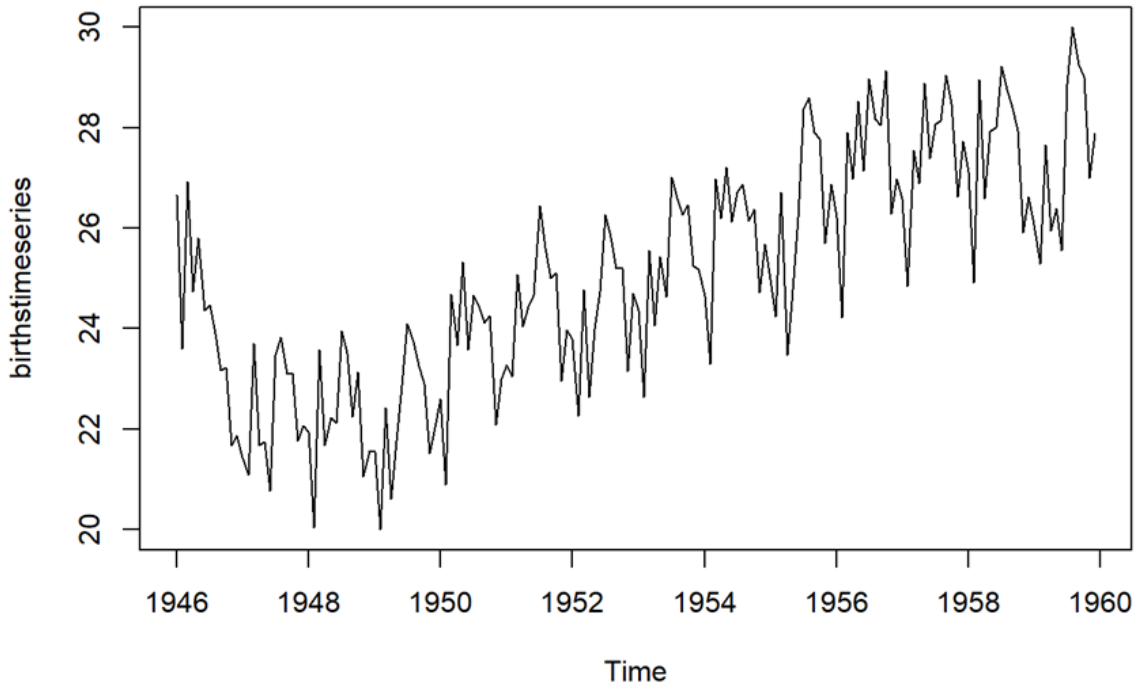
2.2.2.1. Definición

Una serie de tiempo es una secuencia de puntos de datos u observaciones que se recogen y registran a intervalos de tiempo específicos. Estos intervalos pueden ser anuales, mensuales, semanales, diarias, horarias, de minutos e incluso de segundos, dependiendo de la situación. Un ejemplo de gráfica de serie de tiempo se puede observar en la Figura 9.

Asimismo, de acuerdo con Mercado et al. (2015), se define a una serie de tiempo como un conjunto de observaciones, las cuales están ordenadas en el tiempo, además, representa el cambio de una variable que puede ser de tipo económico, físico, biológico, etc. a lo largo de un determinado periodo.

Las series de tiempo se analizan a fin de conocer su patrón de comportamiento, para así poder prever su evolución, teniendo en cuenta que las condiciones no han de variar de forma significativa. Si bien es cierto, el comportamiento de las series de tiempo se puede observar gráficamente en la mayoría de los casos; sin embargo, existen casos en los que no se cumple esta condición, por lo que ciertos movimientos o variaciones características pueden medirse por separado. A dichas variaciones se les conoce como componentes de una serie de tiempo.

Figura 9
Gráfica de una serie de tiempo



Nota. Extraído de Rojas-Jimenez, (2022). Ciencia de Datos para Ciencias Naturales

2.2.2.2. Componentes

2.2.2.2.1. Tendencia

Este componente presenta un movimiento ascendente o descendente constante durante un periodo prolongado. Dichas tendencias pueden ser lineales, no lineales o presentar otros patrones. Como ejemplo se puede mencionar el precio de las acciones de una empresa que aumentan gradualmente a lo largo de los años.

2.2.2.2.2. Estacionalidad

La estacionalidad se produce cuando un fenómeno, el cual ocurre a lo largo del tiempo, tiende a repetirse en cada mismo periodo temporal. (Silva, 2023). Es decir, presenta fluctuaciones regulares a intervalos fijos, que a menudo corresponden a una época en específico, ya sea año, mes, semana, etc.

Por ejemplo, la venta de ciertos productos suele ser estacional, puesto que su demanda se incrementa en determinadas temporadas. Productos tales como los helados, ropa de invierno,

etc., se venden más en ciertas temporadas del año, mientras que el resto del año su venta disminuye.

El análisis de la estacionalidad es importante para poder hacer correlaciones con los datos de otras series de tiempos. Por ejemplo, si se incrementa la demanda del helado, el consumo de madera de palitos de helado también se incrementará.

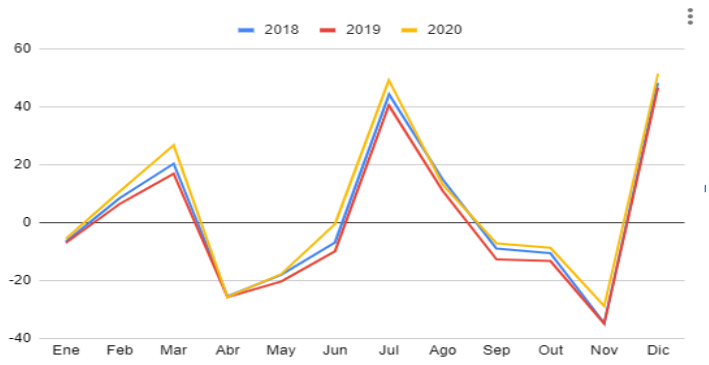
De acuerdo con Silva (2023), existen dos tipos de estacionalidad, los cuales se detallan a continuación:

- **Aditiva**: Una serie de tiempo es aditiva cuando tienen fluctuaciones o variaciones estacionales más o menos constantes, independientemente del nivel global de la serie. (Silva, 2023). Asimismo, no hay variación en la amplitud o frecuencia de las olas, como se puede ver en la Figura 10. Considerando la fórmula matemática, los valores se suman, tal como se puede observar a continuación:

Ecuación 1
Fórmula para una ecuación aditiva

$$y(t) = Nivel + Tendencia + Estacionalidad + Ruido \quad (1)$$

Figura 10
Estacionalidad aditiva



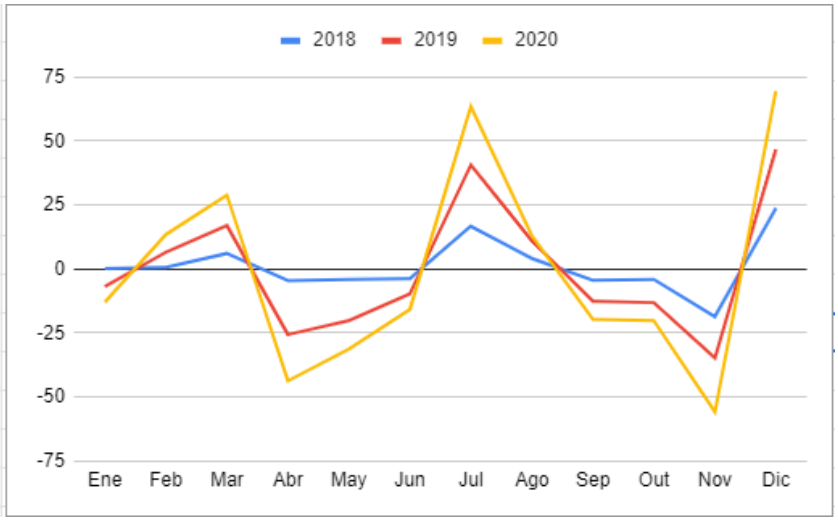
Nota. Extraído de Silva, I. (05 de abril del 2023). Series temporales: Tipos de estacionalidad. *Alura LATAM*. https://www.aluracursos.com/blog/tipos-de-estacionalidad#:~:text=Decimos%20que%20una%20serie%20temporal,mes%20determinado%20a%20cada%20a%C3%B1o.

- **Multiplicativa**: Por otro lado, se dice que una serie de tiempo es multiplicativa cuando el tamaño de sus fluctuaciones estacionales varía según el nivel general de la serie. (Silva, 2023). Asimismo, sí existe variación en la amplitud, pues se multiplica el valor de estacionalidad, como muestra la Figura 11.

Ecuación 2
Fórmula para una ecuación multiplicativa

$$y(t) = Nivel * Tendencia * Estacionalidad * Ruido \quad (2)$$

Figura 11
Estacionalidad multiplicativa



Nota. Extraído de Silva, I. (05 de abril del 2023). Series temporales: Tipos de estacionalidad. *Alura LATAM*. https://www.aluracursos.com/blog/tipos-de estacionalidad#:~:text=Decimos%20que%20una%20serie%20temporal,mes%20determinado%20a%20cada%20a%C3%B1o.

2.2.2.2.3. Residuales

El residual es un componente de la serie de tiempo que recoge las fluctuaciones irregulares que surgen a causa de fenómenos impredecibles, tales como incremento anormal de la demanda de un producto de una empresa, huelgas, desastres, etc.

Son estos valores los que quedan después de eliminar los componentes estacionales, tendenciales y cíclicos de la serie de tiempo. Se calcula empleando el resto de las componentes de la serie y considerando su esquema de composición.

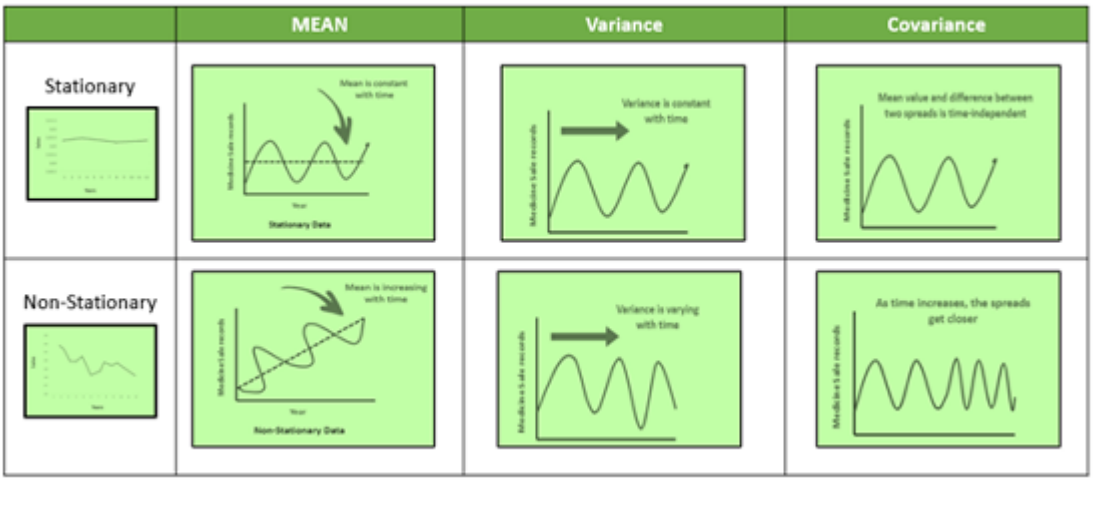
2.2.2.2.4. Estacionariedad

Perkthold et. al. (2023) señalan que la estacionariedad se da cuando las propiedades de una serie de tiempo como la varianza y la covarianza no cambian con el tiempo. Así, dado que algunos modelos requieren que las series sean estacionarias. Al respecto, Shanthababu Pandian (2023) precisa que para que un conjunto de datos sea considerado estacionario debe cumplir con las siguientes pautas generales sin presentar componentes de tendencia, estacionalidad, ciclos o irregularidades en la serie temporal:

- El promedio de los datos debe mantenerse invariable a lo largo del análisis.
- La variación debe permanecer constante en relación al período de tiempo.
- La covarianza se utiliza para cuantificar la relación entre dos variables.

Por el contrario, cuando estas propiedades cambian con el pasar del tiempo, la serie se considera no estacionaria. Esto es importante debido a que, de acuerdo al tipo de estacionalidad, existen modelos que se adaptan mejor. A continuación, en la Figura 12, observamos la diferencia de forma gráfica:

Figura 12
Serie de tiempo estacionaria vs no estacionaria



Nota. Diseñado por Shanthababu Pandian y extraído de Analytics Vidhya. https://av-eks-blogoptimized.s3.amazonaws.com/99388Stationary%20Vs%20Non-Stationary.png

Amat y Escobar (2021) señalan que para determinar si una serie de tiempo es estacionaria, se pueden emplear tres tipos de métodos: inspección de la serie temporal (buscar

una tendencia estacional visualmente), valores estadísticos (comparar valores estadísticos de distintos fragmentos de la serie) y pruebas estadísticas (tests como la prueba Dickey-Fuller aumentada o la prueba Kwiatkowski-Phillips-Schmidt-Shin KPSS).

2.2.2.3. Prueba de Hipótesis

Masaji (2023) define a la Prueba de hipótesis como un proceso estadístico en el que se analiza una teoría o hipótesis correspondiente a una población haciendo uso de datos de muestra. Es decir, se analiza una muestra para evaluar la plausibilidad de una hipótesis de una población.

Parámetros

- **Hipótesis nula (H0):** Se refiere a la afirmación inicial acerca del parámetro de una población, que se asume como verdadera a menos que se disponga de pruebas convincentes que indiquen lo contrario.
- **Hipótesis alternativa (H1):** Es una declaración opuesta a la hipótesis nula y se acepta únicamente si se reúnen pruebas sólidas a su favor, desafiando así la hipótesis nula.
- **Valor p (p-valor):** Es un indicador estadístico que indica la probabilidad de obtener resultados tan extremos o más extremos que los observados, suponiendo que la hipótesis nula sea verdadera. Un valor bajo indica una fuerte evidencia en contra de la hipótesis nula.
- **Nivel de significancia (α):** Es un valor predefinido que determina el punto de corte para decidir si se acepta o rechaza la hipótesis nula. Comúnmente se establece en 0.05, lo que implica que si el valor p es menor que α, la hipótesis nula es rechazada.
- **Estadístico de prueba:** Es un resultado derivado de la información recopilada y se emplea para determinar si la hipótesis nula debe ser descartada. La elección del estadístico de prueba puede variar según el contexto específico.
- **Región crítica:** Se refiere al intervalo de valores del estadístico de prueba en el cual, si este se encuentra dentro de dicho intervalo, conllevará al rechazo de la hipótesis nula.
- **Error tipo I:** Es la equivocación que ocurre al rechazar erróneamente la hipótesis nula cuando es realmente cierta. Está asociado con el nivel de significancia α.
- **Error tipo II:** Es el fallo que se produce al no rechazar la hipótesis nula cuando en realidad es falsa. La probabilidad de incurrir en este error se representa como β.

Interpretación de resultados

Cuando se lleva a cabo una prueba de hipótesis, se analizan la hipótesis nula (H0) y la hipótesis alternativa (H1) utilizando los datos recopilados. Si el valor p calculado es inferior al nivel de significancia (α) previamente establecido, generalmente 0.05, se descarta la hipótesis nula en favor de la hipótesis alternativa. Esto indica que hay pruebas contundentes que respaldan la afirmación de la hipótesis alternativa. Por el contrario, si el valor p es mayor que α, no hay suficiente evidencia para rechazar la hipótesis nula, lo que sugiere que los datos no respaldan la hipótesis alternativa. La interpretación final depende de la relación entre el valor p y el nivel de significancia, y esto determina si se acepta o rechaza la hipótesis nula, influenciando en última instancia las decisiones basadas en los resultados de la prueba.

2.2.2.3.1. Prueba de Augmented Dickey Fuller (ADF)

Según Amat y Escobar (2021), esta prueba se basa en la suposición de que la serie posee una raíz unitaria, lo que implica que no es estacionaria. Es una evaluación de la presencia de una raíz unitaria, donde la hipótesis nula postula que α=1. (Prabhakaran, 2019):

Si la hipótesis nula asume que hay una raíz unitaria (α=1), entonces el valor p obtenido debe ser menor que el nivel de significancia (por ejemplo, 0,05) para rechazar dicha hipótesis. Esto sugiere que la serie es estacionaria. (Prabhakaran, 2019).

Ecuación 3
Ecuación de la Prueba Augmented Dickey Fuller (ADF)

$$y_t = c + \beta t + \alpha y_{t-1} + \Phi_1 \Delta Y_{t-1} + \Phi_2 \Delta Y_{t-2} + \ldots + \Phi_p \Delta Y_{t-p} + e_t \quad (3)$$

Donde:

$\boldsymbol{y_t}$**:** el valor de la serie de tiempo

$\boldsymbol{c}$**:** constante

$\boldsymbol{\beta}$**:** coeficiente que multiplica el tiempo t y modela una tendencia lineal en la serie de tiempo

α**:** coeficiente que representa el efecto de la observación anterior

$\boldsymbol{\Phi_1, \Phi_2, \ldots, \Phi_p}$**:** coeficientes que modelan la dependencia de la serie con las variaciones de sus valores pasados

$\boldsymbol{\Delta Y_{t-1}, \Delta Y_{t-2}, \ldots, \Delta Y_{t-p}}$**:** diferencia en el valor de la serie de tiempo t-1, t-2, etc.

e_t**:** término del error, que representa la parte estocástica o aleatoria de la serie de tiempo

Posterior a la verificación de hipótesis se concluye lo siguiente:

- Hipótesis nula (H0): La serie tiene una raíz unitaria, no es estacionaria.
- Hipótesis alternativa (H1): La serie no tiene raíz unitaria, es estacionaria.

2.2.2.3.2. Prueba Prueba de Kwiatkowski-Phillips-Schmidt-Shin (KPSS)

Amat y Escobar (2021) mencionan que la prueba KPSS evalúa si una serie de tiempo exhibe estacionariedad alrededor de una media o una tendencia lineal. En este análisis, se plantea como hipótesis nula que la serie es estacionaria, por lo tanto, cuando los valores de p son bajos, como, por ejemplo, inferiores a 0.05, se rechaza la hipótesis nula, lo que indica la necesidad de aplicar técnicas de diferenciación. Perktold et. al. (2023), por su lado, señalan que en esta prueba la hipótesis nula y alternativa son opuestas a las de la prueba ADF. Posterior a la verificación de hipótesis se concluye lo siguiente:

- Hipótesis nula (H0): El proceso es estacionario en tendencia.
- Hipótesis alternativa (H1): La serie tiene raíz unitaria (la serie no es estacionaria).

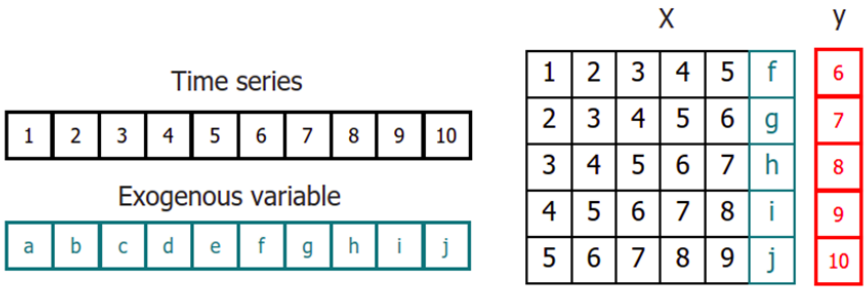
2.2.2.4. Predicciones multi-step

El proceso de forecasting implica predecir resultados futuros de una serie temporal (ST). Existen dos formas: primero modelando la ST en base a su comportamiento anterior o autorregresivo y la segunda consiste en utilizar otras variables externas.

Asimismo, para realizar el entrenamiento del modelo de forecasting, es necesario transformar la ST en una matriz en la que cada valor se asocia a la ventana temporal (lags) que le precede. Adicionalmente, también se puede incluir variables exógenas a la ST. En la Figura 13, la cual se muestra a continuación, se puede observar con más detalle lo anteriormente explicado.

Figura 13

Transformación de una serie temporal junto con una variable exógena.



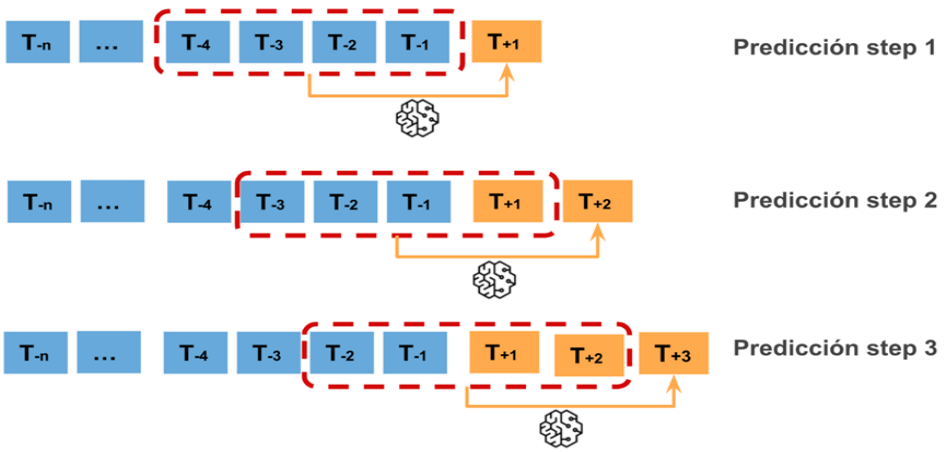
Nota. Extraído de https://cienciadedatos.net/images/matrix_transformation_with_exog_variable.png

En el análisis de series de tiempo, por lo general se busca predecir un intervalo futuro o un punto alejado en el tiempo. Cada paso de predicción es conocido como step. Existen varias estrategias que permiten realizar predicciones:

- **Recursive multi-step forecasting (RMF):** Según Amat y Escobar (2021), para predecir el momento tn - 1 es necesario contar con el valor de tn−1, para lo cual se realiza un proceso en el cual cada nueva predicción hace uso del valor de predicción anterior (también conocido como recursive forecasting o recursive multi-step forecasting). A continuación, se presenta la Figura 14, en la cual se puede observar el diagrama del proceso de predicción multi-step recursivo.

Figura 14

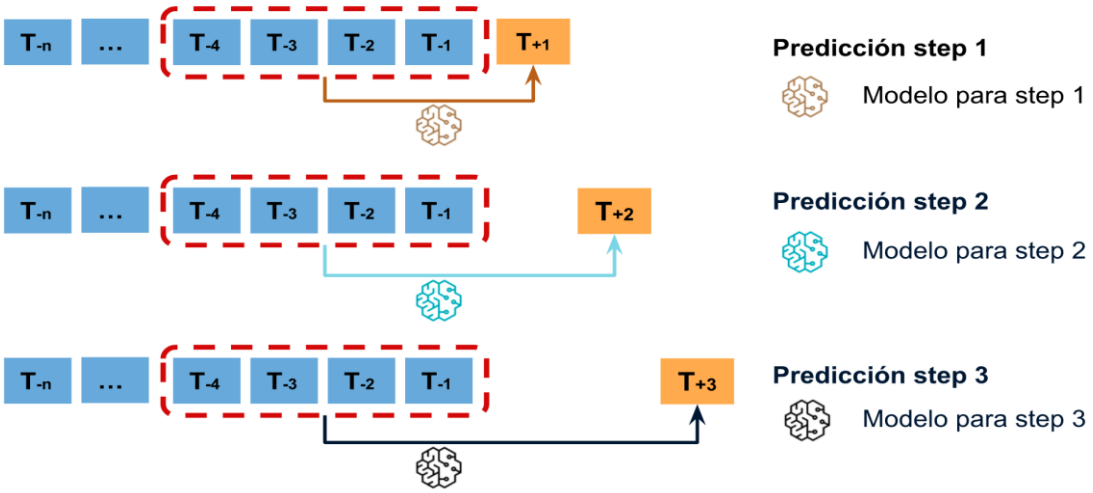
Diagrama del proceso de predicción multi-step recursivo



Nota. Extraído de https://cienciadedatos.net/images/diagrama-multistep-recursiva.png

- **Direct multi-step forecasting (DMF):** En este método, indican Amat y Escobar (2021), se entrena un modelo distinto para cada etapa o step de modo que cada predicción es independiente de las demás. Es decir, si se quiere predecir *n* valores de una serie, se deberán entrenar *n* modelos, como se muestra en la Figura 15.

Figura 15
Diagrama del proceso de predicción direct multi-step



Nota. Extraído de https://cienciadedatos.net/images/diagrama-prediccion-multistep-directa.png

Básicamente, la diferencia entre RMF y DMF radica en que para el primer caso se genera una predicción a la vez tomando en cuenta tanto el comportamiento histórico como las predicciones reales anteriores mientras que para la segunda forma se genera todas las predicciones a la vez sin tomar en cuenta sin depender de las últimas predicciones. No obstante, la principal desventaja de usar el DMF es que es más complejo de implementar debido, consume mayores recursos y presenta sensibilidad a errores iniciales, lo que quiere decir que un error en la predicción inicial puede propagarse a lo largo de todo el horizonte de pronóstico, lo que puede llevar a predicciones menos precisas si no se abordan adecuadamente.

Por último, cabe señalar que la selección de una u otra técnica dependerá de la naturaleza de los datos, los objetivos del pronóstico y las características específicas del problema. Cada enfoque tiene sus propias ventajas y desventajas, y la elección implica analizar cuál se ajusta mejor a las necesidades del usuario y a la idoneidad del modelo de pronóstico.

2.2.2.3. Modelos predictivos

En este apartado se procederá a describir los principales modelos usados en el mercado para predecir la demanda. Para el caso de las regresiones, la explicación será muy general debido a que más adelante en la sección de Machine Learning se profundizará en ello.

2.2.2.3.1. Regresiones

La regresión es una técnica estadística que permite cuantificar la relación entre dos o más variables y predecir el valor de la variable dependiente en función del valor de las variables independientes. No obstante, no sólo explica la relación entre estas variables, sino que también permite cuantificar cómo la variable dependiente cambia cuando cambian las variables independientes.

2.2.2.3.2. Media móvil simple

La media móvil simple es una de las técnicas de análisis de series de tiempo. La Media Móvil (MA) consiste en realizar pronósticos en base a los errores de previsión pasados, cada valor depende de los valores anteriores y de los errores previos. Estos modelos determinan la relación entre el valor actual y los errores de previsión anteriores.

El modelo de media móvil es similar al método de regresión con la diferencia de que se realiza el ajuste de coeficientes θ a los errores pronosticados previamente denominados ε, o conocido también como error de ruido blanco. Asimismo, se adiciona un término constante que es la media (μ) y q corresponde al número de términos de error, o conocido como orden. A continuación, se presenta su expresión matemática:

Ecuación 4
Media Móvil

$$y_t = \mu + \epsilon_t + \theta_1\epsilon_{t-1} + \theta_2\epsilon_{t-2} + \ldots + \theta_q\epsilon_{t-q} \quad (4)$$

Donde:

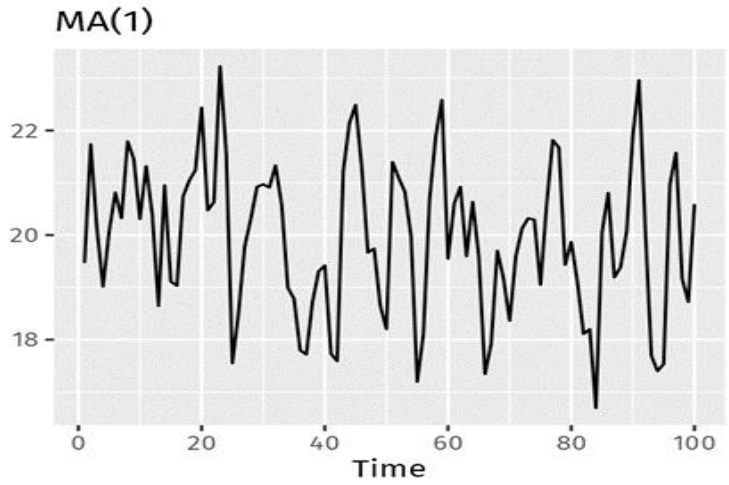
$\boldsymbol{y_t}$ **:** valor de la serie de tiempo en el tiempo t, es decir, la observación actual en la serie.
$\boldsymbol{\mu}$ **:** media de la serie de tiempo y representa el valor promedio de la serie a lo largo del tiempo.

ϵ_t**:** término de error en el tiempo t y representa la parte estocástica o aleatoria de la serie en ese momento.

$\theta_1,\theta_2,...,\theta_p$**:** coeficientes que representan la influencia de los errores pasados en el valor actual de la serie.

Cabe destacar que, un requisito indispensable del modelo MA es que requiere que los datos sean estacionarios, por ende, implica que la varianza y la media deben ser constantes a lo largo del tiempo. Al respecto, es posible estabilizar la varianza mediante una transformación logaritmo o Box-Cox. En la Figura 16, presentada a continuación, se puede observar un modelo de gráfica de Media Móvil.

Figura 16
Modelo de Media Móvil (MA)



Nota. Elaboración propia.

2.2.2.3.3. SARIMAX

El Modelo SARIMAX (Auto Regresivo Integrado de Medias Móvil Estacional con regresores exógenos) es una extensión del Modelo ARIMA (Media Móvil Integrada Autorregresiva) pero que además de tener todas las bondades de este último, también tiene la capacidad de incorporar información acerca de variables exógenas o externas que contribuyen a la comprensión y predicción de la variable principal de interés. (Sandoval, 2022). De este modo tiene la capacidad de modelar tanto la estacionalidad como los factores externos, lo que lo convierte en un modelo más flexible y adecuado para ciertos tipos de previsión de series temporales. Para una única variable x y una única variable y, matemáticamente su relación se define bajo la siguiente fórmula:

Ecuación 5
Relación de x,y en el Modelo SARIMAX

$$yt = \beta 1xt + \cdots + \beta kxtk_{-1} + \phi 1yt - 1 + \cdots + \phi pyt - p + \theta 1zt - 1 + \cdots \theta qzt - q + zt \quad (5)$$

Donde:

x: es una variable exógena

z: es un ruido blanco

Asimismo, de acuerdo con Joaquín Amat y Javier Escobar (2023), este modelo se conforma de tres componentes:

1. **Elemento autorregresivo (AR):** establece una relación entre el valor actual y los valores anteriores (retrasos o lags).
2. **Elemento de media móvil (MA):** asume que el error de predicción es una suma ponderada de los errores de predicción previos.
3. **Componente integrado (I):** señala que los valores de la serie original han sido transformados en la diferencia entre valores consecutivos.

También cabe señalar que en estos modelos se maneja el concepto de orden que básicamente se refiere a la cantidad de componentes autorregresivos (AR), de media móvil (MA) y de diferenciación (I) que se usan en el modelo. Estos órdenes especifican cuántos pasos en el tiempo hacia atrás se consideran al modelar una serie temporal. Es así que la fórmula para un modelo SARIMAX, a nivel de orden, es la siguiente:

Ecuación 6
Modelo SARIMAX

$$SARIMAX(p, d, q) * (P, D, Q, s) \quad (6)$$

Donde (Edel, 2020):

p: el orden de la parte autorregresiva de la serie temporal.
d: el grado de diferenciación de la serie temporal (el número de veces que se han restado los valores consecutivos para lograr la estacionariedad).
q: el orden de la parte de promedios móviles de la serie temporal.
P: el orden de la parte autorregresiva estacional.

D: el grado de diferenciación estacional (el número de veces que se han restado los valores en el mismo período de estacionalidad para lograr la estacionariedad).
Q: el orden de la parte de promedios móviles estacionales.
s: el número de periodos en cada temporada o ciclo estacional.

Cabe señalar que cuando los valores de P, D, Q y m son nulos, y no se incorporan variables exógenas, el modelo SARIMAX se convierte en un ARIMA equivalente. (Amat y Escobar, 2023).

Finalmente, cabe precisar que SARIMAX constituye una versión avanzada de ARIMA y ofrece las siguientes ventajas:

- Posee la capacidad de manejar series de tiempo con patrones estacionales y no estacionales, a través de la incorporación de términos autorregresivos estacionales y de medias móviles.
- Permite incluir variables exógenas con el fin de detectar las influencias externas en las series de tiempo.
- Provee un marco más completo para realizar las predicciones, más aún cuando los datos presentan estacionalidad y factores externos.

2.2.3. Machine Learning

Machine Learning, según Harrington (2012), se define como el proceso de convertir datos en información mediante la aplicación de técnicas computacionales que permiten aprender patrones y tomar decisiones basadas en estos patrones. En la mayoría de los casos, la información deseada no es evidente a simple vista en los datos brutos, y es necesario utilizar algoritmos y modelos de Machine Learning para extraer conocimiento significativo.

Esta disciplina, continúa el autor (Harrington, 2012), se sitúa en la intersección de la informática, la ingeniería y la estadística, y tiene aplicaciones en una amplia variedad de campos, desde la detección de correo no deseado (spam) hasta la geociencia y la política. El Machine Learning utiliza principios estadísticos para abordar problemas en los que la solución no es determinista y no puede ser modelada de manera completa debido a la complejidad del problema o la falta de datos suficientes.

En muchas disciplinas, como las ciencias sociales, no siempre es posible predecir los resultados con certeza absoluta, y alcanzar una tasa de precisión a partir del 70% se considera

un éxito. Esto se debe a la naturaleza compleja y variable de muchos problemas, como el comportamiento humano, que puede ser difícil de modelar determinísticamente debido a las diferencias individuales en las preferencias y valores de las personas.

2.2.3.1. Relación con Inteligencia Artificial

Según Alzubi et. al. (2018), el Machine Learning es una categoría de la Inteligencia artificial. La inteligencia artificial es un campo más amplio que engloba diversas técnicas y enfoques para permitir que las computadoras realicen tareas que de otro modo requerirían la intervención humana y muestren ciertos niveles de inteligencia en su funcionamiento.

En este contexto, el Machine Learning se considera una subdisciplina de la inteligencia artificial que se centra en la capacidad de las computadoras para aprender y mejorar su desempeño en una tarea específica a través de la experiencia o los datos, sin necesidad de una programación explícita. En otras palabras, el Machine Learning es una forma específica en la que la inteligencia artificial se pone en práctica, permitiendo que las computadoras aprendan a partir de datos y experiencias previas para realizar tareas con mayor precisión.

