

**UNIVERSIDAD NACIONAL DE SAN CRISTÓBAL
DE HUAMANGA**

FACULTAD DE INGENIERÍA DE MINAS, GEOLOGÍA Y CIVIL

ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS



TESIS:

**Modelo machine learning para predecir las ventas de una
empresa RETAIL, 2025**

Para optar el título profesional de:
INGENIERA DE SISTEMAS

PRESENTADO POR:
Bach. Keyla Elyneth SERNA BERROCAL

ASESORA:
Mtra. Elinar CARRILLO RIVEROS

AYACUCHO - PERÚ

2025

Dedicatoria

A mi Padre Celestial, por su amor infinito, por guiar mis pasos incluso cuando no los comprendía, y por fortalecerme en cada momento de debilidad. Sin Él, este logro no habría sido posible.

A mi familia, por ser mi apoyo constante, por sus oraciones, su paciencia y su amor incondicional. Cada uno de ustedes ha sido una luz en mi camino, y este trabajo también les pertenece.

Agradecimiento

Expreso mi sincero agradecimiento a la Universidad Nacional de San Cristóbal de Huamanga, y en particular a la Facultad de Ingeniería de Minas, Geología y Civil, por brindarme la formación académica y profesional necesaria para alcanzar este importante logro.

Agradezco profundamente a los docentes de la Escuela Profesional de Ingeniería de Sistemas, quienes, a lo largo de mi formación, compartieron sus conocimientos, valores y experiencias con generosidad y compromiso.

De manera especial, agradezco al Mg. Ing. Elinar Carrillo Riveros, mi asesor de tesis, por su guía constante, sus valiosos aportes técnicos y su paciencia durante el desarrollo de esta investigación.

También extendo mi gratitud a todos quienes, directa o indirectamente, contribuyeron al desarrollo de este trabajo y al cumplimiento de esta etapa tan significativa de mi vida profesional.

Resumen

La tesis titulada “Modelo machine learning para predecir las ventas de una empresa RETAIL, 2025” aborda el problema de la baja precisión en la estimación de ventas en una tienda virtual de una empresa retail peruana. La investigación partió de la necesidad de determinar qué modelo de machine learning, entre Prophet y LSTM, permite una mejor predicción de ventas, considerando que actualmente la empresa fija metas únicamente con base en datos históricos sin métodos de predicción formales.

Para resolver este problema, se desarrolló un estudio de tipo aplicado, con nivel predictivo y diseño no experimental, utilizando datos secundarios. La investigación empleó análisis documental como técnica principal y se analizaron 152,735 registros de ventas históricas comprendidos entre los años 2022 y 2024. Se implementaron y entrenaron ambos modelos (Prophet y LSTM) en Python, utilizando métricas como RMSE, MAE y MAPE para evaluar su desempeño, y se aplicó validación cruzada temporal en el caso del modelo LSTM.

Los resultados mostraron que el modelo LSTM presentó un menor error absoluto y porcentual, con un RMSE de 28,019, MAE de 15,604 y MAPE de 123.35%, frente a Prophet, que alcanzó un RMSE de 33,802.85, MAE de 22,133.22 y MAPE de 182.99%. De esta manera, se concluyó que el modelo LSTM mostró una mejor capacidad para predecir las ventas en el contexto analizado en comparación con Prophet. También, se comprobó que el uso de las variables históricas influye de manera significativa en la precisión del modelo, además, el tipo de algoritmo tiene un impacto directo en el nivel del error, ya que este caso Prophet menos preciso que el LSTM.

El estudio aporta evidencia de que los modelos de machine learning pueden mejorar notablemente las estimaciones de ventas en el canal virtual de una empresa retail. Asimismo, se recomienda a la empresa considerar la aplicación de este tipo de soluciones y, en el futuro, incluir otras variables como promociones o estacionalidades para perfeccionar el modelo.

Sin embargo, una limitación importante fue que se usaron datos de una sola tienda virtual, lo que dificulta extrapolar los resultados a otros canales o empresas del sector retail. Asimismo, no se consideraron variables externas que podrían influir en la demanda, como campañas promocionales, estacionalidad o comportamiento del consumidor. Finalmente, la comparación se centró solo en dos modelos de machine learning, lo que deja abierta la posibilidad de explorar otros enfoques predictivos en futuras investigaciones.

Palabras clave: *Machine Learning, Prophet, LSTM*

Abstract

The thesis, "A Machine Learning Model to Predict Sales for a Retail Company, 2025," addresses the problem of low accuracy in sales estimation in a Peruvian retail company's online store. The research was based on the need to determine which machine learning model, between Prophet and LSTM, allows for better sales prediction, considering that the company currently sets goals solely based on historical data without formal prediction methods.

To address this problem, an applied study was developed, with a predictive level and non-experimental design, using secondary data. The research employed document analysis as the primary technique and analyzed 152,735 historical sales records from the years 2022 to 2024. Both models (Prophet and LSTM) were implemented and trained in Python, using metrics such as RMSE, MAE, and MAPE to evaluate their performance, and temporal cross-validation was applied in the case of the LSTM model. The results showed that the LSTM model had a lower absolute and percentage error, with an RMSE of 28.019, MAE of 15.604, and MAPE of 123.35%, compared to Prophet, which achieved an RMSE of 33,802.85, MAE of 22,133.22, and MAPE of 182.99%. Thus, it was concluded that the LSTM model demonstrated a better ability to predict sales in the analyzed context compared to Prophet. It was also found that the use of historical variables significantly influences the model's accuracy. Furthermore, the type of algorithm has a direct impact on the error level, as Prophet was less accurate than the LSTM.

The study provides evidence that machine learning models can significantly improve sales estimates in a retail company's virtual channel. Likewise, the company is encouraged to consider applying this type of solution and, in the future, include other variables such as promotions or seasonality to refine the model.

However, a major limitation was that data from a single online store was used, making it difficult to extrapolate the results to other channels or companies in the retail sector. Furthermore, external variables that could influence demand, such as promotional campaigns, seasonality, or consumer behavior, were not considered. Finally, the comparison focused only on two machine learning models, leaving open the possibility of exploring other predictive approaches in future research.

Keywords: *Machine Learning, Prophet, LSTM*

INDICE

| | |
|---|------------|
| Dedicatoria | ii |
| Agradecimiento..... | iii |
| Resumen..... | iv |
| Abstract..... | v |
| CAPÍTULO I..... | 11 |
| INTRODUCCIÓN..... | 11 |
| 1.1. PLANTEAMIENTO DEL PROBLEMA..... | 11 |
| 1.2. FORMULACIÓN DEL PROBLEMA..... | 12 |
| 1.2.1. Problema general..... | 12 |
| 1.2.2. Problemas específicos..... | 12 |
| 1.3. JUSTIFICACIÓN DE LA INVESTIGACIÓN | 12 |
| 1.4. LIMITACIONES DE LA INVESTIGACIÓN | 13 |
| 1.5. OBJETIVOS..... | 14 |
| 1.5.1. Objetivo general | 14 |
| 1.5.2. Objetivo específico..... | 14 |
| CAPÍTULO II..... | 15 |
| MARCO TEÓRICO..... | 15 |
| 2.1. ANTECEDENTES DE LA INVESTIGACIÓN | 15 |
| 2.2. MARCO CONCEPTUAL..... | 17 |
| 2.2.1. Ciencia de datos | 17 |
| 2.2.2. Machine learning..... | 18 |
| 2.2.3. Predicción..... | 19 |
| 2.2.4. Predicción de Ventas | 20 |
| 2.2.5. Modelado de series temporales..... | 21 |
| 2.2.6. Data | 21 |
| 2.2.7. Exploratory Data Analysis (EDA)..... | 22 |
| 2.2.8. Dato atípico (outlier)..... | 23 |
| 2.2.9. Media..... | 23 |
| 2.2.10. Suavizado de series temporales | 24 |
| 2.2.11. Tipo de Algoritmo | 24 |
| 2.2.12. Modelo Prophet | 24 |
| 2.2.13. Red Neuronal Recurrente (RNN)..... | 25 |
| 2.2.14. Redes Neuronales de Memoria de Largo Plazo (LSTM) | 26 |
| 2.2.15. Precisión del modelo | 27 |
| 2.2.16. Error | 28 |
| 2.2.17. Métricas de validez..... | 29 |
| 2.2.18. RMSE (Root Mean Squared Error)..... | 30 |
| 2.2.19. MAE (Mean Absolute Error): | 31 |

| | | |
|---------------------------|---|-----------|
| 2.2.20. | MAPE (Mean Absolute Percentage Error):..... | 31 |
| 2.2.21. | Validación temporal..... | 32 |
| 2.2.22. | División de Entrenamiento y Prueba(Train/Test Split)..... | 33 |
| 2.2.23. | Validación cruzada..... | 33 |
| 2.2.24. | TimeSeriesSplit..... | 34 |
| 2.2.25. | Sobreajuste y Subajuste (Overfitting y Underfitting)..... | 34 |
| 2.2.26. | Modelo base (baseline model)..... | 35 |
| 2.2.27. | Evaluación de modelos predictivos..... | 35 |
| 2.2.28. | Ventas..... | 36 |
| 2.2.29. | Unidades vendidas..... | 37 |
| 2.2.30. | Montos diarios..... | 38 |
| 2.2.31. | Datos históricos de ventas..... | 38 |
| 2.2.32. | Python..... | 39 |
| 2.2.33. | Visual Studio Code (VSCode)..... | 39 |
| 2.2.34. | Jupyter Notebook..... | 40 |
| 2.2.35. | NumPy..... | 40 |
| 2.2.36. | Pandas..... | 41 |
| 2.2.37. | Scikit-learn..... | 41 |
| 2.2.38. | Matplotlib..... | 42 |
| 2.2.39. | TensorFlow..... | 43 |
| 2.2.40. | Keras..... | 44 |
| 2.3. | HIPÓTESIS..... | 45 |
| 2.3.1. | Hipótesis general..... | 45 |
| 2.3.2. | Hipótesis específica..... | 45 |
| CAPÍTULO III | | 46 |
| MATERIAL Y MÉTODOS | | 46 |
| 3.1. | TIPO Y NIVEL DE INVESTIGACIÓN..... | 46 |
| 3.1.1. | Tipo de investigación..... | 46 |
| 3.1.2. | Nivel de investigación..... | 46 |
| 3.2. | DISEÑO DE LA INVESTIGACIÓN..... | 47 |
| 3.3. | VARIABLES..... | 47 |
| 3.3.1. | Definición conceptual de variables..... | 47 |
| 3.3.2. | Definición operacional de variables..... | 49 |
| 3.4. | POBLACIÓN Y MUESTRA..... | 49 |
| 3.4.1. | Población..... | 49 |
| 3.4.2. | Muestra..... | 49 |
| 3.5. | TÉCNICAS E INSTRUMENTOS DE LA INVESTIGACIÓN..... | 50 |
| 3.5.1. | Técnica de análisis documental..... | 50 |
| 3.5.2. | Instrumentos: Registro de base de datos..... | 50 |

| | |
|---|-----------|
| 3.5.3. Validez del instrumento..... | 51 |
| 3.5.4. Confiabilidad de instrumento | 51 |
| 3.6. PROCEDIMIENTOS | 51 |
| 3.6.1. Estrategia de prueba de hipótesis | 52 |
| 3.6.2. Técnicas de procesamiento de datos..... | 52 |
| CAPÍTULO IV | 54 |
| RESULTADOS Y DISCUSIÓN | 54 |
| 1.1. EVALUACIÓN DE LA HIPÓTESIS GENERAL | 54 |
| 1.2. EVALUACIÓN DE LA HIPÓTESIS ESPECÍFICA H1 | 55 |
| 1.3. EVALUACIÓN DE LA HIPÓTESIS ESPECÍFICA H2 | 56 |
| 1.4. EVALUACIÓN DE LA HIPÓTESIS ESPECÍFICA H3 | 58 |
| 1.5. DISCUSIÓN | 59 |
| CAPÍTULO V | 64 |
| CONCLUSIONES..... | 64 |
| CAPITULO VI | 65 |
| RECOMENDACIONES | 65 |
| Referencias Bibliográficas | 66 |
| Anexos | 69 |
| Anexo 1. Matriz de consistencia..... | 69 |
| Anexo 2. Instrumento..... | 70 |
| Anexo 3. Preparación del dataset | 71 |
| Anexo 3. Evaluación de la Hipótesis General | 73 |
| Anexo 4. Evaluación de la Hipótesis Específica H1 | 76 |
| Anexo 5. Evaluación de la Hipótesis Específica H2 | 78 |
| Anexo 6. Evaluación de la Hipótesis Específica H3 | 81 |

Lista de tablas

| | | |
|---------|---|----|
| Tabla 1 | <i>Operacionalización de variables</i> | 49 |
| Tabla 2 | <i>Instrumento de la investigación</i> | 51 |
| Tabla 3 | <i>Fases y actividades para el procesamiento de datos</i> | 52 |
| Tabla 4 | <i>Librerías para el desarrollo del modelo LSTM y Prophet</i> | 53 |
| Tabla 5 | Comparación de desempeño de modelos | 54 |
| Tabla 6 | <i>Comparación de errores entre modelos</i> | 55 |
| Tabla 7 | <i>Comparación de errores entre modelos</i> | 57 |
| Tabla 8 | <i>Comparación de predicción de ventas</i> | 57 |
| Tabla 9 | <i>Comparación de errores por fold</i> | 58 |

Lista de Figuras

| | |
|---|----|
| Figura 1 <i>Machine Learning</i> | 19 |
| Figura 2 <i>Análisis Exploratorio de Datos</i> | 23 |
| Figura 3 <i>Modelo Prophet</i> | 25 |
| Figura 4 <i>Red neuronal Recurrente</i> | 26 |
| Figura 5 <i>Error predictivo</i> | 29 |
| Figura 6 <i>Root Mean Squared Error</i> | 30 |
| Figura 7 <i>Mean Absolute error</i> | 31 |
| Figura 8 <i>Mean Absolute Percentage Error</i> | 32 |
| Figura 9 <i>Ventas Online</i> | 37 |
| Figura 10: <i>Ejemplo de Predicción</i> | 42 |
| Figura 11: <i>Ejemplo de gráfico de barras agrupadas</i> | 43 |
| Figura 12 <i>Comparación de modelos para predicción de Ventas</i> | 56 |
| Figura 13 <i>Errores por fold</i> | 59 |

CAPÍTULO I

INTRODUCCIÓN

1.1. PLANTEAMIENTO DEL PROBLEMA

En el contexto nacional peruano, el sector retail ha experimentado un crecimiento sostenido en los últimos años, marcado por una acelerada digitalización tras la pandemia de COVID-19. Según datos de la Cámara Peruana de Comercio Electrónico, en el 2023 el e-commerce en Perú creció más de un 30%, alcanzando un valor que superó los 9,000 millones de soles. Sin embargo, junto con este avance también surgieron retos importantes para las empresas retail del país, sobre todo para aquellas que se encuentran en regiones con menor desarrollo de infraestructura digital y que deben competir con grandes plataformas internacionales. En este contexto, contar con la capacidad de predecir las ventas con precisión se ha convertido en un elemento clave para el crecimiento y la permanencia de las empresas locales.

Además, en el caso de los canales digitales, como las tiendas virtuales, esta capacidad resulta todavía más relevante. Comparado con las tiendas físicas, una tienda virtual de esta empresa retail que cuenta con 30 locales físicos y tiene presencia nacional, muestra las ventas más bajas dentro del grupo y carecen de cualquier sistema de predicción. Actualmente, no se implementan modelos de machine learning ni herramientas para anticipar la demanda, la información disponible se limita a los registros históricos de ventas.

De esta manera, la falta de un modelo predictivo dificulta determinar que enfoque tecnológico podría brindar mejores resultados, limitando la optimización del inventario, la planificación de compras y la capacidad de reaccionar ante cambios en el comportamiento del consumidor. Si esta problemática se mantiene, las consecuencias pueden ser graves: en el corto plazo, se corre el riesgo de mantener niveles de inventario inadecuados que afecten la rentabilidad; en el mediano plazo, la empresa podría perder clientes frente a competidores más ágiles y tecnológicamente preparados; y en el largo plazo, la empresa podría ver comprometida su sostenibilidad debido a la pérdida de competitividad.

En este contexto, la presente investigación se propone comparar dos modelos de machine learning —Prophet y LSTM— aplicados a series temporales, para determinar cuál brinda mayor precisión al momento de predecir las ventas diarias del canal virtual en el año 2025. El estudio se enfoca exclusivamente en los datos históricos

de ventas de la tienda virtual, sin intervención directa sobre los procesos internos de la empresa, con el fin de aportar una solución viable y basada en evidencia al problema de la planificación comercial.

1.2. FORMULACIÓN DEL PROBLEMA

1.2.1. Problema general

¿Qué modelo de machine learning basado en series temporales permite predecir con mayor precisión las ventas de la tienda virtual de una empresa retail en el año 2025?

1.2.2. Problemas específicos

- a. ¿Qué variables históricas de ventas son relevantes para la predicción de comportamientos futuros en el entorno retail?
- b. ¿Qué tipo de algoritmo (Prophet o LSTM) ofrece mejor predicción?
- c. ¿Qué tipo de algoritmo (Prophet y LSTM) tiene mejor rendimiento en la predicción?

1.3. JUSTIFICACIÓN DE LA INVESTIGACIÓN

El presente estudio se justifica desde múltiples dimensiones que respaldan su relevancia académica, técnica y social. En primer lugar, desde el ámbito científico, esta investigación busca aportar a la comprensión del desempeño de dos modelos de machine learning —Prophet y LSTM— aplicados a la predicción de series temporales en un contexto real. Aunque existen investigaciones previas sobre ambos algoritmos, en el caso del retail peruano su aplicación sigue siendo escasa. Trabajar con datos históricos de una tienda virtual permite analizar como responden los modelos frente a escenarios con estacionalidad marcada, ventas negativas y variaciones propias del comercio digital, aspectos poco documentados a nivel local.

Asimismo, desde una perspectiva tecnológica, esta investigación promueve el uso de herramientas avanzadas como Python y librerías especializadas para resolver problemas empresariales. Por ejemplo, el modelo LSTM, es una red neuronal recurrente que puede capturar relaciones temporales complejas, mientras que Prophet ofrece una alternativa más simple de ser implementada y ajustada en función de una distribución de series temporales. Compararlos de forma práctica además de ofrecer información sobre su precisión, también da información sobre la facilidad de uso y los recursos necesarios.

También, en la práctica, la investigación responde a la necesidad específica de mejorar la capacidad de anticipar las ventas en el canal web de la empresa. Actualmente, las decisiones se toman basándose únicamente en datos del año anterior, lo que conlleva riesgos de sobrestock o desabastecimiento. Un modelo predictivo más preciso, permitirá optimizar la planificación, reducir costos y asegurar la disponibilidad de productos, lo que se traduce en mayor eficiencia operativa.

Por último, desde el punto de vista social, los beneficios no se limitan a la empresa analizada. Comparado con grandes cadenas internacionales, muchas pequeñas y medianas empresas del retail enfrentan brechas tecnológicas. La adopción de modelos de predicción de ventas puede ayudarlas a evitar pérdidas, mejorar la satisfacción del cliente y fomentar el uso de inteligencia artificial en regiones donde aún existe rezago digital.

En conjunto, la investigación aporta evidencia empírica, conocimiento técnico y soluciones prácticas que responden a una problemática real del comercio electrónico en el Perú.

1.4. LIMITACIONES DE LA INVESTIGACIÓN

El presente estudio presenta diversas limitaciones que deben ser consideradas al momento de interpretar sus resultados. En primer lugar, la investigación se centró exclusivamente en la tienda virtual de una sola empresa retail peruana. Esta delimitación, aunque necesaria para el enfoque predictivo aplicado, restringe la generalización de los hallazgos a otras empresas con contextos, productos o estrategias de venta distintas.