2.2.3.2. Tipos de Aprendizaje

2.2.3.2.1. Aprendizaje no supervisado

Jung (2022) explica que los métodos de aprendizaje no supervisados en Machine Learning se caracterizan por su capacidad para aprender patrones y estructuras intrínsecas en los conjuntos de datos sin requerir el conocimiento de los valores de etiquetas asociados a cada punto de datos. En este enfoque, no es necesario contar con un maestro o un experto en el dominio que proporcione etiquetas para formar un conjunto de entrenamiento. En particular, Jung (2022) menciona que existen dos grandes familias de métodos de aprendizaje no supervisados:

- **Métodos de clustering**: Su principal tarea es agrupar los puntos de datos en conjuntos o clústeres. La característica fundamental de estos métodos es que los puntos de datos dentro de un mismo clúster deben ser más similares entre sí que con los puntos de datos que se encuentran fuera del clúster.
- **Feature learning**: Se encarga de determinar características numéricas de los datos de manera eficiente. Dos aplicaciones clave de estos métodos son la reducción de dimensionalidad y la visualización de datos. En la reducción de dimensionalidad, se

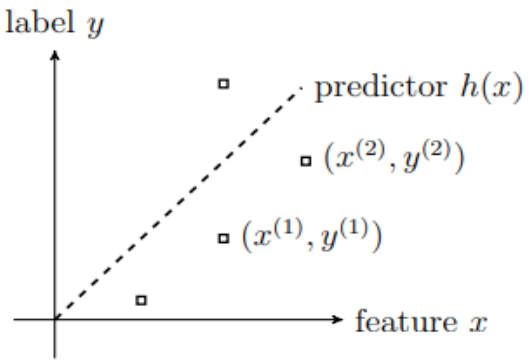
busca representar los datos en un espacio de menor dimensión, lo que ayuda a simplificar y comprimir la información sin perder su esencia. En cuanto a la visualización de datos, se busca encontrar representaciones visuales que permitan comprender mejor la estructura y las relaciones entre los datos.

2.2.3.2.2. Aprendizaje supervisado

Jung (2022) indica que los métodos de aprendizaje supervisados en Machine Learning se centran en la tarea de desarrollar modelos predictivos basados en un conjunto de datos de entrenamiento etiquetados. Cabe precisar que se considera que un punto de datos está etiquetado cuando se conoce su valor de etiqueta o clase asociado y que son asignados a través de humanos expertos.

La tarea fundamental de estos métodos es buscar un modelo que pueda imitar al anotador humano y permita predecir la etiqueta de un punto de datos exclusivamente a partir de sus características o atributos. Gráficamente, como se muestra en la Figura 17, estos métodos buscan ajustar una curva (el gráfico del mapa de predicción) a los puntos de datos etiquetados en el conjunto de entrenamiento, minimizando la discrepancia entre las predicciones del modelo y las etiquetas verdaderas.

Figura 17
Representación Gráfica de Modelo predictor de etiquetas



Nota. Extraído de Jung, Alexander. (2022, p. 30). *Machine Learning: The Basics. Springer*. DOI: https://doi.org/10.1007/978-981-16-8193-6.

Para llevar a cabo este ajuste de curva, se utiliza una función de pérdida (*loss function*) que cuantifica el error de ajuste. Cabe destacar que los métodos de aprendizaje supervisado

pueden diferir en la elección de la función de pérdida utilizada para medir la discrepancia entre la etiqueta predicha y la etiqueta verdadera de un punto de datos, según menciona Jung (2022).

A pesar de que el principio detrás de los métodos de aprendizaje supervisado pueda parecer sencillo, el desafío en las aplicaciones modernas de Machine Learning radica en la gran cantidad de puntos de datos y su complejidad. Estos métodos deben ser capaces de procesar miles de millones de puntos de datos, y cada punto de datos puede estar caracterizado por un número potencialmente vasto de características. Además, menciona Jung (2022), otro desafío es que estos métodos deben ser capaces de ajustar mapas predictivos altamente no lineales. Para abordar esta dificultad, los métodos de aprendizaje profundo (deep learning) utilizan representaciones computacionalmente convenientes de mapas no lineales a través de redes neuronales artificiales (artificial neural networks).

2.2.3.3. Modelos supervisados

Los modelos de Machine Learning, según Jung (2022), son sistemas que buscan aprender patrones y relaciones en los datos para realizar predicciones o clasificaciones sin ser programados explícitamente. Estos modelos se basan en datos previamente etiquetados o etiquetables y se entrenan para desarrollar una hipótesis o regla que relaciona las características de entrada con las etiquetas de salida.

Estos sistemas, continúa Jung (2022), funcionan mediante el aprendizaje de una función o hipótesis que mapea las características de entrada (denotadas como x) a las etiquetas de salida (denotadas como y). Este mapeo se expresa como una función h(x) que estima y a partir de x. El objetivo es encontrar una hipótesis (h) que minimice la diferencia entre sus predicciones y las etiquetas verdaderas de los datos de entrenamiento. Esto se logra ajustando la hipótesis para que se adapte mejor a los datos de entrenamiento.

El uso de funciones y algoritmos para aprender la relación entre las características de entrada y las etiquetas de salida constituyen la parte matemática de los modelos de Machine Learning. Un componente esencial es la función de pérdida (loss function), que cuantifica el error entre las predicciones del modelo y las etiquetas verdaderas. La fórmula general para la función de pérdida es expresada por Jung (2022) como:

Ecuación 7
Fórmula general de función de Pérdida

$$L\ (y, h(x)) \qquad (7)$$

Donde:

L **:** función de pérdida.

y**:** etiqueta verdadera.

$\boldsymbol{h(x)}$**:** predicción del modelo.

2.2.3.3.1 Regresión lineal

Considerando un modelo en el que x es el valor correspondiente a la variable independiente e y el de la variable dependiente, el modelo de regresión simple se representa de acuerdo con la siguiente ecuación:

Ecuación 8
Regresión lineal simple

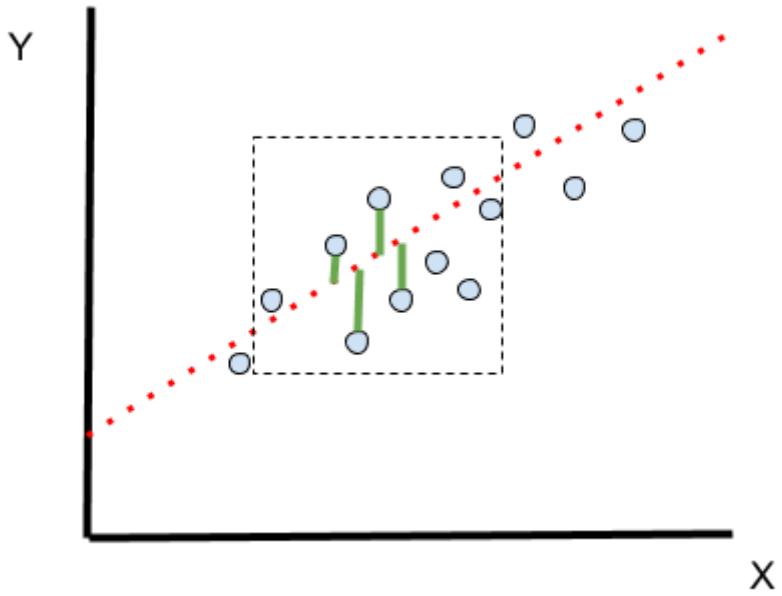
$$Función\ (y) = a + bx \qquad (8)$$

Donde:

y: variable independiente
x: variable dependiente
a: término constante o intercepto
b: coeficiente de la pendiente

A nivel gráfico, dicha relación es presentada en la Figura 18:

Figura 18
Regresión lineal simple



Nota. Extraído de
https://miro.medium.com/v2/resize:fit:846/1*B_z1FIcNre_qgbNJ4gf3Vw.png

Cabe precisar que, de acuerdo a lo mencionado por Zamora, M. & Rending, A. (2011), la regresión lineal debe cumplir de forma obligatoria con cinco (05) supuestos con el fin de realizar inferencias estadísticas a partir de las muestras analizadas, estos supuestos son: Normalidad, Independencia, Homocedasticidad y linealidad. A continuación, se explica cada uno de ellos:

a) **Normalidad:** Los errores exhiben una distribución normal con una media de cero y una varianza (σ^2) igual a 1; es decir, constante. En otras palabras, los valores de Y se comportan de acuerdo a una distribución normal. En el caso de que este supuesto no se cumpla, antes de desarrollar un modelo de regresión, es posible evaluar la opción de aplicar una transformación a la variable Y para que la nueva variable se asemeje más a una distribución normal.
b) **Independencia:** Esto significa que dos observaciones diferentes, como los errores ε_i y ε_j, no tienen una relación estadística entre sí, lo que implica que el valor de un error no guarda ninguna relación con el valor de otro error. Como resultado, los valores de Y seleccionados de una muestra y los valores específicos de X proporcionados son

igualmente independientes. No obstante, es importante señalar que este supuesto puede no mantenerse cuando se obtienen diferentes observaciones en un mismo individuo en momentos distintos. Por ejemplo, si se toma el peso de una persona en diferentes momentos, es probable que exista una correlación entre los pesos de esa misma persona. Cuando este supuesto no se cumple, las conclusiones estadísticas pueden carecer de validez

c) **Homocedasticidad:** Este supuesto indica que la variabilidad en los errores ε_i es uniforme y constante, lo que resulta en que la varianza de Y se mantiene invariable para diferentes valores fijos de X.

d) **Lineal:** Este supuesto establece que, una vez que conocemos los valores constantes de X, las medias de Y siguen una relación lineal. Esto se expresa de manera simbólica a través de la ecuación $Y/X = \beta_0 + \beta_1 X$, en la que β_0 representa el punto en el que la media de la variable de respuesta cruza el eje Y cuando la variable explicativa X tiene un valor de cero. Sin embargo, la interpretación de β_0 no tiene sentido cuando los valores de la variable explicativa analizada no incluyen el valor cero. Por su parte, β_0 representa la pendiente de esta línea.

Respecto a las regresiones simples más usadas, existen tres tipos principales:

- **Modelos lineales:** Para el caso de variables continuas, se aplica el modelo de regresión lineal simple y la función de y es su media aritmética. La ecuación es la misma antes explicada.

Ecuación 9
Modelo Lineal

$$y = a + bx \quad (9)$$

Donde

y: valor de la predicción del modelo de regresión

x: valor de la variable independiente

a: constante

b: constante que indica el impacto de x en la predicción

- **Modelos de regresión logística:** Este modelo se usa en caso de que la variable dependiente sea cualitativa binaria. La ecuación que representa la recta de regresión de la siguiente forma:

Ecuación 10
Modelo regresión logística

$$y = \frac{1}{1 + e^{-x}} \qquad (10)$$

Donde:

y: predicción del modelo

x: valor de la variable independiente

2.2.3.3.2. Regresión lineal múltiple

Para Cristina Dawson (2021), la regresión lineal múltiple (MLR) es la técnica de análisis de regresión más frecuente y relevante, empleada para anticipar el valor de una variable en función de dos o más variables independientes (también llamadas variables explicativas). En tal sentido, su objetivo principal es modelar la relación lineal entre estas variables independientes y la variable dependiente que se desea analizar. (Maulud y Abdulazeez, 2020). Asimismo, señalan que esta estimación se realiza mediante la siguiente ecuación lineal:

Ecuación 11
Regresión Lineal Múltiple

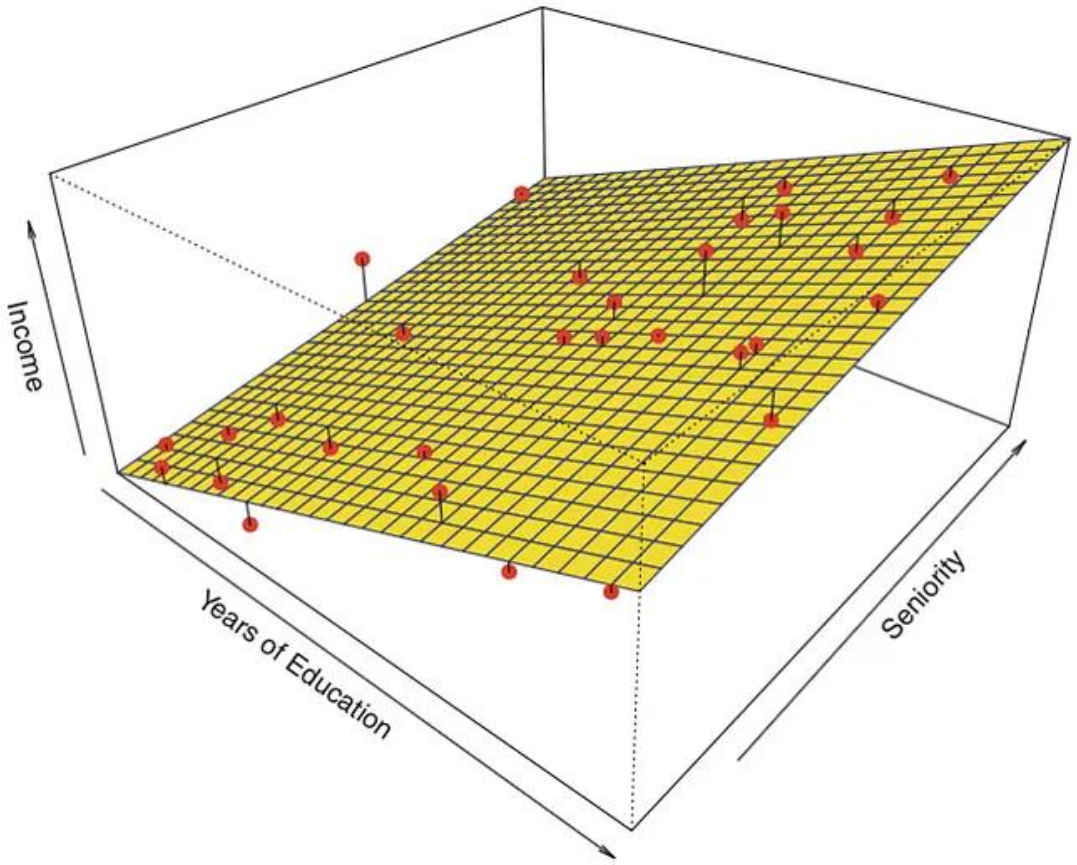
$$y = \beta_0 + \beta_1 x + \ldots + \beta_n x + \epsilon \qquad (11)$$

Donde:

$\boldsymbol{y}$**:** variable dependiente, la cual se desea predecir.
$\boldsymbol{\beta_0}$**:** intercepto o término constante.
$\boldsymbol{\beta_1 - \beta_n}$**:** coeficientes de regresión que representan la contribución de cada variable independiente.
$\boldsymbol{x}$**:** variable independiente.
$\boldsymbol{\epsilon}$**:** término de error que tiene en cuenta las variaciones no explicadas por la relación lineal.

A nivel gráfico, podemos observar esta relación con la Figura 19:

Figura 19
Regresión lineal múltiple



Nota. Extraído de
https://miro.medium.com/v2/resize:fit:720/format:webp/0*qq0yaecNRQiugnif.png

La MLR, según Maulud y Abdulazeez (2020), es especialmente útil cuando se necesita considerar múltiples variables independientes para comprender mejor la relación entre estas variables y la variable dependiente. Permite cuantificar cómo cada variable independiente contribuye a la variación en la variable dependiente.

Al igual en el caso de la regresión lineal simple, Dawson (2021) explica que la regresión múltiple también tiene algunos supuestos que debe cumplir:

- **Homogeneidad de la varianza u homocedasticidad:** esto implica que asumimos que la magnitud del error en nuestras predicciones permanece relativamente constante en diferentes valores de la variable independiente.

- **Independencia de las observaciones:** esta premisa establece que las observaciones en nuestro conjunto de datos son independientes unas de otras y se han obtenido utilizando métodos estadísticamente apropiados. Cuando dos o más variables independientes están fuertemente relacionadas se conoce como multicolinealidad y de no tratarlas se tendría información redundante que podría sesgar los resultados.
- **Normalidad:** damos por sentado que nuestros datos siguen una distribución normal.
- **Linealidad:** presupone que la relación entre las variables independientes y la variable dependiente sigue un patrón lineal.

Regresores

Según Darlington & Hayes (2016) los "regresores" son todas las variables que se utilizan para analizar la relación entre las variables independientes y la variable dependiente, sin hacer una distinción matemática entre variables independientes y covariables. No obstante, en este caso nos referimos como regresores a los modelos o técnicas que se aplican a problemas de regresión. A continuación, explicaremos algunos de los más usados:

Gradient Boosting

Según lo expresado por Abhiroop Choudhury en 2020, el incremento del gradiente se presenta como un algoritmo de potenciación que se fundamenta en la idea de que el próximo modelo más efectivo, al ser amalgamado con los modelos previos, reduce al mínimo el error de predicción global. Para Burrueco (s.f.), se trata de una categoría de algoritmos utilizados tanto en la tarea de clasificación como en la de regresión. Estos algoritmos se caracterizan por su enfoque en combinar modelos predictivos débiles, como los árboles de decisión, con el fin de crear un modelo predictivo robusto. En este proceso, los árboles de decisión débiles se generan de manera secuencial, y cada nuevo árbol se forma para corregir los errores de los árboles previamente construidos. Este procedimiento da como resultado la creación de árboles "poco profundos" con alrededor de dos o tres niveles de profundidad. Un parámetro relevante que se utiliza en este contexto es la tasa de aprendizaje, que mide el grado de mejora de un árbol en comparación con su predecesor.

Choudhury (2020) señala que este tipo de modelo requiere tres elementos:

- Una métrica de error que se busca minimizar.
- Un predictor ineficiente para realizar pronósticos.

- Un enfoque aditivo para combinar los predictores ineficientes y reducir la métrica de error.

A nivel matemático, la idea general consiste en entrenar modelos de forma secuencial, de modo que cada modelo ajusta los errores de los anteriores. Se ajusta un primer weak learner f_1 con el cual se predice la variable respuesta y , como se aprecia a continuación:

Ecuación 12
Modelo Weak learner

$$f_1(x) \approx y \qquad (12)$$

Donde:

$\boldsymbol{f_1(x)}$**:** primer modelo, weak learner
$\boldsymbol{y}$**:** variable de respuesta

Seguidamente se ajusta un nuevo modelo f_2 para predecir los errores del modelo anterior $y - f_1(x)$, es decir, se intenta corregir los errores del modelo f_1 .

Ecuación 13
Modelo de predicción de errores de Weak learner

$$f_2(x) \approx y - f_1(x) \qquad (13)$$

Donde:

$\boldsymbol{f_2(x)}$**:** nuevo modelo
$\boldsymbol{y - f_1(x)}$**:** error del modelo anterior

El siguiente paso es calcular los errores de los dos modelos de form conjunta $y - f_1(x) - f_2(x)$ y se ajusta el tercer modelo f_3 . Este procedimiento se repite M veces, asegurando que los modelos subsiguientes reduzcan los errores previos. Sin embargo, existe la posibilidad de enfrentar el desafío del sobreajuste, que se manifiesta cuando el modelo se ajusta en exceso a los datos de entrenamiento y, por ende, no puede realizar predicciones precisas en nuevas observaciones. Afortunadamente, este problema se aborda utilizando un valor de regularización, también conocido como tasa de aprendizaje (λ), lo que da como resultado el siguiente desenlace:

Ecuación 14
Learning rate.

$$y \approx \lambda f_1(x) + \lambda f_2(x) + \lambda f_3(x) + \ldots + \lambda f_m(x) \quad (14)$$

Donde:

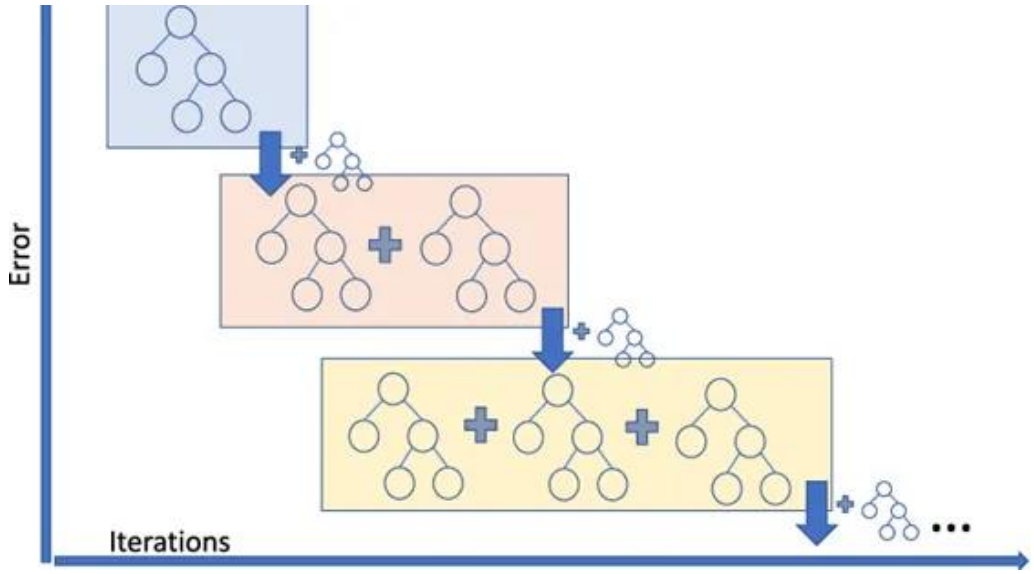
$\boldsymbol{y}$**:** variable de respuesta

$\boldsymbol{\lambda}$ **:** learning rate

$\boldsymbol{f_1(x)}$, $\boldsymbol{f_2(x)}\ldots$ $\boldsymbol{f_m(x)}$ **:** modelos

Como se mencionó anteriormente, este modelo a medida que avanza va corrigiendo los errores generados por los árboles anteriores. Esta relación se puede apreciar a través de la Figura 20, en la cual se ejemplifica gráficamente dicho proceso.

Figura 20
Gradient Boosting



Nota. Extraído de https://miro.medium.com/v2/resize:fit:640/format:webp/1*85QHtH-49U7ozPpmA5cAaw.png

Cabe destacar que, el algoritmo Gradient Boosting suele presentar mejores resultados en escenarios tabulares. Asimismo, se destacan las implementaciones de LightBGM y XGBoost. Con respecto a XGBoost, constituye una versión optimizada de Gradient Boosting. Esta optimización se diseñó específicamente para ofrecer un alto rendimiento y es utilizado frecuentemente en competiciones de ciencia de datos. Se caracteriza por emplear una técnica

de regularización a fin de prevenir el sobreajuste y, asimismo, es capaz de manejar grandes conjuntos de datos y datos de alta dimensionalidad.

Regresor Ridge

Esta técnica de regularización impone una penalización al elevar al cuadrado los coeficientes, lo que resulta en una reducción proporcional de todos los coeficientes del modelo, según señala Joaquín Amat (2020). Sin embargo, esta reducción no llega a hacer que los coeficientes sean cero.

Esta estrategia proporciona varias ventajas, siendo la más destacada la reducción de la variabilidad. Sin embargo, presenta una desventaja significativa: el modelo resultante asigna cierto grado de importancia a todos los valores. Esto se debe a que, a pesar de que la penalización incentiva a los coeficientes a acercarse a cero, nunca los lleva completamente a cero.

En otras palabras, según la explicación de Joaquín Amat en 2022, "Este enfoque logra minimizar la influencia de los predictores menos relevantes en el modelo final en relación a la variable de interés, aunque estos predictores aún mantienen su presencia. Aunque esto no impacta en la precisión del modelo, sí puede complicar su interpretación".

Ecuación 15
Función de costo

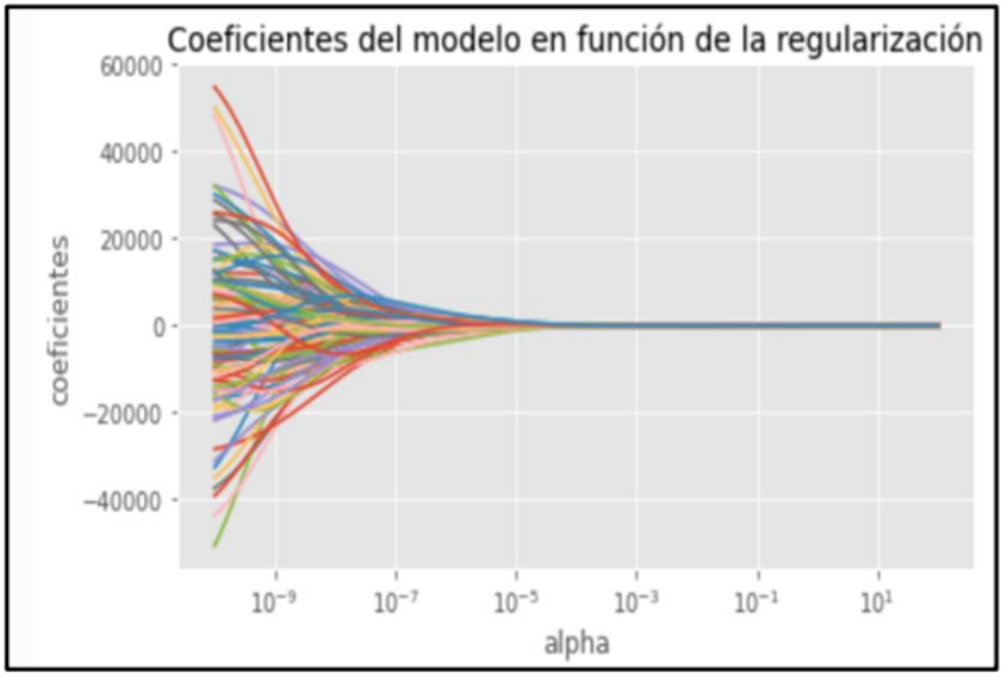
$$J(\theta) = MSE + \alpha \sum i = In\theta i2 \quad (15)$$

Donde:

$\boldsymbol{J(\theta)}$ **:** Es la función de costo.
$\boldsymbol{MSE}$ **:** Es el Error Cuadrático Medio, que mide la diferencia entre las predicciones del modelo y los valores reales.
$\boldsymbol{\alpha}$ **:** Es un hiper parámetro que controla la fuerza de la regularización. Un valor más grande α significa una regularización más fuerte.
$\boldsymbol{\theta\, i}$ **:** Gradiente de iteración actual

A medida que el valor de alpha aumenta, se observa un aumento en el nivel de regularización, lo que conlleva a una disminución en el valor de los coeficientes, lo cual se puede apreciar en la Figura 21.

Figura 21
Coeficiente del modelo en función de la regularización

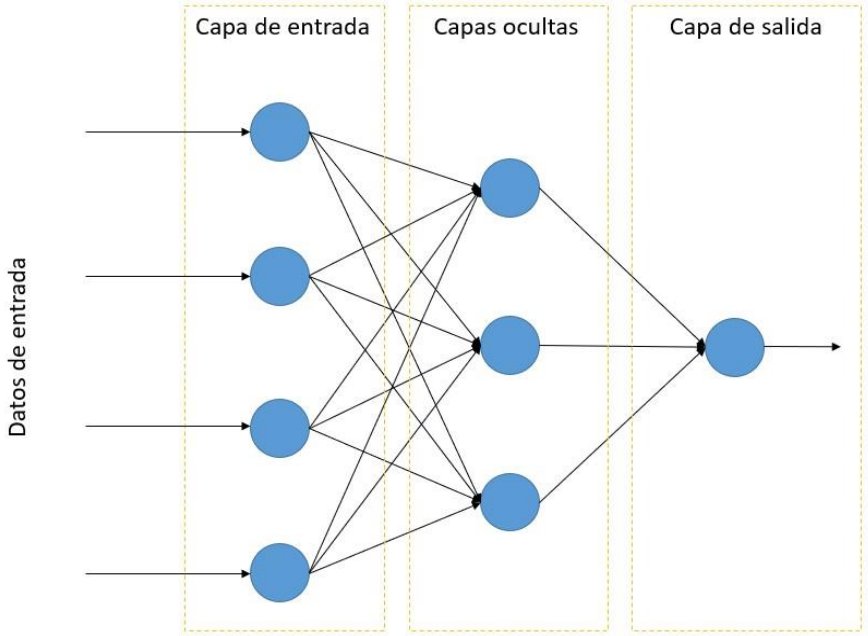


Nota. Extraído de Joaquín, Amat (2022)

Multilayer Perceptron (MLP)

De acuerdo con Serafeim (2021), el perceptrón multicapa (MLP) es un algoritmo de aprendizaje automático supervisado que pertenece a la clase de redes neuronales artificiales feedforward. Este algoritmo se entrena con datos para lograr aprender una función. Dado un conjunto de características y una variable objetivo, el algoritmo aprende una función no lineal ya sea para clasificación o regresión. Un MLP está conformado por al menos tres capas de nodos, exceptuando los nodos de entrada, cada nodo funciona como una neurona que emplea una función de activación no lineal y presenta múltiples capas, tal como se muestra en la Figura 22, la cual se presenta a continuación:

Figura 22
Multilayer Perceptron (MLP)



Nota. Extraído de https://interactivechaos.com/sites/default/files/styles/max_800_px/public/2020-09/tutdl_0044.jpg

Según Bento (2021), la técnica del backpropagation es el mecanismo de aprendizaje que permite al Perceptrón Multicapa ajustar iterativamente los pesos de la red, con el objetivo de minimizar la función de coste. Esto se puede observar en la siguiente fórmula:

Ecuación 16
Multilayer Perceptron (MLP)

$$\Delta_w(t) = -\varepsilon \frac{dE}{dw_{(t)}} + \alpha \Delta_{w(t-1)} \quad (16)$$

Donde:

$\boldsymbol{\Delta_w(t)}$**:** Gradiente de iteración actual

$\boldsymbol{\varepsilon}$**:** Sesgo

$\boldsymbol{dE}$**:** Error

$\boldsymbol{dw_{(t)}}$**:** Vector ponderal

$\boldsymbol{\alpha}$**:** Tasa de aprendizaje

$\boldsymbol{\Delta_{w(t-1)}}$**:** Gradiente de Iteración anterior

2.2.3.3.3. Random Forest

El algoritmo Random Forest, según Schonlau y Zou (2020), es una técnica de Machine Learning que se basa en árboles de decisión y se utiliza para abordar problemas de clasificación y regresión. Esta técnica forma parte de la familia de métodos de conjunto, lo que significa que combina las predicciones de múltiples árboles de decisión individuales para mejorar la precisión y generalización del modelo.

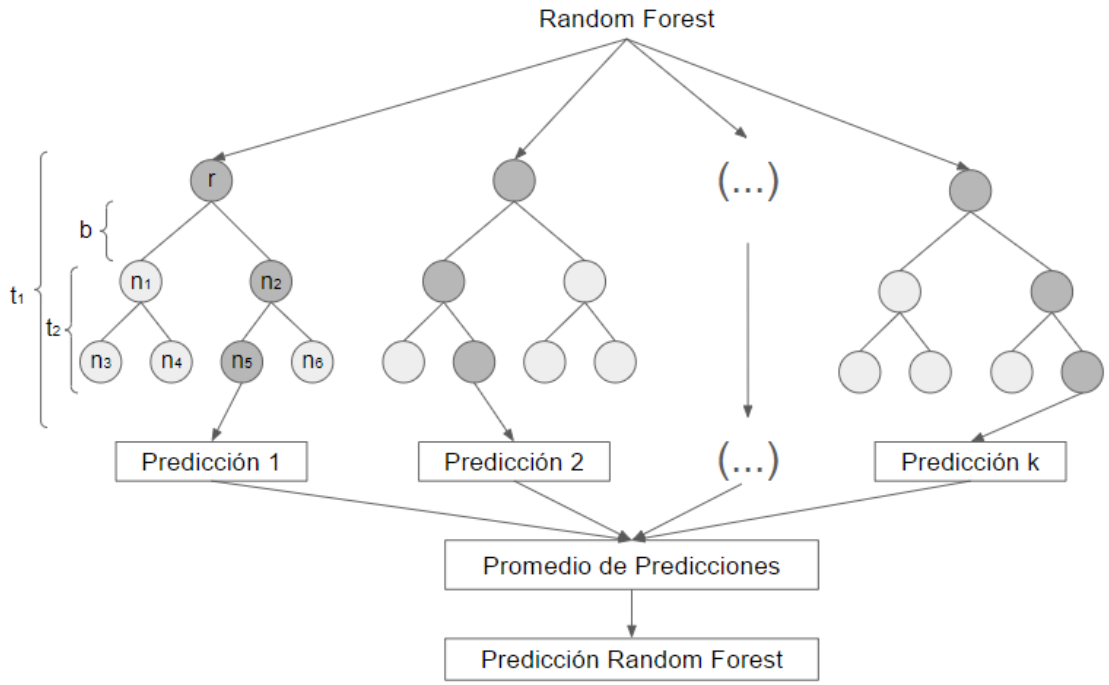
La idea principal detrás del Random Forest es mitigar el problema del sobreajuste (overfitting) que a menudo se encuentra en árboles de decisión individuales. El sobreajuste ocurre cuando un árbol de decisión sigue muy de cerca las peculiaridades de los datos de entrenamiento y, por lo tanto, tiene dificultades para generalizar y hacer predicciones precisas en nuevos datos.

Para abordar este problema, mencionan Schonlau y Zou (2020), el Random Forest construye una colección de árboles de decisión utilizando una técnica llamada "bagging" (bootstrap aggregating). En lugar de utilizar el conjunto de datos completo para construir cada árbol, se crea una muestra aleatoria con reemplazo (bootstrap sample) de los datos de entrenamiento para cada árbol. Esto significa que algunos datos pueden estar presentes en múltiples muestras y otros pueden no estar presentes en absoluto. Esta variabilidad en los datos de entrenamiento ayuda a reducir el sobreajuste.

Además, en cada paso de construcción de un árbol, el Random Forest selecciona aleatoriamente un subconjunto de características (predictoras) para considerar al dividir los nodos internos del árbol. Al considerar solo un subconjunto de características en cada árbol, se reduce la correlación entre los árboles y se mejora la diversidad del conjunto.

Después de la creación de todos los árboles, el Random Forest efectúa pronósticos al combinar las predicciones individuales de cada árbol. En contextos de clasificación, se lleva a cabo un proceso de votación para identificar la clase que es más común entre los árboles. Mientras que, en situaciones de regresión, como se ilustra en la Figura 23, se realiza un promedio de las predicciones provenientes de los árboles.

Figura 23
Representación de Algoritmo Random Forest



Nota. Elaboración propia a partir de información extraída de https://levelup.gitconnected.com/random-forest-regression-209c0f354c84

Variables: La Figura 23 representa cómo se realiza el algoritmo Random Forest en caso de una regresión, en donde $t1$ representa un árbol de decisión y $t2$ un subárbol. Se realizan k predicciones, cada una correspondiente a un árbol distinto y se promedian los resultados de cada árbol para obtener la predicción final del algoritmo. A continuación, se definen las variables que menciona Louppe (2014):

Raíz (r): También llamado nodo raíz, es el nodo inicial de un árbol de decisión y representa la característica que se usa para dividir los datos.

Rama (b): Es una división dentro del árbol de decisión que se crea a partir de una característica específica.

Nodo: Existen dos tipos de nodos

- **Nodo interno (n2)**: Representa una división interna y por lo tanto tiene otros nodos por debajo de sí.

- **Nodo hoja (n5):** Representa una predicción final y por lo tanto no tiene divisiones o ramas. En el caso de una clasificación consiste en una etiqueta y en caso de una regresión, en un valor de regresión.

El Random Forest, según Schonlau y Zou (2020), también proporciona una medida de importancia de las variables, que permite evaluar qué características tienen un mayor impacto en las predicciones del modelo. Esta medida se calcula observando cuánto mejora la función de división (criterio de división) al usar una variable particular en todos los nodos internos de todos los árboles.