Asimismo, se trabajó con datos históricos de ventas correspondientes al periodo 2022-2024. Aunque este rango temporal incluye suficientes registros para el entrenamiento de modelos de machine learning, corresponde a años posteriores a la pandemia de COVID-19, lo que implica un comportamiento de consumo atípico, acelerado por la digitalización. Esta particularidad podría afectar la capacidad de los modelos para capturar patrones de comportamiento estables a largo plazo.

Desde el punto de vista metodológico, el estudio solo considera dos modelos predictivos: Prophet y LSTM. Si bien ambos son ampliamente utilizados para series temporales, existen otros algoritmos —como ARIMA, XGBoost o modelos híbridos— que también podrían ofrecer buenos resultados. Esta exclusión limita el espectro comparativo de la investigación.

Otra limitación importante es que los modelos fueron entrenados únicamente con

variables internas, es decir, los registros históricos de ventas. No se incorporaron variables exógenas como campañas promocionales, cambios estacionales, indicadores macroeconómicos ni feriados, lo cual podría haber mejorado la precisión de los modelos.

Por último, se reconoce que los datos utilizados provienen directamente del sistema de ventas de la empresa. A pesar de su valor como fuente primaria, podrían contener errores de registro, valores atípicos no corregidos o formatos inconsistentes, los cuales pueden haber influido en el rendimiento de los modelos.

En conjunto, estas limitaciones no invalidan los resultados obtenidos, pero sí acotan su alcance y sugieren precaución al momento de extrapolarlos a otros escenarios o sectores. Futuros estudios podrían superar estas restricciones incluyendo más empresas, periodos más amplios, nuevas variables y un repertorio mayor de modelos predictivos.

1.5. OBJETIVOS

1.5.1. Objetivo general

Determinar qué modelo de machine learning basado en series temporales (Prophet o LSTM) brinda mayor precisión para predecir las ventas de una tienda virtual en una empresa retail durante el año 2025.

1.5.2. Objetivo específico

- a. Identificar y recopilar variables históricas de ventas relevantes para la predicción.
- b. Implementar y entrenar los Tipos de algoritmo (Prophet y LSTM) utilizando datos de ventas reales para la predicción.
- c. Evaluar y comparar el rendimiento de predicción de los Tipos de Algoritmo (Prophet y LSTM) mediante métricas estadísticas (RMSE, MAE y MAPE).

CAPÍTULO II

MARCO TEÓRICO

2.1. ANTECEDENTES DE LA INVESTIGACIÓN

2.1.1. Antecedentes Internacionales

De acuerdo con Saini y Sharma (2024), en India, en su artículo “A Comparison Between Long Short-Term Memory and Prophet for Time Series Analysis and Forecasting Technique”, publicado en la revista *Educational Administration: Theory and Practice*, el objetivo fue comparar los algoritmos LSTM y Prophet en tareas de análisis y predicción de series temporales. La metodología consistió en un diseño comparativo con datos de series históricas, evaluando ambos modelos bajo métricas de precisión y eficiencia. Concluyeron que, mientras LSTM ofrece mayor precisión y es eficaz para capturar patrones complejos y dependencias a largo plazo, requiere una mayor cantidad de datos y ajustes. Por otro lado, Prophet se presenta como una alternativa más accesible y rápida, especialmente útil en escenarios con componentes estacionales marcados y cuando se necesita una implementación ágil y con menor necesidad de configuración técnica.

Según Lima y Santos (2024), en Brasil, en su artículo “Avaliação do modelo de previsão Prophet como ferramenta para preenchimento de falha de dados em séries climáticas”, publicado en los *Anais do Simpósio Brasileiro de Computação*, se compararon los modelos Prophet y LSTM para el llenado de datos faltantes en series temporales meteorológicas de Brasil. El objetivo fue evaluar la eficiencia de ambos métodos en la predicción de variables como temperatura, precipitación y humedad relativa. La metodología incluyó el uso de registros climáticos incompletos, donde ambos modelos fueron entrenados y validados. Concluyeron que Prophet resultó ser una opción más sencilla y rápida de implementar, mientras que LSTM, correctamente calibrado, ofreció mejor rendimiento en algunos escenarios con grandes vacíos de datos, especialmente para series con alta variabilidad.

De acuerdo con González Mata (2020), en España, en su trabajo de grado titulado “A Comparison Between LSTM and Facebook Prophet Models: A Financial Forecasting Case Study”, presentado en la Universitat Oberta de Catalunya, el objetivo fue desarrollar y comparar modelos predictivos de precios del índice S&P500 utilizando LSTM y Prophet. Implementó un simulador de trading para validar económicamente las predicciones. La metodología aplicada fue un diseño experimental con simulaciones de trading basadas en los resultados de ambos modelos. Concluyó que LSTM mostró mejor desempeño en precisión y fue un predictor más robusto frente a estrategias simples, mientras que Prophet, aunque ofreció retornos positivos, fue menos eficaz para anticipar el comportamiento bursátil.

De acuerdo con Meng, Yang, Yang y Liu (2021), en China, en su artículo “Comparative Analysis of Prophet and LSTM Model in Drug Sales Forecasting”, publicado en el Journal of Physics: Conference Series, el objetivo fue comparar ambos modelos de predicción aplicados a las ventas farmacéuticas. Utilizando datos reales de una empresa china, analizaron la capacidad de LSTM y Prophet para prever la demanda futura de medicamentos. La metodología consistió en el uso de datos históricos de ventas de medicamentos de una empresa china, implementando ambos algoritmos y evaluando métricas de error. Concluyeron que LSTM fue más preciso que Prophet, mostrando mejor capacidad para modelar dependencias de largo plazo y capturar patrones complejos en la serie temporal.

2.1.2. Antecedentes Nacionales

De acuerdo a Porras (2024), en Perú, en su tesis de pregrado titulado “Machine Learning para predecir la demanda del limón en el Mercado Mayorista de Lima”, el objetivo fue aplicar machine learning para predecir las ventas en el Mercado Mayorista para evitar o reducir el impacto negativo del desabastecimiento del limón, y así contribuir a las personas involucradas con la compra y venta de dicho producto. La metodología consistió en utilizar datos históricos de ventas del mercado mayorista, aplicando modelos Prophet y LSTM para el pronóstico. Concluyó que los modelos temporales LSMT y Prophet son los más idóneos para pronosticar la demanda del limón.

En la investigación de Campos & León (2023), en Perú, titulada “Aplicación para la compra eficiente de materia prima en Jr. Group SAC, utilizando análisis predictivo”, el objetivo fue implementar un modelo de predicción de demanda de materia prima. La metodología empleada fue el análisis de datos históricos de la empresa y el desarrollo de series temporales. Concluyeron que la solución mejora la eficiencia en la planificación de compra de materia prima e influye en el desempeño operativo y competitividad de la

empresa.

Por otra parte, Salazar (2022), en Perú, en su tesis de pregrado titulado: “Pronóstico de Ventas de Productos en la industria del Retail en base a similitud de series temporales”, cuyo objetivo fue investigar la posibilidad de realizar pronósticos en base a similitudes de comportamientos de ventas históricas. La metodología consistió en la construcción de un modelo predictivo basado en registros de al menos tres años y en un análisis exploratorio de datos. Concluyó que contar con un dataset amplio y un adecuado análisis preliminar hace mucho más robusto al modelo predictivo.

2.2. MARCO CONCEPTUAL

2.2.1. Ciencia de datos

Como afirma Cao (2020), la ciencia de datos es una disciplina que busca encontrar patrones significativos dentro de grandes cantidades de información. Su propósito principal es transformar esos datos, que en principio no dicen mucho, en conocimiento útil que apoye la toma de decisiones. Esta área surge como una respuesta natural al crecimiento constante de los volúmenes de datos y también a la necesidad de analizarlos de manera más rápida y efectiva. Además, no se trata solo de aplicar fórmulas o algoritmos, sino de combinar lo estadístico, lo matemático y lo computacional con la comprensión del contexto en el que los datos aparecen. Por eso, hoy en día se aplica tanto en la salud como en el comercio digital.

Asimismo, Hernán et al. (2018) señalan que la ciencia de datos no se limita a una sola tarea, sino que integra diferentes aproximaciones. Hablan de tres ejes principales: lo descriptivo, lo predictivo y lo contrafactual. En lo descriptivo, la meta es organizar la información para entender qué pasó en el pasado; en lo predictivo, lo central es estimar qué podría pasar en el futuro a partir de los datos disponibles; y lo contrafactual, que ya es más hipotético, se refiere a preguntas del tipo “qué hubiera sucedido si una acción no se realizaba”. Estas dimensiones muestran que la ciencia de datos no se limita a analizar cifras, sino que ayuda a responder preguntas prácticas en distintos contextos.

Por otro lado, DataCamp y Data Science UC mencionan que el proceso incluye las etapas de extracción de datos, procesamiento, modelado y comunicación de resultados. Estas fases no siempre se desarrollan de manera lineal, ya que en la práctica suelen superponerse o repetirse según la naturaleza del proyecto. Un ejemplo sencillo sería cuando, en medio del modelado, aparecen inconsistencias en los datos y se debe regresar al proceso de limpieza. Otro punto que resaltan es la importancia de la

comunicación de resultados, que ahora es tan importante como el mismo modelado, porque no alcanza con tener un modelo complejo si los hallazgos no pueden transmitirse de forma clara para convertirse en decisiones útiles para la organización.

En resumen, la ciencia de datos combina técnicas estadísticas, herramientas informáticas y conocimientos del contexto en el que se aplican los datos. Y no solo se limita al procesamiento de datos, sino que también se centra en las tendencias, facilitar decisiones de manera más ágil, y en detectar patrones ocultos. Desempeña un papel estratégico a nivel empresarial, porque ayuda a mejorar las operaciones diarias, personalizar los servicios, y además anticipar la demanda. Este aporte es importante en el comercio electrónico, donde los cambios cambian constantemente.

2.2.2. Machine learning

El machine learning, también llamado aprendizaje automático, es una rama de la inteligencia artificial que busca que las máquinas aprendan de la experiencia. Es decir, en vez de programarlas con reglas rígidas, lo que se hace es entrenarlas con datos para que detecten patrones y puedan hacer predicciones por sí solas. En pocas palabras, se intenta que los sistemas mejoren su desempeño conforme reciben más información y casos reales.

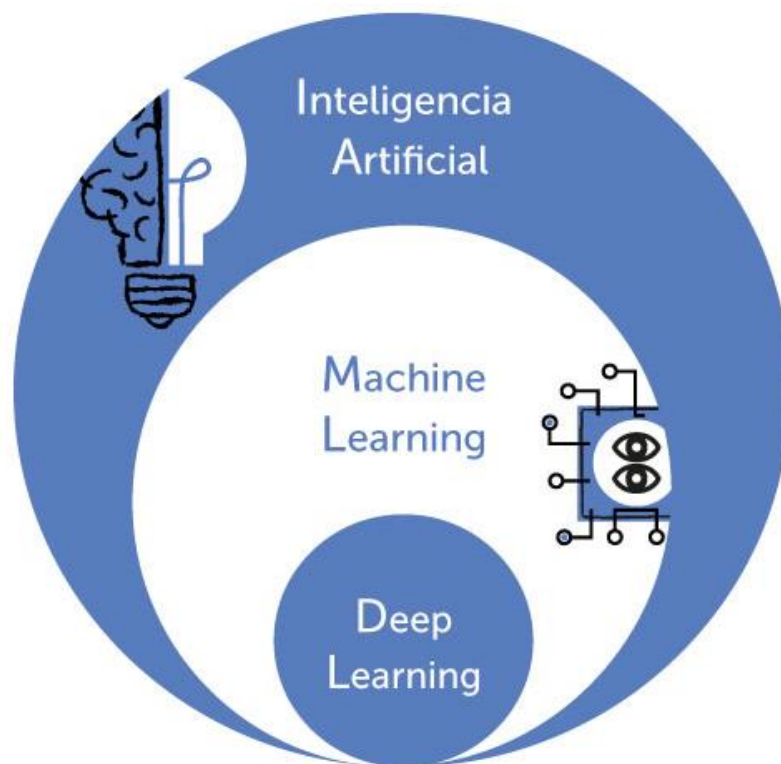
En esa misma línea, Mitchell (1997) lo define como un programa de aprendizaje que es exitoso cuando logra mejorar su rendimiento a medida que acumula experiencia. Es decir, mientras más ejemplos ve el sistema, más aprende. Aunque esta explicación parece sencilla, lo que está detrás son procesos complejos de entrenamiento que permiten a un algoritmo ir ajustando sus “reglas” internas sin intervención humana directa.

De la misma manera, Domingos (2012) menciona que lo innovador del machine learning es que los sistemas dejan de depender completamente de reglas escritas por un programador, y en cambio construyen su propio conocimiento a partir de ejemplos concretos. Esto significa que pueden adaptarse a situaciones nuevas que no estaban previstas al inicio, algo muy útil en contextos como el comercio, donde los patrones de compra cambian constantemente.

Hoy en día, el machine learning se usa en muchísimos campos: salud, finanzas, transporte y, claro, en el comercio electrónico. Su ventaja principal es que permite procesar grandes cantidades de datos de forma rápida y útil, algo que manualmente sería casi imposible. En el caso del retail, por ejemplo, se aprovecha para predecir tendencias de ventas, ajustar inventarios o recomendar productos a los clientes, lo que hace más eficiente la operación y al mismo tiempo mejora la experiencia del consumidor.

Además, Fernández (2021) argumenta que su impacto en la industria se debe a que permite automatizar procesos, identificar anomalías y anticipar comportamientos, incluso cuando las relaciones entre variables no son evidentes. En el caso del sector retail, su uso en la predicción de ventas ha demostrado ser especialmente valioso para anticipar la demanda, optimizar la gestión de inventarios y apoyar en decisiones estratégicas basadas en datos históricos.

Figura 1 *Machine Learning*



Nota: La figura representa la jerarquía de Deep learning, machine learning y la inteligencia artificial. Fuente: Talento Corporativo. (s.f.), (2025)

2.2.3. Predicción

De acuerdo con Russell y Norvig (2021) la predicción se entiende como la capacidad de un sistema para anticipar estados futuros que todavía no se han observado, a partir de ciertas características que si son conocidas. Dicho de otra forma, se busca proyectar lo que puede pasar en base a lo que ya ocurrió. Esto es posible gracias al uso de modelos inductivos que van identificando regularidades estadísticas aprendidas de experiencias previas. Dentro de esta se incluye dos tipos básicos de predicción: la clasificación y la regresión, las cuales se consideran formas elementales para abordar este proceso.

Por su parte, Hyndman y Athanasopoulos (2021), señalan que la predicción no solo es estimar un valor futuro, sino que también implica expresar esta proyección considerando la incertidumbre que siempre existe. En otras palabras, las predicciones deben comunicarse en términos probabilísticos, ya que rara vez se puede tener certeza absoluta. Así, además de dar un valor puntual, se recomienda incluir intervalos de confianza o distribuciones predictivas que ofrezcan un panorama más realista y útil para la toma de decisiones.

En síntesis, dentro del aprendizaje automático, la predicción se entiende como un proceso donde se proyectan valores desconocidos a partir de patrones detectados en los datos históricos. No se trata de adivinar, sino de aprender del pasado para anticipar lo que pueda suceder después, aunque siempre reconociendo que existe un margen de error.

2.2.4. Predicción de Ventas

Según Chopra & Meindl (2019) la predicción de ventas es una tarea esencial dentro de la gestión empresarial, porque influye de directamente en decisiones como el control de inventarios, la planificación logística o la organización de campañas publicitarias.

Asimismo, Armstrong (2001) hace una clasificación de los métodos de predicción en dos grupos: cualitativos y cuantitativos; estos últimos son los que incluyen los modelos de series temporales, los cuales suelen ser muy usados en el sector retail porque permiten reconocer y modelar patrones de demanda en el tiempo.

En ese sentido, la predicción de ventas puede entenderse como un proceso analítico que busca anticipar cómo se comportarán las ventas en el futuro a partir de registros históricos y con el apoyo de modelos matemáticos. Claro que este procedimiento no es exacto, ya que depende mucho de la calidad de los datos y de la naturaleza propia de la serie, pero sigue siendo una herramienta esencial para la toma de decisiones empresariales.

2.2.5. Predicción de Ventas en el Contexto Actual

La predicción de ventas se ha consolidado como una herramienta clave para la planificación estratégica de empresas retail y de e-commerce, en especial en entornos de alta competitividad y variabilidad del mercado. Tradicionalmente, las series temporales se abordaban con modelos estadísticos como ARIMA o SARIMA (Box et al., 2015; Hyndman & Athanasopoulos, 2018). Sin embargo, en los últimos años, los

avances en aprendizaje automático han permitido el uso de modelos más complejos que capturan patrones no lineales y dependencias de largo plazo (Mitchell, 1997; Domingos, 2012).

En este marco, LSTM (Long Short-Term Memory) y Prophet se han convertido en dos de las herramientas más utilizadas. Prophet, desarrollado por Facebook, se caracteriza por su facilidad de uso y su capacidad para modelar tendencias y estacionalidad de forma automática, lo que lo hace atractivo para empresas con recursos limitados en ciencia de datos (Brownlee, 2020). Por su parte, LSTM, al ser una red neuronal recurrente especializada en secuencias, ha demostrado mayor precisión en contextos de alta variabilidad (Brownlee, 2021).

2.2.6. Modelado de series temporales

El modelado de series temporales se refiere al análisis de datos que se van registrando de manera secuencial a lo largo del tiempo. Es decir, no son números aislados, sino que están ordenados cronológicamente y suelen mostrar dependencias entre un momento y el siguiente. Este tipo de análisis se utiliza en muchos ámbitos — desde la economía hasta la salud o el retail digital— porque permite entender cómo cambian las variables con el paso del tiempo y anticipar posibles comportamientos futuros.

Según Box et al. (2015) el modelado de series temporales es "un conjunto de técnicas estadísticas para analizar datos temporales con el objetivo de identificar elementos como tendencias, estacionalidad, ciclos y variaciones irregulares, con el fin de construir modelos que reflejen la dinámica temporal subyacente" (p. 7).

De la misma manera, como señalan Hyndman y Athanasopoulos (2021), el modelado de series temporales combina el rigor de la estadística tradicional con la flexibilidad de los métodos del aprendizaje automático. Gracias a esa integración, es posible abordar fenómenos complejos como relaciones no lineales, heteroscedasticidad o cambios estructurales, que son bastante frecuentes cuando se trabaja con datos de la vida real.

2.2.7. Data

De acuerdo con Provost & Fawcett (2013), el término datos hace referencia a valores individuales, ya sean cualitativos o cuantitativos, estos describen hechos, observaciones o mediciones asociadas a un fenómeno o proceso. En el análisis de datos, constituyen el insumo básico que, a través de métodos estadísticos y computacionales, se transforman en información útil.

Asimismo, Davenport y Harris (2007) enfatizan que la ventaja competitiva de las organizaciones actuales radica en su capacidad para “competir con analítica”, es decir, convertir grandes volúmenes de datos en decisiones estratégicas.

De la misma manera, Kitchin (2014) amplía esta idea al señalar que la era del big data no solo implica un aumento en la cantidad de información disponible, sino también en la aparición de nuevas infraestructuras y prácticas de gestión de datos que redefinen la manera en que se genera y utiliza el conocimiento. De esta forma, los datos dejan de ser solo un recurso operativo para atender necesidades inmediatas y pasan a tener un papel orientador en las decisiones de más largo plazo dentro de las organizaciones.

2.2.8. Exploratory Data Analysis (EDA)

Tukey (1977) menciona que el Análisis Exploratorio de Datos (EDA, por sus siglas en inglés) es básicamente la primera etapa dentro del análisis estadístico. Lo que busca es describir de manera sencilla las principales características de un conjunto de datos, usando tanto gráficos como medidas numéricas que ayudan a tener una primera idea de la información. Esta fase es muy útil porque permite ver patrones, encontrar anomalías y decidir qué modelo puede ser más adecuado para continuar con el análisis.