Hiperparámetros

- **n_estimators.** Se refiere a la cantidad de árboles que tendrá el algoritmo (Koehrsen, 2018). Schonlau y Zou (2020) señalan que se debe establecer una cantidad de árboles lo suficientemente alta para obtener un rendimiento óptimo. Un mayor número de árboles generalmente mejora la estabilidad de las estimaciones de importancia de variables y las predicciones del modelo.
- **max_features.** Según Koehrsen (2018), es el número máximo de características que se usarán para dividir un nodo.
- **max_depth.** Se refiere al máximo número de niveles que tendrá cada árbol de decisión (Koehrsen, 2018).
- **min_samples_split.** Koehrsen (2018) indica que este hiperparámetro se refiere al número mínimo de puntos de datos que se colocan en un nodo antes de dividirlo.
- **min_samples_leaf.** Se define como el número de puntos de datos permitidos en un nodo de hoja (Koehrsen, 2018).
- **bootstrap.** Según Koehrsen (2018), es el método usado para muestrear puntos de datos, que pueden tener o no reemplazo. Schonlau y Zou (2020) indican que no hay una diferencia sustancial en el rendimiento entre muestrear con o sin reemplazo cuando el tamaño de la muestra se configura de manera óptima.

CAPÍTULO III: ENTORNO EMPRESARIAL

3.1 Descripción de la empresa

La empresa, originada en 1948 como una empresa de carácter familiar, tuvo sus raíces en el sector de la venta de aves. En el transcurso de las dos décadas siguientes, la empresa expandió su alcance al suministrar una variedad de aves en la ciudad de Lima. En la época actual, la empresa se enfoca en tres áreas principales: la crianza, producción, incubación, procesamiento y comercialización de diversas especies avícolas, incluyendo pollos, pavos, cerdos, huevos para consumo y aspectos relacionados con la genética avícola. Además de esto, también se dedica a la elaboración y venta de productos alimentarios procesados, y fabrica alimentos equilibrados diseñados para sus actividades de cría.

En la actualidad, la empresa se sitúa como una de las 50 principales en Perú y constituye el eje central de un grupo empresarial compuesto por 13 empresas colaborativas que cooperan para mejorar sus procesos tecnológicos, comerciales y financieros.

3.1.1 Reseña histórica y actividad económica

Desde su fundación en 1948, ha experimentado un viaje caracterizado por el éxito y el progreso, enfocándose en la creación de una sólida dinastía empresarial. En aquel entonces, se concibió la visión de establecer una entidad duradera en el tiempo, y con el transcurso de los años, esta aspiración se ha materializado de manera sobresaliente.

El punto de inflexión en la historia se produjo en 1963, cuando la empresa comenzó sus operaciones con 35 reproductoras en Tomas Marsano. A lo largo de un período de 15 años, la empresa perfeccionó su experiencia en el modelo de negocio de la cría de pollos y, en 1978, dio un paso adelante al ingresar al mercado de pavos. La expansión del negocio continuó, ya que en 1979 se aventuraron en la producción de huevos y se enfocaron en la comercialización de genética de pollos recién nacidos a partir de 1980.

En 1972, inauguraron su primera tienda en Tomas Marsano, lo que marcó el inicio de su incursión en la venta directa de pollos y huevos. El crecimiento integral de la empresa se evidenció en 1974 cuando comenzaron la cría de reproductoras de carne, y en 1977 cuando establecieron su propia instalación de producción de alimentos equilibrados en Lurín.

La presencia y fortalecimiento en el mercado publicitario también fue notable. Para 2011, la empresa ya se destacaba como líder en todos los segmentos en los que competía a nivel nacional. Ese mismo año, lanzaron la exitosa Campaña Lanzamiento de Pavita. No tardaron en llegar importantes reconocimientos, como el premio Gran Effie 2011 y el Effie Oro 2012 y 2013 a la Mejor Campaña en la categoría de alimentos y bebidas.

Gran parte del éxito de la compañía se debe también a los esfuerzos en innovación y creatividad. En 2017, la empresa recibió el Premio Creatividad Empresarial por su línea de abonos orgánicos Mallki, destacando su compromiso con la calidad en productos y servicios intermedios.

Así, la trayectoria de la compañía es ejemplo de una visión firme como parte de la formación de una gran familia empresarial, que ha logrado notables éxitos en la producción avícola, la comercialización, la publicidad y el manejo de la marca corporativa a lo largo de los años.

3.1.2 Descripción de la organización

La organización ha delineado una visión a futuro que sirve como base para su enfoque hacia la sostenibilidad empresarial. Este enfoque se apoya en varios pilares fundamentales, entre los que se incluyen la Responsabilidad Corporativa, la Responsabilidad Social, los Compromisos, la Gestión Ambiental y la Gestión de Seguridad y Salud en el trabajo.

En cuanto a la Responsabilidad Corporativa, se esfuerza por mantener un equilibrio sostenible en todas sus acciones. Esto implica una constante atención a la mejora de la gestión de calidad, la eficiencia de los procesos, la productividad, la conservación del entorno y el apoyo a las comunidades locales.

La Responsabilidad Social es otro aspecto destacado. La empresa muestra preocupación por el bienestar de sus clientes y las comunidades en las que opera. Para lograrlo, implementa un plan de acción sostenido que abarca a los colaboradores, el entorno y la comunidad en general.

En lo que respecta a los Compromisos, especializada en la provisión de alimentos de consumo masivo, se compromete a través de su Sistema Integrado de Gestión. Esto incluye asegurar la seguridad y salud en el trabajo, prevenir lesiones y deterioro de la salud relacionados con las labores, eliminar peligros y riesgos, cumplir con los requisitos de los clientes, promover

la protección del medio ambiente y garantizar la participación y consulta adecuada de los trabajadores.

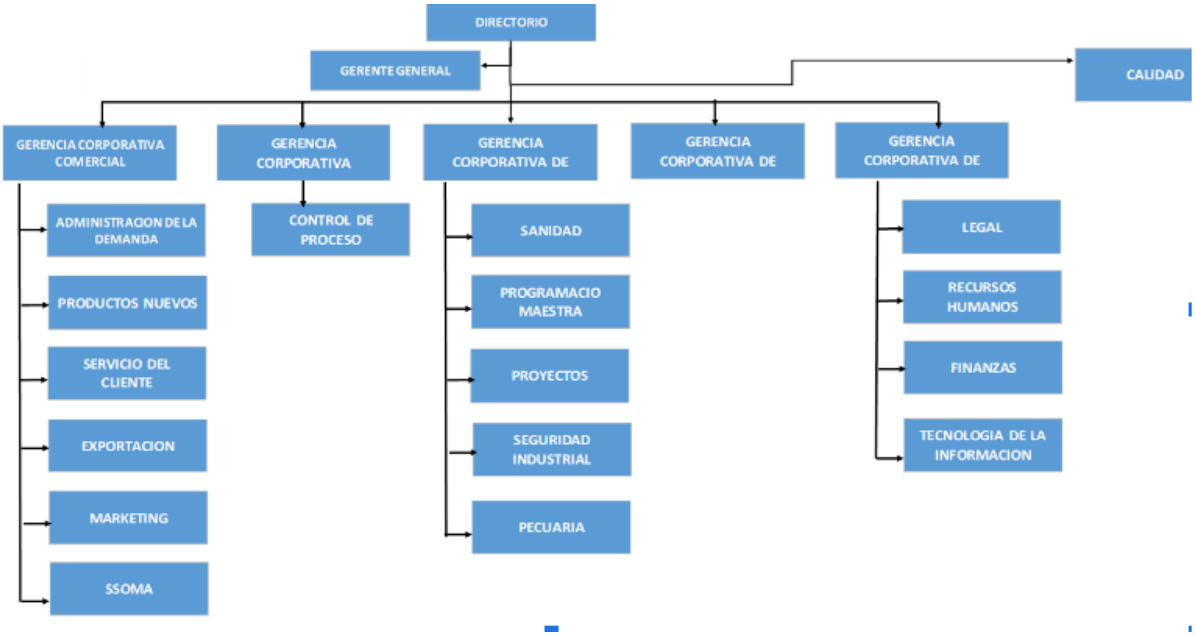
En cuanto a la Gestión Ambiental, demuestra su compromiso con prácticas de producción respetuosas con el entorno, cumpliendo con las regulaciones ambientales y haciendo frente proactivamente los riesgos y oportunidades para reducir posibles impactos negativos. La empresa busca optimizar el uso de recursos naturales y facilitar el reúso de residuos sólidos como plan estratégico para hacer más sostenibles los procesos de producción.

En lo que respecta a la Gestión de Seguridad y Salud en el Trabajo, se compromete a garantizar la protección de sus empleados, contratistas y visitantes. Esto se logra proporcionando entornos seguros y saludables mediante la identificación y control de riesgos. La organización ha implementado con éxito sistemas de gestión de Seguridad y Salud en el Trabajo conforme a estándares reconocidos en algunas de sus unidades operativas.

3.1.2.1 Organigrama

En la Figura 24 se observa cómo se encuentra dividida la organización en cuánto a nivel gerencial corresponde. El liderazgo y la dirección corresponden al Directorio y al Gerente General, quien está a cargo de cinco Gerencias y a su vez del Área de Calidad. El Directorio es quien decide el horizonte que tomará la organización, quienes trasmiten la idea hacia la Gerencia General y a la vez cumple el rol de comunicar a las cinco gerencias para que este alineadas con la planeación estratégica que se ha tomado. La Gerencia General realiza reuniones con sus gerencias todos los lunes de cada semana en las cuales se realiza retroalimentación de las metas semanales y se proponen nuevas metas para la semana activa. La Gerencia Corporativa Comercial se encarga de establecer alianzas comerciales con los clientes y estrechar lazos redituables. La Gerencia corporativa de Seguridad y Sanidad se encarga de velar por la higiene y seguridad ocupacional en la corporación, tanto en la minuciosidad que conlleva la higiene y buenas prácticas alimentarias como la seguridad ocupacional.

Figura 24
Organigrama



Nota. Extraído de "Memoria Anual 2014: S.A.", Recuperado de www.smv.gob.pe/.../temp/Memoria%20SMV%202014_v%20FINAL.pdf

3.1.2.1 Cadena de suministro

La cadena de suministro de la empresa de huevos, mostrada en la Figura 25, ha ido haciéndose más grande y compleja con el paso de los años para poder llevar huevos a todos los hogares del Perú a través de distintos canales de venta. Esta cadena de suministro eficiente y cuidadosamente planificada inicia en dos ubicaciones clave: Chincha y Huaral, en donde se encuentran las granjas de producción.

En estas granjas es donde se producen los huevos frescos, los cuales son recogidos y luego transportados cuidadosamente a la planta de procesamiento. En esta planta, los huevos pasan por una serie de etapas cruciales. Primero, se someten a un riguroso proceso de limpieza y selección para garantizar que cumplan con nuestros estándares de higiene y calidad. Luego, son empaquetados de manera segura y etiquetados con información relevante, incluyendo la fecha de producción.

Una vez que los paquetes de huevos están listos, se distribuyen los lotes de estos productos a los almacenes ubicados en diferentes puntos de Lima Metropolitana. Estos almacenes actúan como centros de distribución clave para atender las demandas de los clientes

en los diferentes distritos coberturados por el servicio de E-commerce, el cual permite a los clientes realizar sus pedidos en línea, seleccionando los productos que desean de nuestro catálogo. Una vez recibidos estos pedidos, el equipo de logística coordina la entrega directa a los hogares de los clientes.

Figura 25
Cadena de Suministro de paquetes de huevos (15 huevos)

Nota. Elaboración propia.

3.1.3 Datos generales estratégicos de la empresa

3.1.3.1 Visión, misión y valores o principios

A continuación, se presentan la visión, misión y principales valores de la empresa:

Visión

Ser competitivos globalmente proporcionando productos de valor agregado para la nutrición humana.

Misión

Contribuir al bienestar de la humanidad proporcionando alimentos a los consumidores masivos en el mercado global.

Valores

- **Honestidad:** Comportarse siempre y expresarse honestamente.
- **Lealtad:** Identificarse siempre con la organización.
- **Respeto:** Cuidar a los demás e integridad organizacional.

- **Laboriosidad**: Respetar las tareas permite alcanzar las metas establecidas.

3.1.3.2 Objetivos estratégicos

Los principales objetivos que la empresa tiene para el largo plazo.

- Responder a las necesidades cambiantes del mercado alimentario nacional y desarrollar procesos que permitan anticiparnos a estas necesidades.
- Atraer clientes internacionales para ampliar el mercado de la organización.
- Aportar agilidad y sensibilidad a las áreas y procesos organizacionales para responder de manera más rápida y efectiva a los cambios ambientales.

3.1.3.3 Evaluación interna y externa: FODA cuantitativo

En la matriz EFE (ver Tabla 7) se ha calculado un puntaje de 2.5 que indica que la empresa está gestionando de manera efectiva las amenazas que enfrenta en su entorno y está aprovechando las oportunidades que se han destacado en la descripción.

Tabla 7
Matriz de Factores Externos (EFE)

	ID	Factor	Peso	Valor	Pond.
Oportunidades	O1	Avance teconlógico en la industria	0.08	4	0.32
	O2	Tendencia mundial de incremento por el consumo de pollo	0.10	3	0.30
	O3	Crecimiento de mercado nacional	0.10	4	0.40
	O4	Apertura de mercados internacionales	0.12	3	0.36
	O5	Preencia de proveedores internacionales	0.08	3	0.24
Amenazas	A1	Volatilidad de precios de insumos (maíz y soya)	0.10	2	0.20
	A2	Riesgos sanitarios (gripe aviar y otras enfermedades)	0.12	2	0.24
	A3	Bajas barreras de entrada	0.10	2	0.20
	A4	Tendencia por dejar de consumir alimentos de origen animal	0.08	2	0.16
	A5	Escasez de agua en zona costera	0.12	1	0.12
	Total		1		2.54

Nota. Elaboración propia.

El puntaje ponderado de 2.70 que se ha obtenido en la matriz EFI (ver Tabla 8) señala que la empresa debe prestar una mayor atención a sus áreas de mejora, ya que estas están teniendo un impacto negativo en su rendimiento.

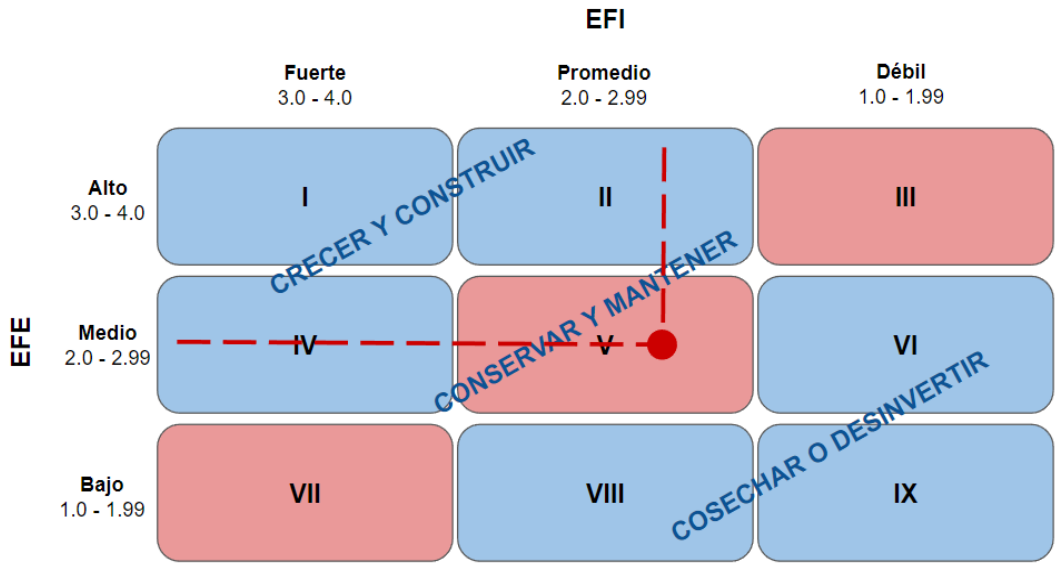
Tabla 8
Matriz de Factores Internos (EFI)

	ID	Factor	Peso	Valor	Pond.
Fortalezas	F1	Nivel de tecnología logística	0.12	3	0.36
	F2	Diversificación de negocios	0.10	4	0.40
	F3	Asociaciones con proveedores estratégicos	0.10	3	0.30
	F4	Posicionamiento de la marca	0.09	4	0.36
	F5	Experiencia en la industria avícola	0.09	4	0.36
Debilidad	D1	Burocracia organizacional	0.10	2	0.20
	D2	Poca presencia internacional	0.12	2	0.24
	D3	Falta de visión estratégica	0.08	1	0.08
	D4	Baja modernización de infraestructura	0.10	2	0.20
	D5	Capacidad de respuesta a variaciones	0.10	2	0.20
	Total		1		2.70

Nota. Elaboración propia.

Ante los puntajes resultantes en las matrices EFE y EFI, la empresa en cuestión se encuentra en el cuadrante V de la matriz IE (ver Figura 26). Eso quiere decir que se recomienda conservar y mantener.

Figura 26
Matriz Interna - Externa (IE)



Nota. Elaboración propia.

Algunas acciones sugeridas para mantener el crecimiento obtenido se muestran en la Figura 27 (FODA cuantitativa) en donde se resalta las oportunidades que se tiene en mercados

extranjeros y pueden ser aprovechadas mediante el renombre y la experiencia que tiene la empresa. Asimismo, se resalta la importancia de fortalecer alianzas para hacer frente a los riesgos sanitarios y asegurar el suministro necesario para el crecimiento.

Figura 27
Matriz FODA Cuantitativa

FODA: San Fernando		F	Fortalezas	D	Debilidades
		1	Nivel de tecnología logística	1	Burocracia organizacional
		2	Diversificación de negocios	2	Poca presencia internacional
		3	Asociaciones con proveedores estratégicos	3	Falta de visión estratégica
		4	Posicionamiento de la marca	4	Baja modernización de infraestructura
		5	Experiencia en la industria avícola	5	Baja capacidad de respuesta a variaciones
O	**Oportunidades**		**Estrategias FO**		**Estrategias DO**
1	Avance teconlógico en la industria	4,5	Captar mercados internacionales aprovechando los años de experiecia en la industria.	3,4	Mejorar la infraestructura para mejorar capacidad productiva y captar el crecimiento de demanda nacional.
2	Tendencia mundial de incremento por el consumo de pollo	3,4	Captar el creciente mercado nacional mediante el renombre de la marca.	5,3	Adoptar una visión estratégica que incluya a los proveedores internacionales de insumos macro para la reducción de costos.
3	Crecimiento de mercado nacional			4,2	Captar clientes internacionales para aumentar la presencia fuera del país.
4	Apertura de mercados internacionales			1,5	Adquirir tecnología que permita agilizar las operaciones y den mejor capacidad de respuesta a la empresa.
5	Preencia de proveedores internacionales				
A	**Amenazas**		**Estrategias FA**		**Estrategias DA**
1	Volatilidad de precios de insumos (maíz y soya)	1,3	Fortelecer alianzas con proveedores para mantener rangos de precios.	1,3	Adoptar una visión estratégica que apunte a la reducción de costos mediante compras de grandes volúmenes de insumos.
2	Riesgos sanitarios (gripe aviar y otras enfermedades)	2,5	Hacer uso de la eperiencia para mitigar los riesgos sanitarios más comunes	2,4	Mejorar la infraestructura para mitigar los riesgos sanitarios presentes.
3	Bajas barreras de entrada	3,2	Aprovechar la diferencia en diferentes negocios para hacer frente a posibles nuevos competidores	3,3	Tomar medidas para adquirir clientes internacionales y mitigar las posibles nuevas entradas de competidores.
4	Tendencia por dejar de consumir alimentos de origen animal	4,4	Hacer uso de la marca para posicionarla como productora con prácticas animalistas.		
5	Escasez de agua en zona costera				

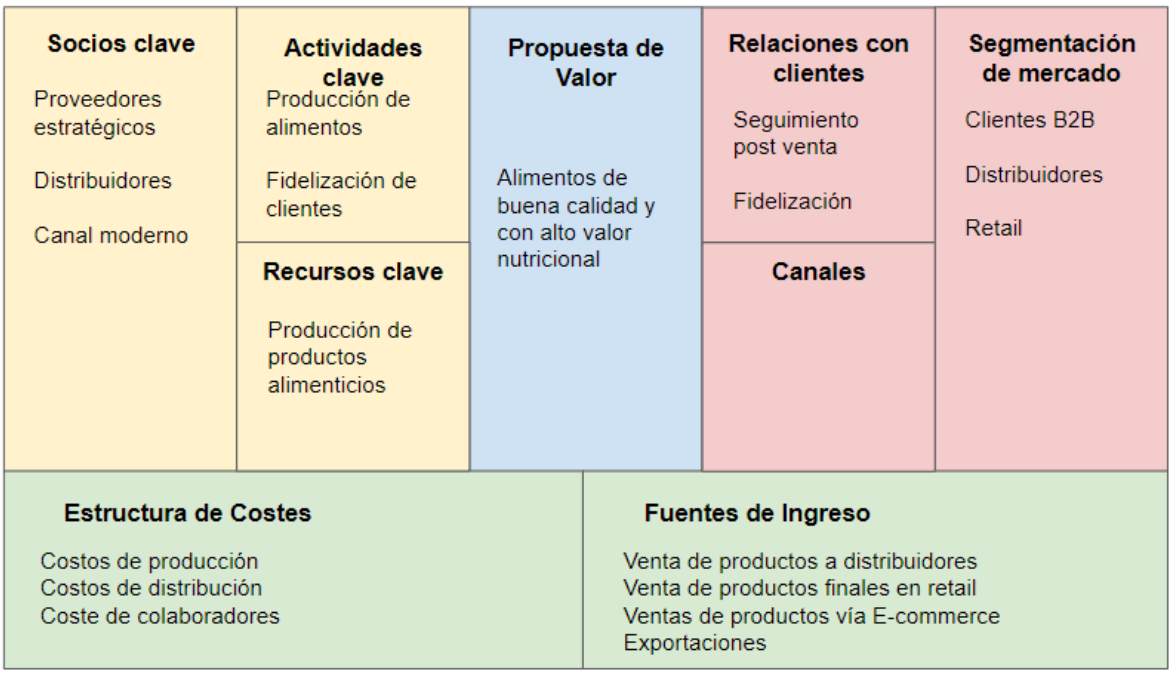
Nota. Elaboración propia.

3.2 Modelo de negocio actual (CANVAS)

En la Figura 28 se puede ver que la propuesta de valor de la empresa de este estudio es la producción de alimentos de calidad y con gran valor nutricional, para lo cual cuenta con proveedores estratégicos y distribuidores que aseguran que dichos productos lleguen a los

consumidores de todo el país e incluso a mercados extranjeros. Asimismo, se puede ver que entre las fuentes de ingreso se encuentra la venta de productos vía E-commerce, servicio que fue lanzado en 2021 a partir del contexto pandémico.

Figura 28
Modelo de Negocio CANVAS



Nota. Elaboración propia.

3.3 Mapa de procesos actual

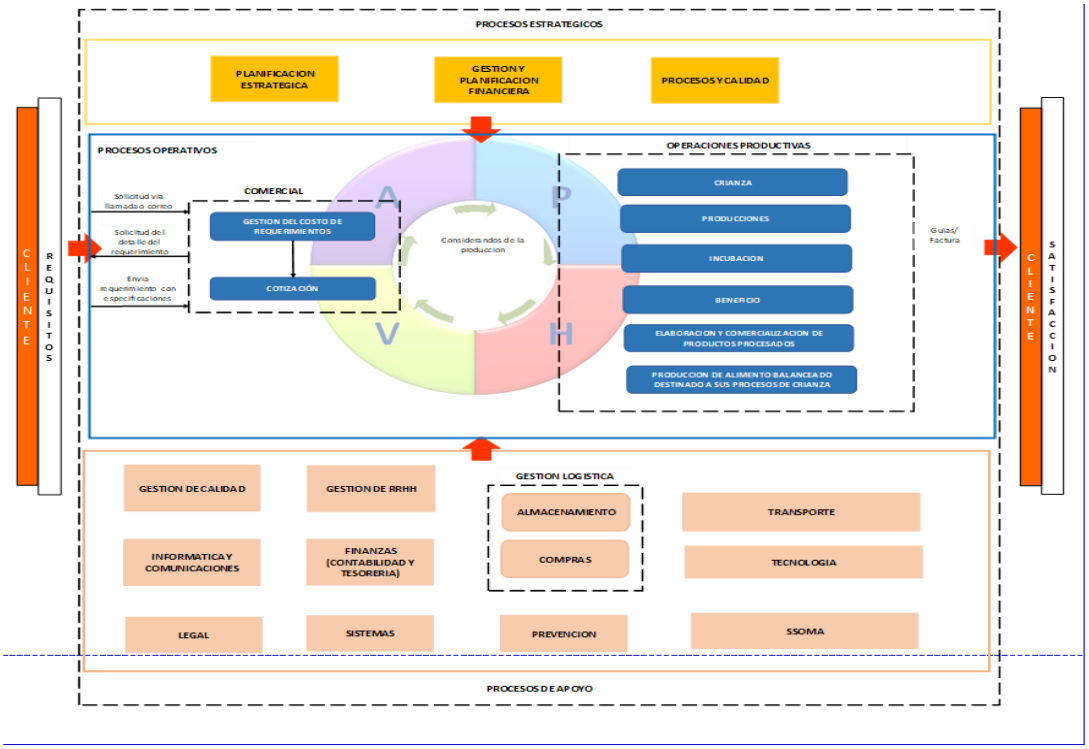
Entre los Procesos Estratégicos de la empresa se encuentran la Planificación Estratégica, la Gestión y Planificación Financiera y los Procesos de Calidad, los cuales son fundamentales para mantener el renombre de la marca y la preferencia de los consumidores.

Los Procesos Operativos, por su parte, se componen de los Procesos Comerciales y las Operaciones Productivas. Los procesos comerciales son la Gestión del costo de requerimientos y la Cotización, los cuales se dan a partir de los requisitos internos de producción que tiene la organización, así como de las solicitudes y necesidades de los clientes. Las Operaciones productivas (Crianza, Incubación, Beneficio, Procesamiento y Producción) son las que permiten la producción animal y su beneficio para la obtención de productos alimenticios.

Estos procesos son realizados gracias a Procesos de apoyo entre los que destacan la Gestión Logística, la Gestión de Calidad, la Gestión de RRHH y las Actividades Legales, lo cual se muestra en la Figura 29, presentada a continuación:

Figura 29

Mapa de procesos actual



Nota. Elaboración propia.

CAPÍTULO IV: METODOLOGÍA DE LA INVESTIGACIÓN

A continuación, se define el diseño de la investigación, cronograma de desarrollo y costo estimado del proyecto:

4.1 Diseño de la Investigación.

4.1.1. Tipo o diseño

Este trabajo es del tipo experimental debido a que se busca predecir la demanda del producto paquetes de huevos (15 huevos) mediante distintos modelos desde los más tradicionales como los estadísticos hasta los más avanzados como los de machine learning.

4.1.2. Enfoque

La presente investigación tiene un enfoque cuantitativo debido a que se trabajará con un conjunto de datos que será analizado y preprocesado mediante técnicas estadísticas. Posteriormente se entrenará modelos tanto estadísticos como de Machine Learning que serán comparados mediante indicadores numéricos para determinar la efectividad de los mismos.

4.1.3. Alcance

Este trabajo tiene un alcance correlacional debido a que se busca encontrar una relación entre la demanda del producto de paquetes de huevos (15 huevos) y una o más variables predictoras como son el precio, campaña, etc.

4.1.4. Población y muestra

Población: todos los registros de órdenes de compra del producto paquetes de huevos (15 huevos) realizados a través del canal E-commerce de la empresa proveedora de productos avícolas.

Muestra: los registros de órdenes de compra del producto paquetes de huevos (15 huevos) realizados a través del canal E-commerce de la empresa proveedora de productos avícolas desde el 17/02/2024 (lanzamiento de canal E-commerce) hasta el 14/09/2023 (fecha de extracción de la data).

4.2 Metodología de implementación de la solución

Luego de explorar las metodologías empleadas en cada uno de los antecedentes de la investigación llegamos a la conclusión de que el orden de sus procedimientos se asemeja mucho a lo que indica el marco de trabajo OSEMN que se usa para proyectos de ciencia de datos.

Figura 30
OSEMN

Nota. Extraído de https://www.datascience-pm.com/osemn/

A continuación, se explica en qué consiste cada uno de los pasos de la metodología que se muestra en la Figura 30.

Obtención de la data: se adquiere la información requerida de las fuentes de datos que están a nuestra disposición. Por lo general se requieren habilidades de manejo de base de datos como SQL para poder combinar y extraer la data. (Han, 2019).

Limpieza de datos: en esta etapa se debe transformar y unificar los datos en un formato estándar. Esto implica consolidar varios archivos CSV en un único repositorio para su posterior análisis, así como depurar los datos al eliminar valores faltantes o incorrectos. También se requiere la manipulación de columnas, como fusionar o dividir datos según sea necesario. (Han, 2019).

Exploración de datos: luego de tener una data limpia es esencial llevar a cabo un proceso de exploración de los datos en el que se pueda examinar detenidamente los datos y sus características. Esto se puede lograr mediante técnicas de estadística descriptiva y apoyo de gráficos. (Han, 2019).

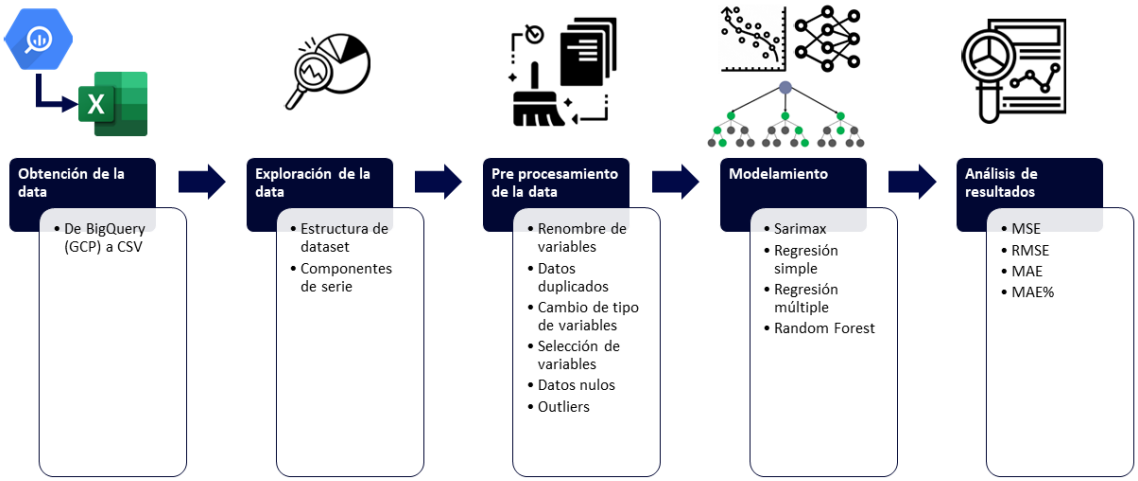
Modelamiento: el Dr. Cher Han Lau (2019) señala que en esta etapa se debe determinar a qué tipo de problema de aprendizaje nos enfrentamos (supervisado o no supervisado) y en base a ello seleccionar los modelos que vamos a entrenar. Un punto clave en esta etapa es seleccionar solo las variables que resultan importantes. Por su parte, Nick Hotz (2023) indica que durante

esta parte se busca identificar el algoritmo que mejor pueda explicar cómo los datos de entrada que ya se conocen pueden ser utilizados para predecir valores de salida que aún no se conocen.

Interpretación: en esta última etapa se lleva a cabo una reflexión sobre las preguntas que inicialmente motivaron la exploración de datos, así como sobre el valor práctico que se obtiene del proceso de investigación y modelado de datos. El propósito fundamental de este proceso es proporcionar información útil a la organización o a las partes interesadas involucradas. (Hotz, 2019). El poder contar una historia con los datos es una habilidad clave en esta etapa para poder darle sentido a todos los hallazgos. (Han, 2019).

Una vez clara esta información se procedió a adaptar dicha metodología a nuestro objetivo, la cual se muestra en la Figura 31. Para ello empezamos igual con la obtención de la data, pero posteriormente intercalamos las siguientes dos etapas porque consideramos que primero es esencial explorar la data para posteriormente limpiarla mediante distintas técnicas de preprocesamiento. En seguida se entrenará distintos modelos para la predicción de demanda que van desde los más sencillos hasta los más complejos que implica el uso de Machine Learning. Finalmente, estos modelos se comparan mediante indicadores específicos y se identifica al mejor. A continuación, se muestra de forma gráfica nuestra metodología y se detalla cada uno de los pasos:

Figura 31
Metodología del proyecto



Nota. Elaboración propia.

4.3 Metodología para la medición de resultados de la implementación

Un punto clave para determinar qué tan bueno es un modelo es seleccionar una métrica correcta. El PhD Jason Brownlee (2021) comenta que cuando se trata de modelos de regresión, a diferencia de los de clasificación, no es posible calcular la precisión (accuracy) de las predicciones por lo que se debe expresar en términos de error en estas.

Al respecto, existe una serie de métricas que miden estos errores, pero también existe un gran desconocimiento sobre cómo funciona, qué miden y qué posibles sesgos puedan ocasionar. Por consecuente, a continuación, se exploran las principales:

4.3.1. Error

Básicamente es la diferencia entre la demanda predicha y la demanda real. Se calcula mediante resta directa:

Ecuación 17
Cálculo del error de predicción

$$et = f_t - d_t \quad (17)$$

Donde:

et: error total

$\boldsymbol{f_t}$**:** predicción total

$\boldsymbol{d_t}$**:** demanda total

No obstante, el error es positivo cuando la predicción es mayor que la demanda y negativo cuando es inferior. Por lo tanto, no es una métrica estandarizada.

4.3.2. Mean Absolute Percentage Error (MAPE)

El error porcentual absoluto medio calcula la suma de los errores absolutos individuales dividida por la demanda por cada período. En otras palabras, calcula el promedio de los errores porcentuales.

Ecuación 18
Mean Absolute Percentage Error (MAPE)

$$MAPE = \frac{1}{n}\sum \frac{|e_t|}{d_t} \quad (18)$$

Donde:

n: total de períodos

$\boldsymbol{e_t}$**:** error total por período

$\boldsymbol{d_t}$**:** demanda total por período

Nicolas Vandeput, científico de datos especialista en predicción de demanda, señala que pese a que este indicador es quizás el más usado por directivos de empresas, no lo recomienda debido a que es impreciso y sesgado. Esto se debe a que al dividir de forma individual cada error por la demanda, en períodos de baja demanda la métrica se ve afectada de forma significativa. (2019). Así, por ejemplo, cuando la demanda es cercana a cero, el MAPE suele tomar valores extremos y cuando la demanda es cero, el MAPE es indefinido.

4.3.3. Mean Absolute Error (MAE)

Es la media del error absoluto y se calcula sumando todos los errores, pero como valor absoluto. Esta métrica es bastante buena pero no está escalada a la demanda promedio por lo que por ejemplo un MAE de 10 podría ser bueno si tu demanda real fue de 200 pero mala si fue de 20. (Vandeput, 2019). Se calcula de la siguiente manera:

Ecuación 19
Mean Absolute Error (MAE)

$$MAPE = \frac{1}{n}\sum|e_t| \qquad (19)$$

Donde:

n: total de períodos

$\boldsymbol{e_t}$**:** error total por período

4.3.4. Porcentual Mean Absolute Error (MAE%)

Con el objetivo de optimizar el MAE, se suele dividir el resultado entre la demanda promedio. De esta manera el MAE% indica en promedio cuánto es el error porcentual entre la demanda real y la pronosticada. Además, al expresar el error en términos porcentuales, se puede

obtener una mejor comprensión de la precisión relativa del modelo en diferentes contextos. Se calcula de la siguiente manera:

Ecuación 20
Porcentual Mean Absolute Error (MAE%)

$$MAE\% = \frac{\frac{1}{n}|e_t|}{\frac{1}{n}\sum|d_t|} \quad (20)$$

Donde:

n: total de períodos

$\boldsymbol{e_t}$**:** error total por período

$\boldsymbol{d_t}$**:** demanda total por período

4.3.5. Mean Squared Error (MSE)

El error cuadrático medio mide la diferencia elevada al cuadrado entre la cantidad predicha y la real para luego dividirla entre la cantidad de registros. Se calcula de la siguiente manera:

Ecuación 21
Mean Squared Error (MSE)

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(f_t - d_t)^2 \quad (21)$$

Donde:

n: total de períodos

$\boldsymbol{f_t}$**:** predicción total

$\boldsymbol{d_t}$**:** demanda total

El MSE es uno de los indicadores más usados debido a que es fácil y rápido de calcular, además de que es más fácil de manipular en comparación a otras métricas. Sin embargo, es importante tener en cuenta que el MSE no se ajusta al error original, ya que este se eleva al cuadrado, generando así un indicador de rendimiento que no puede ser relacionado con la escala de la demanda original. Asimismo, Vandeput (2019) señala que se debe considerar que el

RMSE da mayor importancia a los errores más altos lo cual tiene como costo una sensibilidad a los valores atípicos.

4.3.6. Root Mean Squared Error (RMSE)

El error de raíz cuadrada media mide la desviación estándar de los errores de predicción. Es muy útil y es equivalente a la raíz cuadrado del MSE.

$$RMSE = \sqrt{MSE}$$

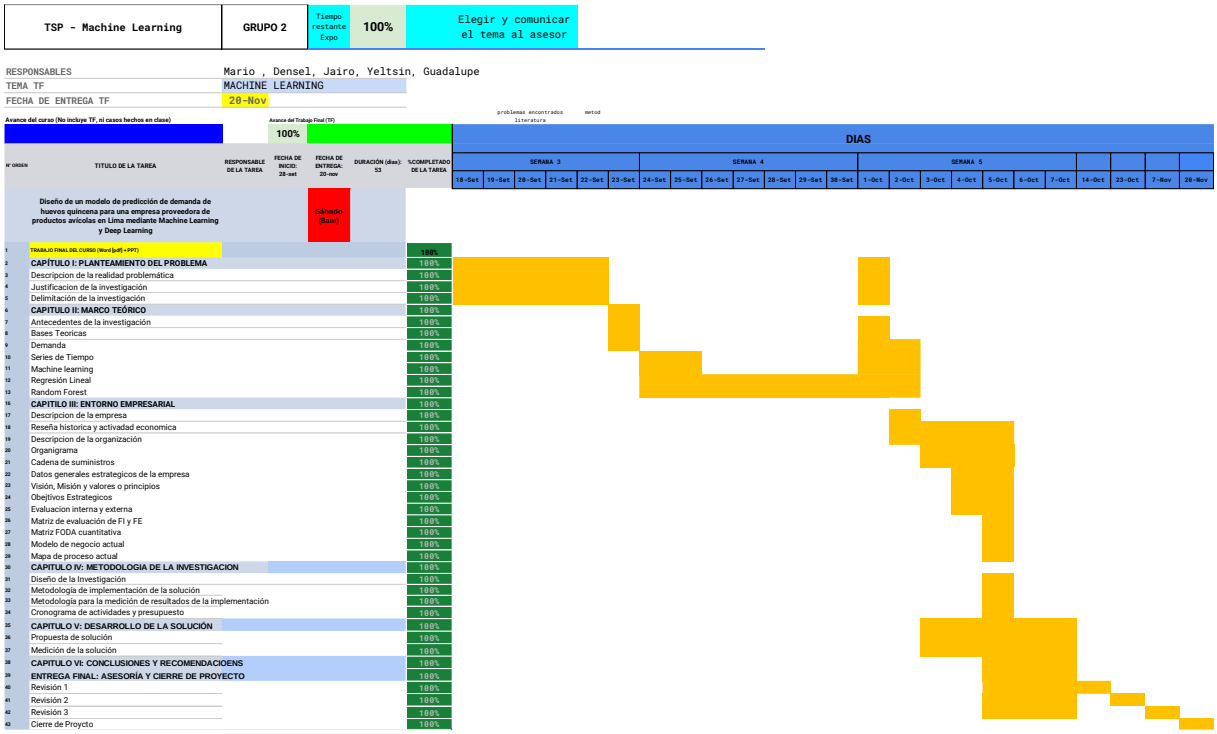
En este punto la pregunta es entonces qué métrica debemos usar para el objetivo de este proyecto. Vandeput concluye que lo más recomendable es usar el MAE o el MSE pero considerando que el MAE preserva la influencia de los valores atípicos, al mismo tiempo que el RMSE garantiza la imparcialidad en las predicciones. Si el conjunto de datos incluye numerosos valores atípicos que puedan sesgar las predicciones, entonces podría ser preferible optar por el MAE.

Por consiguiente, para el presente trabajo se ha decidido medir la eficiencia de los modelos en base al MSE que posteriormente será convertido a RMSE para medir la desviación estándar de los errores de predicción. Asimismo, se usará el MAE para ver si hay impacto de outliers y posteriormente se convertirá a MAE% para tener una idea más exacta del % de error promedio de los modelos.

4.4 Cronograma de actividades y presupuesto

El presente proyecto inició el 18/09/2023 y culminó el 20/11/2023, teniendo así una duración de 63 días; es decir, dos meses. En la Figura 32 se puede apreciar cada una de las etapas desarrolladas, así como el tiempo dedicado.

Figura 32
Cronograma



Nota. Elaboración propia.

Asimismo, se estableció un presupuesto de S/25,000.00 para el estudio planteado, del cual el 48% corresponde a la mano de obra de los integrantes del equipo. Asimismo, la Tabla 9 muestra que el 44% de dicho presupuesto corresponde a la inversión de equipos necesarios para el procesamiento de datos, mientras que el 8% restante se distribuye entre servicios básicos y software.

Tabla 9
Presupuesto

Recurso	Cantidad	Costo unitario	Meses	Costo total
Mano de obra				
Mano de obra	5	S/ 1200	2	S/ 12,000.00
Equipos y softwares				
Laptops	5	S/ 2,200	-	S/ 11,000.00
Software (Colab Pro)	1	S/ 50	2	S/ 100.00
Servicios básicos				

Internet	5	S/ 100	2	S/ 1,000.00
Energía Eléctrica	5	S/ 50	2	S/ 500.00
Pasajes (reuniones presenciales)	5	S/ 40	2	S/ 400.00
			Total	**S/ 25,000.00**

Nota. Elaboración propia.

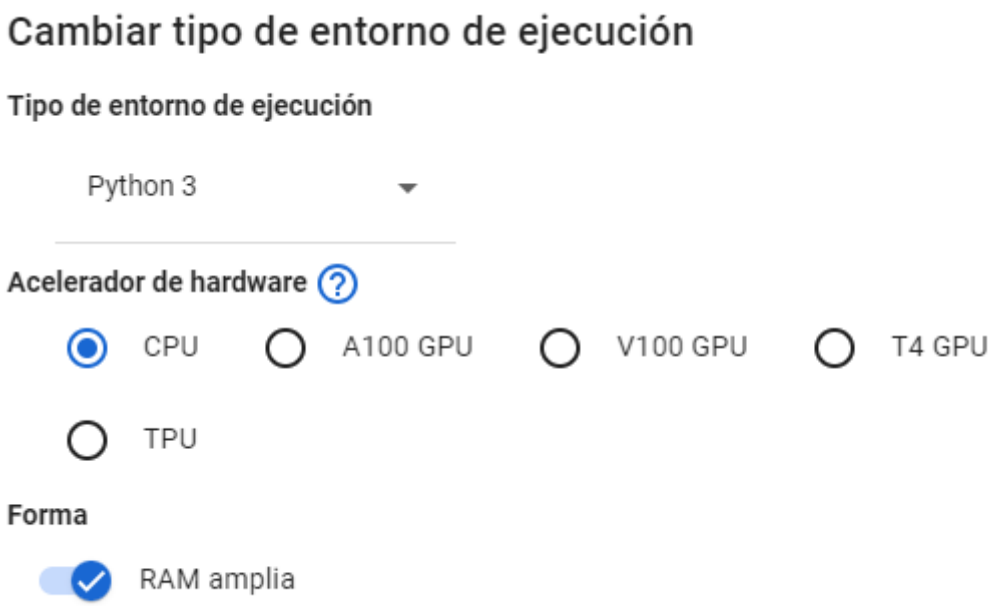
CAPÍTULO V: DESARROLLO DE LA SOLUCIÓN

5.1 Propuesta solución.

El presente trabajo de investigación busca reducir el error de predicción de demanda del producto paquetes de huevos (15 huevos) en Lima para una empresa proveedora de productos avícolas. Para ello se entrenará distintos niveles de modelos predictivos desde los más básicos y estadísticos para que sean nuestra base de comparación hasta los más avanzados de machine learning. Para ello previamente se hará un pre procesamiento exhaustivo de la data.

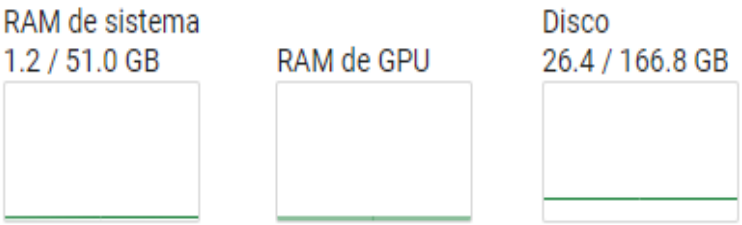
Para el desarrollo del proyecto se ha seleccionado Python como lenguaje de programación y Google Colab como el entorno de programación debido a que brinda la ventaja de que al estar en la nube facilita la edición compartida, así como el consumo de recursos propios de Google. Las opciones del entorno y los recursos de la sesión de Google Colab Pro se pueden apreciar en la Figura 33 y en la Figura 34 respectivamete. Con el fin de optimizar el proceso de entrenamiento se decidió comprar la versión pro por $9.99 que nos ofrece más memoria RAM y el paso de CPU a distintos aceleradores de software como los GPUS.

Figura 33
Opciones de entorno de Google Colab Pro



Nota. Extraído de Google Colab Pro.

Figura 34
Recursos de la sesión en Colab Pro



Nota. Extraído de Google Colab Pro.

Asimismo, disponemos de un GPU Tesla T4 de NVIDIA que está diseñada principalmente para aplicaciones de cómputo de alto rendimiento (HPC), aprendizaje profundo (Deep Learning), inteligencia artificial (IA), y aceleración de cargas de trabajo científicas y técnicas.

5.1.1 Planteamiento y descripción de Actividades

5.1.1. Obtención de la data: la base de datos transaccional del canal de ventas online de la empresa se encuentra alojada en ODOO que es un software ERP integrado; no obstante, se dispone de una copia espejo en BigQuery que es un almacén de datos en la nube de Google Cloud Platform. Para acceder a ello se empleará conocimientos básicos de SQL para poder descargar la base a formato CSV que se usará como input de todo este trabajo.

5.1.2. Exploración de la data: en esta etapa se explorará el esquema de la base de datos para poder entender cada variable disponible, esto implica conocer de qué tipo son y qué valores admiten. Asimismo, dado que queremos predecir la demanda, también resulta necesario poder analizar los distintos componentes de una serie de tiempo y poder identificar si existe alguna tendencia bajo distintos niveles de agregación de la data.

5.1.3. Preprocesamiento de la data: esta es una etapa crucial para el trabajo por lo que se emplearán distintas técnicas que a continuación se muestran:

Figura 35
Preprocesamiento de la data

Nota. Elaboración propia.

5.1.4. Modelamiento: se dividirá la data en 80% para entrenamiento y 20% para test y se entrenará distintos modelos. Los más simples servirán como línea base y los más complejos servirán para demostrar cómo a través del Machine Learning mejoran los resultados. A continuación, en la Tabla 10, se listan los modelos a entrenar según su nivel de complejidad:

Tabla 10
Modelos a entrenar

NIVEL	TIPO	MODELOS	VARIABLES	DETALLE
1	Estadístico	SARIMAX	FechaOrden | CantidadOrdenada	Orden (1,1,0)
2	Estadístico + Machine Learning	Forecast autorregresivo simple	FechaOrden | CantidadOrdenada	Lineal Regresor
				GradientBoosting Regresor
				MLP Regressor
		Forecast autorregresivo múltiple	FechaOrden | PrecioUnitario | CampaniaActiva | CantidadOrdenada	Ridge Regresor
				GradientBoosting Regresor
3	Machine Learning	Regresión lineal simple	FechaOrden | CantidadOrdenada	---
		Regresión múltiple	FechaOrden | PrecioUnitario | CampaniaActiva | CantidadOrdenada	---
		Random Forest	FechaOrden | PrecioUnitario | CampaniaActiva | CantidadOrdenada	Data sin normalizar
				Data normalizada
				Data normalizada + Grid Search + CV
			FechaOrden | PrecioUnitario | CampaniaActiva | CantidadOrdenada | + Weekday	Data normalizada + Grid Search + CV + Nueva variable

Nota. Elaboración propia.

5.1.5. Análisis de resultados: como ya se mencionó anteriormente, se medirá el error de predicción de los modelos en base al MSE, RMSE, MAE y MAE%.

Si bien es cierto, las métricas deberían ser exactas y confiables, Jason Brownlee (2023) menciona lo siguiente:

> Si tiene un modelo de aprendizaje automático y algunos datos, querrá saber si su modelo se ajusta. Puede dividir sus datos en conjuntos de entrenamiento y de prueba. Entrene su modelo con el conjunto de entrenamiento y evalúe el resultado con el conjunto de prueba. Pero usted evaluó el modelo solo una vez y no está seguro de que su buen resultado sea por suerte o no. Desea evaluar el modelo varias veces para poder tener más confianza en el diseño del modelo.

Otro de los grandes problemas de cuando se entrena solo una vez es que los modelos pueden caer en overfitting; es decir que el modelo se ajuste demasiado a los datos de entrenamiento, capturando el ruido y la variabilidad aleatoria en lugar de aprender la verdadera relación subyacente entre las variables. Como resultado, el modelo tiene un rendimiento deficiente cuando se enfrenta a nuevos datos que no forman parte del conjunto de entrenamiento.

Asimismo, debido a que solo contamos con dos años de registros, esto puede influir sobre la robustez del modelo que se espera. Al respecto, Prashant Gupta (2017) precisa que:

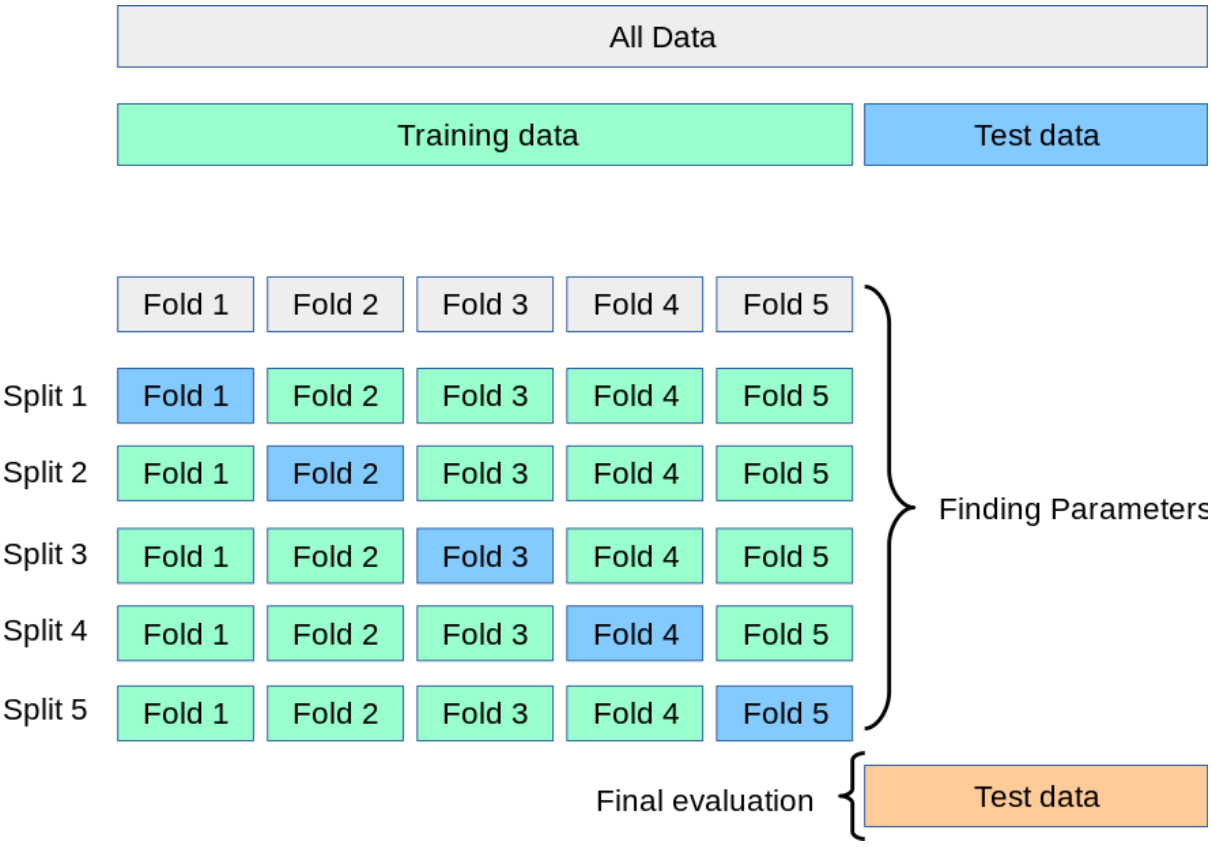
> Como nunca hay suficientes datos para entrenar su modelo, eliminar una parte para su validación plantea un problema de desajuste. Al reducir los datos de entrenamiento, corremos el riesgo de perder patrones/tendencias importantes en el conjunto de datos, lo que a su vez aumenta el error inducido por el sesgo.

Por consiguiente, con el objetivo de que los resultados de las métricas de nuestro mejor modelo sean precisos y que no presente problemas de overfitting, se aplicará una técnica conocida como cross validation. En esta técnica los datos se dividen en K grupos o subconjuntos y luego se entrena un modelo K veces, de manera que, en cada iteración, uno de los K grupos se designa como conjunto de prueba o validación, mientras que los otros K-1 grupos se

combinan para formar un conjunto de entrenamiento. Se calcula el error en cada una de las K pruebas y se promedian estos errores para obtener una medida global de la efectividad de nuestro modelo. De este modo cada observación llega a estar al menos una vez en un conjunto de entrenamiento y un conjunto de pruebas. El único parámetro a definir es K que indica la cantidad de iteraciones.

Esto disminuye de manera considerable el sesgo, dado que aprovechamos la mayor parte de los datos para el ajuste, y también reduce en gran medida la variabilidad, ya que la mayoría de los datos se emplean en el conjunto de validación. Además, alternar entre los conjuntos de entrenamiento y prueba mejora la eficacia de esta técnica. (Gupta, 2017). A continuación, se presenta una vista gráfica de la técnica explicada:

Figura 36
Técnica utilizada



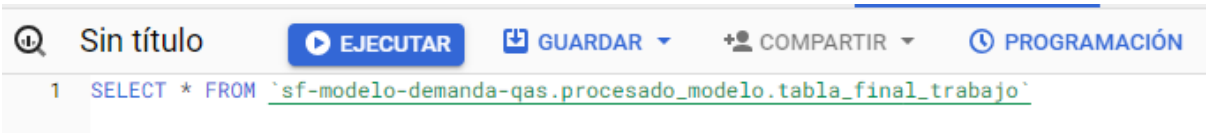
Nota. Elaboración propia.

5.1.2 Desarrollo de actividades

5.1.2.1. Obtención de la data

La base de datos de pedidos de la empresa mencionada se encuentra alojada en ODOO que es un software de ERP integrado. Esta a su vez tiene una copia espejo en BigQuery que es un almacén de datos administrador por Google. La data se almacena en una tabla de nombre tabla_final_trabajo alojado dentro del conjunto de datos procesado_modelo y el proyecto Empresa-Modelo Demanda PRD. Para poder acceder a ella usaremos la sintaxis básica de SQL. Básicamente lo que hacemos es seleccionar todas las columnas y registros contenidos en la tabla de interés. En términos de recursos, la consulta consumió 336.32 MB y demoró menos de un segundo en ejecutarse.

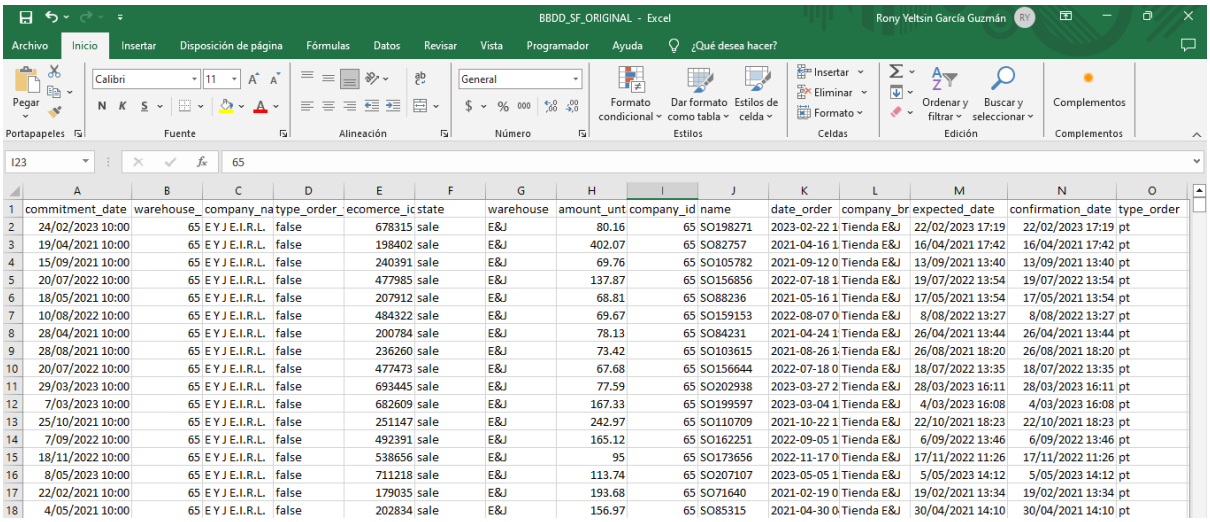
Figura 37
Sentencia SQL para descarga de BBDD



Nota. Elaboración propia.

Posteriormente, la tabla resultante fue descargada en formato CSV. La base descargada pesa un total de 281.67 MB y contiene un total de 802 176 registros de compras de todos sus productos desde el 17 de febrero del 2021 hasta el 14 de septiembre del 2023. A continuación, en la Figura 38, se muestra una vista de la base de datos en formato Excel:

Figura 38
BBDD original



	commitment_date	warehouse_	company_na	type_order_	ecomerce_ic	state	warehouse	amount_unt	company_id	name	date_order	company_br	expected_date	confirmation_date	type_order
2	24/02/2023 10:00	65	E Y J E.I.R.L.	false	678315	sale	E&J	80.16	65	SO198271	2023-02-22 1	Tienda E&J	22/02/2023 17:19	22/02/2023 17:19	pt
3	19/04/2021 10:00	65	E Y J E.I.R.L.	false	198402	sale	E&J	402.07	65	SO82757	2021-04-16 1	Tienda E&J	16/04/2021 17:42	16/04/2021 17:42	pt
4	15/09/2021 10:00	65	E Y J E.I.R.L.	false	240391	sale	E&J	69.76	65	SO105782	2021-09-12 0	Tienda E&J	13/09/2021 13:40	13/09/2021 13:40	pt
5	20/07/2022 10:00	65	E Y J E.I.R.L.	false	477985	sale	E&J	137.87	65	SO156856	2022-07-18 1	Tienda E&J	19/07/2022 13:54	19/07/2022 13:54	pt
6	18/05/2021 10:00	65	E Y J E.I.R.L.	false	207912	sale	E&J	68.81	65	SO88236	2021-05-16 1	Tienda E&J	17/05/2021 13:54	17/05/2021 13:54	pt
7	10/08/2022 10:00	65	E Y J E.I.R.L.	false	484322	sale	E&J	69.67	65	SO159153	2022-08-07 0	Tienda E&J	8/08/2022 13:27	8/08/2022 13:27	pt
8	28/04/2021 10:00	65	E Y J E.I.R.L.	false	200784	sale	E&J	78.13	65	SO84231	2021-04-24 1	Tienda E&J	26/04/2021 13:44	26/04/2021 13:44	pt
9	28/08/2021 10:00	65	E Y J E.I.R.L.	false	236260	sale	E&J	73.42	65	SO103615	2021-08-26 1	Tienda E&J	26/08/2021 18:20	26/08/2021 18:20	pt
10	20/07/2022 10:00	65	E Y J E.I.R.L.	false	477473	sale	E&J	67.68	65	SO156644	2022-07-18 0	Tienda E&J	18/07/2022 13:35	18/07/2022 13:35	pt
11	29/03/2023 10:00	65	E Y J E.I.R.L.	false	693445	sale	E&J	77.59	65	SO202938	2023-03-27 2	Tienda E&J	28/03/2023 16:11	28/03/2023 16:11	pt
12	7/03/2023 10:00	65	E Y J E.I.R.L.	false	682609	sale	E&J	167.33	65	SO199597	2023-03-04 1	Tienda E&J	4/03/2023 16:08	4/03/2023 16:08	pt
13	25/10/2021 10:00	65	E Y J E.I.R.L.	false	251147	sale	E&J	242.97	65	SO110709	2021-10-22 1	Tienda E&J	22/10/2021 18:23	22/10/2021 18:23	pt
14	7/09/2022 10:00	65	E Y J E.I.R.L.	false	492391	sale	E&J	165.12	65	SO162251	2022-09-05 1	Tienda E&J	6/09/2022 13:46	6/09/2022 13:46	pt
15	18/11/2022 10:00	65	E Y J E.I.R.L.	false	538656	sale	E&J	95	65	SO173656	2022-11-17 0	Tienda E&J	17/11/2022 11:26	17/11/2022 11:26	pt
16	8/05/2023 10:00	65	E Y J E.I.R.L.	false	711218	sale	E&J	113.74	65	SO207107	2023-05-05 1	Tienda E&J	5/05/2023 14:12	5/05/2023 14:12	pt
17	22/02/2021 10:00	65	E Y J E.I.R.L.	false	179035	sale	E&J	193.68	65	SO71640	2021-02-19 0	Tienda E&J	19/02/2021 13:34	19/02/2021 13:34	pt
18	4/05/2021 10:00	65	E Y J E.I.R.L.	false	202834	sale	E&J	156.97	65	SO85315	2021-04-30 0	Tienda E&J	30/04/2021 14:10	30/04/2021 14:10	pt

Nota. Elaboración propia.

5.1.2.2. Exploración de la data

5.1.2.2.1. Esquema de base de datos

En la Tabla 11 se muestra la variable Target. Por otro lado, la base de datos cuenta con un total de 43 variables que se componen de tres variables tipo Fecha, mostradas en la Tabla 12; nueve variables numéricas mostradas en la Tabla 11 y treinta y una variables categóricas mostradas en la Tabla 13. Por otro lado, en las Tablas 14 y 15 se muestran las variables agrupadas por categoría.

Tabla 11
Variable Target

VARIABLE	DEFINICIÓN	TIPO
qty_ordered	Cantidad ordenada	Numérica

Nota. Elaboración propia.

Tabla 12
Variables Fecha

VARIABLE	DEFINICIÓN	TIPO
date_order	Fecha de orden	Fecha
commitment_date	Fecha de compromiso	Fecha
expected_date	Fecha de espera	Fecha
confirmation_date	Fecha confirmada	Fecha

Nota. Elaboraciones categóricas

Tabla 13
Variables numéricas

VARIABLE	DEFINICIÓN	TIPO
amount_untaxed	Monto sin impuestos	Numérica
amount_total	Monto total	Numérica
price_unit	Precio unitario	Numérica
qty_delivered	Cantidad entregada	Numérica
qty_ordered	Cantidad ordenada	Numérica
units_per_product	Unidades por producto	Numérica

standard_price	Precio estándar	Numérica
list_price	Precio de lista	Numérica
Weight	Peso	Numérica

Nota. Elaboración propia.

Tabla 14
Variables categóricas

VARIABLE	DEFINICIÓN	TIPO
warehouse_id	ID de almacén	Categórica
company_name	Nombre de compañía	Categórica
type_order_vale	Tipo de orden	Categórica
ecomerce_id	ID de ecommerce	Categórica
State	Estado de pedido	Categórica
Warehouse	Nombre de almacén	Categórica
company_id	ID de compañía	Categórica
Name	Nombre de comprador	Categórica
company_branch	Marca de tienda	Categórica
type_order	Tipo de orden	Categórica
order_source	Fuente de orden	Categórica
partner_id	Id de partner	Categórica
company_branch_id	ID de tienda	Categórica
warehouse_code	Código de almacén	Categórica
Id	ID	Categórica
price_unit	Precio unitario	Numérica
Product	Nombre de producto	Categórica
product_uom	Unidad de producto	Categórica
product_id	ID de producto	Categórica
default_code	Código por defecto	Categórica
ESTADO_PEDIDO	Estado de pedido	Categórica

sku_producto	SKU de producto	Categórica
CAMPANA_ACTIVA	Campaña activa	Categórica
CYBER	Oferta Cyber	Categórica
category_id	ID de categoría de producto	Categórica
category_name	Categoría de producto	Categórica
product_type	Tipo de producto	Categórica
unidad_medida	Unidad de medida	Categórica
id_sap	ID SAP	Categórica
ZONA_COBERTURA	Zona de cobertura	Categórica
estado_afiliada	Estado de tienda	Categórica

Nota. Elaboración propia.

5.1.2.2.1. Componentes de serie

Debido a que nuestro objetivo es predecir la demanda, resulta sumamente necesario evaluarla en forma de serie de tiempo para poder comprender su comportamiento.

5.1.2.2.1.1. Preparación de serie de tiempo

Empezamos filtrando nuestra base de datos para solo quedarnos con los registros de compras del producto Huevo Quincena. No obstante, existen múltiples compras para un mismo día por lo que necesitamos agrupar las ventas por fecha. Es así que de lo mostrado en la base de datos inicial en Excel pasamos a lo siguiente:

Figura 39
Agrupación de demanda por fecha

```
# Nuevo dataframe con suma de pedidos por semana
df_producto = data.groupby(['FechaOrden'])['CantidadOrdenada'].sum().reset_index()
df_producto.head()
```

	FechaOrden	CantidadOrdenada
0	1/01/2022	3
1	1/01/2023	33
2	1/02/2022	93
3	1/02/2023	23
4	1/03/2021	226

Nota. Elaboración propia.

Posteriormente, debemos asegurarnos de que la serie esté completa porque puede que en algunos días no haya habido compras. Para ello primero hacemos que la Fecha de la Orden sea el index de la Tabla y que se identifique una frecuencia de avance de un día mediante el siguiente

Figura 40
Asignación de frecuencia de serie

```
df_producto['FechaOrden'] = pd.to_datetime(df_producto['FechaOrden'], format='%Y/%m/%d')
df_producto = df_producto.set_index('FechaOrden')
df_producto = df_producto.asfreq('1D')
df_producto = df_producto.sort_index()
df_producto.head()
```

Nota. Elaboración propia.

Ahora sí podemos contar si para alguna fecha no hay registro de la demanda y lo reemplazamos por cero que es lo que realmente le corresponde. Es así que identificamos 10 fechas en las que se tuvo que aplicar este procedimiento de la siguiente manera:

Figura 41
Completud de la serie

```
# Cantidad de fechas con demanda nula
df_producto.isnull().sum().sort_values(ascending = False)
```

```
CantidadOrdenada    10
dtype: int64
```

```
# Reemplazo de demanda desconocida por 0
df_producto['CantidadOrdenada'].fillna(0,inplace=True)
```

```
# Verificamos que el índice temporal está completo
# ==============================================================
(df_producto.index == pd.date_range(
                    start = df_producto.index.min(),
                    end   = df_producto.index.max(),
                    freq  = df_producto.index.freq)
).all()
```

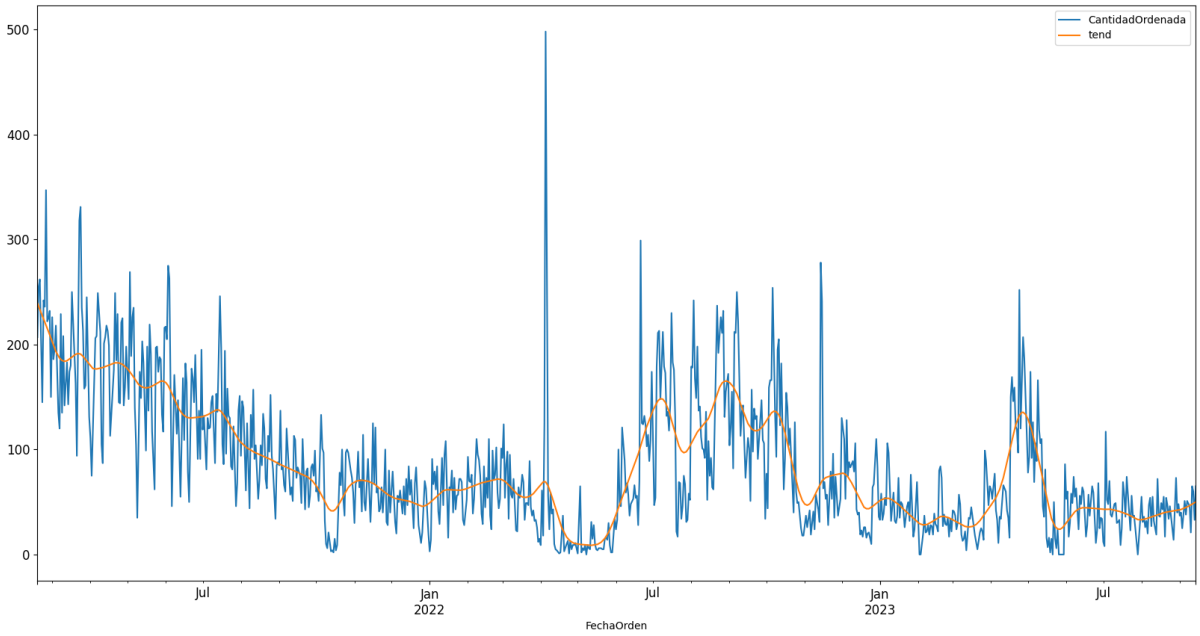
```
True
```

Nota. Elaboración propia.

5.1.2.2.1.2. Tendencia general

Una vez con la serie completa ya podemos analizar la tendencia general a lo largo de toda la serie. De ello podemos corroborar de forma visual que existe una tendencia no lineal pero muy irregular debido a que muestra múltiples curvas de subida y bajada de demanda.