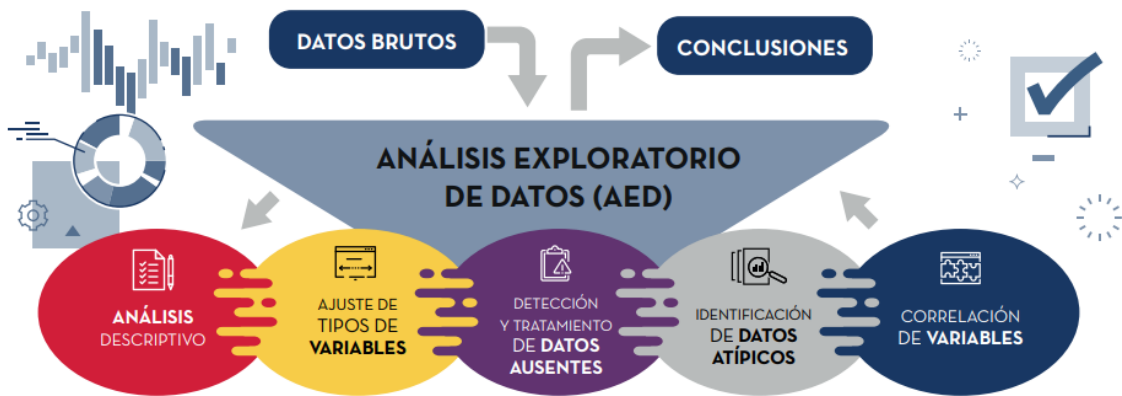
Según el NIST (2013), los datos no deberían usarse solo para resolver problemas inmediatos, sino que también cumplen un papel estratégico. Dicho de otra forma, además de servir para responder preguntas puntuales, se convierten en un apoyo para planificar y tomar decisiones dentro de la organización.

Cleveland (1993) resalta la importancia de las representaciones gráficas, sobre todo las interactivas, porque hacen más fácil detectar relaciones y comportamientos que a simple vista o con solo números podrían pasar desapercibidos.

Behrens (1997) agrega que el EDA no debe entenderse como un proceso rígido, sino más bien flexible. Es decir, las hipótesis iniciales pueden cambiar a medida que avanza el estudio, lo que permite un aprendizaje más adaptativo y dinámico.

Finalmente, Wickham y Grolemund (2016) destacan que con herramientas modernas como ggplot2 y dplyr este tipo de análisis se ha fortalecido bastante. Gracias a estas librerías, la visualización se combina con la manipulación de grandes volúmenes de datos, lo que facilita hacer análisis más ordenados, reproducibles y aplicables tanto en la academia como en los negocios.

Figura 2 *Análisis Exploratorio de Datos*



Nota: La figura representa los pasos o fases de un Análisis exploratorio de datos. Fuente: Iniciativa Aporta. (2024).

2.2.9. Dato atípico (outlier)

Aggarwal (2017) señala que un dato atípico es una observación que se aparta de manera clara del comportamiento que siguen la mayoría de los datos. Estos valores pueden aparecer por distintas razones: a veces son errores de medición, otras veces son variaciones extremas pero válidas, y también pueden reflejar fenómenos poco frecuentes que vale la pena analizar con más detalle. Detectarlos es importante porque, si se ignoran, pueden alterar los resultados y llevar a conclusiones poco fiables.

Barnett y Lewis (1994) coinciden en que identificar este tipo de valores no solo ayuda a que los modelos estadísticos sean más precisos, sino que también puede dar pistas sobre eventos poco comunes que enriquecen la interpretación de la información.

Por su parte, Chandola, Banerjee y Kumar (2009) resaltan que la detección de outliers es especialmente importante en áreas sensibles como la salud, la seguridad o el monitoreo industrial. En estos casos, una sola anomalía puede ser la señal temprana de un problema mayor que necesita ser atendido de inmediato.

2.2.10. Media

Según Triola (2018) la media aritmética es probablemente la medida de tendencia central más utilizada. Se obtiene al dividir la suma de todos los valores entre el número total de observaciones y permite tener una idea del “promedio” de una variable. Sin embargo, es importante señalar que puede verse afectada cuando existen valores extremos.

Por su parte, Moore y McCabe (2014) describen la media como un “punto de equilibrio” en la distribución, lo que la convierte en una medida adecuada cuando los

datos presentan cierta simetría.

Finalmente, Montgomery y Runger (2018) añaden que, cuando se combina con medidas de dispersión como la desviación estándar, la media permite caracterizar con mayor detalle la variabilidad de un conjunto de datos y así obtener una visión más completa del fenómeno estudiado.

2.2.11. Suavizado de series temporales

El suavizado con media móvil es una técnica estadística empleada para reducir las fluctuaciones aleatorias de corto plazo en una serie temporal y, con ello, resaltar patrones más claros como tendencias o ciclos.

Frost (2020) señala que este procedimiento consiste en calcular el promedio de valores cercanos dentro de una ventana móvil, lo cual ayuda a eliminar el “ruido” y hace que la interpretación sea más sencilla.

De acuerdo con Gundersen (2022), esta técnica se vuelve especialmente útil en contextos donde los datos son muy variables, ya que permite seguir la evolución de la serie sin perder la memoria de los cambios recientes.

En el caso de esta investigación, se aplicó un suavizado con media móvil de 7 días como parte del preprocesamiento de datos para el modelo LSTM. Con ello se buscó que el algoritmo reconociera los patrones diarios sin que los picos extremos o las oscilaciones aleatorias afectaran su desempeño.

2.2.12. Tipo de Algoritmo

El tipo de algoritmo hace referencia a la familia de modelos que se emplea en aprendizaje automático, cada una con fortalezas y limitaciones propias. En este estudio, la comparación se centra en dos enfoques distintos: Prophet, un modelo basado en procesos aditivos que descompone la serie en estacionalidad, tendencia y eventos especiales; y LSTM, una red neuronal recurrente diseñada para capturar dependencias temporales de largo plazo.

La elección del algoritmo constituye un aspecto central en esta investigación, pues permite determinar cuál de los dos ofrece un mejor desempeño predictivo en el contexto de ventas online.

2.2.13. Modelo Prophet

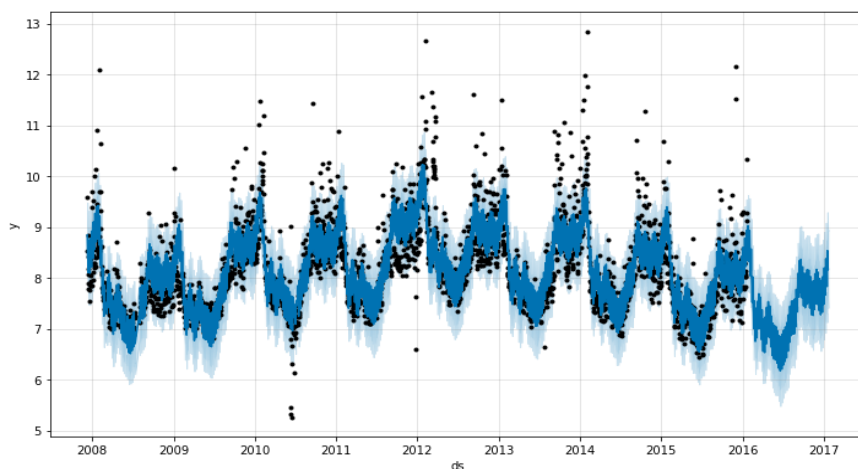
Prophet, desarrollado por Taylor y Letham (2018) en Facebook, es un modelo pensado para la predicción de series temporales que utiliza un enfoque aditivo. Su lógica consiste en descomponer la serie en tres elementos principales: tendencia,

estacionalidad y efectos de días festivos. Gracias a esta estructura, resulta especialmente útil en escenarios donde los datos presentan patrones estacionales marcados, cambios abruptos o comportamientos que tienden a repetirse, algo bastante común en el comercio electrónico.

De acuerdo con Villalobos (2021), Prophet muestra un desempeño sólido cuando la serie presenta estacionalidad anual o tendencias no lineales, además de que es capaz de ajustarse ante cambios repentinos. Otro punto a favor es que está disponible tanto en Python como en R, lo que ha facilitado su uso tanto en investigaciones académicas como en aplicaciones empresariales.

En el marco de este estudio, Prophet se considera una herramienta adecuada para modelar las ventas de una tienda virtual, ya que permite identificar picos relacionados con campañas de marketing o temporadas de mayor consumo. Asimismo, su capacidad para incluir variables externas, como promociones o feriados, lo convierte en un modelo flexible y bien adaptado a las dinámicas del sector retail, donde la estacionalidad y las variaciones diarias tienen un impacto directo en las ventas.

Figura 3 *Modelo Prophet*



Nota: La figura representa una predicción de series temporales realizada con el modelo Prophet. Fuente: Taylor, S. J., & Letham, B. (2018).

2.2.14. Red Neuronal Recurrente (RNN)

Las redes neuronales recurrentes (RNN, por sus siglas en inglés) son un tipo de modelo de aprendizaje profundo pensado para trabajar con datos que siguen un orden en el tiempo. A diferencia de otros algoritmos que procesan los datos como si fueran independientes, las RNN toman en cuenta la secuencia, lo que les da ventaja en situaciones donde el contexto importa, como ocurre con el texto, el audio o las series temporales.

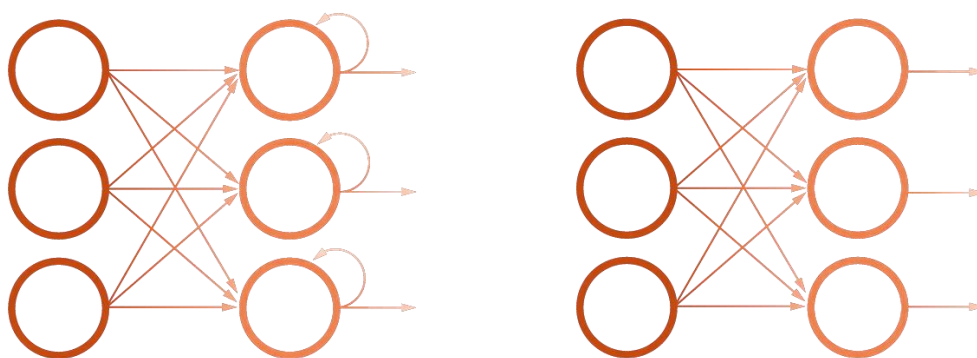
Según Goodfellow, Bengio y Courville (2016), la clave de estas redes está en que incluyen conexiones cíclicas en sus capas ocultas. Esto hace que la salida de un instante se use también como parte de la entrada del siguiente, lo que les permite “recordar” lo ocurrido antes y modelar dependencias de corto y también de más largo plazo.

De manera similar, Schuster y Paliwal (1997) mencionan que estas redes funcionan mejor que las tradicionales cuando se trabaja con datos secuenciales, ya que el orden en que aparecen los elementos cambia totalmente su interpretación.

Ahora bien, como comenta Schmidhuber (2015), no todo son ventajas. Uno de los principales problemas de las RNN es el desvanecimiento del gradiente, que complica mucho el aprendizaje cuando las dependencias entre los datos son demasiado largas. Para enfrentar esto, se propusieron mejoras como el backpropagation through time (BPTT) y también variantes más avanzadas, como las LSTM y las GRU, que logran extender la memoria y manejar relaciones más prolongadas.

Por otra parte, Graves (2013) muestra que la capacidad de las RNN para captar patrones en secuencias permitió grandes avances en áreas como el reconocimiento de voz y la traducción automática. Finalmente, Bengio, Simard y Frasconi (1994) resaltan que su mayor potencial está en modelar distribuciones probabilísticas de datos secuenciales, lo que las convierte en una herramienta valiosa para tareas como la predicción de series, el análisis de sentimientos o incluso la generación de texto.

Figura 4 Red neuronal Recurrente



Nota: La figura representa la retropropagación para una RNN. Fuente: <https://datademia.es/blog/que-es-deep-learning-y-que-es-una-red-neuronal> (2025).

2.2.15. Redes Neuronales de Memoria de Largo Plazo (LSTM)

Según Hochreiter y Schmidhuber (1997) las redes de memoria a largo plazo (Long Short-Term Memory, LSTM) son una variante de las redes neuronales recurrentes

(RNN), fueron creadas para resolver el problema que tienen las RNN tradicionales cuando intentan aprender dependencias largas en los datos. A diferencia de las RNN tradicionales, las LSTM incorporan tres compuertas principales: entrada, olvido y salida. Estas compuertas funcionan como filtros que ayudan al modelo a decidir que información guardar y cuál dejar de lado, de modo que pueda enfocarse en los datos que si son importantes para el aprendizaje.

Este diseño les da una ventaja frente a las RNN básicas, sobre todo en escenarios donde la información relevante no puede perderse rápidamente, como en la predicción de ventas, el análisis de la conducta de los usuarios o el seguimiento de precios a lo largo del tiempo.

Brownlee (2021) menciona que, gracias a esta arquitectura, las LSTM no sufren tanto del problema del gradiente que desaparece, algo que solía limitar bastante a las RNN clásicas. En otras palabras, logran aprender secuencias largas sin “olvidarse” tan rápido, lo que las vuelve más estables y efectivas para trabajar con series temporales que presentan relaciones complejas.

Por esta razón, las LSTM no se quedaron solo en la teoría, sino que empezaron a usarse en distintos campos: desde la traducción automática y el análisis financiero, hasta el procesamiento del lenguaje natural y la predicción de la demanda. En el caso particular de esta investigación, se optó por este modelo para estimar las ventas diarias de una tienda virtual, ya que su capacidad de reconocer secuencias no lineales y de adaptarse a cambios frecuentes lo convierte en una herramienta muy adecuada

2.2.16. Precisión del modelo

Chai y Draxler (2014) explican que, en el contexto de la predicción, la precisión se entiende como el grado de cercanía entre los valores estimados por un modelo y los valores reales observados. En otras palabras, un modelo preciso es aquel que no solo acierta en sus cálculos, sino que también da resultados confiables y útiles para la toma de decisiones.

Por su parte, Hyndman y Koehler (2006) comentan que esa precisión se puede medir con distintas métricas, como el error absoluto medio (MAE), el error cuadrático medio (MSE) y el coeficiente de determinación (R^2). Cada una muestra algo distinto: el MAE dice en promedio cuánto se alejan las predicciones de lo real, el MSE pone más peso a los errores grandes y el R^2 indica qué tanto logra el modelo explicar la variabilidad de los datos.

Al final, usar varias de estas métricas en conjunto ayuda a tener una idea más completa y realista de cómo se está comportando el modelo. Esto facilita comparar

diferentes opciones y elegir la que mejor se ajuste, sobre todo cuando lo que se busca es una herramienta confiable para la gestión en las empresas.

2.2.17. Comparación Prophet y LSTM

Ecevit et al. (2023) compararon LSTM, Prophet y SARIMA en la predicción de ventas horarias en una plataforma de e-commerce. La metodología incluyó validación hold-out y métricas RMSE, MAPE y R^2 . Los resultados confirmaron que LSTM alcanzó mayor precisión que Prophet, aunque con mayor complejidad computacional.

Ensafi et al. (2022) aplicaron Prophet, LSTM y CNN en la predicción de ventas de productos estacionales, concluyendo que los modelos de aprendizaje profundo capturan mejor la estacionalidad y la variabilidad no lineal, lo que resulta especialmente relevante en mercados de consumo masivo.

Ni, Peng & Liu (2022) desarrollaron un modelo Bi-LSTM para la predicción de demanda logística en e-commerce de alimentos frescos. Este estudio evidenció que la incorporación de variables externas (condiciones climáticas, calendario) mejora la precisión de las predicciones, lo que abre una línea futura de investigación para el retail peruano.

PeerJ (2025) publicó un estudio de predicción de ventas en retail mediante modelos híbridos de redes neuronales, concluyendo que la combinación de métodos mejora la precisión y estabilidad de los pronósticos.

Kizgin et al. (2025) destacaron que los modelos de machine learning deben ajustarse a periodos de crisis o alta incertidumbre, ya que eventos extraordinarios pueden reducir la precisión de las predicciones. Este hallazgo aporta a la discusión de las limitaciones de los modelos aplicados.

2.2.18. Error

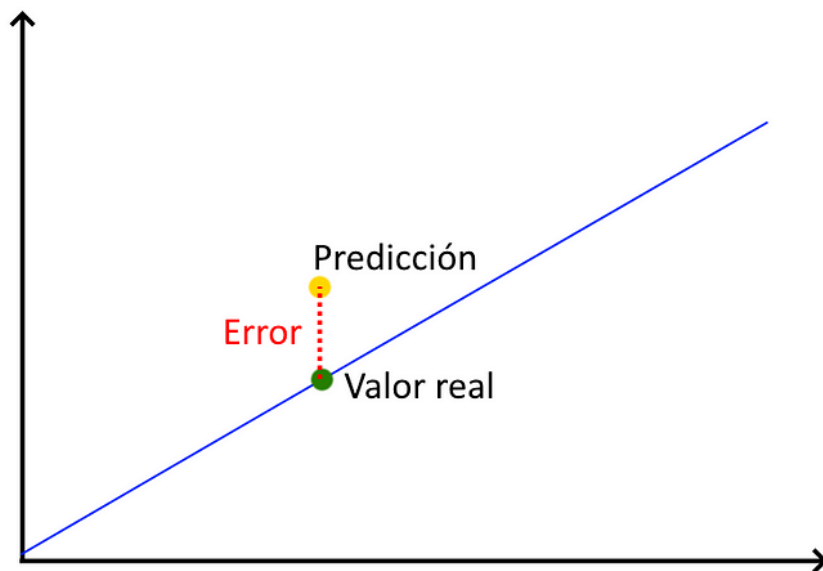
Hyndman y Athanasopoulos (2018) explican que, dentro de los modelos predictivos, el error se entiende como la diferencia entre lo que realmente ocurre y lo que el modelo estima. En otras palabras, es la medida que nos dice qué tan lejos estamos de la realidad. Analizarlo resulta fundamental porque no solo revela el grado de desviación, sino que también permite ajustar el modelo y, con ello, mejorar su rendimiento en futuras predicciones.

No obstante, no todos los errores se miden de la misma manera. Willmott y Matsuura (2005) diferencian que el RMSE (Root Mean Squared Error) da mayor importancia a los errores grandes, mientras que el MAE (Mean Absolute Error) asigna el mismo peso a todas las desviaciones, siendo menos sensible a valores extremos.

En la misma línea, Draper y Smith (1998) destacan que examinar con detalle los errores durante la validación cruzada ayuda a prevenir problemas como el sobreajuste o una visión demasiado optimista del modelo. Por su parte, James, Witten, Hastie y Tibshirani (2013) sugieren que no basta con revisar el error en el conjunto de entrenamiento, porque eso puede dar una idea equivocada del rendimiento. Ellos recomiendan también calcularlo sobre los datos de prueba, ya que de esa manera se obtiene una visión más cercana de cómo funcionará el modelo cuando se enfrente a información nueva.

En definitiva, una gestión cuidadosa del error no solo contribuye a que el modelo generalice mejor, sino que también aumenta la confianza en sus predicciones, lo que es clave tanto en aplicaciones empresariales como en investigaciones científicas.

Figura 5 *Error predictivo*



Nota: La figura representa el error como diferencia entre la predicción y el valor real. Fuente: Arrijoa, N. (2021).

2.2.19. Métricas de validez

Hyndman y Athanasopoulos (2018) comentan que, al momento de evaluar un modelo predictivo, no basta con mirar el resultado final. Lo importante es apoyarse en métricas que permitan ver cómo está funcionando el modelo y si de verdad ayuda a resolver el problema planteado. En el caso de las series temporales, destacan que algunas de las más usadas son la raíz del error cuadrático medio (RMSE) y el error porcentual absoluto medio (MAPE), porque dan una idea bastante clara de qué tan cerca están las predicciones respecto a los valores reales.

Por otro lado, Kuhn y Johnson (2018) señalan que estas métricas no solo sirven

para medir qué tan exacto es el modelo, sino que también ayudan a evaluar aspectos igual de importantes, como su estabilidad frente a cambios en los datos, si el modelo está bien calibrado o si realmente logra diferenciar patrones. Además, recuerdan que existen varios indicadores que complementan la evaluación, como el error absoluto medio (MAE), el error cuadrático medio (MSE), el coeficiente de determinación (R^2) y el propio MAPE. Cada uno muestra una parte distinta del desempeño y, al usarlos en conjunto, se hace más fácil comparar modelos y quedarse con el que mejor se ajuste a cada situación.

2.2.20. RMSE (Root Mean Squared Error)

Hyndman y Athanasopoulos (2018) explican que la raíz del error cuadrático medio (RMSE) se ha convertido en una de las métricas más utilizadas cuando se analizan modelos de predicción. Su utilidad radica en que resume en un solo valor la magnitud promedio de las diferencias entre lo que el modelo predice y lo que realmente ocurre, aunque dando un mayor peso a los errores grandes. Esta particularidad hace que se considere un buen referente de la precisión de los modelos, sobre todo cuando los desajustes importantes no deben pasar desapercibidos.

En la misma línea, Chai y Draxler (2014) destacan que el RMSE es muy sensible frente a valores atípicos, lo que en algunos casos resulta beneficioso porque permite detectar desviaciones importantes en el comportamiento del modelo. Sin embargo, Willmott y Matsuura (2005) advierten que confiar únicamente en esta métrica puede llevar a interpretaciones sesgadas. Por ello, recomiendan complementarla siempre con otras medidas de error para obtener una evaluación más completa y equilibrada del desempeño predictivo.

Figura 6 *Root Mean Squared Error*

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (y_i - x_i)^2}{n}}$$

Nota: La figura representa la fórmula del RMSE, donde Y son los valores predichos y X los valores reales. Fuente: Arrijoja, N. (2021).

2.2.21. MAE (Mean Absolute Error):

Willmott y Matsuura (2005), explican que el error absoluto medio (MAE) es el promedio de las diferencias absolutas entre los valores reales y los pronosticados. A diferencia del RMSE, este indicador otorga el mismo peso a todos los errores, sin importar su magnitud, lo que lo convierte en una alternativa adecuada cuando se busca reducir la media del error sin dar prioridad a las desviaciones más grandes.

Por su parte, Hyndman y Koehler (2006) explican que el MAE tiene la ventaja de ser más fácil de interpretar que el RMSE, sobre todo en escenarios donde no es necesario penalizar de manera adicional los errores grandes. En la misma línea, Armstrong (2001) resalta que su sencillez y claridad facilitan su comunicación en entornos empresariales, especialmente cuando los resultados deben ser presentados a públicos no técnicos.

Figura 7 Mean Absolute error

$$MAE = \frac{\sum_{i=1}^n |y_i - x_i|}{n}$$

Nota: La figura representa la fórmula del MAE, donde Y son los valores predichos y X los valores reales. Fuente: Arriola, N. (2021).

2.2.22. MAPE (Mean Absolute Percentage Error):

De acuerdo con Makridakis, Wheelwright y Hyndman (1998), el error porcentual absoluto medio (MAPE) se usa para calcular el promedio del error absoluto, pero expresado como porcentaje del valor real. Esto hace que sea una métrica bastante práctica, ya que permite comparar modelos aun cuando las series tengan diferentes escalas o magnitudes.

Además, Kim y Kim (2016) comentan que otra de sus ventajas es que resulta fácil de interpretar, lo cual es importante sobre todo en ámbitos empresariales, donde se necesita comunicar los resultados de manera clara a personas que no siempre dominan la parte técnica.

Por otro lado, Flores (1986) advierte que el MAPE no siempre funciona bien. El

problema aparece cuando los valores reales se acercan mucho a cero, porque en esos casos el indicador puede dar resultados engañosos o distorsionados. Por esa razón, el autor sugiere que siempre se complemente con otras métricas que no tengan ese mismo inconveniente.

Figura 8 *Mean Absolute Percentage Error*

$$MAPE = \frac{100}{n} \sum_{i=1}^n \left| \frac{y_i - x_i}{y_i} \right|$$

Nota: La figura representa la fórmula del MAPE, donde Y son los valores predichos y X los valores reales. Fuente: Arrijoja, N. (2021).

2.2.23. Validación temporal

Según Hyndman y Athanasopoulos (2018), la validación temporal es un procedimiento diseñado para evaluar modelos de series de tiempo manteniendo intacto el orden cronológico de los datos. La lógica detrás de esto es bastante clara: en un escenario real, la información futura nunca está disponible al momento de entrenar un modelo, por lo que resulta necesario que los datos de entrenamiento siempre antecedan a los de prueba. Si no se respeta este principio, las métricas obtenidas perderían realismo y darían una falsa idea del desempeño del modelo.

De acuerdo con Gholamy, Kreinovich y Kosheleva (2018), la validación temporal cobra una relevancia especial en contextos donde los datos tienen un peso directo en la toma de decisiones, como en las proyecciones financieras o en la estimación de la demanda. En estos escenarios, incluso un error pequeño en el procedimiento puede dar la impresión de que el modelo es más preciso de lo que realmente es, lo cual se traduce en riesgos concretos. Por ejemplo, en el caso de las ventas digitales, como las analizadas en esta investigación, un modelo mal validado podría generar predicciones demasiado optimistas y llevar a la empresa a sobrevalorar sus ingresos, tomando decisiones de stock o inversión que finalmente resulten insostenibles.

Asimismo, Raschka y Mirjalili (2020) mencionan que los métodos más tradicionales de validación, como el k-fold, no son apropiados en este contexto. La razón es que al dividir los datos en particiones aleatorias se rompe la secuencia natural de la serie, como si cada observación fuese independiente de las demás, cuando en realidad

existe una fuerte autocorrelación. En otras palabras, el valor de un día depende de lo ocurrido en los días previos, y si se ignora esa dependencia, los resultados obtenidos no reflejarán la realidad.

Por su parte, Chollet (2021) señala que, al trabajar con redes neuronales aplicadas a series temporales, es indispensable utilizar validación basada en tiempo. Este procedimiento no solo contribuye a disminuir sesgos y prevenir el sobreajuste, sino que también otorga una visión más clara de cómo se comportará el modelo frente a nuevos escenarios. En la práctica, esto significa que un modelo validado correctamente no se limita a “memorizar” los datos históricos, sino que desarrolla cierta capacidad de adaptación para afrontar observaciones que todavía no existen en la base de datos.

2.2.24. División de Entrenamiento y Prueba (Train/Test Split)

De acuerdo con Brownlee (2020), el método de dividir los datos en entrenamiento y prueba consiste básicamente en separar la información en dos partes: una que se usa para que el modelo aprenda y otra, formada por datos más recientes, que sirve para comprobar qué tan bien funciona. La idea es simular un escenario real de predicción, donde no se cuenta con información futura y se necesita evaluar si el modelo es capaz de dar buenos resultados con datos que nunca “vio” antes.

Este procedimiento tiene como punto fuerte la sencillez y la rapidez, lo que explica que sea tan utilizado en la práctica. Sin embargo, también tiene limitaciones importantes, ya que al trabajar con una sola división no muestra cómo podría variar el rendimiento del modelo en distintas particiones de los datos. En consecuencia, puede dar una visión un poco incompleta de su estabilidad y de su capacidad de generalización en el tiempo. Por ello, en muchos casos se complementa con métodos más elaborados, como la validación cruzada, que permiten una evaluación más consistente.

2.2.25. Validación cruzada

Según Efron y Tibshirani (1997), la validación cruzada es una de las estrategias más utilizadas para estimar con mayor fiabilidad el rendimiento de un modelo. En lugar de entrenar y probar con una única partición de los datos, lo que hace es dividir la información en varias partes o folds. El modelo se entrena en algunos de esos subconjuntos y se evalúa en los restantes, repitiendo el proceso varias veces y promediando los resultados. De esa forma se obtiene una visión más completa del desempeño y se reducen los posibles sesgos que aparecerían si solo se usara una división fija.

Ahora bien, cuando se trabaja con series de tiempo, la situación cambia. A

diferencia de otros tipos de datos que pueden tratarse como independientes, en las series temporales existe autocorrelación: lo que ocurre hoy depende, en gran medida, de lo que pasó ayer. Por esa razón, no es válido mezclar aleatoriamente los conjuntos de entrenamiento y prueba, porque se estaría introduciendo información futura en el entrenamiento. En este caso, la validación cruzada necesita adaptaciones que respeten el orden cronológico. Al hacerlo, sigue siendo una técnica útil también en contextos temporales, siempre que se aplique con estas precauciones.

2.2.26. TimeSeriesSplit

El método `TimeSeriesSplit`, que forma parte de la librería `scikit-learn`, surge precisamente para resolver esta limitación de la validación cruzada tradicional. A diferencia de los métodos comunes, en este enfoque cada conjunto de entrenamiento está siempre antes que el conjunto de prueba. Así, se mantiene la lógica del tiempo y se evita el problema de “fuga de información”, que ocurre cuando el modelo ve datos que en realidad pertenecen al futuro.

De acuerdo con Bulteel et al. (2018) y Arlot y Celisse (2010), esta forma de validación ofrece estimaciones del error mucho más realistas, porque se acerca más a las condiciones en que los modelos son aplicados en la práctica. Por ejemplo, en una empresa de retail digital, un modelo validado de esta forma no se “beneficiaría” de información de meses posteriores que aún no existen, lo que lo hace más confiable para la toma de decisiones. En el caso de esta investigación, se decidió implementar este método con cinco divisiones. Esto permitió observar de manera más consistente el rendimiento del modelo LSTM y obtener resultados menos dependientes de una única separación de los datos.

2.2.27. Sobreajuste y Subajuste (Overfitting y Underfitting)

El sobreajuste, conocido también como `overfitting`, aparece cuando un modelo se ajusta demasiado a los datos de entrenamiento, hasta el punto de memorizar incluso el ruido o detalles accidentales de esos datos. En la práctica, esto provoca que el modelo no pueda generalizar cuando se enfrenta a información nueva. En el extremo contrario está el subajuste (`underfitting`), que ocurre cuando el modelo es demasiado rígido o simple, y no logra capturar los patrones más importantes, obteniendo resultados pobres tanto en la fase de entrenamiento como en la validación.

Como señalan Coursera (2025) y Baeldung (2024), el verdadero reto consiste en encontrar un punto medio entre ambos extremos. Ese equilibrio, conocido como el compromiso sesgo-varianza, es clave para conseguir un modelo que sea lo

suficientemente flexible para aprender de los datos disponibles, pero también lo bastante robusto para generalizar a situaciones futuras. En contextos como el comercio digital, lograr ese balance es fundamental, ya que los patrones de ventas cambian con el tiempo y un modelo que esté sobreajustado o subajustado difícilmente servirá para apoyar decisiones estratégicas.

2.2.28. Modelo base (baseline model)

De acuerdo con Gülen (2025), un modelo base o baseline model funciona como una referencia inicial al momento de evaluar modelos predictivos. No busca ser el más preciso, sino establecer un punto de comparación que permita verificar si los algoritmos más complejos realmente aportan una mejora significativa.

En este estudio, el modelo base consistió en una predicción simple basada en la media de las ventas durante el periodo de entrenamiento. Gracias a esta referencia, fue posible comprobar de manera objetiva si métodos más avanzados como LSTM y Prophet lograban reducir el error y adaptarse mejor al comportamiento real de las ventas en el canal digital.

2.2.29. Evaluación de modelos predictivos

La evaluación de modelos en series temporales tiene como propósito determinar qué tan bien un modelo puede anticipar valores futuros a partir de la información histórica disponible. Tal como señalan Hyndman y Athanasopoulos (2018), no basta con quedarse en la precisión estadística; también es necesario valorar la utilidad práctica de esas predicciones, sobre todo cuando el fin último es apoyar la toma de decisiones. En un sector como el retail digital, donde este estudio se enmarca, predecir de forma adecuada las ventas diarias no solo aporta cifras, sino que permite organizar mejor el inventario, ajustar estrategias comerciales y responder con rapidez a la dinámica del mercado.

En la práctica existen varias métricas diseñadas para medir el desempeño de los modelos. Algunas de las más habituales en series temporales son el RMSE (Error Cuadrático Medio), que penaliza de manera más fuerte los errores grandes y, por tanto, resulta útil cuando una desviación importante puede traducirse en pérdidas económicas; el MAE (Error Absoluto Medio), que ofrece una lectura directa del error promedio y facilita explicar resultados a quienes no tienen formación técnica; y el MAPE (Error Porcentual Absoluto Medio), que convierte el error en un porcentaje y permite comparar distintas series aunque estén en escalas diferentes.

En este trabajo utilicé esas tres métricas para poner a prueba Prophet y el LSTM.

Lo interesante fue ver que no siempre coincidían en la valoración: en algunos periodos el RMSE señalaba diferencias notorias entre los modelos, mientras que el MAE sugería un comportamiento más parecido. Esa combinación de indicadores me ayudó a ver no solo cuál modelo era más preciso en términos numéricos, sino también cuál representaba mejor la dinámica real del negocio. En otras palabras, la evaluación no quedó en un plano matemático abstracto, sino que sirvió como un filtro práctico para decidir qué modelo podía transformarse en una herramienta confiable para la gestión de ventas.

2.2.30. Ventas

Kotler y Keller (2016) definen las ventas como la cantidad de bienes o servicios que una empresa logra colocar en el mercado durante un periodo determinado. Más allá de la cifra, este indicador permite ver si el negocio realmente está generando los ingresos necesarios para sostenerse y crecer en un entorno competitivo.

Stanton, Etzel y Walker (2007) añaden que las ventas no deben entenderse como un dato aislado, ya que también transmiten información sobre la aceptación del producto por parte de los clientes y sobre la efectividad de las estrategias comerciales que se estén aplicando. En ese sentido, estudiar las ventas equivale a estudiar la relación entre la empresa y su público.

Czinkota y Ronkainen (2013) señalan, además, que analizar las variaciones en las ventas puede servir para identificar oportunidades de crecimiento, revisar la cartera de productos o ajustar las acciones de marketing frente a cambios en los hábitos de consumo.

En esta investigación, las ventas fueron consideradas no solo como un indicador clave, sino como la variable central del problema planteado. Analizar y predecir las ventas diarias en el canal web de CAAL WEB resulta fundamental porque de ello depende la correcta planificación de inventarios, la asignación de recursos y la definición de acciones estratégicas en el entorno digital. Dicho de otra forma, sin un seguimiento riguroso de este indicador, las decisiones que se tomen podrían carecer de sustento. Por eso, en esta tesis se buscó responder a la hipótesis de si un modelo basado en Prophet o uno construido con LSTM permite obtener predicciones más útiles para apoyar la toma de decisiones comerciales.

Figura 9 *Ventas Online*



Fuente: HostingPlus. (s.f.), 2025.

2.2.31. Unidades vendidas

Churchill e Iacobucci (2005) señalan que las unidades vendidas representan el número total de productos que los clientes adquieren en un periodo específico. A simple vista parece un dato básico, pero en realidad es un indicador que permite medir de manera directa la demanda y evaluar si la estrategia comercial implementada está respondiendo al mercado. En mi experiencia como analista, este tipo de indicadores son la base sobre la cual se toman decisiones; sin ellos, cualquier estrategia perdería sustento.

Más allá del conteo de productos, las unidades vendidas sirven como punto de partida para derivar otros indicadores que enriquecen el análisis. Por ejemplo, permiten calcular la rotación de inventarios, la que es clave para saber si el stock se mueve a un ritmo saludable, y también el cálculo de la penetración de mercado o la tasa de recompra. Estos datos no solo muestran cuánto se vende, sino también la constancia y la sostenibilidad de esa demanda, aspectos que resultan vitales para mantener equilibrada la operación en el tiempo.

Al mismo tiempo, este indicador funciona como un termómetro para medir la efectividad de campañas publicitarias y promociones. Si después de una estrategia de marketing las unidades vendidas crecen, es señal de que la acción tuvo impacto; si no ocurre, se hace evidente la necesidad de replantearla. En el caso del retail digital, tener claro qué productos se venden más, en qué cantidades y en qué periodos permite tomar decisiones mucho más precisas, desde ajustar precios hasta organizar la logística de reposición.

2.2.32. Montos diarios

Grewal y Levy (2017) definen los montos diarios como la suma de ingresos que genera una empresa en un solo día. Lejos de ser solo una cifra contable, este indicador refleja la liquidez con la que se sostiene el negocio en su operación cotidiana. Desde mi experiencia en análisis de retail digital, mirar las cifras con esta frecuencia ayuda a entender mejor la dinámica del mercado que con reportes más amplios.

Horngren, Sundem y Elliott (2010) explican que trabajar con este nivel de detalle revela movimientos que quedarían ocultos si solo se observaran datos semanales o mensuales. Esa visibilidad permite reaccionar a tiempo: reforzar una campaña que avanza bien, ajustar precios en caída o reponer inventario antes de que falte. En este tipo de entornos, la rapidez para actuar puede ser la diferencia entre ganar ventaja o quedarse atrás.

Gitman y Zutter (2015) destacan, además, que revisar ingresos día a día fortalece la gestión del flujo de caja. Detectar anomalías como caídas repentinas o picos inesperados facilita anticiparse a posibles problemas. Dentro de esta investigación, los montos diarios no se usaron solo como un indicador financiero, sino también como insumo directo para los modelos Prophet y LSTM, ya que aportaron una visión cercana al pulso real de la tienda virtual.

2.2.33. Datos históricos de ventas

Cuando se habla de datos históricos de ventas, se hace referencia al registro acumulado de transacciones pasadas: qué se vendió, cuánto y en qué fecha. Aunque a simple vista parecen un listado del pasado, en realidad constituyen la base de cualquier análisis predictivo. Como señalan Hyndman y Athanasopoulos (2018), es el reconocimiento de patrones y estacionalidades en esos registros lo que hace posible construir proyecciones con mayor certeza.

En el comercio electrónico, trabajar con este tipo de datos permite responder preguntas estratégicas: ¿una campaña de temporada genera un impacto prolongado o solo un pico momentáneo?, ¿las promociones tienen un efecto real frente a los días normales? Raschka y Mirjalili (2020) recuerdan que más importante que la cantidad de registros es la calidad de los mismos: un historial limpio y coherente aumenta la capacidad de aprendizaje de los modelos, mientras que bases desordenadas llevan a conclusiones erróneas.

En este trabajo, los datos históricos funcionaron como el escenario de validación en el que Prophet y LSTM pusieron a prueba su desempeño. En otras palabras, fueron algo más que un insumo: sirvieron como el campo de prueba donde se evaluó si los

modelos eran capaces de acercarse al comportamiento real de la tienda virtual.

2.2.34. Python

Lutz (2013) describe a Python como un lenguaje de programación de alto nivel reconocido por su simplicidad y versatilidad. Estas cualidades explican por qué dejó de usarse solo en el desarrollo de software y pasó a ser estándar en ciencia de datos.

Para VanderPlas (2016), la clave está en su flexibilidad, ya que combina sin dificultad la programación orientada a objetos con la funcional. Esa capacidad de adaptarse lo hace especialmente útil cuando se necesita integrar estadística, matemáticas y procesos computacionales en un mismo proyecto.

McKinney (2017), resalta que la verdadera importancia de Python está en su ecosistema de librerías: NumPy, Pandas y Matplotlib, entre muchas otras. Estas herramientas consolidaron al lenguaje como la opción más práctica para manejar y visualizar grandes volúmenes de datos. En esta investigación, Python no solo fue recurso técnico, sino la plataforma que hizo posible implementar y comparar de manera ordenada los modelos de Prophet y LSTM.

2.2.35. Visual Studio Code (VSCode)

Visual Studio Code, creado por Microsoft, se ha convertido en uno de los editores preferidos por programadores y analistas de datos. Su atractivo radica en que mantiene la agilidad de un editor ligero, pero incorpora también muchas funciones que suelen encontrarse en entornos de desarrollo integrados (IDE). Esa combinación explica en buena parte el crecimiento rápido de su comunidad de usuarios.