Figura 42
Tendencia de la serie



Nota. Elaboración propia.

5.1.2.2.1.3. Estacionariedad

Con el objetivo de conocer si es que las propiedades de la serie como la media, la varianza y covarianza cambian o no en función del tiempo, se procedió a hacer pruebas de hipótesis para conocer su estacionariedad.

Figura 43
Test de estacionariedad

```
# Test estacionariedad
# ==============================================================================
warnings.filterwarnings("ignore")

datos_diff_1 = train.diff().dropna()
datos_diff_2 = datos_diff_1.diff().dropna()

print('Test estacionariedad serie original')
print('------------------------------------')
adfuller_result = adfuller(df_producto)
kpss_result = kpss(df_producto)
print(f'ADF Statistic: {adfuller_result[0]}, p-value: {adfuller_result[1]}')
print(f'KPSS Statistic: {kpss_result[0]}, p-value: {kpss_result[1]}')
```

```
Test estacionariedad serie original
------------------------------------
ADF Statistic: -2.831235819293908, p-value: 0.05394861653854464
KPSS Statistic: 1.8116425986921152, p-value: 0.01
```

Nota. Elaboración propia.

Prueba de Dickey-Fuller aumentada (ADF)

- Hipótesis nula (HO): La serie tiene una raíz unitaria, no es estacionaria.
- Hipótesis alternativa (HA): La serie no tiene raíz unitaria, es estacionaria.

De aplicar la ecuación 3 se obtuvo un p-value de 0.054 por lo que al ser mayor que 0.05 (umbral) aceptamos la hipótesis nula y confirmamos que la serie no es estacionaria.

Prueba Kwiatkowski-Phillips-Schmidt-Shin (KPSS)

- Hipótesis nula (HO): La serie es estacionaria.
- Hipótesis alternativa (HA): La serie no es estacionaria.

Se obtuvo un p-value de 0.01 por lo que al ser menor que 0.05 (umbral) rechazamos la hipótesis nula y confirmamos que la serie no es estacionaria.

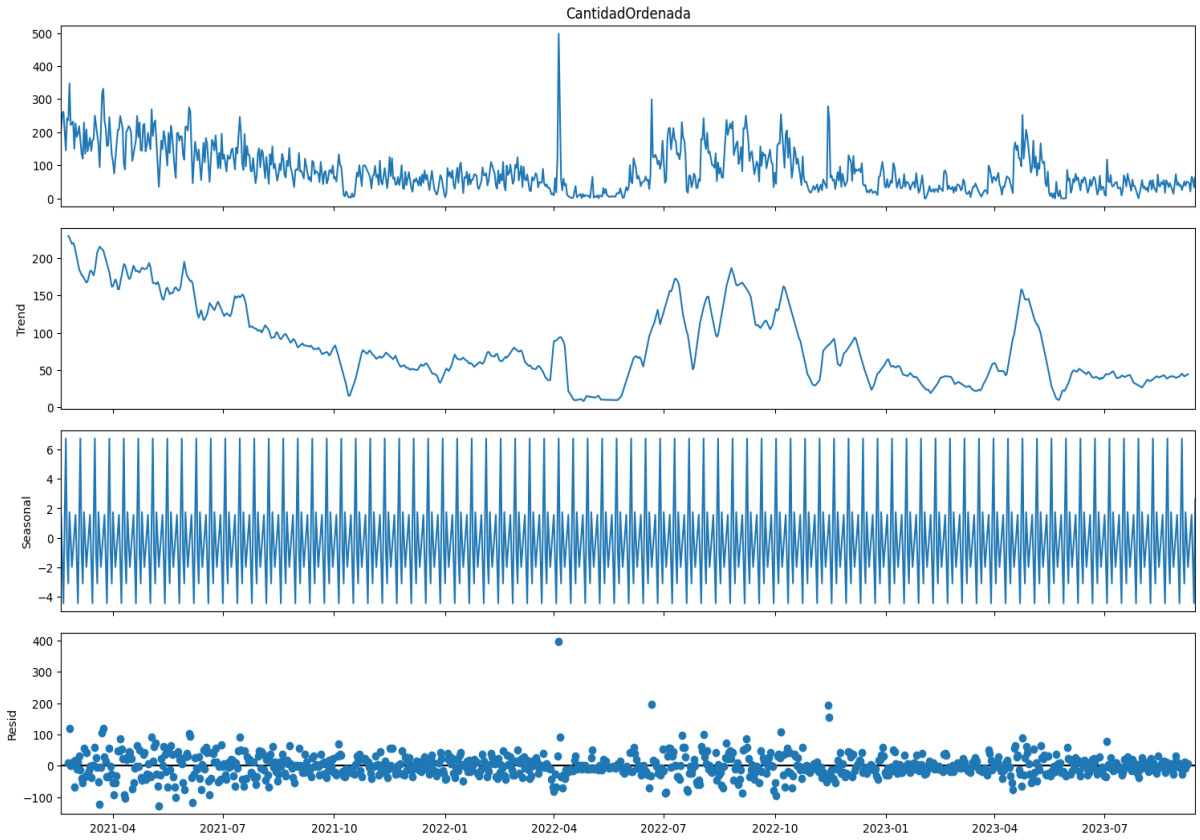
De las dos formas anteriores confirmamos que estamos trabajando con una serie cuya demanda no es estacionaria por lo que sus propiedades cambian con el tiempo. Por ende, podemos corroborar de forma visual lo siguiente:

- **Tendencia:** aunque es irregular, se nota que puede ir aumentando o descendiendo con el tiempo.
- **Estacionalidad:** se podría encontrar patrones relacionados con eventos regulares como épocas del año, meses, semanas, etc. Por ende, más adelante se harán algunas pruebas gráficas.
- **Variaciones irregulares o aleatorias:** la serie evidentemente tiene variaciones de este tipo que no pueden preverse ni modelarse con facilidad. Esto se confirmará más adelante con una gráfica de residuos.

5.1.2.1.1.4. Descomposición aditiva

Dado que en el análisis anterior se conoció que la serie no presenta estacionariedad entonces procedemos a realizar con certeza una descomposición aditiva debido a que permitirá identificar y separar la tendencia ascendente o descendente y los patrones estacionales de tu serie de tiempo. A continuación, se muestra de forma gráfica lo obtenido:

Figura 44
Descomposición aditiva



Nota. Elaboración propia.

Del gráfico se puede apreciar lo siguiente:

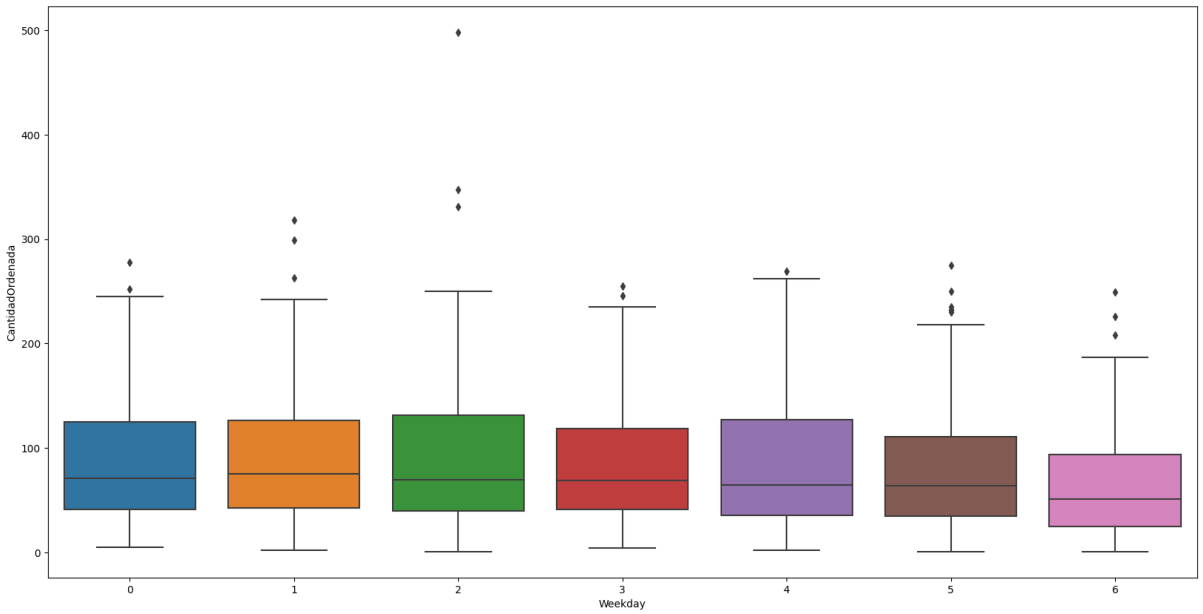
- **Tendencia:** es completamente irregular.
- **Estacionalidad:** sí existe, pero no se puede apreciar la frecuencia de forma exacta por lo que más adelante se identificará.
- **Residuos:** muestra variaciones aleatorias o no sistemáticas que no siguen un patrón predecible.

5.1.2.2.2. Tendencias

Tendencia por día de semana

Figura 45

Demanda de paquetes de huevos (15 huevos) por día de la semana



Nota. Elaboración propia.

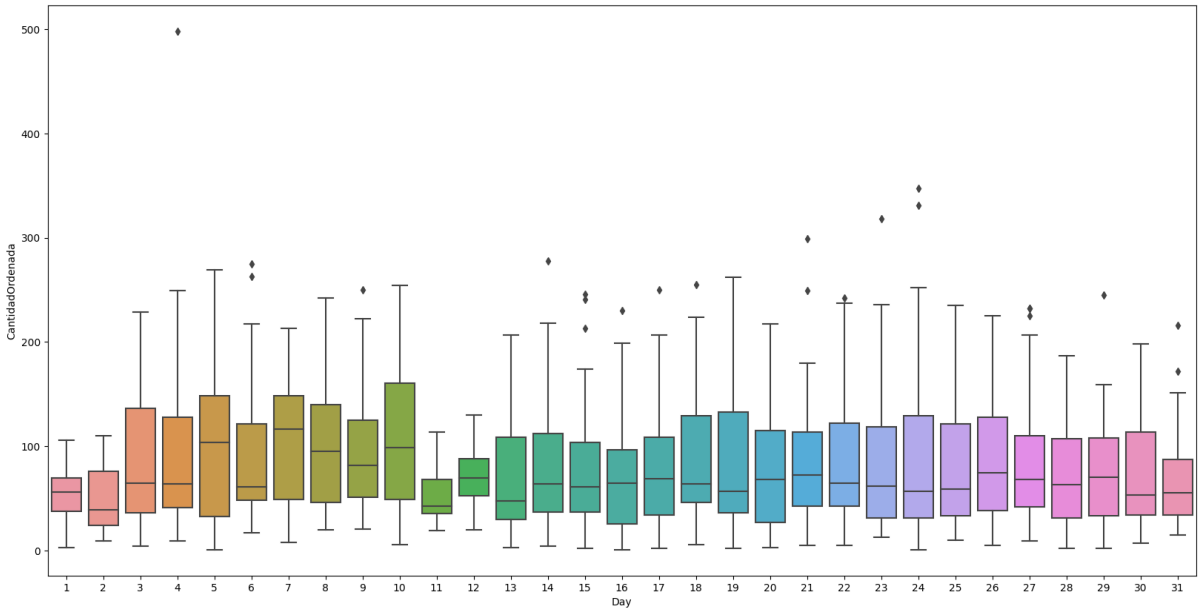
En la Figura 45, se observa la cantidad de paquetes de huevos (15 huevos) que son ordenados por día durante la semana, comenzando por el día domingo, seguido del día lunes y así sucesivamente. A partir del gráfico, se puede observar que en los días domingos y lunes, la demanda de paquetes de huevos (15 huevos) es mayor en comparación con los demás días de la semana.

Tendencia por día de mes

Con respecto a la demanda por mes, en la Figura 46 se aprecia que tiene una tendencia irregular. Se observa que los primeros días del mes, la demanda tiene una tendencia creciente hasta llegar a su máximo pico y luego empieza a decrecer aproximadamente a mitad del mes. Durante los días posteriores la tendencia es creciente otra vez hasta que, finalmente, decrece durante los últimos días del mes.

Figura 46

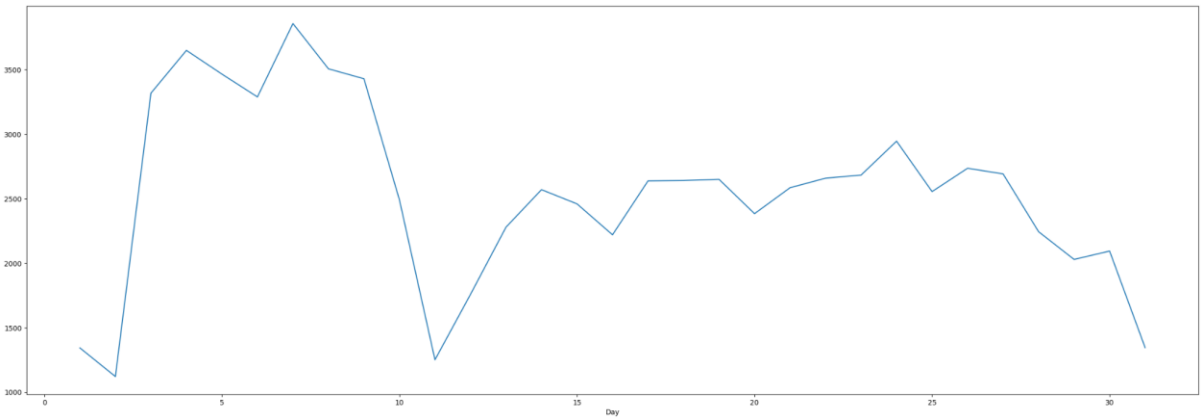
Boxplot de demanda de paquetes de huevos (15 huevos) por día del mes



Nota. Elaboración propia.

Figura 47

Distribución de demanda paquetes de huevos (15 huevos) por día del mes



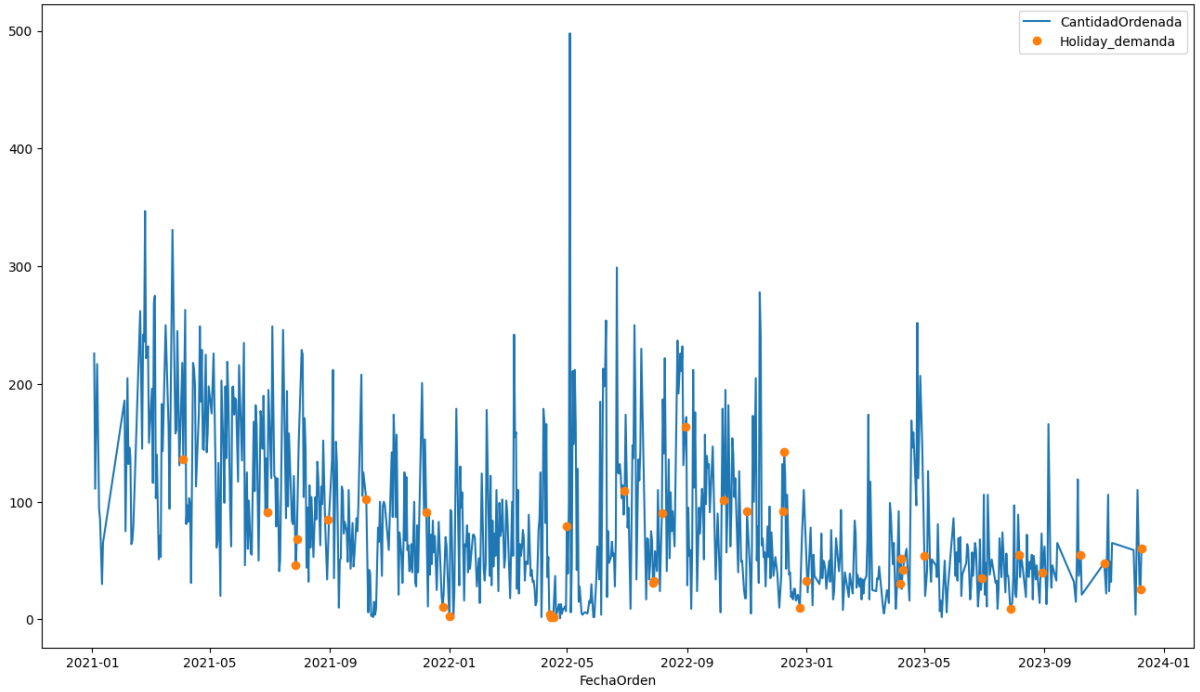
Nota. Elaboración propia.

Tendencia por fechas festivas

De acuerdo con la Figura 48, en el año 2021, la tendencia se incrementa durante los meses de febrero y marzo aproximadamente. Tanto para el año 2022 y 2023 se alcanza el pico más alto durante el mes de mayo.

Figura 48

Demanda de paquetes de huevos (15 huevos) en fechas festivas



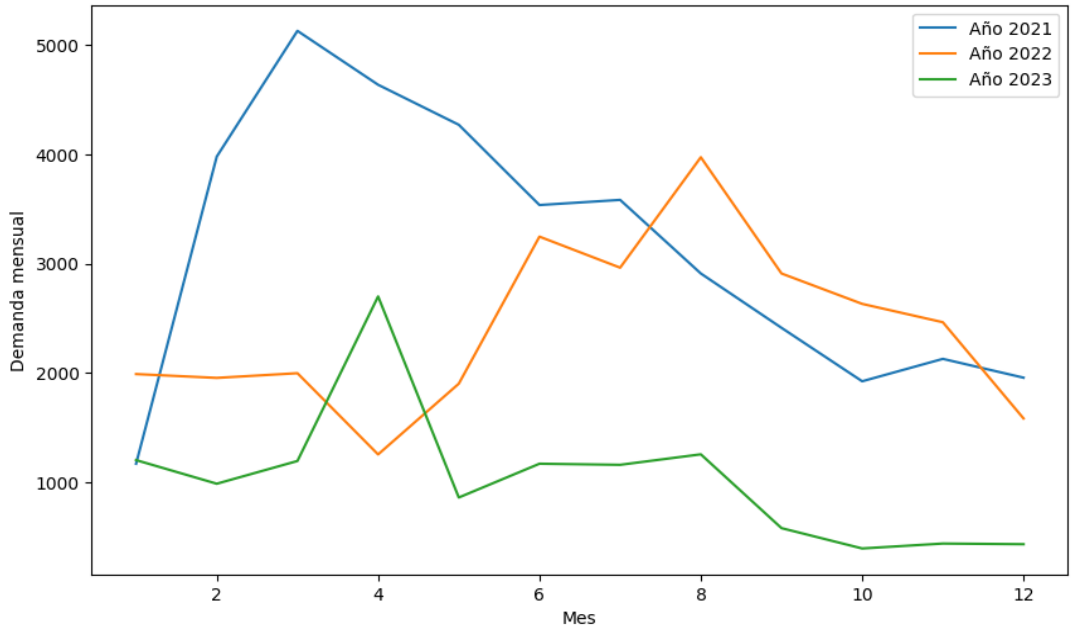
Nota. Elaboración propia.

Tendencia por meses

Al comparar la demanda de paquetes de huevos (15 huevos) desde el lanzamiento del canal E-commerce en febrero de 2021 en la Figura 49, se evidencia que no existe una tendencia o estacionalidad por mes y los requerimientos recibidos fueron variando año a año. Esta volatilidad se explica, por ejemplo, por la alta demanda de servicios de delivery en 2021 durante la pandemia y a aspectos coyunturales que hicieron que varíen los precios de huevo y, por lo tanto, su demanda. Así, a inicios de este año los elevados precios de soya y la amenaza de la gripe aviar hizo que los costos de producción incrementaran.

Figura 49

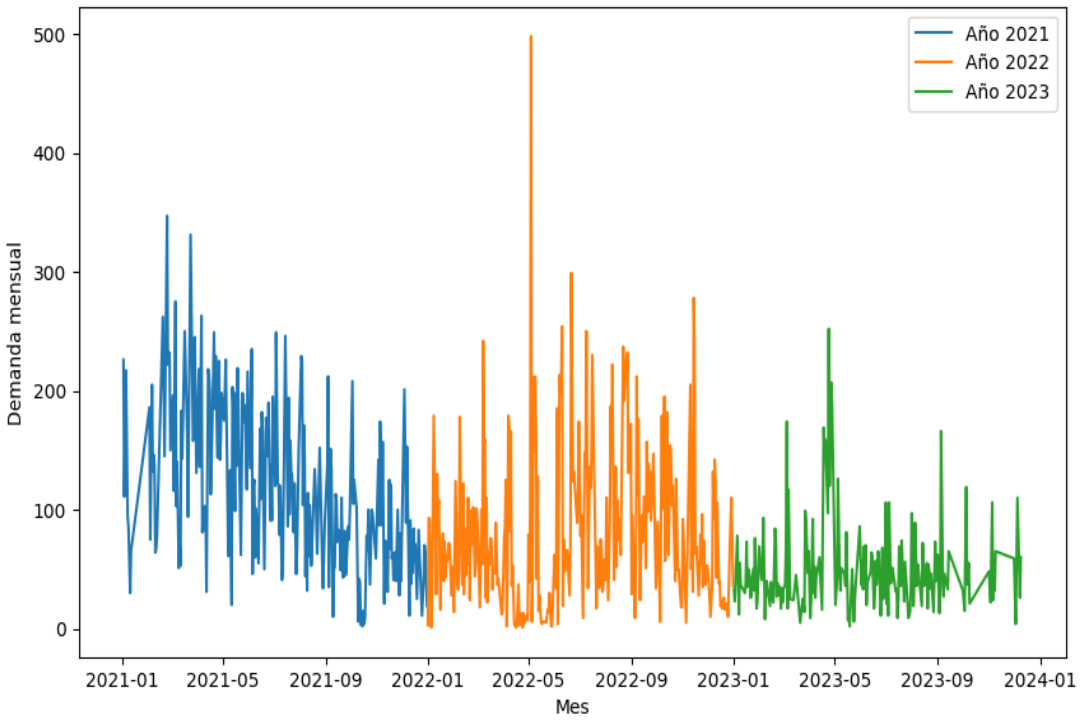
Comparación de Demanda histórica por meses 2021-2023



Nota. Elaboración propia.

Figura 50

Demanda Histórica de paquetes de huevos (15 huevos) 2021 - 2023



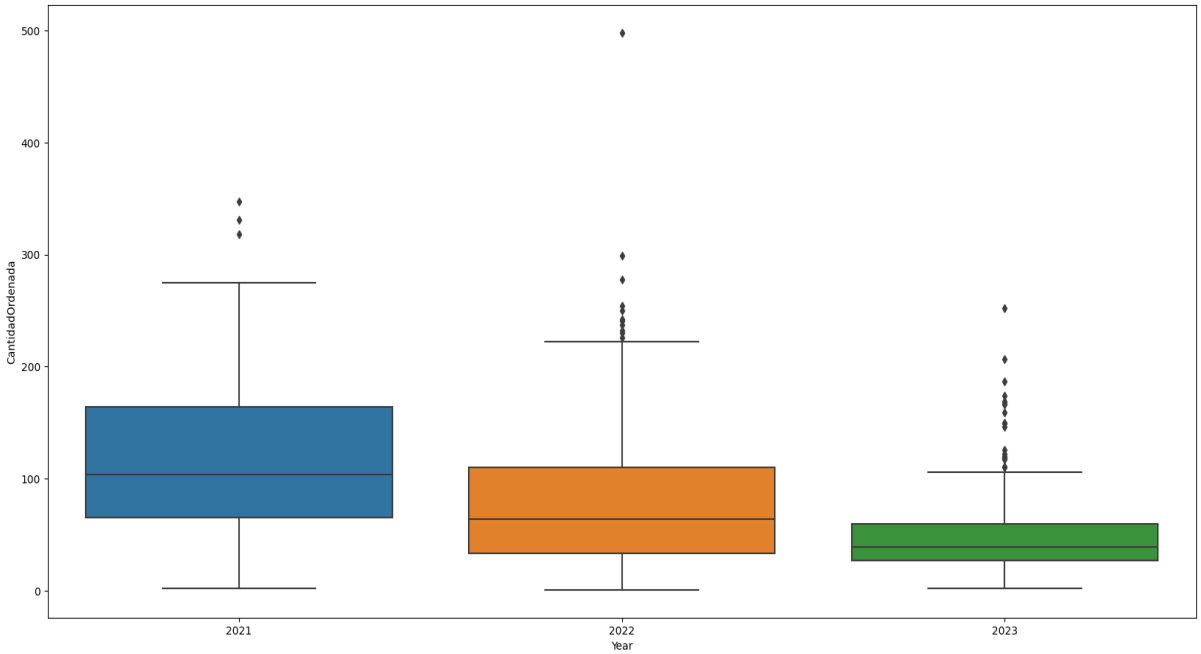
Nota. Elaboración propia.

Tendencia por año

A partir de las Figuras 51 y 52, se puede observar que durante el año 2021 se tuvo una mayor demanda de paquetes de huevos (15 huevos). No obstante, esta fue decreciendo durante el año 2022 y esta tendencia continuó hasta la fecha. Asimismo, se puede apreciar el gráfico de la demanda histórica de paquetes de huevos (15 huevos) la cual tiene una tendencia decreciente.

Figura 51

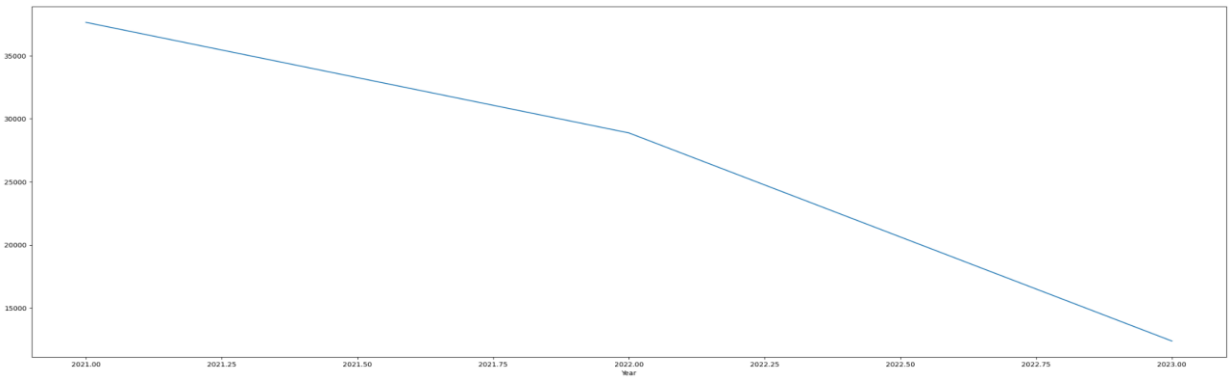
Demanda diaria de paquetes de huevos (15 huevos) por año



Nota. Elaboración propia.

Figura 52

Demanda histórica de paquetes de huevos (15 huevos)



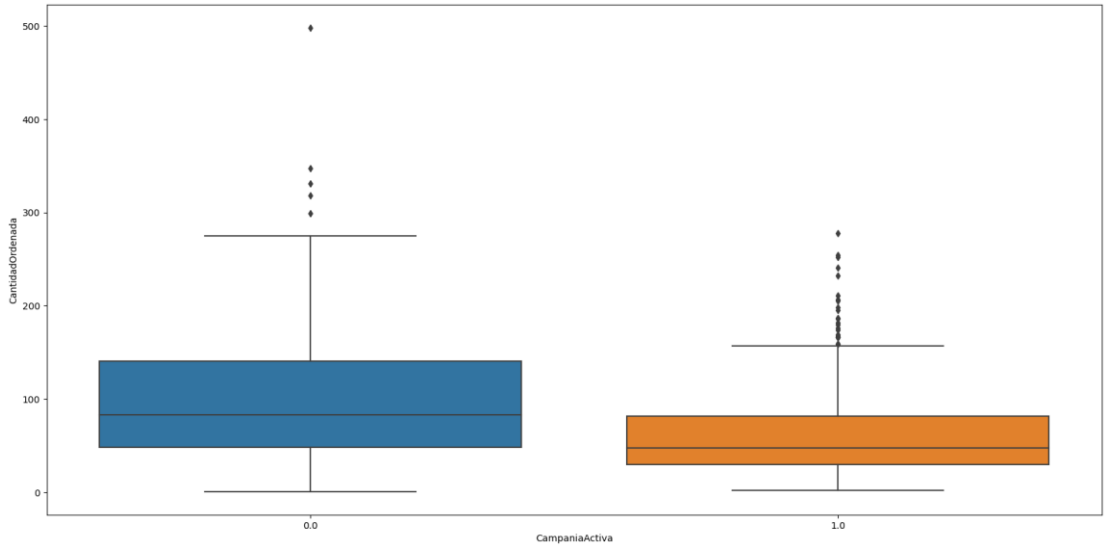
Nota. Elaboración propia.

Tendencia por activación de campañas

Se puede observar en las Figuras 53 y 54 que hay más ventas cuando no se ha realizado campañas y esto se debe a que tenemos pocas campañas en la base de datos. Cabe señalar que solo disponemos de campañas a partir del año 2022.

Figura 53

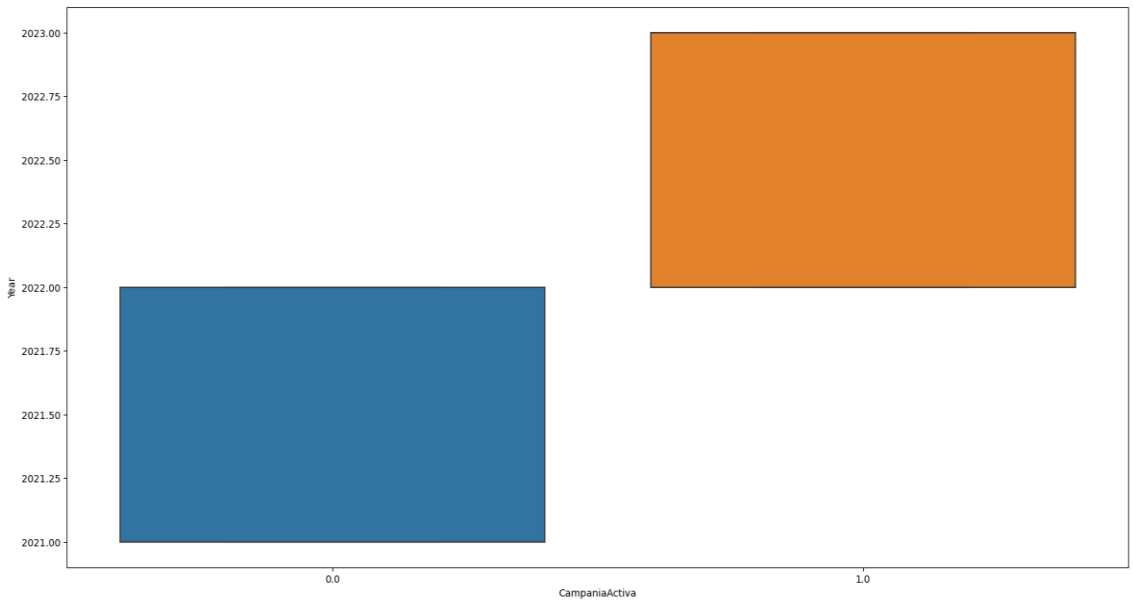
Boxplot de demanda de paquetes de huevos según Activación de campañas



Nota. Elaboración propia.

Figura 54

Demanda de paquetes de huevos (15 unid.) según Activación de campañas

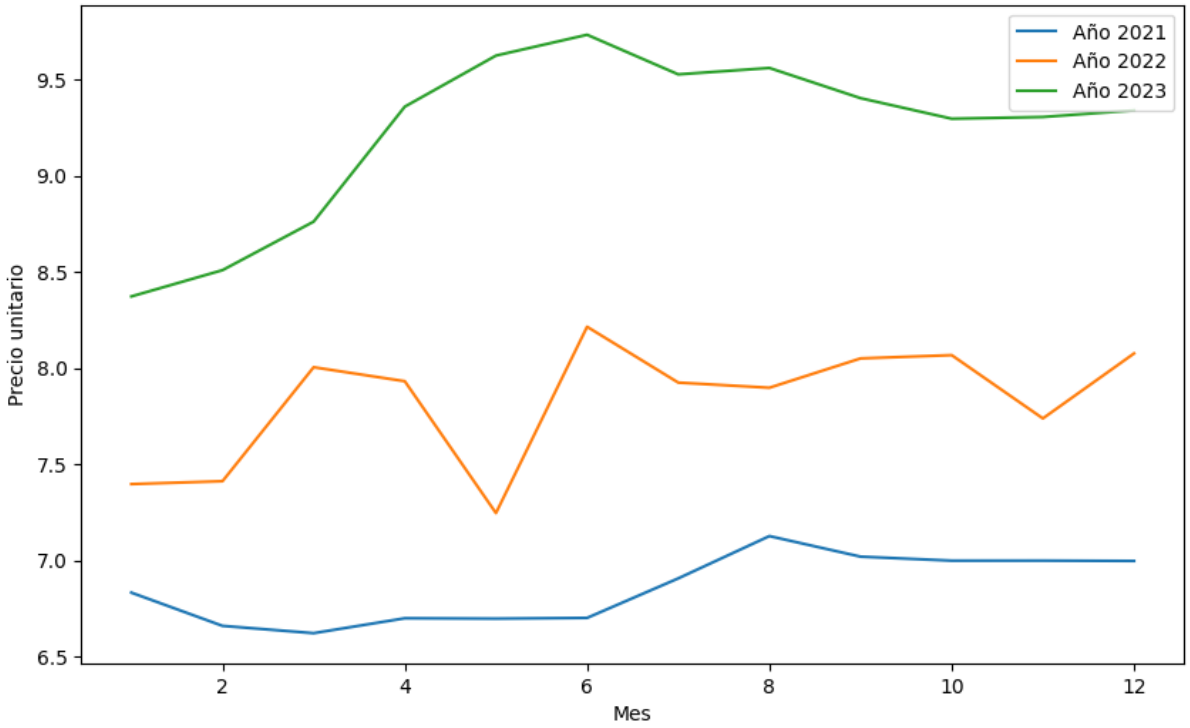


Nota. Elaboración propia.

Evolución del precio

Con respecto al precio, se puede observar en la Figura 55 que conforme van pasando los años, el precio ha aumentado. No obstante, en las gráficas anteriores podemos corroborar que los aumentos del precio no siempre han tenido un impacto directo sobre la demanda debido a temas coyunturales como la pandemia, inflación, etc. Mencionar que esto tiene que ver mucho con la parte de la teoría económica de oferta y demanda.

Figura 55
Evolución del precio de venta de paquetes de huevos (15 huevos)



Nota. Elaboración propia.

5.1.2.3. Pre procesamiento de la data

5.1.2.3.1. Renombre de variables

Con el fin de que el manejo de variable sea más entendible a nivel semántico, se procedió a cambiar el nombre de todas. Se detallan los cambios de la Variable Target en la tabla 15, mientras que los cambios de las Variables de Fecha se indican en la Tabla 16, los de las variables numéricas en la Tabla 17 y los cambios de las variables categóricas en la Tabla 18.

Tabla 15
Variable Target renombrada

NOMBRE ORIGINAL	NOMBRE NUEVO	TIPO
qty_ordered	CantidadOrdenada	Numércia

Nota. Elaboración propia.