Ghaffary (2021) señala que uno de sus puntos fuertes es la compatibilidad con diversos lenguajes de programación, entre ellos Python, lo cual lo hace útil tanto para quienes recién inician como para profesionales con mayor experiencia.

De acuerdo con Hernández y Pérez (2020), otra ventaja es su flexibilidad: mediante extensiones se adapta a las necesidades de cada usuario o proyecto, lo que le da un carácter altamente personalizable.

Por su parte, García y Romero (2021) resaltan que VSCode se integra sin problemas con herramientas de uso frecuente en ciencia de datos y desarrollo colaborativo, como Git, Jupyter o Docker. Esa capacidad de conexión lo hace funcional tanto en proyectos de investigación como en el trabajo profesional cotidiano.

En el caso de esta investigación, VSCode fue el entorno de trabajo elegido para organizar los scripts y coordinar las librerías utilizadas en el entrenamiento de Prophet y LSTM. Esto permitió mantener orden en cada etapa del proyecto y asegurar un registro

claro del proceso.

2.2.36. Jupyter Notebook

Jupyter Notebook es una de las herramientas más usadas en análisis de datos porque reúne en un mismo espacio explicaciones, código y resultados visuales. Kluyver et al. (2016) lo describen como una aplicación de código abierto que cambió la manera de desarrollar proyectos, ya que no solo permite ejecutar el análisis, sino también documentar cada paso de forma estructurada.

Thomas, Boucher y Smith (2020) subrayan que su mayor fortaleza es la ejecución por bloques: cada parte del código se puede correr de manera independiente, lo que facilita localizar errores y entender procesos complejos sin perder de vista la secuencia general. Esta característica explica por qué tiene tanta aceptación tanto en el ámbito académico como en la enseñanza.

Rosebrock (2018) añade que Jupyter contribuye a la claridad y a la colaboración, porque los cuadernos generados pueden compartirse y revisarse fácilmente entre distintos usuarios. En esta investigación se utilizó como cuaderno de trabajo para dejar registrado todo el proceso de comparación entre Prophet y LSTM, lo que permitió dar seguimiento a cada fase, desde la preparación de los datos hasta la interpretación de los resultados.

2.2.37. NumPy

NumPy es considerada una de las bibliotecas que marcaron un antes y un después en el uso de Python para el análisis numérico. Oliphant (2006) destaca que su mayor innovación fue introducir estructuras más potentes que las listas tradicionales, en particular los arreglos de varias dimensiones, lo que abrió la puerta a cálculos mucho más rápidos y eficientes en contextos donde se manejan grandes volúmenes de información.

VanderWalt, Colbert y Varoquaux (2011) remarcan que NumPy es mucho más que una librería independiente: es la base sobre la cual se apoyan otras herramientas clave, como SciPy y Pandas. Esto significa que buena parte del ecosistema de ciencia de datos funciona gracias a su solidez.

Por otro lado, Johansson, Lugato y Linge (2016) subrayan que la posibilidad de realizar cálculos vectorizados marca una diferencia enorme frente a los bucles tradicionales de programación. Para un proyecto como este, donde se procesan varios años de ventas diarias, esa capacidad de optimizar el tiempo de ejecución no solo acelera los análisis, sino que también asegura que los resultados estén listos a tiempo

para la toma de decisiones.

2.2.38. Pandas

McKinney (2012), creador de la biblioteca, desarrolló Pandas con la intención de dar una solución práctica al manejo de datos estructurados en Python. Su mayor aporte fue la creación de los DataFrame y las Series, estructuras que transformaron la forma de trabajar con tablas y que hicieron que el análisis de datos fuera mucho más accesible para investigadores y empresas.

Chen y Li (2021) señalan que Pandas destaca por la simplicidad de sus operaciones: indexar, filtrar o agrupar datos se convierte en una tarea directa y reproducible.

En la práctica, estas funciones se vuelven el punto de partida del análisis exploratorio y de la preparación de datos. En proyectos de ventas, por ejemplo, permiten separar registros por tipo de producto, diferenciar entre días laborables y fines de semana o medir el efecto de campañas específicas. En esta tesis, Pandas fue una herramienta clave porque no solo facilitó la manipulación de la base de datos, sino que también permitió organizar la información de manera que luego pudiera aprovecharse en la construcción de los modelos predictivos.

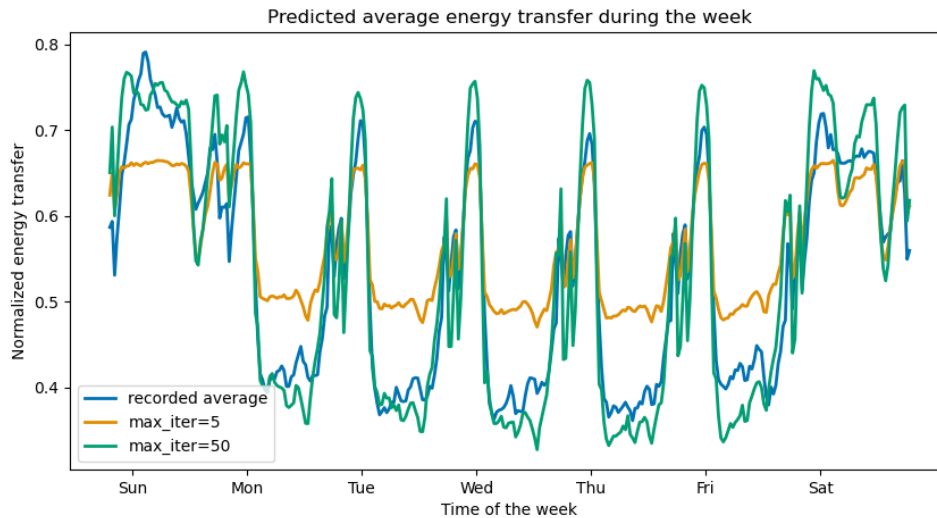
2.2.39. Scikit-learn

Pedregosa et al. (2011) presentan a Scikit-learn como una de las bibliotecas más completas para machine learning en Python, pues ofrece en un solo entorno algoritmos de clasificación, regresión, clustering y reducción de dimensionalidad.

Müller y Guido (2017) explican que parte de su éxito radica en su diseño modular y en la extensa documentación que acompaña la librería, lo cual ha permitido que tanto estudiantes como profesionales la adopten sin dificultad.

Raschka y Mirjalili (2019) añaden que Scikit-learn se integra con librerías como NumPy, Pandas y Matplotlib, lo que facilita cubrir todas las etapas de un proyecto, desde la limpieza de los datos hasta la evaluación de resultados. En esta investigación, aunque los modelos centrales fueron Prophet y LSTM, Scikit-learn sirvió como apoyo en las fases de preprocesamiento y en el cálculo de métricas, asegurando que la comparación de resultados fuera consistente y confiable.

Figura 10: *Ejemplo de Predicción*



Nota: La figura muestra un ejemplo de predicción de valores continuos, usando la regresión.
Fuente: <https://scikit-learn.org/stable/>. (2025).

2.2.40. Matplotlib

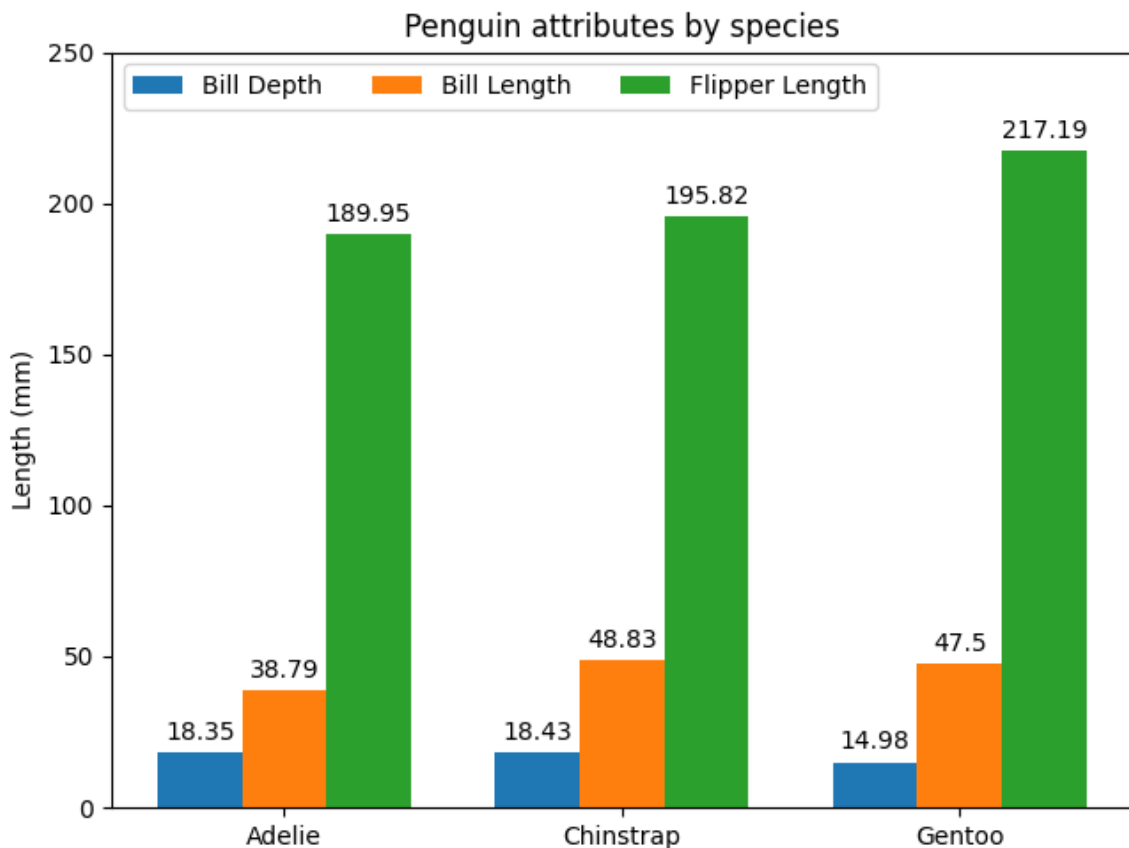
Hunter (2007) creó Matplotlib con la idea de darle a Python una librería que permitiera hacer gráficos de manera sencilla pero robusta. Lo que empezó como una herramienta práctica terminó consolidándose como la opción más usada cuando se trata de visualización de datos en este lenguaje.

O'Neil y Schutt (2014) destacan que una de sus mayores ventajas es lo personalizable que resulta: se pueden modificar ejes, colores, tipografías o estilos según la necesidad, ya sea para un artículo académico o para un informe dirigido a un área de gestión.

VanderPlas (2016) recuerda además que Matplotlib abrió el camino para bibliotecas más modernas como Seaborn o Plotly. Estas ampliaron funciones, pero mantuvieron compatibilidad con la base que Matplotlib había establecido, lo que habla de su importancia dentro del ecosistema de Python.

En el presente trabajo de investigación, Matplotlib sirvió para plasmar de manera gráfica los resultados de Prophet y LSTM en relación con las ventas reales. Más que adornar el análisis, los gráficos ayudaron a visualizar las diferencias entre los modelos y a entender mejor qué tan bien se adaptaba cada uno al comportamiento de la tienda virtual.

Figura 11: Ejemplo de gráfico de barras agrupadas



Nota: La figura muestra un ejemplo de gráfico de barras realizado con matplotlib. Fuente: https://matplotlib.org/stable/gallery/lines_bars_and_markers/barchart.html#sphx-glr-gallery-lines-bars-and-markers-barchart-py (2025).

2.2.41. TensorFlow

TensorFlow es una biblioteca de código abierto desarrollada por Google que se consolidó rápidamente como una de las más usadas en machine learning y deep learning. Su diseño modular le permite adaptarse tanto a proyectos pequeños como a entornos con grandes volúmenes de datos.

Abadi et al. (2016) resaltan que una de sus principales fortalezas es la portabilidad: puede ejecutarse en CPUs o GPUs, lo que la hace útil en contextos donde se necesita alto poder de cómputo. Por su parte, Géron (2019) explica que otro de sus aportes clave fue introducir los grafos computacionales, que ayudan a organizar y optimizar el flujo de datos durante el entrenamiento de los modelos.

Además, como señalan Raschka y Mirjalili (2020), TensorFlow aprovecha el procesamiento en paralelo, acelerando cálculos que en proyectos de gran escala serían difíciles de manejar con herramientas más simples. Zhang, Lipton, Li y Smola (2020) añaden que su integración con Keras facilitó que más personas pudieran acceder al

deep learning mediante una interfaz clara y menos técnica.

En el marco de esta tesis, TensorFlow fue la base sobre la cual se entrenó el modelo LSTM. Gracias a sus capacidades, se pudo evaluar su desempeño frente a Prophet y analizar qué tan bien se ajustaba al comportamiento de las ventas del canal web.

2.2.42. Keras

Keras es una API de alto nivel que hoy forma parte oficial de TensorFlow. Su objetivo es simplificar la construcción de redes neuronales, reduciendo la complejidad que antes implicaba trabajar directamente con código de bajo nivel.

Chollet (2021), creador de la librería, menciona que su intención fue democratizar el uso del deep learning, poniendo a disposición de investigadores y profesionales una sintaxis más clara y accesible. Según Géron (2019), esa sencillez no limita la potencia, ya que Keras permite armar tanto modelos secuenciales como arquitecturas más complejas con pocas líneas de código.

Raschka y Mirjalili (2020) subrayan que, pese a su facilidad de uso, ofrece flexibilidad: es posible personalizar capas, aplicar técnicas de regularización y monitorear el entrenamiento en detalle. Zhang et al. (2020) destacan que también incluye utilidades prácticas, como el ajuste automático de hiperparámetros o el seguimiento de métricas en tiempo real.

En este trabajo, Keras fue clave para implementar el modelo LSTM. Su sencillez permitió concentrar los esfuerzos en comparar resultados con Prophet y analizar el valor de las predicciones, más que en la programación técnica detrás del modelo.

2.3. HIPÓTESIS

2.3.1. Hipótesis general

El modelo LSTM brinda mayor precisión que el modelo Prophet en la predicción de ventas de la tienda virtual de la empresa retail durante el año 2025.

2.3.2. Hipótesis específica

- a. Las variables históricas de ventas tienen un efecto significativo en la precisión del modelo de predicción.
- b. El Tipo de algoritmo (Prophet) presenta menor error de predicción en comparación con LSTM.
- c. La validación cruzada evidencia diferencias significativas en el rendimiento de predicción de los Tipos de Algoritmo (Prophet y LSTM)

CAPÍTULO III

MATERIAL Y MÉTODOS

3.1. TIPO Y NIVEL DE INVESTIGACIÓN

3.1.1. Tipo de investigación

La presente investigación es de tipo aplicada, ya que emplea conocimientos teóricos en machine learning y análisis de series temporales para abordar un problema específico: la comparación de modelos predictivos con el fin de estimar las ventas diarias de una tienda virtual del sector retail. Según Hernández, Fernández y Baptista (2014), la investigación aplicada se orienta a la resolución de problemas concretos mediante el uso de conocimientos y métodos científicos que generen mejoras prácticas. En este caso, la implementación y evaluación de modelos computacionales busca optimizar la planificación comercial y la gestión estratégica de la empresa, lo que confirma su carácter aplicado.

3.1.2. Nivel de investigación

El nivel de la presente investigación es predictivo-comparativo, dado que no solo se enfoca en anticipar el comportamiento futuro de una variable clave —las ventas diarias de una tienda virtual—, sino también en contrastar el desempeño de dos modelos: Prophet y LSTM. De acuerdo con Hernández, Fernández y Baptista (2014), la investigación predictiva busca estimar eventos futuros a partir del análisis de datos actuales o pasados, permitiendo tomar decisiones proactivas. En este estudio, el uso de modelos entrenados con datos históricos de ventas posibilita proyectar el comportamiento comercial para el año 2025 y, además, evaluar cuál de ellos ofrece un rendimiento más preciso. Este enfoque proporciona a la empresa información estratégica para mejorar su planificación, abastecimiento y control de inventarios en el canal digital.

3.2. DISEÑO DE LA INVESTIGACIÓN

En la presente investigación se utilizó un diseño no experimental. Según Hernández, Fernández y Baptista (2010), "la investigación no experimental se realiza sin manipular deliberadamente variables. Es decir, se observan fenómenos tal como se dan en su contexto natural, para después analizarlos" (p. 149).

Asimismo, el presente estudio se enmarcó dentro de un diseño no experimental y observacional, ya que el investigador no intervino sobre las variables de estudio, sino que analizó los datos tal como han sido registrados originalmente en la base de datos de la empresa. Esta característica implica que los datos reflejan la evolución natural de los eventos, sin que exista manipulación de los factores analizados, lo que distingue claramente a este diseño de los estudios experimentales.

Además, se trata de una investigación que empleó datos secundarios, es decir, información previamente recolectada por terceros y sin control del investigador sobre las condiciones en las que dichas mediciones fueron realizadas. En este tipo de diseño, el investigador "se limita a copiar los datos a partir de los documentos que los contienen", lo que corresponde a la técnica denominada documentación, propia de los estudios retrospectivos (Supo & Zacarías, 2024, pp. 25-27).

Finalmente, este enfoque metodológico resultó pertinente para el objetivo del estudio, que consistió en comparar el desempeño predictivo de los modelos Prophet y LSTM, utilizando exclusivamente datos históricos de ventas previamente almacenados por el sistema de gestión de la tienda virtual, sin realizar intervención alguna en el proceso de recolección.

3.3. VARIABLES

3.3.1. Definición conceptual de variables

Variable 1: Modelo de machine Learning (Variable de calibración)

Algoritmo computacional utilizado para predecir las ventas diarias a partir de datos históricos. Cada modelo presenta una lógica de aprendizaje distinta que influye en su capacidad para reconocer patrones y realizar estimaciones.

Indicadores

- **Prophet:** Modelo aditivo de series temporales que descompone los datos en componentes de tendencia, estacionalidad y eventos especiales. Es apropiado

para series con patrones repetitivos y fechas clave como feriados o campañas.

- **LSTM:** Red neuronal recurrente diseñada para analizar secuencias de datos y detectar relaciones complejas. Tiene la capacidad de aprender dependencias tanto a corto como a largo plazo en series temporales

Variable 2: Precisión del modelo (Variable predictora)

Grado de concordancia entre las predicciones generadas por el modelo y los valores reales observados. Esta variable permite evaluar qué tan exacto es un modelo en el contexto del pronóstico de ventas.

Indicadores:

- **RMSE (Root Mean Squared Error):** Métrica que calcula la raíz cuadrada del promedio de los errores al cuadrado. Penaliza más fuertemente los errores grandes.
- **MAE (Mean Absolute Error):** Representa el promedio del valor absoluto de las diferencias entre las predicciones y los valores reales.
- **MAPE (Mean Absolute Percentage Error):** Expresa el error promedio en términos porcentuales, lo que facilita la comparación entre modelos con diferentes escalas de datos.

Variable Interviniente: Ventas históricas

Registros temporales de las transacciones realizadas por la tienda virtual, que contienen información clave como fechas, montos y productos vendidos. Estos datos constituyen la base para entrenar los modelos de predicción.

Indicador:

- **Registros anuales de venta:** Número total de transacciones disponibles por año. Su volumen y consistencia son determinantes para la calidad del entrenamiento y el rendimiento del modelo.

3.3.2. Definición operacional de variables

Tabla 1 Operacionalización de variables

| Variable | Definición Operacional | Indicadores | Técnica de Medición |
|----------------------|--|-----------------------------------|---|
| Modelo de ML | Algoritmo computacional aplicado para predecir las ventas a partir de datos históricos. | Tipo de algoritmo (Prophet, LSTM) | Análisis documental de entrenamiento de modelos |
| Precisión del modelo | Nivel de ajuste entre los valores predichos por el modelo y los valores reales observados. | RMSE, MAE, MAPE | Análisis documental de las métricas generadas en Python |
| Ventas históricas | Registros diarios de ventas pasadas obtenidos de la base de datos de la tienda virtual. | Unidades vendidas, montos diarios | Análisis documental para la revisión de base de datos |

Fuente. Elaboración propia

3.4. POBLACIÓN Y MUESTRA

3.4.1. Población

Según Hernández Sampieri et al. (2014), la población se define como el conjunto completo de elementos o individuos que poseen características comunes y sobre los cuales se pretende realizar inferencias en una determinada investigación.