Tabla 16
Variables Fecha renombradas

NOMBRE ORIGINAL	NOMBRE NUEVO	TIPO
date_order	FechaOrden	Fecha
commitment_date	FechaCompromiso	Fecha
expected_date	FechaEsperada	Fecha
confirmation_date	FechaConfirmada	Fecha

Nota. Elaboración propia.

Tabla 17
Variables numéricas renombradas

NOMBRE ORIGINAL	NOMBRE NUEVO	TIPO
amount_untaxed	MontoSinImpuestos	Numérica
amount_total	MontoTotal	Numérica
price_unit	PrecioUnitario	Numérica
qty_delivered	CantidadEntregada	Numérica
qty_ordered	CantidadOrdenada	Numérica
units_per_product	UnidadesPorProducto	Numérica
standard_price	PrecioEstandar	Numérica
list_price	PrecioLista	Numérica
Weight	Peso	Numérica

Nota. Elaboración propia.

Tabla 18
Variables categóricas renombradas

VARIABLE	DEFINICIÓN	TIPO

warehouse_id	AlmacenID	Categórica
company_name	Compania	Categórica
type_order_vale	ValorTipoOrden	Categórica
ecomerce_id	EcommerceID	Categórica
State	EstadoOrden	Categórica
Warehouse	Almacen	Categórica
company_id	CompaniaID	Categórica
Name	SocioID	Categórica
company_branch	Tienda	Categórica
type_order	TipoOrden	Categórica
order_source	OrigenPedido	Categórica
partner_id	Socio	Categórica
company_branch_id	TiendaID	Categórica
warehouse_code	AlmacenCodigo	Categórica
Id	ID	Categórica
price_unit	PrecioUnitario	Numérica
Product	Producto	Categórica
product_uom	ProductoUnidad	Categórica
product_id	ProductoID	Categórica
default_code	CodigoPredeterminado	Categórica
ESTADO_PEDIDO	EstadoOrden2	Categórica
sku_producto	ProductoSKU	Categórica
CAMPANA_ACTIVA	CampaniaActiva	Categórica
CYBER	Cyber	Categórica
category_id	CategoriaID	Categórica
category_name	Categoria	Categórica
product_type	ProductoTipo	Categórica
unidad_medida	UnidadMedida	Categórica

id_sap	SAPID	Categórica
ZONA_COBERTURA	ZonaCobertura	Categórica
estado_afiliada	EstadoAfiliada	Categórica

Nota. Elaboración propia.

A continuación, se muestra la codificación del cambio de nombres para una muestra de las variables:

Figura 56
Renombre de variables

```
# Renombramos variables
df_SF.rename(columns={'commitment_date':'FechaCompromiso'},inplace = True)
df_SF.rename(columns={'warehouse_id':'AlmacenID'},inplace = True)
df_SF.rename(columns={'company_name':'Compania'},inplace = True)
df_SF.rename(columns={'type_order_vale':'ValorTipoOrden'},inplace = True)
df_SF.rename(columns={'ecomerce_id':'EcommerceID'},inplace = True)
```

Nota. Elaboración propia.

5.1.2.3.2. Tratamiento de data duplicada

Se identificaron 417 registros de ventas duplicados. Se procedió a eliminarlos para que no altere la demanda real.

Figura 57
Tratamiento de data duplicada

```
# Cantidad de data duplicada
df_SF.duplicated().sum()
```

```
417
```

```
df_SF = df_SF.drop_duplicates()
df_SF.duplicated().sum()
```

```
0
```

Nota. Elaboración propia.

5.1.2.3.3. Cambio de tipo de variables

Si bien es cierto, anteriormente se categorizaron a las variables de acuerdo a su naturaleza, esto no quiere decir que python entienda lo mismo. Por ende, algunas variables tuvieron que ser convertidas, a continuación, se detalla las conversiones.

- **De tipo objetc a tipo date:** FechaOrden
- **De tipo int a tipo object:** AlmacenID, CompaniaID, TiendaID, ID, ProductoID, CampaniaActiva, Cyber y CategoriaID

Para ello se empleó el siguiente código:

Figura 58
Cambio de tipo de variables

```
# Cambiamos tipos de datos
# De object a datetime
df_SF['FechaOrden'] =  pd.to_datetime(df_SF['FechaOrden'])
df_SF['FechaOrden'] = df_SF['FechaOrden'].dt.date
# De int/float a object
df_SF['AlmacenID'] = df_SF['AlmacenID'].astype('object')
df_SF['CompaniaID'] = df_SF['CompaniaID'].astype('object')
df_SF['TiendaID'] = df_SF['TiendaID'].astype('object')
df_SF['ID'] = df_SF['ID'].astype('object')
df_SF['ProductoID'] = df_SF['ProductoID'].astype('object')
df_SF['CampaniaActiva'] = df_SF['CampaniaActiva'].astype('object')
df_SF['Cyber'] = df_SF['Cyber'].astype('object')
df_SF['CategoriaID'] = df_SF['CategoriaID'].astype('object')
```

Nota. Elaboración propia.

5.1.2.3.4. Eliminación de variables

Variables Fecha

Existen cuatro variables de tipo fecha que son FechaOrden, FechaCompromiso, Fecha Esperada y FechaConfirmada. Comparando estas variables, solo se encontró 16 registros con diferencias entre FecaEsperada y FechaConfirmada. No obstante, en algunos casos la diferencia podía ser de hasta 593 días debido a que por algún error de codificación se consideró una fecha esperada de hasta años después de la fecha confirmada. Por otra parte, solo el 10.3% de los pedidos en total presentan una diferencia mayor a un día entre la fecha de orden y la fecha esperada y en un 1.31% omitieron digitar la fecha esperada, para el 89% restante se espera que la fecha de entrega sea el mismo día que la fecha del pedido. Por ende, solo conservaremos la

variable FechaOrden. A continuación, se muestra a nivel de código cómo se eliminaron las demás variables y servirá de ejemplo para posteriores eliminaciones:

Figura 59
Eliminación de variables

```
df_SF.drop(['FechaCompromiso','FechaEsperada','FechaConfirmada'], axis=1, inplace=True)
df_SF.head()
```

Nota. Elaboración propia.

Variables equivalentes

Existen variables que aportan la misma información o muy parecida por lo que están fuertemente correlacionadas y en algunos casos hasta pueden generar confusión.

En primer lugar, se identificó que la variable Almacen y las variables AlmacenID y AlmacenCodigo representan la misma información. Por ende, se optó por mantener por ahora la variable Almacén debido a que es la única que almacena datos textuales.

Figura 60
Variable Almacen vs AlmacenID/Almacen

```
result = df_SF.groupby(['AlmacenID', 'Almacen']).size().reset_index(name='count')
print(result)
```

```
    AlmacenID                      Almacen   count
0           1                   My Company       6
1          16                      ARIDEL   10092
2          17                NUTRIABASTOS   71155
3          18          COMERCIAL CELA EIRL   19592
4          22                       DYCVAR    7687
5          23                     ANFERSEB   22335
6          33                      APETERM   17463
7          60      CASTELLARES ARAMBURÃ▯Â▯     869
8          62                     CRISELY   53880
9          65                         E&J   74623
10         67                GRUPO VERTIZ   26747
11         72                VALLE CEDRELA   97036
12         73          VERÃ▯Â▯NICA PALOMINO   53795
```

Nota. Elaboración propia.

El mismo comportamiento fue encontrado para las relaciones entre las variables Compania y CompaniaID, Tienda y TiendaID, Categoria y CategoriaID, Producto y ProductoID, ProductoUnidad y UnidadMedida y Socio y SocioID. En todos estos casos nos quedamos con la variable textual y eliminamos el ID.

Algo parecido se encontró entre las variables EstadoOrden y EstadoOrden2 donde se detctó que el 95.3% de los pedidos presentan un EstadoOrden corresponden a una venta concretada. No obstante, en términos de demanda lo que importa es tener el producto disponible para cuando el cliente lo pide independientemente de si luego modifica o cancela el pedido. Por ende, nos quedaremos con todos los registros y eliminaremos ambas variables. A continuación, se muestra el detalle numérico de ello:

Figura 61
Variable EstadoOrden vs EstadoOrden2

```
result = df_SF.groupby(['EstadoOrden', 'EstadoOrden2']).size().reset_index(name='count')
print(result)
```

```
  EstadoOrden EstadoOrden2   count
0      cancel    CANCELADO   10531
1       draft  MODIFICIADO     522
2       false    CANCELADO       1
3        sale  MODIFICIADO  388332
4        sale        VENTA  402363
5        sent  MODIFICIADO      10
```

Nota. Elaboración propia.

Por último, para el caso de las variables CampaniaActiva y Cyber se detectó que todas los cybers están contenidos dentro de CampaniaActiva por lo que solo nos quedaremos con esta debido a que en un futuro puede admitir más posibilidades de campañas y no solo cybers. A continuación, se muestra de forma numérica lo identificado:

Figura 62
Variable CampaniaActiva vs Cyber

```
result = df_SF.groupby(['CampaniaActiva', 'Cyber']).size().reset_index(name='count')
print(result)
```

```
   CampaniaActiva  Cyber   count
0               0      0  725497
1               1      1   76262
```

Nota. Elaboración propia.

5.1.2.3.5. Cardinalidad

A continuación, se analiza de forma numérica y/o gráfica la distribución de valores de las variables que fueron descartadas del análisis por distintas razones como, por ejemplo:

- Variables que no tienen datos
- Variables que solo tienen una categoría
- Variables cuyo mayor porcentaje se centra en solo una categoría o se acumulan en pocas categorías

Partimos con la variable EcommerceID en la que se tiene no solo un total de 163 551 valores distintos, sino que también 17 441 valores son de tipo false. Debido a la alta cardinalidad de esta variable se procedió a retirarla.

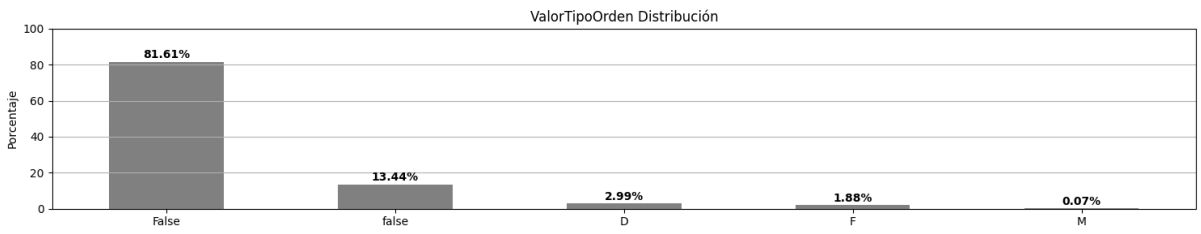
Figura 63
Cardinalidad de variable EcommerceID

```
df_SF.groupby(['EcommerceID'])['EcommerceID'].count()

EcommerceID
177869        5
177888        7
177889        9
177958        3
177984        5
             ...
744911        8
744926        8
744933        4
744937        9
false     17441
Name: EcommerceID, Length: 163551, dtype: int64
```

Nota. Elaboración propia.

Posterior a ello se identificó que para la variable ValorTipoOrden, según su gráfico de distribución mostrado en la Figura 64, que casi el total de pedidos (81.6%) tienen la clase "False". Debido a que la significancia de la variable se concentra en una sola variable y se tiene desconocimiento del significado de las otras categorías minoritarias, se procederá a retirar dicha variable de la base de datos.

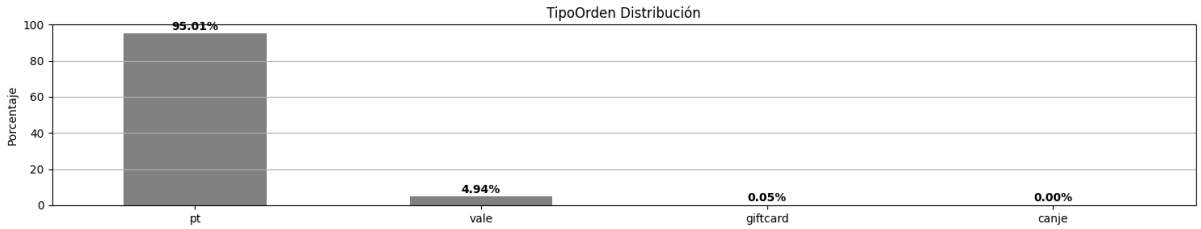
Figura 64
Distribución de valores - Variable ValorTipoOrden



Nota. Elaboración propia.

Un caso similar sucede con la variable TipoOrden, mostrado en la Figura 65, en la que se observa que el 95.01% de las órdenes fueron pagadas en su totalidad (pt) mientras que el 4.94% fueron vales, el 0.05% giftcard y el resto canje. Por ende, casi la totalidad de los pedidos se centran en una sola clase por lo que esta variable no se considerará como parte del modelo.

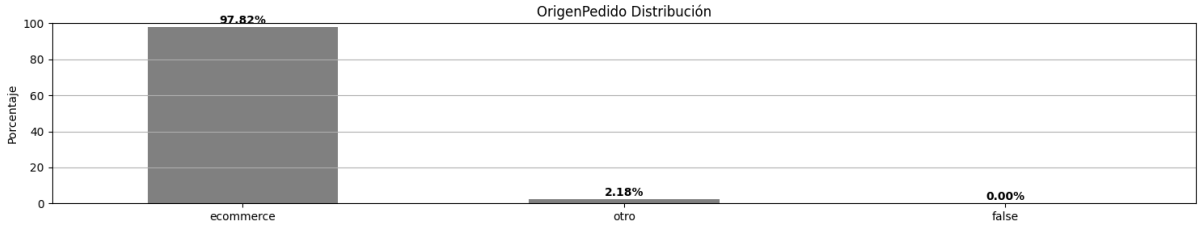
Figura 65
Distribución de valores - Variable TipoOrden



Nota. Elaboración propia.

Lo mismo sucede para la variable OrigenPedido en la que todo se centra en una clase debido a que el 97.8% de los valores corresponde a origen ecommerce, como se muestra en la Figura 66. No obstante, toda la data es de su canal ecommerce por lo que otros valores podrían ser el resultado de una mala digitación. Por esta razón, se descarta esta variable del modelamiento.

Figura 66
Distribución de valores - Variable OrigenPedido



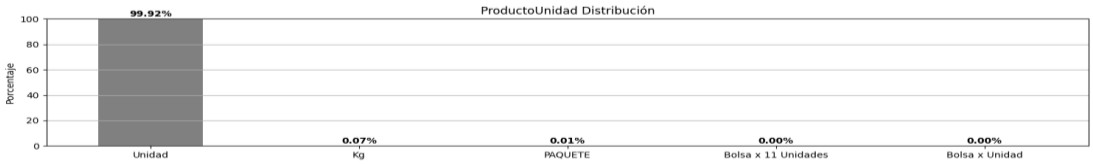
Nota. Elaboración propia.

Para el caso de la variable Socio, tenemos a 46 471 personas que vienen realizando compras y aunque algunos pueden tener hasta 60 compras en total, la cardinalidad es extremadamente alta y esto quita precisión a los modelos por lo que se excluirá esta variable.

Respecto a la variable ProductoUnidad, cuya distribución es mostrada en la Figura 67, esta se descarta debido a que el el 99.9% de los valores pertenecen a la clase “unidad” además

de que, al filtrar solo por el producto de interés, esta variable se volverá completamente homogénea no aportando información alguna para el modelo.

Figura 67
Distribución de valores - Variable ProductoUnidad

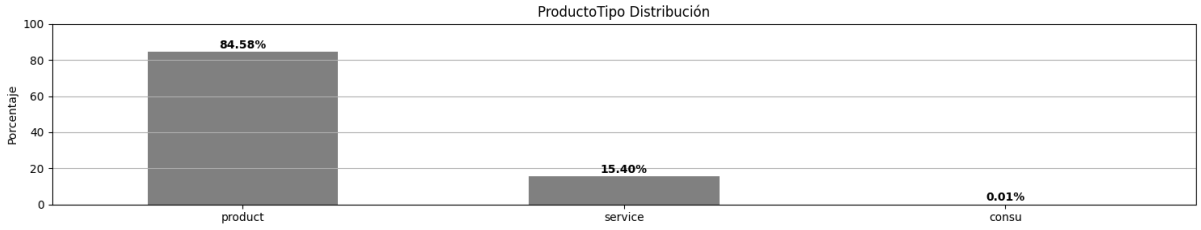


Nota. Elaboración propia.

Asimismo, tenemos el caso de la variable CodigoPredeterminado que teóricamente identifica al producto; sin embargo, este código no se debe usar debido a que 118 presentan un valor false, un grupo incluyen la palabra [OLD] haciendo referencia a que están desactualizados y otro grupo incluye de forma directa la indicación (no usar). Por todas estas razones, esta variable se retirará del modelo.

Para la variable Categoría del producto, mostrada en la Figura 68, esta se volverá homogénea cuando se filtre por producto y, de igual manera, sucede con la variable ProductoTipo en la que incluso se observa que el 84.6% de valores se concentra en la clase “product”. Por ende, ambas variables serán eliminadas.

Figura 68
Distribución de valores - Variable ProductoTipo



Nota. Elaboración propia.

A continuación, se muestran todas las variables que fueron eliminadas en esta etapa:

Figura 69
Eliminación de variables por cardinalidad

```
# Eliminamos variables no relevantes
df_SF.drop(['ValorTipoOrden','EcommerceID','TipoOrden','OrigenPedido','Socio','ProductoTipo',
            'ProductoUnidad','CodigoPredeterminado','Categoria'], axis=1, inplace=True)
df_SF.head()
```

Nota. Elaboración propia.

Variables postpedido

Existen variables que al generarse luego de concretar un pedido no nos ayudan a predecir nuestro objetivo. Este es el caso de la variable CantidadEntregada por lo que será eliminada.

Variables desconocidas

Por otra parte, existen variables de las que se desconoce su significado como el caso de la variable ID que además también presenta una alta cardinalidad. Por ende, será retirada.

5.1.2.3.6. Tratamiento de valores perdidos

Se identificó la cantidad de datos nulos por variable y su porcentaje respecto al total de registros. En base a ello se tomaron decisiones de eliminación o imputación mediante los siguientes criterios:

- % NAs < = 8%: eliminación de valores nulos
- 8% < % NAs < = 12%: imputación de variable
- % NAs > = 13%: eliminación de variable

De los cálculos, se obtuvo y decidió lo siguiente:

Figura 70
Valores nulos

```
# RATIO DE VALORES NULOS
na_ratio = ((df_SF.isnull().sum() / len(df_SF))*100).sort_values(ascending = False)
print("% TOTAL DE NAs:", sum(((df_SF.isnull().sum() / len(df_SF))*100)))
print("-----------------------------------")
print("% INDIVIDUAL DE NAs")
print(na_ratio)
```

```
% TOTAL DE NAs: 261.8841821544878
-----------------------------------
% INDIVIDUAL DE NAs
ProductoSKU             90.488164
SAPID                   59.956171
Peso                    50.880751
PrecioEstandar          35.610576
UnidadesPorProducto     18.033224
PrecioLista              6.296905
ZonaCobertura            0.309195
EstadoAfiliada           0.309195
Producto                 0.000000
Almacen                  0.000000
CampaniaActiva           0.000000
CantidadOrdenada         0.000000
PrecioUnitario           0.000000
MontoTotal               0.000000
Tienda                   0.000000
FechaOrden               0.000000
MontoSinImpuestos        0.000000
Compania                 0.000000
dtype: float64
```

Nota. Elaboración propia.

- Eliminar variables: ProductoSKU, SAPID, Peso, PrecioEstandar y UnidadesPorProducto.
- Eliminar, aunque % NAs < 8%: la variable PrecioLista es el precio sugerido por la empresa pero no es el fijo, por lo tanto la eliminamos y en su reemplazo tenemos a PrecioUnitario.
- Imputación de variables: se eliminaron los registros con NAs para ZonaCobertura y EstadoAfiliada. Esto último se hizo de la siguiente manera:

Figura 71
Imputación de NAs

```
# CRITERIO: %NAs <= 8%
# DECISIÓN: eliminar registros
# JUSTIFICACIÓN: no afecta a la significancia de la data
df_SinNA = df_SF.dropna()
df_SinNA = df_SinNA.reset_index(drop=True)
print ("Shape of dataset before drop variables: ", df_SF.shape)
print ("Shape of dataset after drop variables: ", df_SinNA.shape)
```

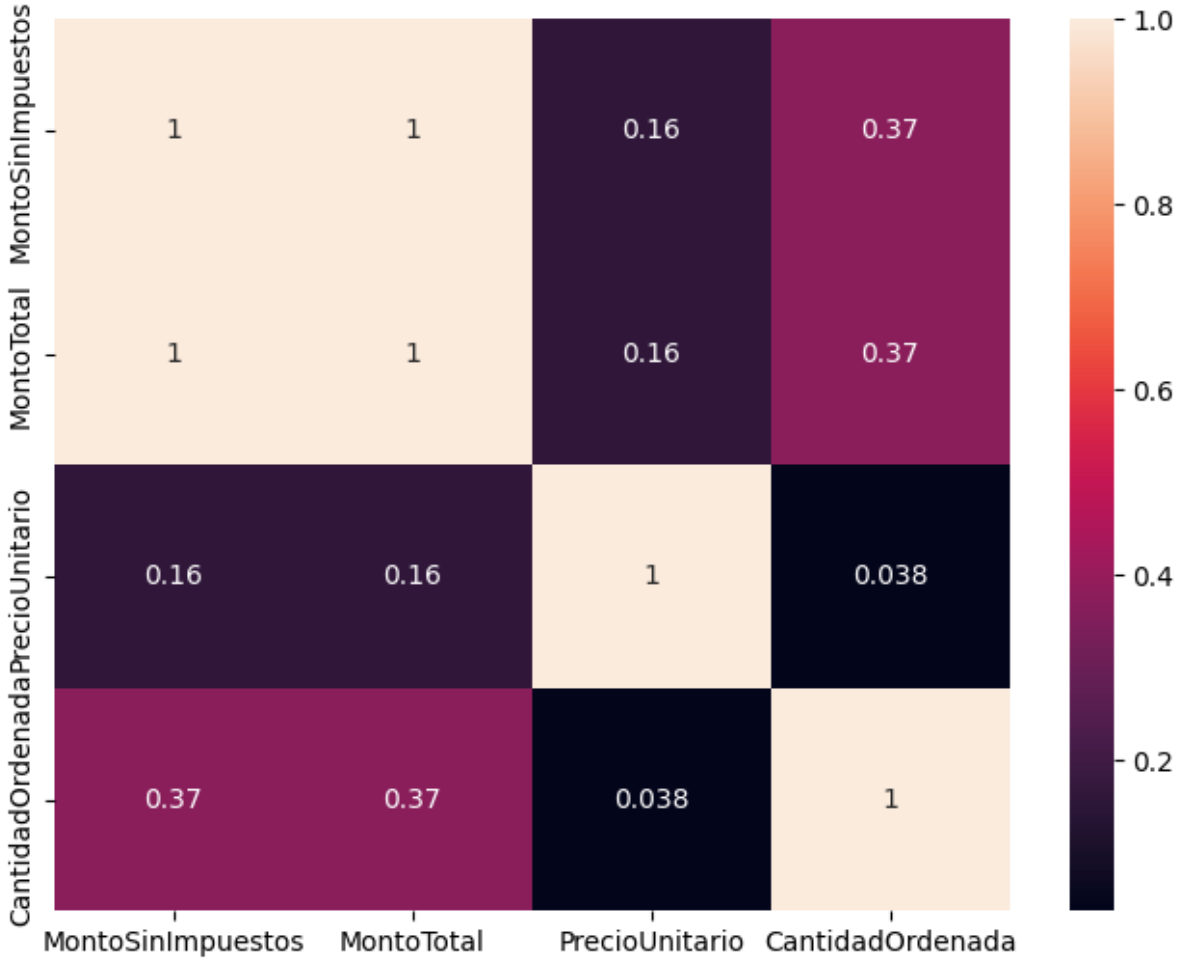
```
Shape of dataset before drop variables:  (801759, 12)
Shape of dataset after drop variables:  (799280, 12)
```

Nota. Elaboración propia.

5.1.2.3.5. Correlación de variables

Empecemos analizando la correlación entre las variables numéricas, para ello usaremos el coeficiente de correlación de Pearson. De acuerdo a la teoría, cuando estos valores son cercanos a uno las variables se correlacionan fuertemente por lo que se debe eliminar al menos una de ellas.

Figura 72
Matriz de correlación de variables numéricas

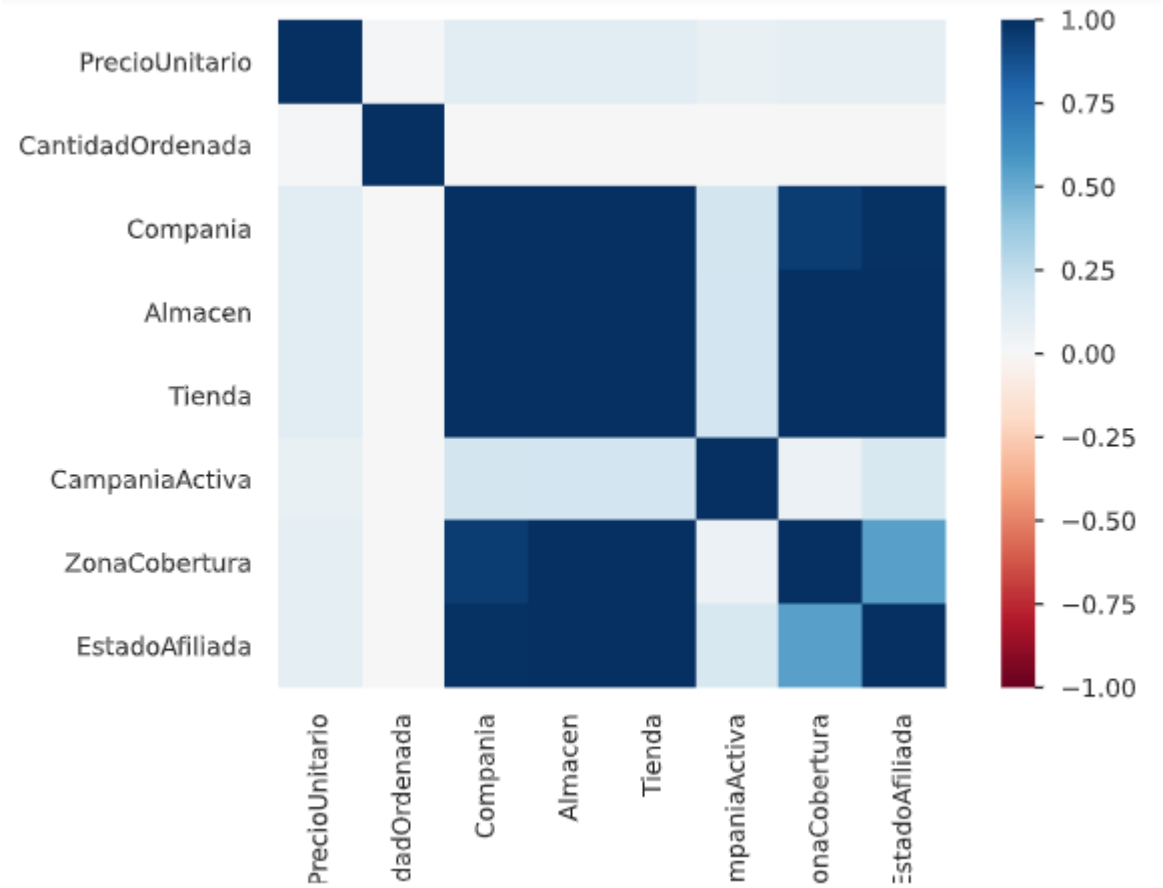


Nota. Elaboración propia.

De la Figura 72 se evidencia que la variable MontoTotal tiene una correlación igual a uno con la variable MontoSinImpuestos lo cual es bastante lógico debido a que la primera es la suma de la segunda más los impuestos. Asimismo, aunque en menor medida, ambas variables dependen de la cantidad ordenada y el precio del producto por lo que se eliminarán ambas al considerarse variables derivada o casi postpedido.

Posterior a ello se hará el mismo análisis para el resto de variables que quedan, incluyendo las categóricas. Debido a que en este caso el cálculo es más complejo, nos apoyaremos de la librería pandas_profiling la cual nos arroja el siguiente resultado:

Figura 73
Vista gráfica de matriz de correlación de variables restantes



Nota. Elaboración propia.

A continuación, en la Figura 74, se puede apreciar las mismas relaciones, pero a nivel numérico:

Figura 74
Vista numérica de matriz de correlación de variables

	PrecioUnitario	CantidadOrdenada	Compania	Almacen	Tienda	CampaniaActiva	ZonaCobertura	EstadoAfiliada
PrecioUnitario	1	0.008	0.11	0.112	0.112	0.071	0.101	0.101
CantidadOrdenada	0.008	1	0.003	0.007	0.007	0	0.007	0.002
Compania	0.11	0.003	1	1	1	0.182	0.951	0.991
Almacen	0.112	0.007	1	1	1	0.188	1	1
Tienda	0.112	0.007	1	1	1	0.188	1	1
CampaniaActiva	0.071	0	0.182	0.188	0.188	1	0.055	0.157
ZonaCobertura	0.101	0.007	0.951	1	1	0.055	1	0.546
EstadoAfiliada	0.101	0.002	0.991	1	1	0.157	0.546	1

Nota. Elaboración propia.

De la matriz se puede identificar que en rojo se encuentran todas las variables que se encuentran fuertemente correlacionadas con un valor de coeficiente de correlación de pearson igual a 1. A continuación se listan estas relaciones:

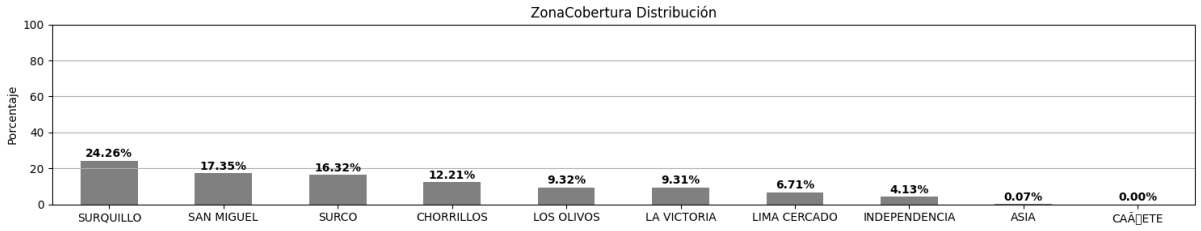
- Compania con las variables Almacen y Tienda.
- Almacen con las variables Compania, Tienda, ZonaCobertura y EstadoAfiliada.
- Tienda con Compania, Almacen, ZonaCobertura y EstadoAfiliada.
- ZonaCobertura con Almacen y Tienda.
- EstadoAfiliada con Almacen y Tienda.

Asimismo, de la matriz se puede identificar que en amarillo se encuentran todas las variables que se encuentran fuertemente correlacionadas, pero con un valor de coeficiente de correlación de pearson entre 0.7 y 0.9. A continuación se listan estas relaciones:

- Compania con ZonaCobertura y EstadoAfiliada
- ZonaCobertura con Compania.
- EstadoAfiliada con Compania.

De todas estas correlaciones fuertes se optó por eliminar todas las variables excepto ZonaCobertura que parecía ser la menos correlacionada. No obstante, al analizar su distribución que es bastante heterogénea en donde los distritos con mayor número de órdenes registradas son Surquillo, San Miguel, Surco y Chorrillos, concentrando casi el 70% de los casos. Por ende y dado que buscamos predecir la demanda para todo Lima en general, se optó por también eliminar estas variables.

Figura 75
Distribución de valores - Variable ZonaCobertura



Nota. Elaboración propia.

Ya teniendo el pre procesamiento casi listo, se filtró la data correspondiente solo a Producto = Paquete de huevos (15 huevos) que es el producto más demandado y en ese momento la variable Producto se volvió homogénea por lo que también se eliminó. Es así que pasamos de tener un total de 43 variables al inicio del pre procesamiento, a tener solo 4 al final de este proceso exhaustivo.

Por último, se procedió a hacer un análisis de outliers debido a que las variables predictoras resultantes son numéricas al igual que la variable a predecir. Para ello nos apoyamos de diagramas de cajas o también conocidos como boxplots que básicamente muestran la distribución de los valores de una variable a través de sus cuantiles. Para ello nos ayudamos del siguiente código:

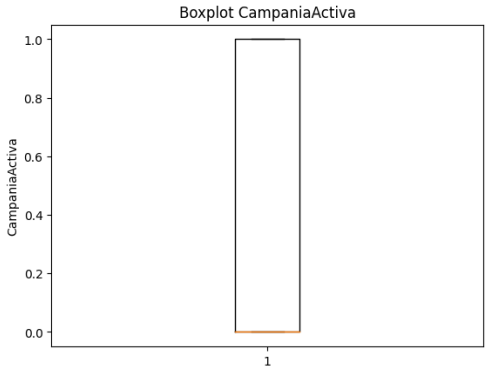
Figura 76
Construcción de boxplot

```
import matplotlib.pyplot as plt
plt.boxplot(df_producto["CampaniaActiva"])
plt.ylabel("CampaniaActiva")
plt.title("Boxplot CampaniaActiva")
```

Nota. Elaboración propia.

En primer lugar, analizamos la variable CampaniaActiva; no obstante, no se encontraron outliers debido a que esta variable solo admite dos posibles valores: un cero cuando no hay campaña y un uno cuando sí existe. Como se aprecia en el siguiente boxplot, la distribución no presenta anomalías.

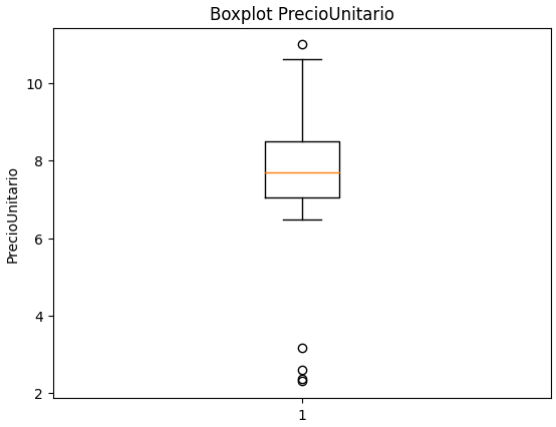
Figura 77
Boxplot – Campaña Activa



Nota. Elaboración propia.

El mismo ejercicio se hizo para la variable PrecioUnitario y tal como se ve en el boxplot, se identificó que existen algunos outliers para cuando el precio estuvo por debajo de los S/4.00 así como para cuando estuvo por encima de los S/10.00. No obstante, no se aplicó ningún tratamiento debido a que como ya se analizó anteriormente, el precio ha ido subiendo con el tiempo y no siempre ha tenido una relación esperada con la demanda como lo dice las leyes económicas.

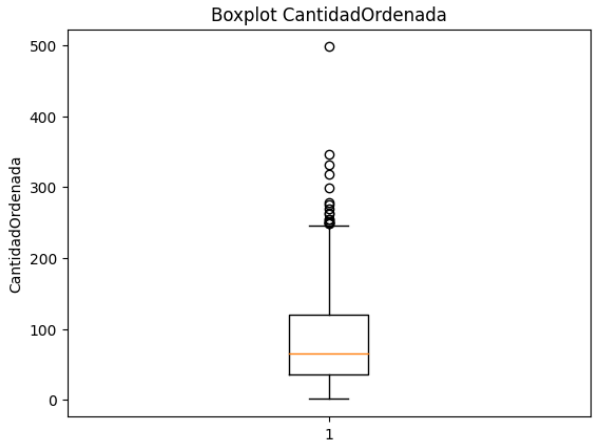
Figura 78
Boxplot – Precio Unitario



Nota. Elaboración propia.

Por último, analizamos la distribución de la variable target. Tal como se ve en el siguiente diagrama de cajas, existen varios outliers por encima de los 250 pedidos. No obstante, en este trabajo se defiende la postura de que un incremento de pedidos corresponde a la realidad de ventas por lo que aportan información importante a los modelos para futuras predicciones. Por lo tanto, no se aplicará ninguna técnica de tratamiento. Caso similar sucede para cuando no existieron compras.

Figura 79
Boxplot – Cantidad Ordenada



Nota. Elaboración propia.

Con el fin de saber la cantidad exacta de outliers, se aplicó el siguiente código de python. En este se empieza escalando la variable CantidadOrdenada utilizando el método Z-score. Luego, se establece un umbral = 2 en valor absoluto para ser comparado con las órdenes escaladas y los resultados se almacenan en la nueva columna de nombre out_SP. Es así que si es un outlier se asigna un true y caso contrario un false. Por último, se cuenta cuántos outliers se encontraron en los datos al contar cuántas filas tienen el valor. Como resultado se contaron 48 trues, lo que representa el 5.16% de la data en cuanto a días. Al ser un % bastante bajo y sumado a la justificación mencionada anteriormente, se mantiene la postura de incluir dichos valores.

Figura 80
Conteo de outliers

```
# Z-SCORE: 48 outliers
from sklearn import preprocessing
# Escalo variable con Z-score
p1=preprocessing.scale(df_producto["CantidadOrdenada"])
# Establesco límite para outlier
df_producto["out_SP"]=abs(p1)>2
len(df_producto[df_producto["out_SP"]==True])
```

```
48
```

Nota. Elaboración propia.

5.1.2.4. Modelamiento

En esta etapa y ya con la data limpia, se entrenaron una serie de modelos de distinto nivel de complejidad.

SARIMAX

Para ello primero se convirtió la data a una serie de tiempo siguiendo los mismos pasos que se detallaron al inicio de la etapa de Exploración de la data. Posterior a ello se segmentó la data en 80% para entrenamiento y 20% para test; es decir, los primeros 752 días para el entrenamiento y los últimos 188 para el test. Para ello se estableció un orden de entrenamiento de (1,1,0), debido a que es un buen valor base, y se usó el siguiente código de python:

Figura 81
Entrenamiento de modelo SARIMAX - orden (1,1,0)

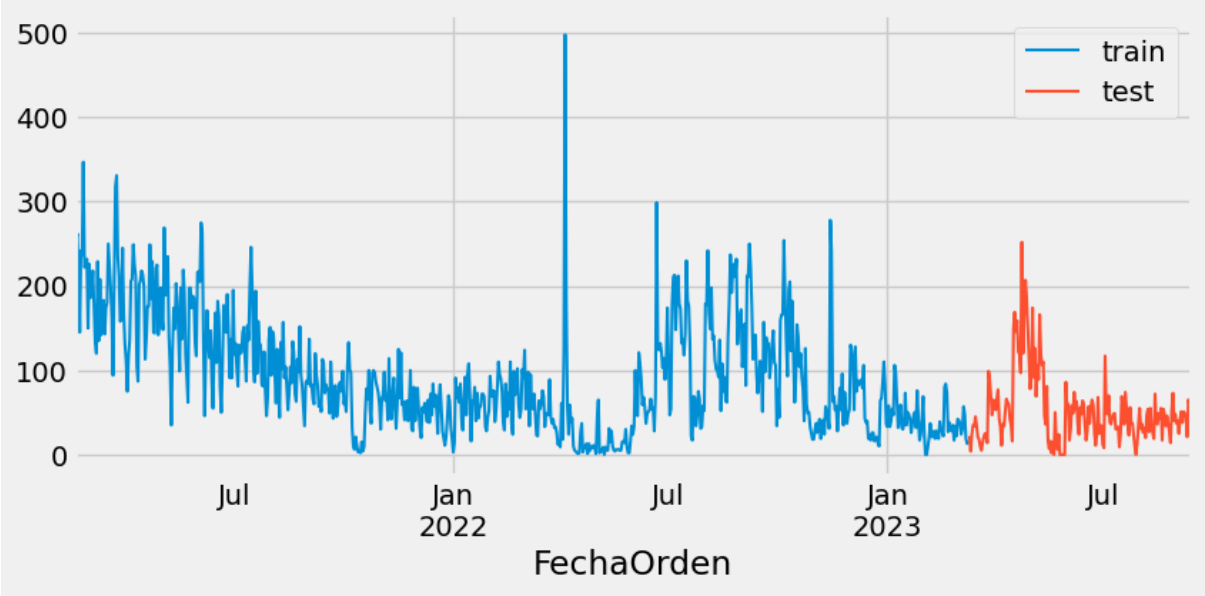
```
from statsmodels.tsa.arima.model import ARIMA
train = df_producto["CantidadOrdenada"][:752]
test = df_producto["CantidadOrdenada"][752:]
# p,d,q ARIMA Model
model = ARIMA(train, order=(1,1,0))
model_fit = model.fit()
print(model_fit.summary())
```

Nota. Elaboración propia.

Forecast autorregresivo simple

Se entrenó tres modelos autorregresivos recursivos que son un tipo de modelo de series temporales que predice en base a valores pasados, pero también predicciones previas. Para ello primero se completó la serie al igual que en el modelo anterior. Posterior a ello se dividió la data en train y test con la relación 80% y 20% respectivamente.

Figura 82
Distribución de data en train y test



Nota. Elaboración propia.

Para el entrenamiento del modelo se consideraron tres regresores: LinearRegression, GradientBoostingRegressor y MLPRegressor que se reemplazaron en donde aparece el recuadro verde de la siguiente Figura 83. Asimismo, se estableció un lags = 15 para que también tome en cuenta la demanda de los últimos 7 días. El código usado para el entrenamiento fue el siguiente:

Figura 83
Entrenamiento de modelo ForecasterAutoreg con Regresor Lineal

```
# Crear y entrenar forecaster
# ==============================================
forecaster = ForecasterAutoreg(
                regressor = LinearRegression(),
                lags = 15
             )

forecaster.fit(y=datos_train['CantidadOrdenada'])
forecaster
```

Nota. Elaboración propia.

Regresión lineal simple

Para este modelo se buscó establecer la relación entre la fecha y la demanda. Esto podría sonar que es similar a lo antes explicado en las series de tiempo, pero la diferencia es que se toma a la fecha como una variable y no como un index. Aclarar que no se tomó al precio como variable independiente debido a que sus fluctuaciones son pocas y en el caso de campaña activa solo se dispone de data a partir del segundo año y sus valores solo son 0 y 1.

En primer lugar y dado que la fecha en su formato natural no puede ser leída por una regresión lineal, se hizo una conversión numérica de tal forma que la primera fecha es equivalente a 1 y se va incrementando de uno en uno. Para ello usamos el siguiente código:

Figura 84
Conversión de fecha en número ordinal

```
# Agrega una nueva columna numérica 'Numero_Dia' desde 1 en adelante
df_producto['FechaOrden'] = range(1, len(df_producto) + 1)
```

Nota. Elaboración propia.

Posterior a ello se dividió la data en 80% para entrenamiento y 20% para test. En esta ocasión se usó una forma más directa. Asimismo, se usó un valor semilla de 123 que se viene poniendo en todos los modelos que admiten ese parámetro y un shuffle = True que nos asegura que los datos se mezclen antes de ser divididos en train y test. El código se muestra a continuación:

Figura 85
Train y test - forma directa

```
# División de los datos en train y test
# ======================================================================
X = df_producto[['FechaOrden']]
y = df_producto['CantidadOrdenada']

X_train, X_test, y_train, y_test = train_test_split(
                                       X.values.reshape(-1,1),
                                       y.values.reshape(-1,1),
                                       train_size   = 0.80,
                                       random_state = 123,
                                       shuffle      = True
                                   )
```

Nota. Elaboración propia.

Posteriormente se entrenó el modelo de la siguiente manera:

Figura 86
Entrenamiento de regresión lineal simple

```
# Creación del modelo
# ==============================================================
modelo = LinearRegression()
modelo.fit(X = X_train.reshape(-1, 1), y = y_train)
```

Nota. Elaboración propia.

Regresión múltiple

Luego de haber entrenado series de tiempos y regresiones lineales, se procedió a entretener regresiones múltiples. Esto quiere decir que ahora se tomará en cuenta más de una variable independiente. Para ser exacto, todas las que quedaron luego de la etapa de pre procesamiento: PrecioUnitario y CampaniaActiva.

Para ello en primer lugar se agrupó la demanda por día, pero ahora con la aclaración de que se tomaron más criterios para agrupar las otras variables independientes. Es así que el PrecioUnitario será el promedio de todos los precios del día y lo mismo sucederá para CampaniaActiva aunque en este caso no generará cambio alguno debido a que se tiene la certeza de que por día hubo o no hubo campaña. Esto se logra mediante el siguiente código:

Figura 87
Agrupación de demanda para regresión múltiple

```
# Nuevo dataframe con suma de pedidos por día
df_producto = data.groupby('FechaOrden').agg({
    'CantidadOrdenada': 'sum',
    'PrecioUnitario': 'mean',
    'CampaniaActiva': 'mean'
}).reset_index()
df_producto.head()
```

Nota. Elaboración propia.

Posterior a ello verificamos que las fechas estén completas y en caso se carezca de datos para alguna, se reemplazará la CantidadOrdenada con cero debido a que no hubo compra, el PrecioUnitario por el promedio de todos los precios y la CampaniaActiva con cero que es el equivalente a que no hubo. Para ello usamos el siguiente código:

Figura 88
Completitud de datos para fechas sin datos

```
df_producto['CantidadOrdenada'].fillna(0,inplace=True)
df_producto['PrecioUnitario'].fillna(df_producto['PrecioUnitario'].mean(),inplace=True)
df_producto['CampaniaActiva'].fillna(0,inplace=True)
```

Nota. Elaboración propia.

Finalmente se entrenó el modelo con el mismo método que se vio durante el curso de Machine Learning con la siguiente programación:

Figura 89
Entrenamiento de regresión lineal múltiple

```
# Creando a mi alumno
LR_model = LinearRegression()
# Estudia
LR_model.fit(X_train,Y_train)
# Examen
Y_pred = LR_model.predict(X_test)
```

Nota. Elaboración propia.

Forecast autorregresivo múltiple

Similar a la regresión múltiple, primero nos aseguramos de que las fechas estén completas mediante el mismo procedimiento del modelo anterior. Posterior a ello entrenamos el modelo con dos regresores: Ridge y GradientBoostingRegressor que se colocarán en el recuadro verde señalado en la Figura 90. Asimismo, consideramos una semilla de 123 y 7 lags para tomar en cuenta la data de los últimos 7 días.

Figura 84

Entrenamiento de Forecast autorregresivo múltiple

```
# Crear y entrenar forecaster
# ======================================================
forecasterR = ForecasterAutoreg(
                regressor     = Ridge(),
                lags          = 7
             )

forecasterR.fit(y=datos_train['CantidadOrdenada'])
forecasterR
```

Nota. Elaboración propia.

Random Forest

Un modelo que suele tener buenos resultados es el Random Forest debido a que suelen ser muy precisos, son resistentes a los sobreajustes, pueden manejar tanto datos faltantes como outliers y además son robustos ante la presencia de multicolinealidad. Por ende, optamos por entrenar este modelo.

En un principio, luego de asegurarnos de que las fechas estén completas como en los dos últimos modelos, se procedió a entrenar con 100 estimadores que indican el número de árboles individuales. Es importante destacar que estos datos se encuentran en su estado original después del preprocesamiento y no han sido sometidos aún a ningún proceso de normalización. Para ello usamos el siguiente código:

Figura 90

Entrenamiento de Random Forest con data sin normalizar

```
# Fitting Random Forest Regression to the dataset
# import the regressor
from sklearn.ensemble import RandomForestRegressor

# create regressor object
RF_model = RandomForestRegressor(n_estimators=100,
                                 random_state=0)
# Estudia
RF_model.fit(X_train,Y_train)
# Examen
Y_pred = RF_model.predict(X_test)
```

Nota. Elaboración propia.

Al ver que los resultados eran mejores que los anteriores modelos, decimos incluir una nueva variable que es weekday debido a que, durante el análisis de series de tiempo, encontramos que había una estacionalidad a nivel de día de semana. Los resultados fueron mejores; no obstante, con el objetivo de obtener mejores resultados se procedió a volver a entrenar el modelo, pero normalizando las variables independientes para que tanto la variable PrecioUnitario, CampanaActiva y Weekday se encuentren en el mismo rango numérico. Para tal fin se usó un escalamiento estándar que en esencia convierte todas las variables para que tengan una media cercana a cero y una desviación estándar cercana a uno. Esto lo logramos mediante una línea simple de código:

Figura 91
Escalamiento de variables

```
from sklearn.preprocessing import StandardScaler
# Crear un objeto Scaler
scaler = StandardScaler()
```

Nota. Elaboración propia.

Finalmente se volvió a entrenar el modelo con un cross validation de K = 5 que quiere decir que el modelo se entrenará cinco veces y en cada iteración los datos de entrenamiento y testeo serán diferentes. Cabe señalar que previamente se hizo un Grid Search que en esencia significa indicarle al modelo que asuma distintas combinaciones de parámetros para que pueda determinar cuáles son los mejores en base al cálculo del MSE. En este caso se indicaron distintos valores para el número de estimadores, máxima profundidad, el número mínimo de muestras requeridas para dividir un nodo interno y número mínimo de muestras requeridas en una hoja del árbol. Esto se logra mediante el siguiente código:

Figura 92
Grid Search de Random Forest

```
from sklearn.model_selection import GridSearchCV
from sklearn.ensemble import RandomForestRegressor
from sklearn.preprocessing import StandardScaler
from sklearn.pipeline import Pipeline

# Crear un modelo de Random Forest
RF_model = RandomForestRegressor(random_state=0)

# Crear un objeto Scaler
scaler = StandardScaler()

# Definir el espacio de búsqueda de hiperparámetros, incluyendo los del modelo
param_grid = {
    'regressor__n_estimators': [50, 100, 150],
    'regressor__max_depth': [None, 10, 20, 30],
    'regressor__min_samples_split': [2, 5, 10],
    'regressor__min_samples_leaf': [1, 2, 4]
}

# Crear una tubería que incluye la normalización y el modelo
pipeline = Pipeline([
    ('scaler', scaler),
    ('regressor', RF_model)
])

# Crear un objeto GridSearchCV
grid_search = GridSearchCV(estimator=pipeline, param_grid=param_grid,
                           scoring='neg_mean_squared_error', cv=5, n_jobs=-1)

# Ajustar GridSearchCV a tus datos
grid_search.fit(X_train, Y_train)
```

Nota. Elaboración propia.

5.1.2.3. Análisis de resultados

SARIMAX

Del modelo se calcularon sus respectivos indicadores y se obtuvo un MSE de 3118.40 y por ende un RMSE de 55.84. Estos valores, sobre todo, el MSE es muy alto por lo que nos indica errores bastantes altos. Para ello utilizamos el siguiente código:

Figura 93
Cálculo de MSE - SARIMAX

```
predictions = model_fit.forecast(len(test))
print(f'SARIMAX Model Test Data MSE: {np.mean((predictions.values - test.values)**2):.3f}')
```

```
SARIMAX Model Test Data MSE: 3118.399
```

Nota. Elaboración propia.

Asimismo, se obtuvo un MAE de 39.188 y un MAE% de 46.40% que quiere decir que ese es el error porcentual que se espera en las predicciones. Para ello utilizamos el siguiente código:

Figura 94
Cálculo de MAE - SARIMAX

```
predictions = model_fit.forecast(len(test))
mae = np.mean(np.abs(predictions.values - test.values))
print(f'SARIMAX Model Test Data MAE: {mae:.3f}')
```

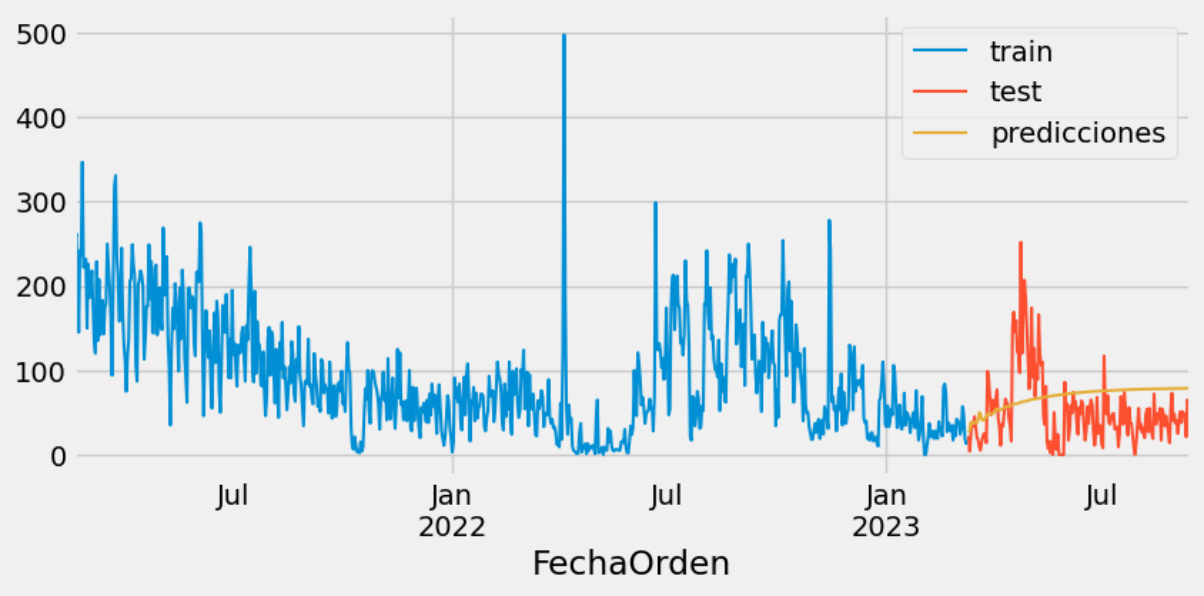
```
SARIMAX Model Test Data MAE: 39.188
```

Nota. Elaboración propia.

Forecast autorregresivo

Para el caso del LinearRegression se obtuvo un MSE de 2242.55 y un MAE de 38.80. A simple vista se puede ver que la predicción no se ajusta bien dado que las predicciones tienen a una subida fuera del rango normal de la demanda real.

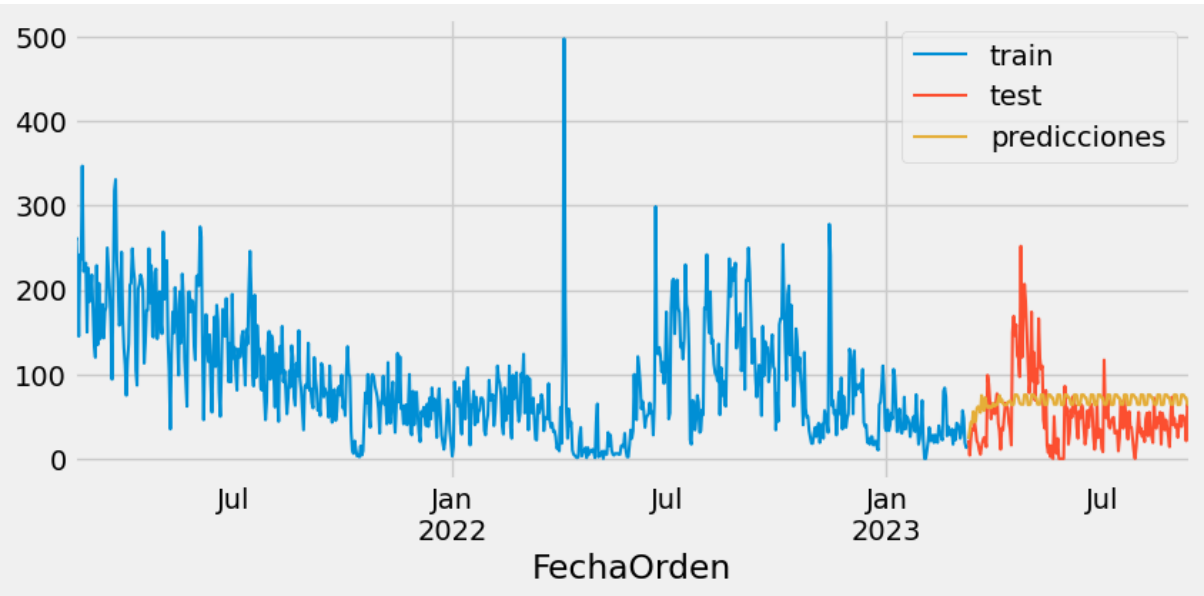
Figura 95
Predicción Forecast autorregresivo con LinearRegression



Nota. Elaboración propia.

Para el caso del GradientBoostingRegressor se obtuvo un MSE de 2013.91 y un MAE de 36.30. Se puede ver que la predicción no se ajusta bien a la demanda real dado que es un poco plana y tiende a una ligera subida.

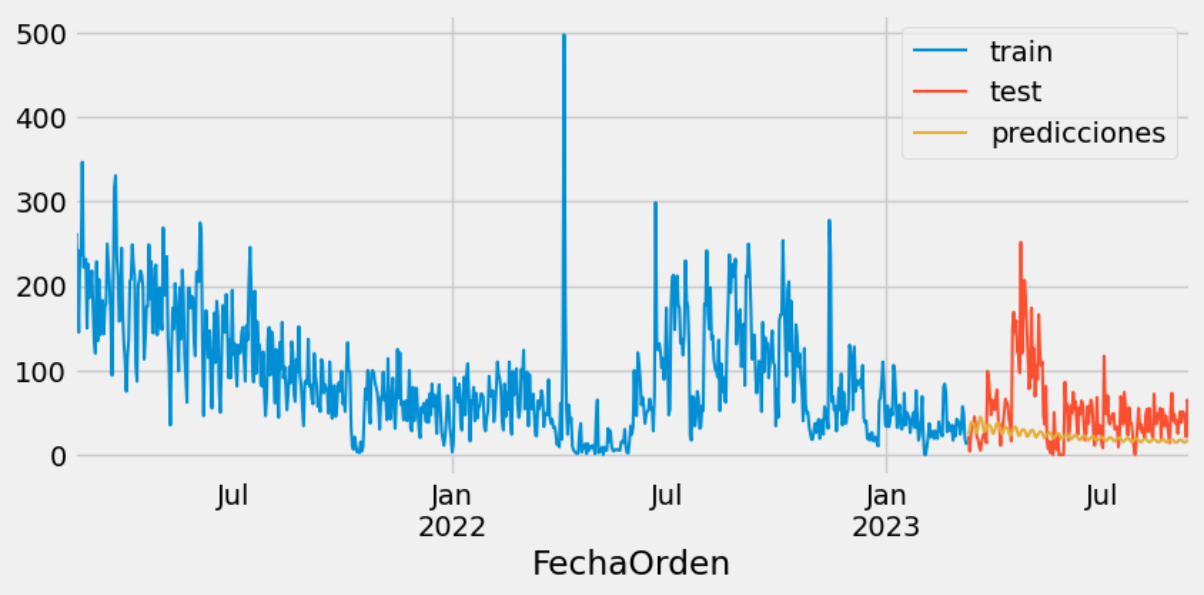
Figura 96
Predicción Forecast autorregresivo con GradientBoostingRegressor



Nota. Elaboración propia.

Para el caso del MLPRegressor se obtuvo un MSE de 2526.97 y un MAE de 35.19. Claramente la predicción no se ajusta a la demanda real dado que tiende a la caída.

Figura 97
Predicción Forecast autorregresivo con MLPRegressor



Nota. Elaboración propia.

Regresión lineal simple

Al tratarse de una regresión lineal, en primer lugar, medimos el coeficiente de correlación de Pearson entre la variable dependiente e independiente. Este valor fue de -0.45 que indica una correlación negativa moderada y por lo que sí podemos usar estas variables para entrenar este modelo. Para su cálculo usamos el siguiente código:

Figura 98
Cálculo de coeficiente de pearson para regresión lineal simple

```
# Correlación lineal entre las dos variables
from scipy.stats import pearsonr
# ================================================================================
corr_test = pearsonr(x = df_producto['FechaOrden'], y =  df_producto['CantidadOrdenada'])
print("Coeficiente de correlación de Pearson: ", corr_test[0])
print("P-value: ", corr_test[1])
```

```
Coeficiente de correlación de Pearson:  -0.45041269728192074
P-value:  1.1796307386948612e-47
```

Nota. Elaboración propia.

Asimismo, obtuvimos un MSE de 3536.99 y un MAE de 45.19. Hasta ahora el peor modelo, incluso peor que el SARIMAX que es nuestra línea base de comparación. El código usado fue el siguiente:

Figura 99
Cálculo de MSE y MAE para regresión lineal simple

```
from sklearn.metrics import mean_squared_error, mean_absolute_error

# Realiza las predicciones en el conjunto de prueba
predicciones = modelo.predict(X_test.reshape(-1, 1))

# Calcula el MSE
mse = mean_squared_error(y_test, predicciones)

# Calcula el MAE
mae = mean_absolute_error(y_test, predicciones)

print(f"El MSE del modelo es: {mse}")
print(f"El MAE del modelo es: {mae}")
```

```
El MSE del modelo es: 3536.9876895312154
El MAE del modelo es: 45.19479850558143
```

Nota. Elaboración propia.

Regresión lineal múltiple

Obtuvimos un MSE de 3699.47 y un MAE de 47.38. Por ende, este modelo es peor que el anterior. Para su cálculo se usó de forma directa las métricas de la librería sklearn.

Figura 100
Cálculo de MSE y MAE para regresión lineal simple

```
from sklearn.metrics import mean_squared_error as mse
from sklearn.metrics import mean_absolute_error as mae
```

```
# Nota
mse(Y_test, Y_pred)
```

```
3699.4727397070174
```

```
# Nota
mae(Y_test, Y_pred)
```

```
47.379360900258
```

Nota. Elaboración propia.

Forecast autorregresivo múltiple

Se obtuvo un MSE de 2505.74 y un MAE de 31.06 para el regresor Ridge y un MSE de 1883.50 y un MAE de 31.06 para el regresor GradientBoosting. Como vemos, los MSEs siguen siendo elevados. Para calcular estas métricas se uso las funciones de la librería sklearn de python del mismo modo que se muestra en la figura 100.

Random Forest

Como se mencionó anteriormente, primero se entrenó un modelo de Random Forest teniendo como variables predictoras la fecha de la orden, el precio de venta y la existencia de campaña. Con ello se obtuvo un MSE de 1819.25 y un MAE de 31.20.

Debido a que los resultados obtenidos fueron sustancialmente mejores que los otros modelos entrenados, se optó por agregar la variable día de semana (weekday) con el objetivo de que aporte significancia a la predicción. Esto se fundamenta en que durante la etapa de exploración de series de tiempo se identificó que la demanda se incrementa los días domingos y lunes, y en adelante empieza a disminuir.

Como resultado de incluir esta cuarta variable predictora se obtuvo un MSE de 1793.59 y un MAE de 30.30. No obstante, hasta ahora se había entrenado con la data pre procesada y sin normalizar, por lo que como siguiente estrategia se escaló la data y al volver a entrenar se obtuvo un MSE de 1696.14 y un 29.78.

Al ver que los resultados seguían mejorando se hizo un Grid Search y se obtuvo que los mejores hiper parámetros para este modelo son 150 estimadores, una profundidad máxima de 10 por cada árbol para evitar el sobreajuste, número mínimo de 10 muestras requeridas para dividir un nodo interno del árbol y número mínimo de 2 muestras requeridas en una hoja del árbol. Estos valores se pueden conocer mediante la siguiente línea de código:

Figura 101
Mejores hiperparámetros - Grid Search para Random Forest

```
# Obtener los mejores hiperparámetros encontrados
best_params = grid_search.best_params_
print(f"Mejores hiperparámetros: {best_params}")
```

Nota. Elaboración propia.

De este mejor modelo se obtuvo un MSE de 1491.70 y un MAE de 28.94. Estos resultados parecen ser mucho mejores.

5.2 Medición de la solución.

Como ya se había mencionado anteriormente, se usará el MSE y el MAE para poder evaluar de forma independiente qué modelo es mejor. Asimismo, dado que ya se tiene ambos valores, se calculará el RMSE y el MAE%.

5.2.1 Análisis de Indicadores cuantitativo y/o cualitativo.

Del modelo SARIMAX, que es el usado como base comparativa, se obtuvo que en promedio las predicciones del modelo difieren de los valores reales en 3118.40 (MSE) unidades al cuadrado y un error promedio de aproximadamente 55.84 (RMSE) unidades en la misma escala que la variable de interés. Asimismo, del MAE se obtuvo que las predicciones difieren de los valores reales en aproximadamente 39.188 unidades y en términos porcentuales, un 46.40%.

Por otra parte, nuestro mejor modelo obtuvo que en promedio las predicciones del modelo difieren de los valores reales en 1491.70 (MSE) unidades al cuadrado y un error promedio de aproximadamente 38.62 (RMSE) unidades en la misma escala que la variable de interés. Asimismo, se obtuvo que las predicciones difieren de los valores reales en aproximadamente 28.94 (MAE) unidades y en términos porcentuales, un 34.27% (MAE %).

En la Tabla 19 se puede observar el resumen de todas las métricas para todos los modelos entrenados.

Tabla 19
Resultado finales de entrenamiento de modelos

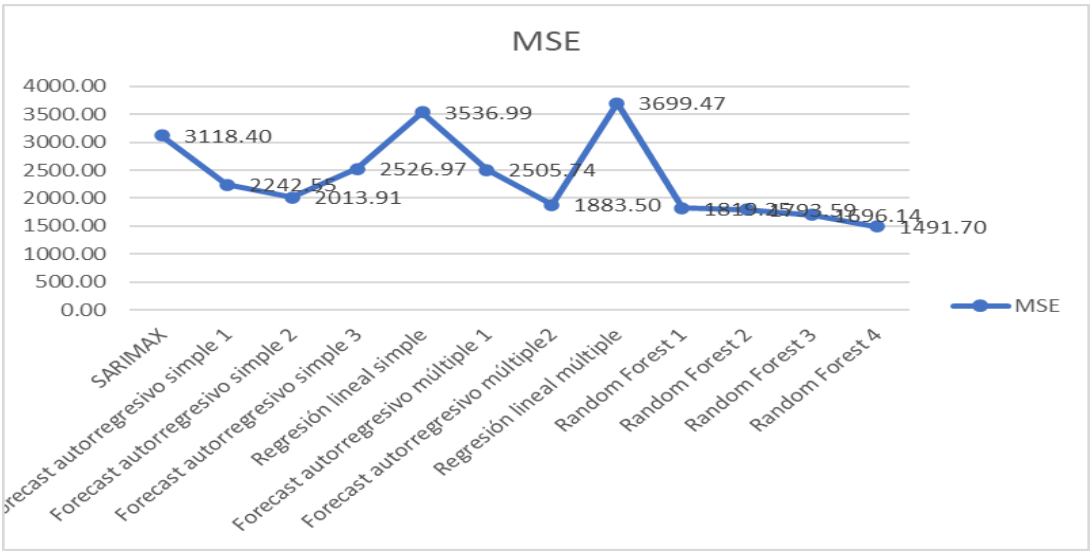
MODELO	REGRESOR / DETALLE	MSE	RMSE	MAE	MAE%
SARIMAX	Orden (1,1,0)	3118.40	55.84	39.19	46.40%
Forecast autorregresivo simple	Lineal	2242.55	47.36	38.80	45.94%
	GradientBoosting	2013.91	44.88	36.30	42.99%

	MLPRegressor	2526.97	50.27	35.19	41.67%
Regresión lineal simple	---	3536.99	59.47	45.19	53.52%
Forecast autorregresivo múltiple	Ridge	2505.74	50.06	31.06	36.78%
	GradientBoosting	1883.50	43.40	31.06	36.78%
Regresión lineal múltiple	---	3699.47	60.82	47.38	56.10%
Random Forest	Data pre procesada	1819.25	42.65	31.20	36.94%
	Inclusión de weekday	1793.59	42.35	30.30	35.87%
	Weekday + Data normalizada	1696.14	41.18	29.78	35.26%
	Weekday + Gread serch + Data normalizada + CV	**1491.70**	**38.62**	**28.94**	**34.27%**

Nota. Elaboración propia.

Asimismo, estas diferencias entre indicadores se presentan de forma gráfica a continuación:

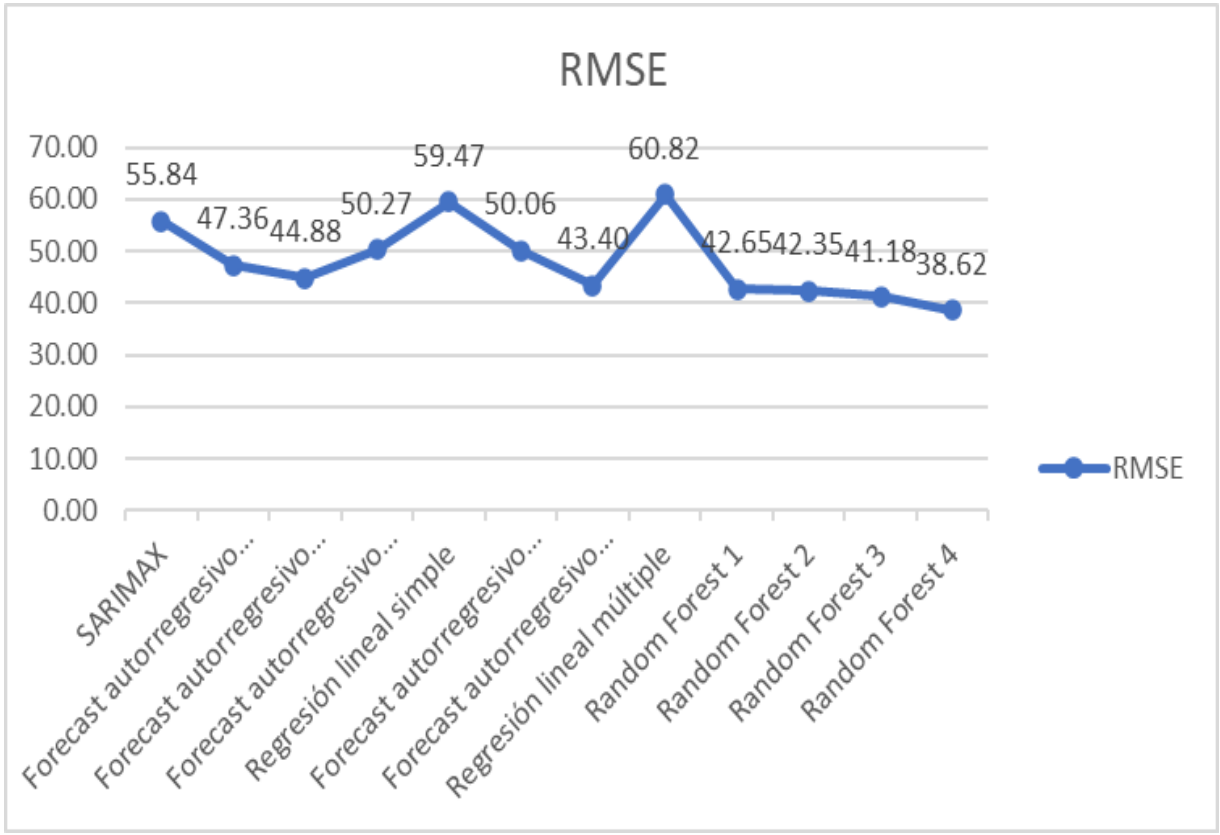
Figura 102
MSE por modelo



Nota. Elaboración propia.