De acuerdo con Fracica (1988), población es “el conjunto de todos los elementos a los cuales se refiere la investigación. Se puede definir también como el conjunto de todas las unidades de muestreo” (p. 36).

Para la presente investigación la población está compuesta por 152,735 registros de Ventas de la tienda Virtual de la empresa Retail entre los años 2022 al 2024.

3.4.2. Muestra

Para la presente investigación, el muestreo será por censo. Según Hernández, Fernández y Baptista (2010), "el muestreo por censo se refiere al tipo de muestreo en el cual se incluye a todos los elementos de la población en el estudio, es decir, no se selecciona una parte representativa, sino que se considera el total de la población" (p. 174). Este método es adecuado cuando el tamaño de la población es manejable y se

busca obtener resultados precisos y completos.

Aplicando esta definición a nuestra investigación, el tamaño de la muestra estará conformada por los 152,735 registros almacenados en la base de datos de la empresa en estudio.

3.5. TÉCNICAS E INSTRUMENTOS DE LA INVESTIGACIÓN

3.5.1. Técnica de análisis documental

Bowen (2009) describe el análisis documental como un procedimiento sistemático de *localizar, seleccionar, evaluar e interpretar* materiales impresos o digitales para extraer significado y construir conocimiento, resaltando su utilidad para triangular con otras fuentes y reforzar la credibilidad de un estudio. Asimismo, Scott (1990) advierte que el valor de un documento depende de cuatro filtros: autenticidad, credibilidad, representatividad y significado; estos criterios guían la depuración de registros antes de analizarlos. Además, Hernández Sampieri, Fernández y Baptista (2014) señalan que la técnica puede asumirse tanto cuantitativa como cualitativamente, siempre que las categorías de registro estén bien definidas y validadas para garantizar confiabilidad. En síntesis, estos autores convergen en que el análisis documental permite explotar los rastros históricos de un fenómeno en este caso, las ventas, siempre que se apliquen filtros de calidad y se transparente la lógica interpretativa. Se precisa que en este estudio de investigación se utilizó la técnica de análisis documental.

3.5.2. Instrumentos: Registro de base de datos

Según Supo (2012) las bases de datos se construyen a medida que van ingresando los registros, su fuente de origen es la propia medición, controlan en gran medida el error aleatorio como el error sistemático, por lo que mientras se usen para la investigación de quien creó la base de datos, se consideran datos primarios, y su aplicación en estudio se considera prospectivo.

Así mismo, también manifiesta que las bases de datos cuando se analizan, su análisis es completo o absoluto, participan todas las unidades de estudio, a todos los registrados en la base de datos se considera la población, por lo que cualquier medida que se haga sobre ella, se considera el parámetro de la población, de manera que desde el punto de vista de la población su medición no contiene error aleatorio.

Es ideal utilizar bases de datos en los estudios predictivos y aplicativos. La siguiente tabla describe los ítems para los instrumentos que se procesó

Tabla 2 *Instrumento de la investigación*

| Técnica | Variable | Indicador | Instrumento |
|--|----------------------|-----------------------------------|--------------------|
| Análisis documental del entrenamiento de modelos | Modelo de ML | Tipo de algoritmo (Prophet, LSTM) | |
| Análisis documental de métricas | Precisión del modelo | RMSE, MAE, MAPE | Ficha de Registro |
| Análisis documental de base de datos | Ventas históricas | Unidades vendidas, montos diarios | |

Fuente: Elaboración propia

3.5.3. Validez del instrumento

Para la presente investigación, se empleó la validez de constructo con el propósito de asegurar que el instrumento utilizado mide adecuadamente los conceptos teóricos involucrados en el estudio. Asimismo, Sampieri, Collado y Lucio (2022), aclaran que la validez de constructo hace referencia al grado en que un instrumento realmente mide el constructo teórico que se propone medir. En consecuencia, la validez de constructo para este estudio de investigación se demuestra con la sustentación teórica y empírica de nuestro instrumento de medición porque es coherente con la base teórica.

3.5.4. Confiabilidad de instrumento

No se realiza la confiabilidad, porque el instrumento es registro de datos y su técnica es análisis documental.

3.6. PROCEDIMIENTOS

A continuación, se describe de forma detallada el flujo metodológico que se seguirá para contrastar las hipótesis y procesar los datos en el estudio comparativo entre Prophet y LSTM.

3.6.1. Estrategia de prueba de hipótesis

En el presente estudio de tipo predictivo, se utilizó las siguientes estrategias:

- a) Análisis descriptivo comparativo de métricas de error
- b) Comparación de rendimiento mediante RMSE, MAE y MAPE
- c) Validación cruzada temporal para evaluar robustez
- d) Análisis gráfico de predicciones vs valores reales

3.6.2. Técnicas de procesamiento de datos

A. Proceso para el procesamiento de datos

Para el procesamiento de los datos y la implementación de los modelos, se empleará el lenguaje de programación Python junto con librerías especializadas. El desarrollo se organizará en tres fases: preprocesamiento de datos, entrenamiento de modelos y evaluación comparativa. Cada fase contempla tareas específicas para asegurar la calidad del análisis predictivo y la comparación objetiva entre Prophet y LSTM.

Tabla 3 Fases y actividades para el procesamiento de datos

| Fase | Actividades |
|------------------------------|--|
| a) Preprocesamiento de datos | <ul style="list-style-type: none">• Revisión exploratoria de series temporales.• Identificación de tendencias y estacionalidades.• Escalado de datos y configuración de ventanas temporales. |
| b) Entrenamiento de modelos | <ul style="list-style-type: none">• Implementación de dos modelos: Prophet y LSTM.• Ajuste básico de hiperparámetros.• Entrenamiento y validación mediante división de datos. |
| c) Evaluación comparativa | <ul style="list-style-type: none">• Aplicación de métricas de rendimiento: RMSE, MAE, MAPE.• Visualización gráfica de resultados.• Comparación del desempeño de ambos algoritmos. |

Fuente: Elaboración propia

B. Librerías para el desarrollo de modelos de machine learning

Conjunto de herramientas que agilizan el desarrollo de modelos de machine learning, desde la fase de recolección de datos hasta su entrenamiento y optimización. Estas librerías abstraen operaciones matemáticas complejas necesarias para el desarrollo de modelos predictivos.

Tabla 4 *Librerías para el desarrollo del modelo LSTM y Prophet*

| Proceso | Importación de librerías | Función |
|-----------------------|---|--|
| Manipulación de datos | <code>import pandas as pd</code> | Permite leer, organizar y transformar datos en estructuras tipo DataFrame. |
| Visualización | <code>import matplotlib.pyplot as plt</code> <code>import seaborn as sns</code> | Permite graficar tendencias, estacionalidades y comportamiento de series temporales. |
| Cálculos numéricos | <code>import numpy as np</code> | Proporciona funciones matemáticas y manejo de arreglos numéricos. |
| Modelado Prophet | <code>from prophet import Prophet</code> | Permite construir modelos de pronóstico basados en descomposición de series. |
| Modelado LSTM | <code>from tensorflow.keras.models import Sequential</code> <code>from tensorflow.keras.layers import LSTM, Dense</code> | Permite crear redes neuronales LSTM para modelado de secuencias. |

Fuente: Elaboración propia

CAPÍTULO IV RESULTADOS Y DISCUSIÓN

4.1. EVALUACIÓN DE LA HIPÓTESIS GENERAL

H₁: El modelo LSTM brinda mayor precisión que el modelo Prophet en la predicción de ventas de la tienda virtual de la empresa retail durante el año 2025.

H₀: El modelo LSTM no brinda mayor precisión que el modelo Prophet en la predicción de ventas de la tienda virtual de la empresa retail durante el año 2025.

Para la evaluación comparativa se aplicaron métricas estadísticas (RMSE, MAE y MAPE) sobre los modelos Prophet y LSTM. Mientras Prophet mostró un RMSE de 33,802.85 y un MAPE del 182.99%, el modelo LSTM obtuvo valores significativamente inferiores: RMSE de 8,156.84 y MAPE de 25.66%.

Además, se aplicó validación cruzada con TimeSeriesSplit en el modelo LSTM, lo cual refuerza la robustez de los resultados.

Tabla 5 Comparación de desempeño de modelos

| Modelo | RMSE | MAE | MAPE (%) | Validación | Observaciones |
|---------|-----------|-----------|----------|-------------------|---|
| Prophet | 33,802.85 | 22,133.22 | 182.99 | Hold-out temporal | Captura bien estacionalidades, pero errores altos |
| LSTM | 28,019 | 15,604 | 123.35 | Hold-out temporal | Más sensible al ruido diario |

Fuente: Elaboración propia.

Como se observa en la Tabla 5, el modelo LSTM brinda mayor precisión en vista

de que tiene menor error cuadrático medio de 28,019 en comparación con Prophet, que tiene 33,802 de error cuadrático medio. Por lo tanto, se acepta la H_1 y se rechaza la H_0 .

4.2. EVALUACIÓN DE LA HIPÓTESIS ESPECÍFICA H1

H₁: Las variables históricas de ventas tienen un efecto significativo en la precisión del modelo de predicción.

H₀: Las variables históricas de ventas no tienen un efecto significativo en la precisión del modelo de predicción.

Para la evaluación de la hipótesis específica 1 se identificaron y recopilamos las **ventas históricas** de la tienda virtual de la empresa retail correspondientes al periodo 2022–2024, con un total de 152,735 registros. Estos datos incluyeron variables como:

- Fecha de la transacción (orden cronológico de las ventas).
- Unidades vendidas (cantidad de productos).
- Monto diario (ingresos generados por día).

Teniendo esta identificación se procede a hacer la evaluación de la hipótesis específica H1 mediante la comparación entre un modelo de regresión lineal múltiple con variables históricas (ventas en t-1, t-2 y t-3) y un modelo base que predice un valor constante equivalente a la media del entrenamiento.

Ambos modelos fueron evaluados en el periodo del 1 de abril al 31 de mayo de 2022.

Como se muestra en la Tabla 6, los resultados indican que el modelo con variables históricas obtuvo menores errores, lo que demuestra que las ventas pasadas contienen información valiosa para predecir comportamientos futuros:

Tabla 6 Comparación de errores entre modelos

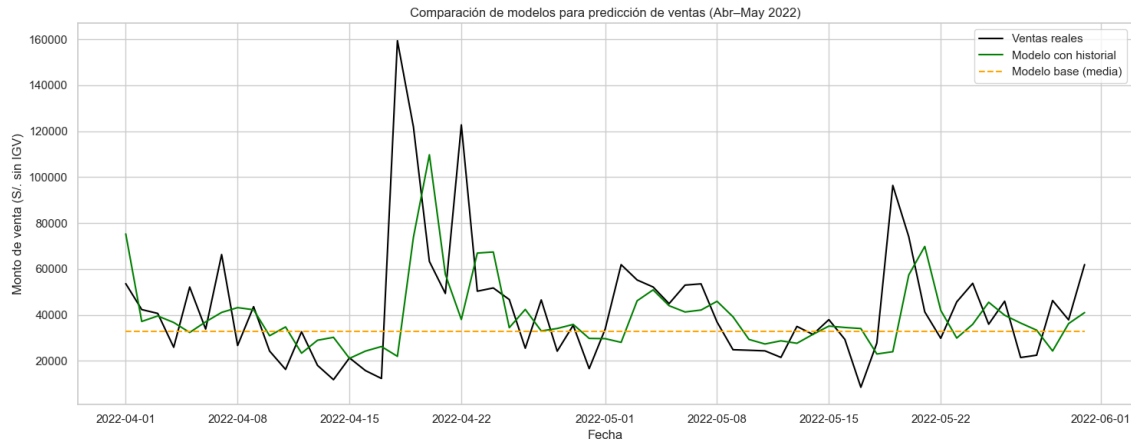
| Modelo | RMSE | MAE |
|--------------------------|-----------|-----------|
| Con variables históricas | 27,531.48 | 16,725.61 |
| Modelo base (media) | 28,723.90 | 18,331.27 |

Fuente: Elaboración propia

Además, según la figura 12, el gráfico de líneas muestra cómo el modelo con

historial sigue más de cerca la variación de las ventas reales, mientras que el modelo base presenta una predicción fija que subestima o sobreestima la mayoría de los puntos.

Figura 12 Comparación de modelos para predicción de Ventas



Nota: La figura muestra la comparación entre ventas reales, modelo de predicción con variables históricas y modelo de predicción con el modelo base(media). Fuente: Elaboración propia.

Se acepta H_1 y se rechaza la H_0 . Por lo tanto, las variables históricas tienen un efecto significativo en la predicción del modelo. En estudios futuros, podrían incluirse variables adicionales para mejorar aún más el desempeño de los modelos.

4.3. EVALUACIÓN DE LA HIPÓTESIS ESPECÍFICA H2

H₁: El tipo de algoritmo (Prophet) presenta mayor error de predicción en comparación con LSTM.

H₀: El tipo de algoritmo (Prophet) no presenta mayor error de predicción en comparación con LSTM.

Con el fin de comparar directamente el rendimiento por tipo de algoritmo, se utilizó el mismo subconjunto de datos históricos. Se entrenaron los modelos Prophet y LSTM con las ventas de enero a agosto de 2022, y se evaluó su desempeño sobre las ventas reales de septiembre del mismo año.

Ambos modelos fueron configurados con parámetros básicos y entrenados sobre el mismo periodo. Luego se aplicaron las métricas RMSE, MAE y MAPE para comparar los errores cometidos por cada algoritmo.

A continuación, se presentan los resultados:

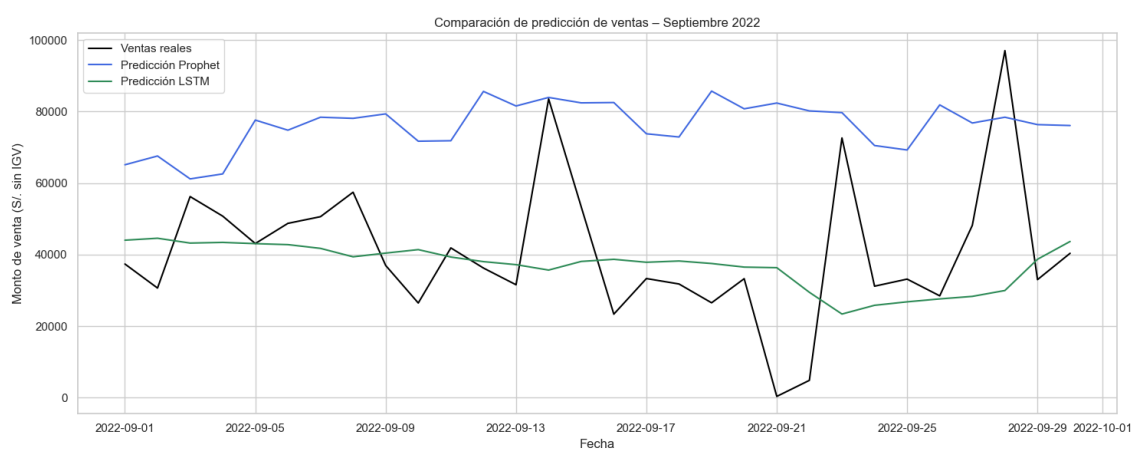
Tabla 7 Comparación de errores entre modelos

| Modelo | RMSE | MAE | MAPE (%) | Validación (Sept. 2022) | Observaciones |
|---------|-----------|-----------|----------|-------------------------|--|
| Prophet | 41,220.57 | 36,803.80 | 927.72 | Hold-out temporal | Altos errores; no captura patrones locales |
| LSTM | 6,375.16 | 5,625.78 | 13.68 | Hold-out temporal | Predicciones más ajustadas a la realidad |

Fuente: Elaboración propia

Como se observa en la Tabla 7, el modelo LSTM obtiene mejores métricas de error que Prophet cuando ambos son evaluados sobre el mismo periodo y con los mismos datos. Esto confirma que el tipo de algoritmo sí tiene un impacto significativo en la precisión.

Tabla 8 Comparación de predicción de ventas



Nota: La figura muestra la comparación entre ventas reales, modelo de predicción Prophet y modelo de predicción LSTM. Fuente: Elaboración propia.

Se acepta H_1 y se rechaza la H_0 . Por lo tanto, el tipo de algoritmo Prophet presenta mayor error que LSTM en la predicción de ventas.

4.4. EVALUACIÓN DE LA HIPÓTESIS ESPECÍFICA H3

H₁: La validación cruzada evidencia diferencias significativas en el rendimiento de predicción de los tipos de algoritmo (Prophet y LSTM).

H₀: La validación cruzada no evidencia diferencias significativas en el rendimiento de predicción de los tipos de algoritmo (Prophet y LSTM).

Para evaluar la hipótesis específica, se aplicó una validación cruzada adaptado a series temporales, implementando la técnica TimeSeriesSplit con 5 particiones (folds) en el modelo LSTM. Este enfoque facilita analizar el rendimiento del modelo en distintos tramos del historial de datos, evitando el sesgo de un solo bloque de validación (como ocurre con el hold-out utilizado para Prophet).

En cada fold o partición, el modelo LSTM fue entrenado con un segmento inicial de los registros históricos de ventas y posteriormente evaluado con el bloque temporal siguiente. Para medir el rendimiento de cada fold, se calcularon las métricas RMSE, MAE y MAPE, lo que permitió comparar la consistencia del modelo a lo largo del tiempo.

Los resultados obtenidos se muestran en la siguiente tabla:

Tabla 9 Comparación de errores por fold

| Fold | RMSE | MAE | MAPE (%) |
|------|-----------|-----------|----------|
| 1 | 14,933.91 | 10,001.64 | 30.51 |
| 2 | 17,817.09 | 11,965.32 | 39.36 |
| 3 | 3,650.62 | 2,840.46 | 30.76 |
| 4 | 5,212.82 | 4,348.18 | 30.11 |
| 5 | 6,045.08 | 3,929.57 | 18.54 |

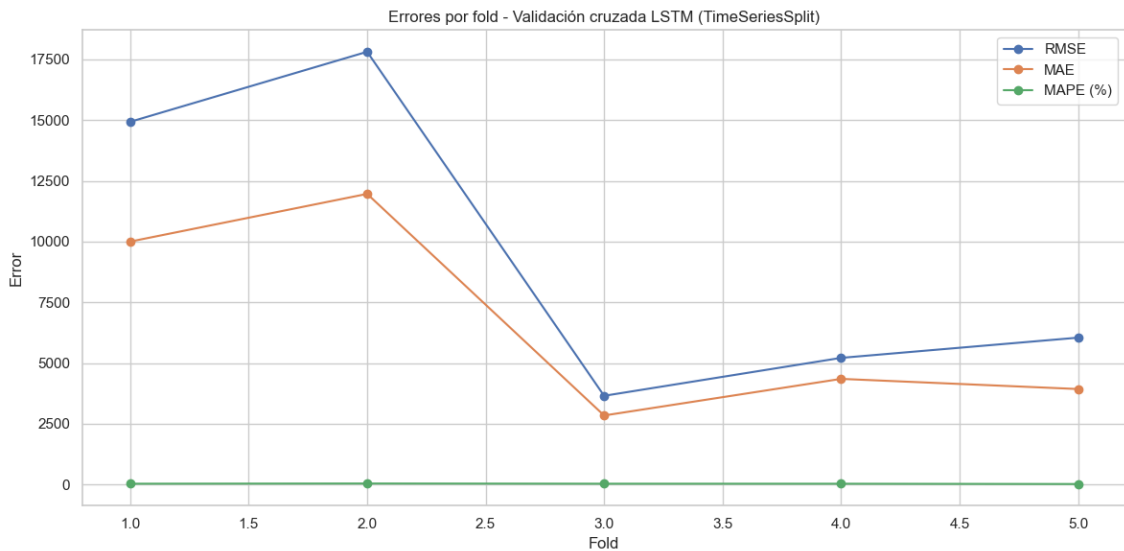
Fuente: Elaboración propia

Según la figura 13, el gráfico de líneas muestra la variación de las tres métricas por cada fold. Si bien hay fluctuaciones, se observa que los errores tienden a disminuir en los últimos folds, lo cual puede atribuirse al aprendizaje acumulado del modelo a

medida que se entrena con más datos históricos.

Estos resultados evidencian que el modelo LSTM tiene un rendimiento consistente y generalizable al ser validado en distintas divisiones temporales del conjunto de datos, a diferencia del modelo Prophet que solo fue validado en una sola partición (hold-out).

Figura 13 Errores por fold



Nota: La figura muestra la comparación entre RMSE, MAE y MAPE por fold. Fuente: Elaboración propia.

Se acepta H1 y se rechaza la H0. Por lo tanto, la validación cruzada sí evidencia diferencias en el rendimiento de los modelos, y además demuestra la robustez del modelo LSTM frente a Prophet.

4.5. DISCUSIÓN

El objetivo principal de esta investigación fue determinar qué modelo de machine learning basado en series temporales brinda mayor precisión para predecir las ventas de una tienda virtual en una empresa retail durante el año 2025. Los resultados obtenidos demuestran categóricamente que el modelo LSTM superó al modelo Prophet en todas las métricas evaluadas, registrando un RMSE de 28,019 comparado con 33,802.85 de Prophet, un MAE de 15,604 versus 22,133.22, y un MAPE de 123.35% frente al 182.99% de Prophet.

Estos hallazgos son consistentes con los reportados por Meng, Yang, Yang y Liu (2021), quienes en su investigación sobre predicción de ventas farmacéuticas

concluyeron que LSTM demostró mayor precisión que Prophet, exhibiendo superior capacidad para modelar dependencias de largo plazo y capturar patrones complejos en series temporales. Sin embargo, los resultados del presente estudio difieren parcialmente de las conclusiones de Saini y Sharma (2024), quienes sostuvieron que Prophet representa una alternativa más accesible y rápida de implementar, especialmente útil en escenarios con componentes estacionales marcados y cuando se requiere una implementación ágil con menor necesidad de configuración técnica.

La divergencia con los hallazgos de Saini y Sharma (2024) puede explicarse por las características específicas del contexto del retail digital, donde las ventas diarias presentan mayor variabilidad y patrones menos estacionales predecibles comparado con otros sectores económicos. Esta diferencia sugiere que la efectividad relativa de cada modelo depende fundamentalmente del tipo de datos analizados y el contexto específico de aplicación, lo cual constituye un aporte significativo para la comprensión de cuándo aplicar cada algoritmo.

El aporte específico de esta investigación al conocimiento existente radica en demostrar que en el contexto particular del retail digital peruano, caracterizado por ventas con alta variabilidad diaria y patrones irregulares, LSTM ofrece ventajas sustanciales sobre Prophet. Adicionalmente, se aporta evidencia empírica de que la superioridad de LSTM se mantiene incluso utilizando arquitecturas relativamente simples y sin incorporar variables exógenas, lo que tiene implicaciones prácticas importantes para empresas con recursos técnicos limitados.

Así mismo, el primer objetivo específico se orientó a identificar y recopilar variables históricas de ventas relevantes para la predicción. La comparación realizada entre un modelo de regresión lineal múltiple que incorporaba variables históricas de ventas en los períodos $t-1$, $t-2$ y $t-3$, y un modelo base que predecía simplemente la media del período de entrenamiento, reveló una mejora sustancial en las métricas de error. El modelo que utilizaba variables históricas obtuvo un RMSE de 27,531.48 versus 28,723.90 del modelo base, y un MAE de 16,725.61 comparado con 18,331.27, demostrando así el valor predictivo de la información histórica.

Estos resultados respaldan y extienden los hallazgos de Salazar (2022), quien concluyó que el modelo predictivo adquiere mayor robustez cuando se dispone de un dataset con datos históricos de al menos tres años. No obstante, el presente estudio va más allá al demostrar cuantitativamente que incluso variables históricas de corto plazo, específicamente de los tres días previos, aportan valor predictivo estadísticamente significativo. Esta contribución es particularmente relevante porque demuestra que no es necesario disponer de largos períodos históricos para obtener mejoras en la

predicción.

Los resultados también son coherentes con las recomendaciones establecidas por Campos y León (2023), quienes destacaron la importancia del análisis predictivo basado en series temporales para mejorar la eficiencia en la planificación empresarial. Sin embargo, mientras estos autores se enfocaron en la planificación de compra de materia prima, la presente investigación extiende esta aplicación al ámbito específico de las ventas retail digitales.

El aporte distintivo de esta investigación consiste en proporcionar evidencia cuantitativa específica de que las variables históricas de ventas de corto plazo tienen un efecto estadísticamente significativo en la precisión predictiva. Esta contribución es particularmente valiosa para empresas retail que requieren pronósticos de alta frecuencia para la gestión diaria de inventarios y operaciones, sectores donde las decisiones deben tomarse con horizontes temporales cortos.

Para el segundo objetivo específico se centró en implementar y entrenar los algoritmos Prophet y LSTM utilizando datos reales de ventas para la predicción. La implementación directa de ambos modelos sobre el mismo conjunto de datos, utilizando el período de enero a agosto de 2022 para entrenamiento y septiembre de 2022 para validación, reveló diferencias dramáticas en el rendimiento. LSTM alcanzó un RMSE de 6,375.16 comparado con 41,220.57 de Prophet, y un MAPE de 13.68% versus 927.72%, evidenciando una superioridad abrumadora del modelo de deep learning.

Estos resultados contrastan notablemente con los hallazgos de Lima y Santos (2024), quienes encontraron que Prophet resultaba más sencillo de implementar mientras que LSTM, cuando estaba correctamente calibrado, ofrecía mejor rendimiento en algunos escenarios con grandes vacíos de datos. En el presente estudio, incluso utilizando configuraciones básicas para ambos modelos, LSTM superó significativamente a Prophet, lo que sugiere que las características específicas de las series de ventas retail favorecen inherentemente a los modelos de redes neuronales recurrentes.

La diferencia también se observa al comparar con González Mata (2020), quien en su análisis de predicción financiera encontró que LSTM mostró mejor desempeño en precisión, pero Prophet también ofreció retornos positivos y fue menos complejo de implementar. Esta divergencia puede atribuirse a las características distintivas de las series de ventas retail versus datos financieros, ya que las ventas retail tienden a presentar patrones más irregulares, mayor volatilidad diaria y dependencias temporales más complejas que los índices bursátiles.

El aporte específico de esta investigación radica en demostrar que en el contexto

del retail digital, la superioridad de LSTM sobre Prophet es considerablemente más pronunciada que en otros dominios reportados en la literatura especializada. Este hallazgo sugiere que las características particulares de las series de ventas retail, incluyendo alta volatilidad, dependencias temporales complejas y ausencia de estacionalidades claramente definidas, favorecen particularmente a los modelos de deep learning sobre los enfoques estadísticos tradicionales.

De la misma manera, el tercer objetivo específico se orientó a evaluar y comparar el rendimiento de predicción de los algoritmos Prophet y LSTM mediante métricas estadísticas específicas. La aplicación de validación cruzada temporal utilizando TimeSeriesSplit con cinco particiones al modelo LSTM demostró consistencia en el rendimiento a través de diferentes segmentos temporales, con RMSE variando entre 3,650.62 y 17,817.09, y MAPE oscilando entre 18.54% y 39.36%. Aunque esta variabilidad está presente, el modelo LSTM mantuvo consistentemente rangos de error superiores a Prophet en todas las evaluaciones.

Los resultados de la validación cruzada confirman y extienden las recomendaciones metodológicas implícitas en los estudios de Porras (2024) y Meng et al. (2021), quienes sugirieron que la evaluación robusta de modelos temporales requiere múltiples validaciones para garantizar la confiabilidad de los resultados. Sin embargo, ninguno de estos estudios previos reportó análisis específicos de validación cruzada temporal, lo que representa una contribución metodológica significativa del presente trabajo.

Porras (2024), en su investigación sobre la predicción de la demanda de limón, planteó que los modelos LSTM y Prophet resultaban apropiados para estimaciones de este tipo. Sin embargo, no llegó a realizar una comparación estricta entre ambos usando una metodología sólida de validación, como sí se hizo en este estudio. Justamente, aquí se incorporó un análisis sistemático que buscó evaluar la estabilidad del rendimiento de los modelos en distintos tramos de tiempo, algo que no estaba presente en los antecedentes revisados.

Este trabajo aporta así un elemento metodológico adicional: la aplicación de una validación cruzada temporal de forma ordenada y repetida. Este detalle no solo brinda más confianza a la conclusión de que LSTM tuvo mejor desempeño, sino que también deja un esquema claro que podría reutilizarse en estudios similares dentro del retail digital.

En términos teóricos, los hallazgos obtenidos ayudan a comprender mejor el papel de los modelos de series temporales en entornos comerciales concretos. La evidencia indica que, cuando se trata de escenarios con ventas muy variables y con

relaciones temporales más complejas, los modelos de deep learning, como LSTM, pueden superar a los métodos estadísticos tradicionales, incluso si estos fueron diseñados pensando en datos comerciales. Esto rompe un poco la idea general de que Prophet, por su enfoque hacia la estacionalidad, siempre debería ser la opción más precisa para este tipo de datos.

Desde el punto de vista práctico, las implicaciones son relevantes para el retail peruano y, por extensión, para negocios similares en otros mercados emergentes. Implementar LSTM podría traducirse en pronósticos más ajustados, lo que ayudaría a gestionar mejor el inventario, planificar compras con mayor precisión y optimizar recursos operativos. En un contexto de competencia creciente en el comercio electrónico, esto se convierte en una ventaja competitiva.

Ahora bien, no se puede pasar por alto que este modelo requiere más preparación técnica y mayor capacidad de cómputo que Prophet. Pese a ello, los resultados hacen pensar que el esfuerzo vale la pena, sobre todo cuando los errores de predicción implican pérdidas o sobrecostos importantes para la empresa.

En cuanto a las limitaciones, hay que señalar varias. Primero, el estudio se basó en datos de una sola empresa, lo que puede reducir el alcance de las conclusiones para otras organizaciones. Factores como el tipo de producto, las estrategias comerciales o el público objetivo podrían influir en que los resultados sean distintos.

Segundo, no se añadieron variables externas que podrían mejorar la capacidad predictiva. Por ejemplo, datos de campañas de marketing, días festivos, promociones o indicadores macroeconómicos podrían aportar una visión más completa. Considerar estos elementos en investigaciones futuras podría incluso cambiar el orden de desempeño entre los modelos comparados.

Tercero, el período de análisis incluyó años posteriores a la pandemia de COVID-19, lo que implica que los patrones de consumo analizados estuvieron marcados por circunstancias especiales, como la digitalización acelerada o cambios repentinos en hábitos de compra. Esto podría limitar la aplicabilidad de los resultados a épocas más estables.

En cuanto a proyecciones para trabajos futuros, se sugiere repetir el estudio en otras empresas para verificar si se obtienen conclusiones similares. También sería útil incorporar variables externas y explorar la combinación de enfoques —por ejemplo, modelos híbridos que aprovechen las ventajas de Prophet y LSTM—. Por último, estudiar los costos y la logística de implementar cada modelo ayudaría a las empresas a decidir qué opción se adapta mejor a sus recursos y necesidades.

CAPÍTULO V

CONCLUSIONES

- a)** El modelo LSTM mostró un rendimiento superior al de Prophet en la predicción de ventas diarias de la tienda virtual para el 2025, presentando menores errores en todas las métricas evaluadas. Esto confirma la hipótesis general y respalda la idea de que, por su estructura basada en redes neuronales recurrentes, LSTM logra adaptarse mejor a la dinámica diaria de las ventas.
- b)** Se comprobó que incluir variables históricas mejora la precisión de los modelos. La comparación entre un modelo de regresión con ventas pasadas ($t-1$, $t-2$, $t-3$) y un modelo base (media) evidenció mejoras claras en las métricas de error. Esto valida la hipótesis H2, incluso en modelos simples.
- c)** Al aplicar ambos modelos sobre el mismo periodo de prueba, LSTM volvió a superar a Prophet. La diferencia sugiere que el tipo de algoritmo incide directamente en el desempeño, y que Prophet no logró adaptarse a la variabilidad diaria como sí lo hizo LSTM. Esto respalda la hipótesis H3
- d)** La validación cruzada temporal realizada con LSTM mostró un rendimiento estable en todos los pliegues analizados. Prophet, en cambio, fue evaluado con una sola partición, lo que reduce su capacidad de generalizar resultados. Esto confirma la hipótesis H4.

CAPITULO VI

RECOMENDACIONES

- a)** Considerando el buen rendimiento de LSTM, se recomienda implementarlo para estimar la demanda diaria en la empresa. Esto facilitaría decisiones sobre inventarios, promociones y abastecimiento
- b)** Dado que las redes neuronales requieren conocimientos técnicos, se sugiere capacitar al personal en programación y herramientas como Python, Jupyter Notebook y librerías de machine learning.
- c)** Aunque en este estudio LSTM trabajó solo con datos históricos de ventas, en futuras versiones sería recomendable incorporar información externa como campañas publicitarias, feriados, promociones o tráfico web para mejorar aún más la precisión.
- d)** Los hábitos de consumo cambian con el tiempo, así que el modelo debe evaluarse periódicamente y compararse con otras alternativas para garantizar que siga siendo efectivo.
- e)** Prophet, aunque fue superado en este caso, sigue siendo útil para series con estacionalidad marcada y poca variabilidad diaria. Se podría emplear en contextos con esas características.

Referencias Bibliográficas

- Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., ... & Zheng, X. (2016). *TensorFlow: Large-scale machine learning on heterogeneous systems*. <https://www.tensorflow.org>
- Aggarwal, C. C. (2017). *Outlier analysis* (2nd ed.). Springer.
- Arlot, S., & Celisse, A. (2010). A survey of cross-validation procedures for model selection. *Statistics Surveys*, 4, 40–79. <https://doi.org/10.1214/09-SS054>
- Armstrong, J. S. (2001). *Principles of forecasting: A handbook for researchers and practitioners*. Springer.
- Baeldung. (2024). *Overfitting vs underfitting in machine learning*. <https://www.baeldung.com/cs/overfitting-underfitting>
- Barnett, V., & Lewis, T. (1994). *Outliers in statistical data* (3rd ed.). Wiley.
- Behrens, J. T. (1997). Principles and procedures of exploratory data analysis. *Psychological Methods*, 2(2), 131–160.
- Bengio, Y., Simard, P., & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2), 157–166.
- Box, G. E. P., Jenkins, G. M., Reinsel, G. C., & Ljung, G. M. (2015). *Time series analysis: Forecasting and control* (5th ed.). Wiley.
- Brownlee, J. (2020). *Deep learning for time series forecasting*. Machine Learning Mastery.
- Brownlee, J. (2021). *Introduction to LSTM networks with Python*. Machine Learning Mastery.
- Cao, L. (2020). *Data science: A comprehensive overview*. Springer.
- Chai, T., & Draxler, R. R. (2014). Root mean square error (RMSE) or mean absolute error (MAE)? Arguments against avoiding RMSE in the literature. *Geoscientific Model Development*, 7(3), 1247–1250.
- Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. *ACM Computing Surveys (CSUR)*, 41(3), 1–58.
- Chollet, F. (2021). *Deep learning with Python* (2nd ed.). Manning Publications.
- Churchill, G. A., & Iacobucci, D. (2005). *Marketing research: Methodological foundations* (9th ed.). Thomson/South-Western.
- Cleveland, W. S. (1993). *Visualizing data*. Hobart Press.
- Coursera. (2025). *Overfitting and underfitting in deep learning*. <https://www.coursera.org/learn/deep-learning>
- Czinkota, M. R., & Ronkainen, I. A. (2013). *International marketing* (10th ed.). Cengage Learning.
- Davenport, T. H., & Harris, J. G. (2007). *Competing on analytics: The new science of winning*. Harvard Business Press.
- Domingos, P. (2012). A few useful things to know about machine learning. *Communications of the ACM*, 55(10), 78–87.
- Ecevit, A., Öztürk, M., Dağ, A., & Özcan, A. (2023). Short-term sales forecasting using LSTM and Prophet based models in e-commerce. *Uluslararası Yönetim Bilişim Sistemleri ve Bilgisayar Bilimleri Dergisi*, 7(2), 41–54. <https://dergipark.org.tr/tr/download/article-file/2985051>
- Efron, B., & Tibshirani, R. (1997). Improvements on cross-validation: The .632+ bootstrap method. *Journal of the American Statistical Association*, 92(438), 548–560.
- Ferrell, O. C., & Hartline, M. D. (2011). *Marketing strategy* (5th ed.). Cengage Learning.
- Ensafi, M., Sahin, Y., & Duman, E. (2022). Time-series forecasting of seasonal items sales using Prophet, LSTM, and CNN. *Discover Artificial Intelligence*, 2(1), 2. <https://doi.org/10.1007/s44163-022-00002-7>
- Flores, B. E. (1986). A pragmatic view of accuracy measurement in forecasting. *Omega*, 14(2), 93–98.

- Frost, J. (2020). *Introduction to time series analysis*. Statistics by Jim. <https://statisticsbyjim.com>
- García, F., & Romero, A. (2021). *Programación en Visual Studio Code*. Ediciones Alpha.
- Géron, A. (2019). *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow* (2nd ed.). O'Reilly Media.
- Ghaffary, A. (2021). *Visual Studio Code: Complete guide for beginners*. Leanpub.
- Gitman, L. J., & Zutter, C. J. (2015). *Principles of managerial finance* (14th ed.). Pearson.
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.
- Graves, A. (2013). *Speech recognition with deep recurrent neural networks*. arXiv:1303.5778.
- Grewal, D., & Levy, M. (2017). *Marketing* (5th ed.). McGraw-Hill Education.
- Gülen, S. (2025). *Baseline models in machine learning*. Towards Data Science. <https://towardsdatascience.com>
- Hernán, M. A., Hsu, J., & Healy, B. (2018). A second chance to get causal inference right: A classification of data science tasks. *Chance*, 32(1), 42–49.
- Hernández, R., Fernández, C., & Baptista, P. (2014). *Metodología de la investigación* (6.ª ed.). McGraw-Hill.
- Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735–1780.
- Horngren, C. T., Sundem, G. L., & Elliott, J. A. (2010). *Introducción a la contabilidad financiera* (9.ª ed.). Pearson.
- Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, 9(3), 90–95.
- Hyndman, R. J., & Athanasopoulos, G. (2018). *Forecasting: Principles and practice* (2nd ed.). OTexts.
- Hyndman, R. J., & Koehler, A. B. (2006). Another look at measures of forecast accuracy. *International Journal of Forecasting*, 22(4), 679–688.
- James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An introduction to statistical learning*. Springer.
- Johansson, R., Lugato, M., & Linge, J. (2016). *Numerical Python: Scientific computing and data science applications with NumPy*. Springer.
- Kim, S., & Kim, H. (2016). A new metric of absolute percentage error for intermittent demand forecasts. *International Journal of Forecasting*, 32(3), 669–679.
- Kizgin, H., Aksoy, E., & Demir, S. (2025). Machine learning-based sales forecasting during crises: Evidence from the retail sector. *Journal of Retailing and Consumer Services*, 78, 103783. <https://doi.org/10.1016/j.jretconser.2025.103783>
- Kluyver, T., Ragan-Kelley, B., Pérez, F., Granger, B. E., Bussonnier, M., Frederic, J., ... & Willing, C. (2016). Jupyter Notebooks – A publishing format for reproducible computational workflows. In *Positioning and Power in Academic Publishing: Players, Agents and Agendas* (pp. 87–90). IOS Press.
- Kotler, P., & Keller, K. L. (2016). *Marketing management* (15th ed.). Pearson.
- Kuhn, M., & Johnson, K. (2018). *Applied predictive modeling*. Springer.
- Lutz, M. (2013). *Learning Python* (5th ed.). O'Reilly Media.
- Makridakis, S., Wheelwright, S. C., & Hyndman, R. J. (1998). *Forecasting: Methods and applications* (3rd ed.). Wiley.
- McKinney, W. (2012). *Python for data analysis*. O'Reilly Media.
- Mitchell, T. M. (1997). *Machine learning*. McGraw-Hill.
- Moore, D. S., & McCabe, G. P. (2014). *Introduction to the practice of statistics* (8th ed.). W. H. Freeman.
- Müller, A. C., & Guido, S. (2017). *Introduction to machine learning with Python*. O'Reilly Media.