De la Figura 102 se aprecia que el MSE más alto fue de 3699.47 cuando se entrenó el modelo de regresión lineal múltiple y el MSE más bajo fue de 1491.70 para el random forest 4. Por ende, hay una diferencia de 2207.77 unidades al cuadrado.

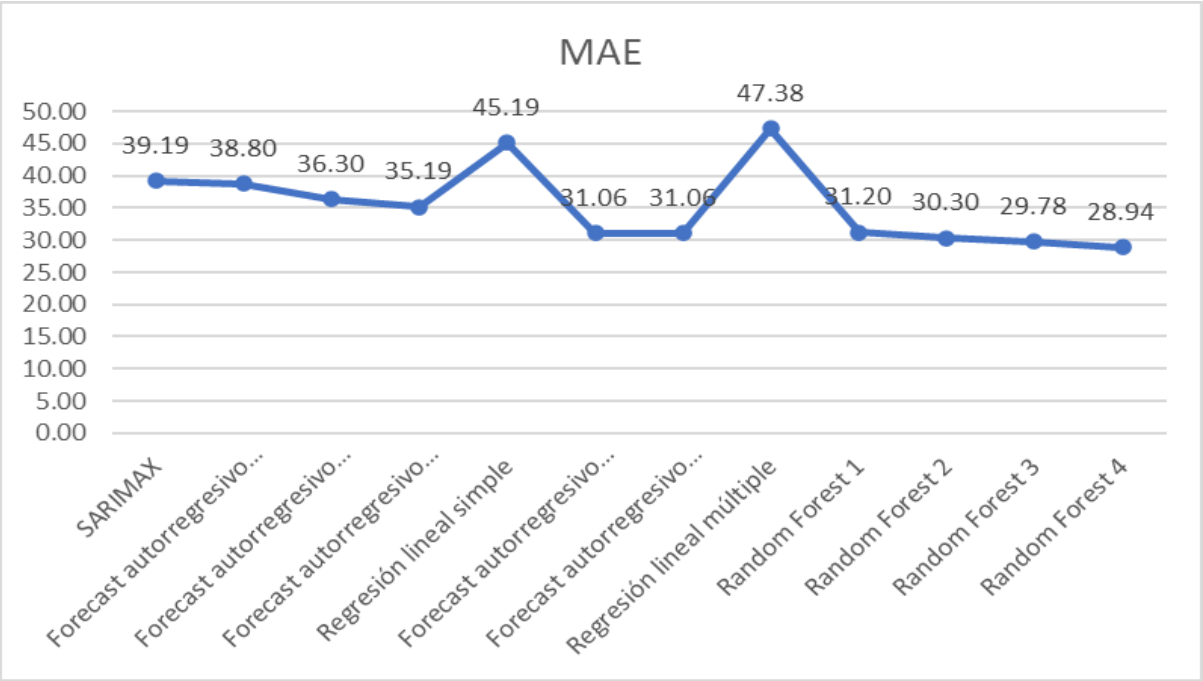
Figura 103
RMSE por modelo



Nota. Elaboración propia.

De la Figura 103 se aprecia que el RMSE más alto y más bajo fue de 60.82 y 38.62 para los modelos de regresión lineal múltiple y random forest respectivamente. Por lo tanto, hay una diferencia de 22.2 unidades en la misma escala de la variable de interés.

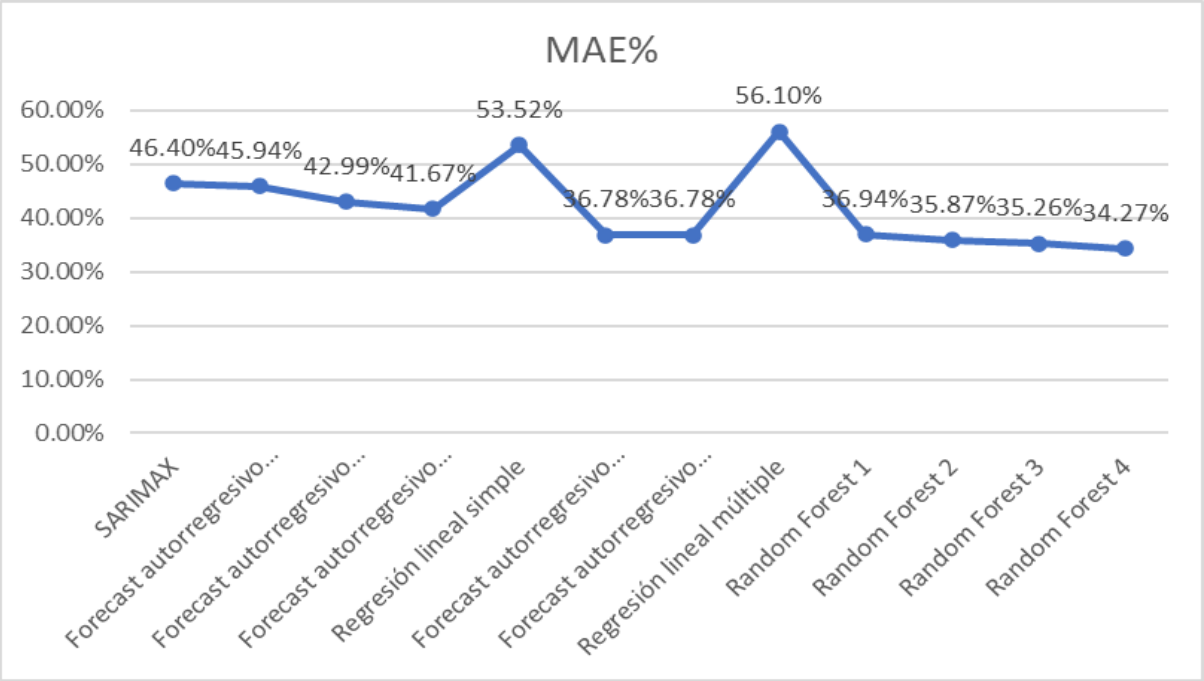
Figura 104
MAE por modelo



Nota. Elaboración propia.

Respecto al MAE, según la Figura 104, el mayor fue de 47.38 para la regresión lineal simple y el menor fue de 28.94 para el random forest en su configuración número cuatro. Es decir que hay una diferencia de 18.44 unidades.

Figura 105
MSE por modelo



Nota. Elaboración propia.

Para el caso del MAE%, en la figura 105, al igual que en los casos anteriores, el mayor indicador fue para la regresión lineal múltiple con un valor del 56.10% y un valor menor de 34.27% para la mejor configuración de random forest. Es así que hay una diferencia considerable de 21.83%.

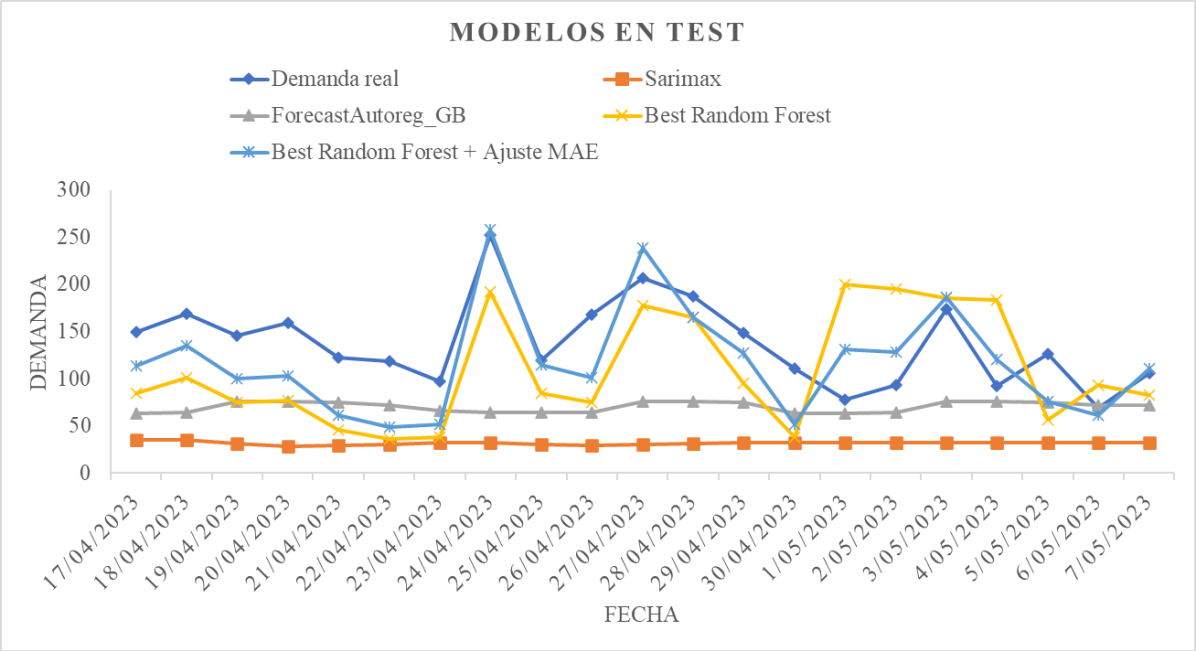
Por consiguiente, de las Tabla y los gráficos podemos observar que los mejores resultados a nivel de todos los indicadores los ha obtenido el modelo Random Forest con optimización de hiper parámetros luego de normalizar la data y aplicar cross validation. El segundo mejor modelo fue el Random Forest solo con data normalizada y el tercer mejor modelo fue el Random Forest con data sin normalizar. En general, los modelos que aplican Machine Learning son mejores que los que solo son estadísticos, con excepción de la regresión lineal múltiple que fue la que obtuvo los peores resultados.

Con el fin de poder dar mayor claridad de lo que representan todos estos indicadores respecto al performance de los modelos entrenados, se seleccionaron tres modelos para generar la figura 106 en la que se muestra la demanda predicha vs la demanda real para las fechas del

17/04/2023 al 07/05/2023. Este período fue seleccionado bajo el criterio de que las fechas se encontraban dentro de la data de test que es en dónde se evalúa el MSE y MAE.

Respecto a los modelos seleccionados, se eligió el SARIMAX debido a que es nuestra línea base, el mejor Random Forest y un modelo intermedio que en este caso fue el forecast autoregresivo entrenado con gradient boosting.

Figura 106
Desempeño de modelos en data de test



Nota. Elaboración propia.

De la figura 106 se puede observar que el modelo SARIMAX tiene un comportamiento casi plano con ligeros puntos de inflexión, pero en todo momento con una predicción muy por debajo de la demanda real. Por su parte, el Forecast autoregresivo con Gradien Boostig logra predecir mucho mejor que el SARIMAX, pero sigue estando, en general, por debajo de la demanda real y no logra detectar tendencias de picos o valles. Ahora bien, la situación es muy distinta cuando se trata del mejor modelo de Random Forest que se muestra en amarillo, en este caso sigue muy bien el comportamiento de la demanda real y sus puntos de inflexión son bastante certeros. Con el fin de poder reducir esta diferencia de unidades, se simuló el comportamiento del personal de planificación y en base a la demanda de la semana anterior se procedió a hacer ajustes con ayuda del MAE que nos decía que en promedio hay una diferencia

de predicción del 34.27%. El resultado se puede ver en celeste y se logró que el modelo siga la tendencia de manera muy buena para periodos como los comprendidos del 17/04/2023 al 26/04/203, 28/04/2023 al 30/04/2023 y 01/05/2023 al 04/05/2023. Asimismo, se detectaron picos casi perfectos para las fechas del 24/04/2023, 27/04/2023 y 03/05/2023.

Siguiendo en la línea de la eficiencia demostrada por el mejor modelo de Random Forest, se procedió a generar el gráfico completo de la demanda predicha para la data de test que está conformada por 188 días al azar y a compararla con la demanda real. El código usado para este objetivo se muestra a continuación en la figura 107:

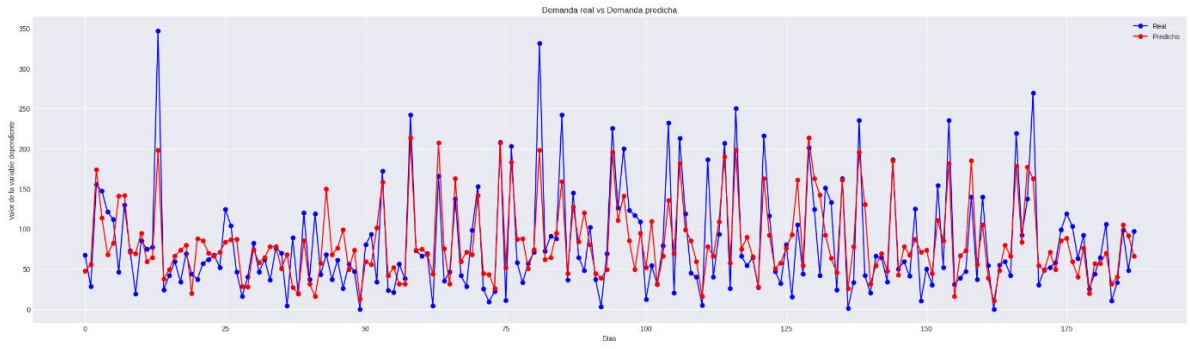
Figura 107
Código para visualización de predicción - Random Forest

```
import matplotlib.pyplot as plt
import numpy as np

# Ajustar el tamaño de la figura y la opacidad
plt.figure(figsize=(30, 8))

# Graficar valores reales con líneas
plt.plot(np.arange(len(Y_test)), Y_test, label='Real', color='blue', marker='o')

# Graficar valores predichos con líneas
plt.plot(np.arange(len(Y_test)), Y_pred, label='Predicho', color='red', marker='o')

plt.title('Demanda real vs Demanda predicha')
plt.xlabel('Días')
plt.ylabel('Valor de la variable dependiente')
plt.legend()
plt.show()
```

Nota. Elaboración propia.

El resultado de dicho código se puede apreciar a continuación:

Figura 108
Visualización de predicción en test - Random Forest



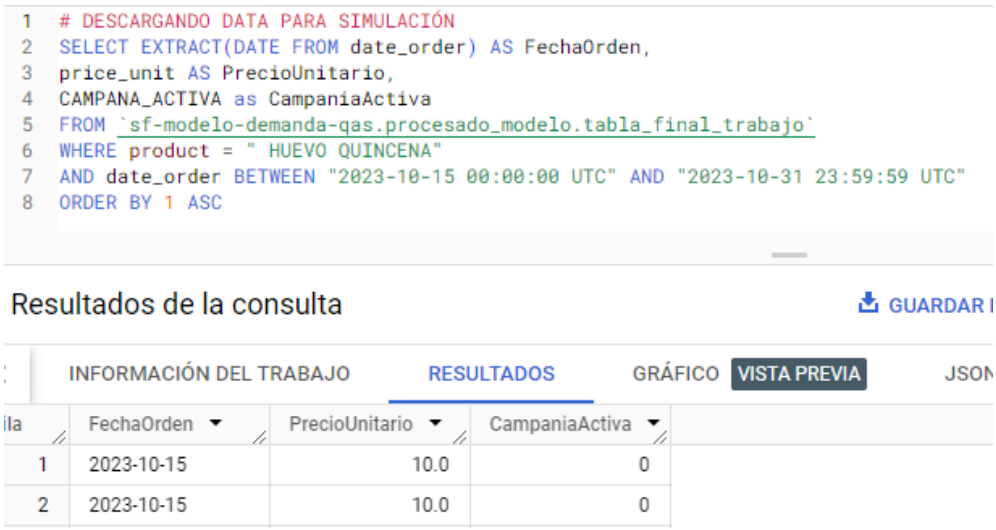
Nota. Elaboración propia.

De la figura 108, se puede reafirmar que el modelo sigue muy bien el comportamiento de la demanda real y que además lograr identificar casi todos los picos, aunque para el caso de los más elevados la diferencia puede ser un poco mayor que para el resto de fechas.

5.2.2. Simulación de solución

Luego de entrenar 12 modelos e identificar cuál es el mejor, se procedió a hacer una simulación para predecir la demanda en base a datos de precio y campaña disponible para el período del 15/10/2023 al 31/10/2023. Cabe señalar que este rango de fechas no ha sido entrenado ni testeado antes por el modelo. Estos datos requeridos se descargaron desde BigQuery y se logró mediante la consulta que se muestra a continuación en la figura 109:

Figura 109
Descarga de data para simulación



Nota. Elaboración propia.

Respecto a la programación, el primer paso fue guardar el mejor modelo en formato pkl mediante la librería joblib que proporciona herramientas para la serialización de objetos Python, lo que es útil para guardar modelos entrenados, entre otros objetos, para poder usarlos más adelante sin tener que volver a entrenarlos. Esto se logra mediante las siguientes líneas de código mostradas en la figura 110:

Figura 110
Guardado de mejor modelo en formato pkl

```
import joblib
# Guardar el modelo entrenado y el objeto de Scaler
joblib.dump(best_RF_model, 'best_RF_model.pkl')
```

```
['best_RF_model.pkl']
```

Nota. Elaboración propia.

Posteriormente, se creó una función de nombre procesar_dataset en la que se almacenara todo el pre procesamiento necesario para disponibilizar la data. A continuación, se presenta el código usado:

Figura 111
Función de pre procesamiento de data - simulación

```
def procesar_dataset(data):
    # Creación de variable weekday
    data['datetime'] = data['FechaOrden'].apply(pd.to_datetime)
    data['Weekday'] = data['datetime'].apply(lambda x: x.dayofweek)

    # Agrupación por suma de cantidad ordenada
    df_producto = data.groupby('FechaOrden').agg({
        'PrecioUnitario': 'mean',
        'CampaniaActiva': 'mean',
        'Weekday': 'mean'
    }).reset_index()

    # Formato de fecha
    #df_producto['FechaOrden'] = pd.to_datetime(df_producto['FechaOrden'], format='%d/%m/%Y').dt.strftime('%Y/%m/%d')

    # Frecuencia de fecha
    df_producto['FechaOrden'] = pd.to_datetime(df_producto['FechaOrden'])
    df_producto = df_producto.set_index('FechaOrden')
    df_producto = df_producto.asfreq('1D')
    df_producto = df_producto.sort_index()

    # Tratamiento de fechas faltantes
    df_producto['PrecioUnitario'].fillna(df_producto['PrecioUnitario'].mean(), inplace=True)
    df_producto['CampaniaActiva'].fillna(0, inplace=True)
    df_producto['Weekday'] = df_producto.index.weekday

    # Escalamiento de data
    # Crear un objeto Scaler
    scaler = StandardScaler()

    # Ajustar y transformar los datos de entrenamiento
    df_scaled = pd.DataFrame(scaler.fit_transform(df_producto), columns=df_producto.columns, index=df_producto.index)

    return df_producto
```

Nota. Elaboración propia.

En base a la figura 111, a continuación, se explica paso a paso lo que hace dicha función:

Conversión de fechas:

- La columna 'FechaOrden' se convierte en un objeto de fecha y hora mediante la función pd.to_datetime.
- Se agrega una nueva columna 'Weekday' que representa el día de la semana correspondiente a cada fecha.

Agrupación y agregación:

- El conjunto de datos se agrupa por la columna 'FechaOrden'.
- Se realiza una agregación utilizando la función agg, calculando la media de 'PrecioUnitario', 'CampaniaActiva', y 'Weekday' para cada grupo de fechas.
- El resultado se almacena en un nuevo DataFrame llamado df_producto.

Formato y frecuencia de fecha:

- Se formatea la columna 'FechaOrden' como objeto datetime.
- Se establece esta columna como el índice del DataFrame df_producto.
- Se utiliza asfreq para rellenar los días faltantes y establecer la frecuencia a '1D' (día).
- El DataFrame se ordena según el índice.

Tratamiento de datos faltantes:

- Los valores nulos en 'PrecioUnitario' se llenan con la media de 'PrecioUnitario'.
- Los valores nulos en 'CampaniaActiva' se llenan con 0.
- La columna 'Weekday' se actualiza con el valor correspondiente al día de la semana basado en el índice de fechas.

Escalamiento de datos:

- Se utiliza la clase StandardScaler de scikit-learn para realizar el escalamiento estándar de los datos.
- Se ajusta y transforma el DataFrame df_producto utilizando el scaler.
- El resultado se almacena en un nuevo DataFrame llamado df_scaled.

Retorno del resultado:

- La función retorna el DataFrame df_producto, que contiene los datos procesados y transformados.

Como paso siguiente, se procedió a leer la data descargada de BigQuery y se le aplicó la función antes explicada. En la figura 112 se puede apreciar el código usado, como ven basta de solo una línea para pre procesar la data de manera inmediata.

Figura 112
Aplicación de función de pre procesamiento de data - simulación

```
df_procesado = procesar_dataset(data)
df_procesado.head()
```

Nota. Elaboración propia.

Posterior a ello se cargó el modelo previamente guardado y se aplicó a la data pre procesada en el paso anterior. El código y los resultados se pueden apreciar en la figura 113:

Figura 113
Aplicación de mejor modelo - simulación

```
best_model = joblib.load('best_RF_model.pkl')
```

```
# Realiza predicciones con el modelo cargado
Y_pred_test = best_model.predict(df_procesado)
# Crea un DataFrame con las fechas y las predicciones
df_predictions = pd.DataFrame({'Fecha': df_procesado.index, 'Predicciones': Y_pred_test})
# Imprime el DataFrame con las predicciones
print(df_predictions)
```

```
        Fecha  Predicciones
0  2023-10-15     20.729551
1  2023-10-16     56.555054
2  2023-10-17     52.521772
3  2023-10-18     45.055597
4  2023-10-19     53.959107
5  2023-10-20     40.777167
```

Nota. Elaboración propia.

Con el objetivo de visualizar de una mejor forma los resultados, se generó un diagrama de líneas que a continuación se presenta. Cabe señalar que, por tratarse de unidades, todas las cantidades fueron redondeadas hacia arriba.

Figura 114
Predicción de mejor modelo - simulación

Nota. Elaboración propia.

De la figura 114 se aprecia que para el período en cuestión se predice una demanda bastante regular con ligeras subidas y bajadas comprendidas entre los 21 y 58 paquetes, a excepción del 30/10/2023 en el que se espera un pico un poco alto de 125 paquetes. Dicho pico puede responder a que, al día siguiente en el Perú, se celebra Halloween y el Día de la Canción Criolla. Una vez más se comprueba que el mejor modelo no solo detecta el comportamiento normal de la demanda, sino que también puede identificar comportamientos un tanto anómalos para fechas puntuales.

CAPÍTULO VI: Conclusiones y recomendaciones

5.1. Conclusiones

Durante la etapa de adquisición de la data se descargó la base de datos de BigQuery en formato CSV. Se tuvo acceso a un total de 802 176 ventas en general, de las cuales 32 452 correspondía a paquetes de huevos de 15 unidades. En términos de tiempo se dispuso de 940 días a partir del 17 de febrero del 2021.

Durante la etapa de exploración de la data se detectó que el producto en cuestión presenta una tendencia irregular y no presenta estacionariedad por lo que su predicción constituye un desafío mayor.

Del análisis de tendencia de la demanda por distintos niveles de agregación se identificó que solo existe una una estacionalidad por día de la semana dado que se evidenció que las mayores compras se dan los días domingos y lunes, en adelante la demanda decrece. Esto responde a que los clientes, por lo general, suelen planificar sus compras de forma semanal.

En la etapa de preprocesamiento se emplearon técnicas exhaustivas para tratamiento de datos y se logró pasar de 43 variables a solo 4 que son FechaOrden, PrecioUnitario, CampaniaActiva y CantidadOrdenada. Adicionalmente, para el mejor modelo se añadió la variable Weekday debido a que se identificó que presentaba una estacionalidad.

En la etapa de modelamiento se entrenaron tres niveles de modelos. SARIMAX como modelo de tipo estadístico y cuyos resultados sirvieron de base para comparar con los demás modelos, modelos estadísticos impulsador por Machine Learning y modelos netamente de Machine Learning. En total se entrenaron 6 tipos de modelos y contando variaciones se entrenaron 12 modelos.

De la etapa de análisis e interpretación de resultados se concluyó que el mejor modelo a nivel de MSE, RMSE, MAE y MAE% fue el Random Forest optimizado y entrenado mediante cross validation con data normalizada. Mediante este modelo se logró disminuir el MSE de 3118.40 a 1491.70 mejorando así es un 52.16% y pasando de un MAE de 39.19 a 28.94 mejorando en un 26.15%.

Por otra parte, nuestro mejor modelo asume una reducción de recursos en el equipo de planificación pasando de requerir cinco personas a solo dos: un analista de planificación Sr. y un analista de nivel Jr. que básicamente se centarían en hacer pequeños ajustes a las

predicciones en base al comportamiento de la demanda de la última semana. Esto en términos de personal implica la reducción de un 60%.

En términos de hora de trabajo se pasaría de ocho horas a solo cuatro horas, lo que significa una reducción del 50%. Si bien es cierto la predicción de la demanda mediante nuestro modelo demora apenas unos segundos, se está considerando cuatro horas por la cantidad de productos en los que se debe hacer ajustes.

En términos de costos, debido a la reducción de personal y tiempo empleado para la proyección de demanda, se pasará de gastar S/ 3200 a gastar S/ 1200. Esto representa un ahorro del 62.5%.

Por último, cabe mencionar que la incorporación de Machine Learning en la proyección de demanda permitirá a la empresa obtener predicciones más precisas y basadas en datos históricos, patrones estacionales y factores externos. Esto impulsará la eficiencia operativa al reducir errores y optimizar inventarios, brindando a la empresa una ventaja competitiva al adaptarse de manera más ágil a las necesidades del mercado y mejorando la toma de decisiones estratégicas.

5.2. Recomendaciones

Si bien es cierto el mejor modelo representa una mejora consistente en comparación al método estadístico que se usó como línea base, se recomienda entrenar más modelos y más complejos como redes neuronales del tipo MLP o LSTM que de acuerdo con los antecedentes pueden dar resultados excelentemente buenos, llegando incluso a obtener RMSEs entre 0 (antecedente 4) y 1.5 (antecedente 1). Por temas de tiempo y complejidad, debido a que requiere conocimientos especializados, no se exploró esta solución.

Dado que su base de datos de pedidos se almacena de BigQuery dentro de GCP, se sugiere implementar el modelo en la misma nube con el fin de que se pueda retroalimentar con los nuevos datos de pedidos reales y en adelante mejorar cada vez más su nivel de precisión.

Por último, se sugiere crear un dashboard en el que se pueda centralizar información como predicción de demanda vs demanda real, top de productos más pedidos, top de productos con riesgo de quiebre de stock, indicadores del modelo, etc.

Referencias bibliográficas

Amat, J. & Escobar, J. (febrero de 2021). Skforecast: forecasting series temporales con Python y Scikit-learn. https://cienciadedatos.net/documentos/py27-forecasting-series-temporales-python-scikitlearn.html

Alzubi, J., Nayyar, A., & Kumar, A. (2018). Machine learning from theory to algorithms: an overview. In *Journal of physics: conference series* (Vol. 1142, p. 012012). IOP Publishing. DOI: https://doi.org/10.1088/1742-6596/1142/1/012012

Asociación Peruana de Avicultura (13 de octubre de 2022). *APA compartirá más de 26 mil huevos con poblaciones vulnerables del Perú.* https://apa.org.pe/2022/10/13/apa-compartira-26mil-huevos-poblacionesvulnerables-1-2/#:~:text=De%20acuerdo%20al%20Ministerio%20de,de%20huevo%20(304%20unidades).

Bento, C. (21 de setiembre de 2021). *Multilayer Perceptron Explained with a Real-Life Example and Python Code: Sentiment Analysis*. TowrdsDataScience. https://towardsdatascience.com/multilayer-perceptron-explained-with-a-real-life-example-and-python-code-sentiment-analysis-cb408ee93141

Brownlee, J. (16 de febrero del 2021). Regression Metrics for Machine Learning. *Machine Learning Mastery*. https://machinelearningmastery.com/regression-metrics-for-machine-learning/

Burrueco, D. (s.f.). *El Perceptrón Multicapa*. Interactive Chaos, making things simple. https://interactivechaos.com/sites/default/files/styles/max_800_px/public/2020-09/tutdl_0044.jpg

Darlington, R. B., & Hayes, A. F. (2016). *Regression analysis and linear models: Concepts, applications, and implementation*. Guilford Publications.

Gestión (3 de marzo de 2023). *Avisur alerta escasez de huevo y pollos en mercados peruanos. Diario Gestión.* https://gestion.pe/economia/gripe-aviar-falta-de-soya-avisur-alerta-escasez-de-huevo-y-pollos-en-mercados-peruanos-noticia/

Han, C. (03 de enero del 2019). 5 Steps of a Data Science Project Lifecycle. *Towards Data Science*. https://towardsdatascience.com/5-steps-of-a-data-science-project-lifecycle-26c50372b492

Haselbeck, F., Killinger, J., Menrad, K., Hannus, T. & Grimm, D. G. (2022). Machine Learning Outperforms Classical Forecasting on Horticultural Sales Predictions. *Machine Learning with Applications* , *7*. https://doi.org/10.1016/j.mlwa.2021.100239

Harrington, P. (2012). *Machine learning in action*. Simon and Schuster.

Holbrook, M. B. & Hirschman, E. C. (1982). The Experimental Aspects of Consumption Consumer Fantasies Feelings and Fun. Journal of Consumer Research , 9, 132- 140.

Hotz, N. (19 de enero del 2023). OSEMN Data Science Life Cycle. *Data Science Process Alliance*. https://www.datascience-pm.com/osemn/

Joaquin Amat, Rodrigo (Noviembre 2020) Regularización de Ridge, Lasso, Elastic Net Con Python.Obtenido de https://www.cienciadedatos.net/documentos/py14-ridge[1]lasso-elastic-net-python.html

Jung, Alexander. (2022). *Machine Learning: The Basics*. Springer. DOI: https://doi.org/10.1007/978-981-16-8193-6

Koehrsen, W. (9 de enero de 2018). *Hyperparameter Tuning the Random Forest in Python.* https://towardsdatascience.com/hyperparameter-tuning-the-random-forest-in-python-using-scikit-learn-28d2aa77dd74

Krugman, P., & Wells, R. (2009). Economics (2nd ed.). Worth Publishers.

Louppe, G. (2014). Understanding random forests: From theory to practice. https://arxiv.org/pdf/1407.7502.pdf,

Masaji, C. (22 de junio de 2023). Hypothesis to be Tested: Definition and 4 Steps for Testing with Example. https://www.investopedia.com/terms/h/hypothesistesting.asp#:~:text=Key%20Takeaways-,Hypothesis%20testing%20is%20used%20to%20assess%20the%20plausibility%20of%20a,of%20the%20population%20being%20analyzed.

Maulud, D., & Abdulazeez, A. M. (2020). A review on linear regression comprehensive in machine learning. Journal of Applied Science and Technology Trends, 1(4), 140-147. https://doi.org/10.38094/jastt1457

Medina, A. (2009). Fundamentación de las competencias discentes y docentes. En A. Medina (ed.), Formación y desarrollo de la competencia básica (pp. 11-44). Madrid: Universitas.

Perktold, J., Seabold, S., Sheppard, K., Fulton, C., Shedden, K., Quackenbush, P., Arel-Bundock, V., McKinney, W., Langmore, I., Baker, B., Gommers, R., Zhurko, E., Brett, M., Giampieri, E., Liu, Y., & Halchenko, Y. (17 de octubre de 2023). Stationarity and detrending (ADF/KPSS). https://www.statsmodels.org/dev/examples/notebooks/generated/stationarity_detrending_adf_kpss.html

Prabhakaran, S. (2 de noviembre de 2019). Augmented Dickey Fuller Test (ADF Test) – Must Read Guide. https://www.machinelearningplus.com/time-series/augmented-dickey-fuller-test/

Probst, P., Wright, M. N., & Boulesteix, A. L. (2019). Hyperparameters and tuning strategies for random forest. *Wiley Interdisciplinary Reviews: data mining and knowledge discovery*, 9(3), e1301. https://arxiv.org/pdf/1804.03515.pdf

Reding Bernal, A., Zamora Macorra, M., & López Alvarenga, J. C. (2011). ¿Cómo y cuándo realizar un análisis de regresión lineal simple? Aplicación e interpretación. *Dermatologia Revista Mexicana*, *55*(6), 414–421.

Rojas-Jimenez, K. (2022). Capítulo 8. Análisis de series de tiempo. *Ciencia de Datos para Ciencias Naturales*. https://bookdown.org/keilor_rojas/CienciaDatos/an%C3%A1lisis-de-series-de-tiempo.html?q=Serie%20de%20tiempo#an%C3%A1lisis-de-series-de-tiempo

Ruiz, Benjamín (12 de mayo de 2023). *Ranking latinoamericano de consumo de pollo y huevo.* https://catedralatam.com/ranking-latinoamericano-de-consumo-de-pollo-y-huevo/#:~:text=En%20el%20caso%20del%20huevo,ahora%20cada%20producto%20por%20pa%C3%ADses.

Samuelson, P. A., & Nordhaus, W. D. (2010). Economía (19ª ed.). McGraw-Hill.

Sarramona, J. (2007). Las competencias profesionales del profesorado de secundaria. Estudios Sobre Educación, 12, 31-42.

Schonlau, M., & Zou, R. Y. (2020). The random forest algorithm for statistical learning. *The Stata Journal*, 20(1), 3-29. DOI: https://doi.org/10.1177/1536867X20909688

Serafeim, L. (2021). *Classifying Handwritten Digits Using A Multilayer Perceptron Classifier (MLP)*. Towards Data Scince. https://towardsdatascience.com/classifying-handwritten-digits-using-a-multilayer-perceptron-classifier-mlp-bc8453655880

Sevillano, M. L. (Dir.) (2009). Competencias para el uso de herramientas virtuales en la vida, trabajo y formación permanente. Madrid: Pearson, Prentice Hall.

Spiliotis, E., Makridakis, S., Semenoglou, A. A., & Assimakopoulos, V. (2020). Comparison of statistical and machine learning methods for daily SKU demand forecasting. *Operational Research*, 1-25.

Vandeput, N. (05 de julio del 2019). Forecast KPIs: RMSE, MAE, MAPE & Bias. *Towards Data Science*. https://towardsdatascience.com/forecast-kpi-rmse-mae-mape-bias-cdc5703d242d

Villa, A. y Poblete, M. (2004). Práctica y evaluación de competencias. Profesorado: Revista de Currículum y Formación del Profesorado, 8 (2). Disponible en: http://www.ugr.es/local/recfpro/rev82ed.pdf.