- Ni, Y., Peng, Y., & Liu, W. (2022). Logistics demand forecast of fresh food e-commerce based on Bi-LSTM model. *Journal of Computer and Communications*, 10(5), 40–55. <https://doi.org/10.4236/jcc.2022.105004>
- Oliphant, T. E. (2006). *A guide to NumPy*. Trelgol Publishing.
- PeerJ Computer Science. (2025). Sales forecasting for retail stores using hybrid neural network models. *PeerJ Computer Science*, 11, e3058. <https://doi.org/10.7717/peerj-cs.3058>
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.
- Provost, F., & Fawcett, T. (2013). *Data science for business*. O'Reilly Media.
- Raschka, S., & Mirjalili, V. (2019). *Python machine learning* (3rd ed.). Packt Publishing.
- Rosebrock, A. (2018). *Deep learning for computer vision with Python*. PyImageSearch.
- Russell, S. J., & Norvig, P. (2021). *Artificial intelligence: A modern approach* (4th ed.). Pearson.
- Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks*, 61, 85–117.
- Schuster, M., & Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11), 2673–2681.
- Stanton, W. J., Etzel, M. J., & Walker, B. J. (2007). *Fundamentals of marketing* (14th ed.). McGraw-Hill.
- Taylor, S. J., & Letham, B. (2018). Forecasting at scale. *The American Statistician*, 72(1), 37–45.
- Thomas, J., Boucher, M., & Smith, L. (2020). *Data science with Jupyter*. Packt Publishing.
- Triola, M. F. (2018). *Estadística* (13.^a ed.). Pearson Educación.
- Tukey, J. W. (1977). *Exploratory data analysis*. Addison-Wesley.
- VanderPlas, J. (2016). *Python data science handbook*. O'Reilly Media.
- VanderWalt, S., Colbert, S. C., & Varoquaux, G. (2011). The NumPy array: A structure for efficient numerical computation. *Computing in Science & Engineering*, 13(2), 22–30.
- Villalobos, A. (2021). *Modelos de series temporales aplicados con Prophet*. *Revista de Estadística Aplicada*, 7(2), 15–24.
- Willmott, C. J., & Matsuura, K. (2005). Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. *Climate Research*, 30(1), 79–82.
- Zhang, Z., Lipton, Z. C., Li, M., & Smola, A. J. (2020). *Dive into deep learning*. <https://d2l.ai>


Anexos

Anexo 1. Matriz de consistencia

Título: Modelo machine learning para predecir las ventas de la empresa RETAIL,2025

| PROBLEMAS | OBJETIVOS | HIPOTESIS | VARIABLES | METODO |
|--|--|--|--|---|
| General | General | General | | |
| ¿Qué modelo de machine learning basado en series temporales permite predecir con mayor precisión las ventas de la tienda virtual de una empresa retail en el año 2025? | Determinar qué modelo de machine learning basado en series temporales (Prophet o LSTM) brinda mayor precisión para predecir las ventas de una tienda virtual en una empresa retail durante el año 2025. | El modelo LSTM brinda mayor precisión que el modelo Prophet en la predicción de ventas de la tienda virtual de la empresa retail durante el año 2025. | Variable1 (Variable de calibración) Modelo de machine Learning Indicadores: Tipo de Algoritmo (Prophet, LSTM) | Tipo de investigación Aplicada Nivel de investigación Predictivo Diseño No experimental, Observacional |
| Específico | Específico | Específico | Variable2 (Variable evaluativa) Y: Predecir Indicadores: RMSE MAE MAPE Variable Interviniente: Datos históricos de ventas Indicadores: Registros anuales de venta | Población La población está compuesta por 152,735 registros de Ventas que pertenecen a los años del 2022 al 2024. Muestra Se tomará toda la población, utilizando un muestreo censal. Técnica Análisis documental (base de datos de ventas) Instrumento Se utilizará el registro de ventas de la base de datos Herramienta computacional Python |
| P1: ¿Qué variables históricas de ventas son relevantes para la predicción de comportamientos futuros en el entorno retail? P2: ¿Qué Tipo de algoritmo (Prophet o LSTM) ofrece mejor predicción? P3: ¿Qué Tipo de algoritmo Prophet y LSTM) tiene mejor rendimiento en la predicción? | O1: Identificar y recopilar variables históricas de ventas relevantes para la predicción. O2: Implementar y entrenar los Tipos de algoritmo (Prophet y LSTM) utilizando datos de ventas reales para la predicción. O3: Evaluar y comparar el rendimiento de predicción de los Tipos de Algoritmo (Prophet y LSTM) mediante métricas estadísticas (RMSE, MAE y MAPE). | H1: Las variables históricas de ventas tienen un efecto significativo en la precisión del modelo de predicción. H2: El Tipo de algoritmo (Prophet) presenta mayor error de predicción en comparación con LSTM. H3: La validación cruzada evidencia diferencias significativas en el rendimiento de predicción de los Tipos de Algoritmo (Prophet y LSTM) | | |

Anexo 2. Instrumento

|  Ficha de Registro | |
|---|-------------------------|
| Periodo | 2022-2024 |
| Fecha (ds) | Venta Soles Sin IGV (y) |
| 1/01/2022 | 3422.004 |
| 2/01/2022 | 6833.405 |
| 3/01/2022 | 7307.252 |
| 4/01/2022 | 8633.481 |
| 5/01/2022 | 49217.776 |
| 6/01/2022 | 10603.224 |
| 7/01/2022 | 13642.188 |
| 8/01/2022 | 9611.4 |
| 9/01/2022 | 7007.899 |
| 10/01/2022 | 15368.502 |
| 11/01/2022 | 7373.899 |
| ... | ... |
| 20/12/2024 | 40265.836 |
| 21/12/2024 | 33019.201 |
| 22/12/2024 | 46264.607 |
| 23/12/2024 | 26028.863 |
| 24/12/2024 | 30271.641 |
| 25/12/2024 | 24282.335 |
| 26/12/2024 | 5886.416 |
| 27/12/2024 | 15204.852 |
| 28/12/2024 | 20732.709 |
| 29/12/2024 | 19344.954 |
| 30/12/2024 | 27730.73 |
| 31/12/2024 | 15328.084 |

Anexo 3. Preparación del dataset

```
# Ver columnas del archivo
print(df.columns)
```

✓ 0.0s Python

```
Index(['Fecha', 'VentaSolesSinIGV'], dtype='object')
```

```
# Renombramos si es necesario para estandarizar (ajusta esto según tus nombres reales)
df.rename(columns={"Fecha": "fecha", "VentaSolesSinIGV": "monto"}, inplace=True)
```

✓ 0.0s Python

```
# Convertir 'fecha' a datetime si no lo está
df["fecha"] = pd.to_datetime(df["fecha"])
```

✓ 0.0s Python

```
# Asegurarnos de que esté ordenado por fecha
df = df.sort_values("fecha")
```

✓ 0.0s Python


```
# Estadísticas generales
print("\nResumen estadístico:")
print(df["monto"].describe())
```

✓ 0.0s Python

```
Resumen estadístico:
count      1096.000000
mean       30613.383241
std        33031.532697
min        -23552.827000
25%        12310.667000
50%        22451.184000
75%        37341.623500
max         285879.859000
Name: monto, dtype: float64
```

```
# Revisión visual de la serie
plt.figure(figsize=(15, 5))
plt.plot(df["fecha"], df["monto"], label="Monto diario")
plt.title("Ventas diarias (sin IGV)")
plt.xlabel("Fecha")
plt.ylabel("Monto")
plt.legend()
plt.tight_layout()
plt.show()
```

✓ 1.3s Python



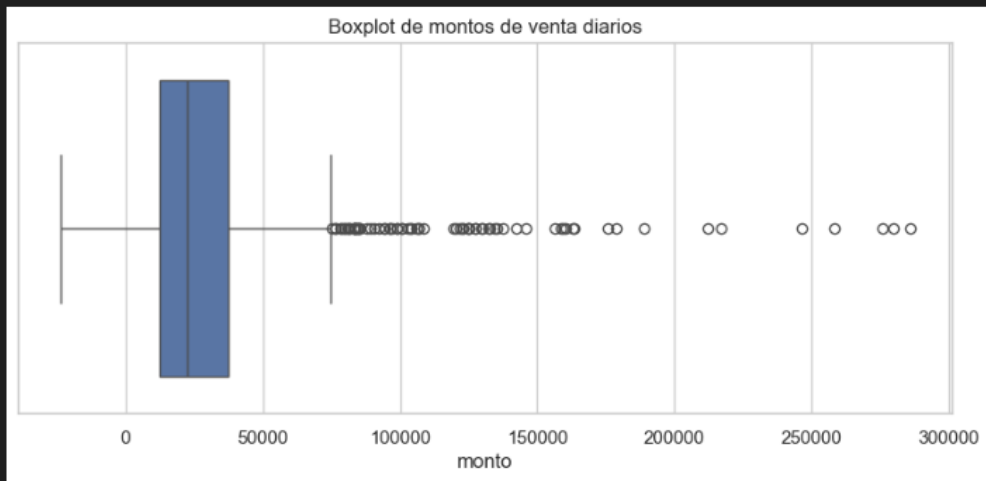
```

# Gráfico de caja (boxplot)
plt.figure(figsize=(10, 4))
sns.boxplot(x=df["monto"])
plt.title("Boxplot de montos de venta diarios")
plt.show()

```

✓ 0.1s

Python



```

# Método IQR para detectar outliers
Q1 = df["monto"].quantile(0.25)
Q3 = df["monto"].quantile(0.75)
IQR = Q3 - Q1
limite_inferior = Q1 - 1.5 * IQR
limite_superior = Q3 + 1.5 * IQR

outliers = df[(df["monto"] < limite_inferior) | (df["monto"] > limite_superior)]
print(f"Cantidad de outliers detectados: {outliers.shape[0]}")

```

✓ 0.0s

Python

Cantidad de outliers detectados: 75

```

# Suavizado con media móvil de 7 días
df["suavizado_7d"] = df["monto"].rolling(window=7).mean()

plt.figure(figsize=(15, 5))
plt.plot(df["fecha"], df["monto"], alpha=0.3, label="Original")
plt.plot(df["fecha"], df["suavizado_7d"], color='red', label="Suavizado (7 días)")
plt.title("Ventas diarias vs. Suavizado")
plt.legend()
plt.show()

```

✓ 0.2s

Python



Anexo 3. Evaluación de la Hipótesis General

```
# 1. Importar librerías
from prophet import Prophet
from sklearn.metrics import mean_squared_error, mean_absolute_error
import numpy as np
import pandas as pd

✓ 11.5s Python

c:\Users\Dumbledore\AppData\Local\Programs\Python\Python311\Lib\site-packages\tqdm\auto.py:21: T
from .autonotebook import tqdm as notebook_tqdm
Importing plotly failed. Interactive plots will not work.

# 2. Preparar los datos
# SeleccionaR solo fecha y monto, y renombraR columnas según Prophet
df_prophet = df[["fecha", "monto"]].rename(columns={"fecha": "ds", "y": "monto"})
df_prophet.rename(columns={"monto": "y"}, inplace=True)

✓ 0.0s Python

# 3. Dividir en entrenamiento y test (20%)
train = df_prophet[df_prophet["ds"] < "2024-01-01"]
test = df_prophet[df_prophet["ds"] >= "2024-01-01"]

✓ 0.0s Python

# 4. Entrenar modelo Prophet
modelo = Prophet(yearly_seasonality=True, daily_seasonality=True)
modelo.fit(train)

✓ 2.2s Python

# 5. Crear fechas futuras
futuras_fechas = modelo.make_future_dataframe(periods=len(test), freq='D')
predicciones = modelo.predict(futuras_fechas)

✓ 0.2s Python

# 6. Extraer solo predicciones sobre test
y_pred = predicciones[predicciones['ds'].isin(test['ds'])]["yhat"].values
y_true = test["y"].values

✓ 0.0s Python

# 7. Calcular métricas
rmse = np.sqrt(mean_squared_error(y_true, y_pred))
mae = mean_absolute_error(y_true, y_pred)
mape = np.mean(np.abs((y_true - y_pred) / y_true)) * 100

print("♦ RMSE:", round(rmse, 2))
print("♦ MAE:", round(mae, 2))
print("♦ MAPE (%):", round(mape, 2))

✓ 0.0s Python

♦ RMSE: 33802.85
♦ MAE: 22133.22
♦ MAPE (%): 182.99
```

```

# 1. Importar librerías
import numpy as np
import pandas as pd
from sklearn.preprocessing import MinMaxScaler
from sklearn.model_selection import TimeSeriesSplit
from sklearn.metrics import mean_squared_error, mean_absolute_error
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense
from tensorflow.keras.optimizers import Adam

```

✓ 35.1s

Python

```

# 2. Crear serie suavizada
df_lstm = df.copy()
df_lstm["monto_suavizado"] = df_lstm["monto"].rolling(window=7).mean()
df_lstm.dropna(inplace=True)

```

✓ 0.0s

Python

```

# 3. Escalar entre 0 y 1
scaler = MinMaxScaler()
df_lstm["scaled"] = scaler.fit_transform(df_lstm[["monto_suavizado"]])

```

✓ 0.0s

Python

```

# 4. Función para crear secuencias de entrada (X) y salida (y)
def crear_secuencias(data, n_ventana):
    X, y = [], []
    for i in range(n_ventana, len(data)):
        X.append(data[i - n_ventana:i])
        y.append(data[i])
    return np.array(X), np.array(y)

```

✓ 0.0s

Python

```

# 5. Preparar datos
ventana = 30
serie = df_lstm["scaled"].values
X, y = crear_secuencias(serie, ventana)
X = X.reshape((X.shape[0], X.shape[1], 1)) # Requerido por LSTM

```

✓ 0.0s

Python

```

# 6. Validación cruzada con TimeSeriesSplit
tscv = TimeSeriesSplit(n_splits=5)
rmse_list, mae_list, mape_list = [], [], []

for train_index, test_index in tscv.split(X):
    X_train, X_test = X[train_index], X[test_index]
    y_train, y_test = y[train_index], y[test_index]

    # 7. Definir modelo LSTM
    model = Sequential()
    model.add(LSTM(50, activation='relu', input_shape=(ventana, 1)))
    model.add(Dense(1))
    model.compile(optimizer=Adam(learning_rate=0.001), loss='mae')

    # 8. Entrenar
    model.fit(X_train, y_train, epochs=30, batch_size=32, verbose=0)

    # 9. Predecir
    y_pred = model.predict(X_test)
    y_pred_inv = scaler.inverse_transform(y_pred)
    y_test_inv = scaler.inverse_transform(y_test.reshape(-1, 1))

    # 10. Calcular métricas
    rmse = np.sqrt(mean_squared_error(y_test_inv, y_pred_inv))
    mae = mean_absolute_error(y_test_inv, y_pred_inv)
    mape = np.mean(np.abs((y_test_inv - y_pred_inv) / y_test_inv)) * 100

    rmse_list.append(rmse)
    mae_list.append(mae)
    mape_list.append(mape)

```

```

# 11. Promedios finales
print("◆ Promedio RMSE:", round(np.mean(rmse_list), 2))
print("◆ Promedio MAE:", round(np.mean(mae_list), 2))
print("◆ Promedio MAPE (%):", round(np.mean(mape_list), 2))

```

✓ 0.0s

Python

- ◆ Promedio RMSE: 8156.84
- ◆ Promedio MAE: 5491.61
- ◆ Promedio MAPE (%): 25.66

Anexo 4. Evaluación de la Hipótesis Específica H1

```
# 1. Importar librerías necesarias
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, mean_absolute_error
import numpy as np
import matplotlib.pyplot as plt
```

✓ 0.0s

Python

```
# 2. Crear variables t-1, t-2, t-3
df_h2 = df[["fecha", "monto"]].copy()
df_h2["t-1"] = df_h2["monto"].shift(1)
df_h2["t-2"] = df_h2["monto"].shift(2)
df_h2["t-3"] = df_h2["monto"].shift(3)
df_h2.dropna(inplace=True)
```

✓ 0.0s

Python

```
# 3. Definir rangos de entrenamiento y prueba
train_h2 = df_h2[(df_h2["fecha"] >= "2022-01-01") & (df_h2["fecha"] < "2022-04-01")]
test_h2 = df_h2[(df_h2["fecha"] >= "2022-04-01") & (df_h2["fecha"] <= "2022-05-31")]
```

✓ 0.0s

Python

```
# 4. Preparar los datos
X_train = train_h2[["t-1", "t-2", "t-3"]]
y_train = train_h2["monto"]

X_test = test_h2[["t-1", "t-2", "t-3"]]
y_test = test_h2["monto"]
```

✓ 0.0s

Python

```
# 5. Modelo 1: con variables históricas (regresión lineal)
modelo_hist = LinearRegression()
modelo_hist.fit(X_train, y_train)
pred_hist = modelo_hist.predict(X_test)
```

```
# 6. Modelo 2: base (media del entrenamiento)
media_entrenamiento = y_train.mean()
pred_base = [media_entrenamiento] * len(y_test)
```

✓ 0.0s

Python

```
# 7. Calcular métricas
rmse_hist = np.sqrt(mean_squared_error(y_test, pred_hist))
mae_hist = mean_absolute_error(y_test, pred_hist)

rmse_base = np.sqrt(mean_squared_error(y_test, pred_base))
mae_base = mean_absolute_error(y_test, pred_base)

print(" ♦ Modelo con variables históricas:")
print("  RMSE:", round(rmse_hist, 2))
print("  MAE:", round(mae_hist, 2))

print("\n ♦ Modelo base (media):")
print("  RMSE:", round(rmse_base, 2))
print("  MAE:", round(mae_base, 2))
```

✓ 0.0s Python

♦ Modelo con variables históricas:
RMSE: 27531.48
MAE: 16725.61

♦ Modelo base (media):
RMSE: 28723.9
MAE: 18331.27

```
# 8. Gráfico comparativo
plt.figure(figsize=(15,6))
plt.plot(test_h2["fecha"], y_test.values, label="Ventas reales", color="black")
plt.plot(test_h2["fecha"], pred_hist, label="Modelo con historial", color="green")
plt.plot(test_h2["fecha"], pred_base, label="Modelo base (media)", color="orange", linestyle=
plt.title("Comparación de modelos para predicción de ventas (Abr-May 2022)")
plt.xlabel("Fecha")
plt.ylabel("Monto de venta (S/. sin IGV)")
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()
```

✓ 0.4s Python

Anexo 5. Evaluación de la Hipótesis Específica H2

```
from prophet import Prophet
from sklearn.metrics import mean_squared_error, mean_absolute_error
import numpy as np
import pandas as pd

# Filtrar datos de enero a septiembre 2022
df_sub = df[["fecha", "monto"]].copy()
df_sub = df_sub[(df_sub["fecha"] >= "2022-01-01") & (df_sub["fecha"] <= "2022-09-30")]

# Preparar para Prophet
df_prophet = df_sub.rename(columns={"fecha": "ds", "monto": "y"})
train_prophet = df_prophet[df_prophet["ds"] < "2022-09-01"]
test_prophet = df_prophet[(df_prophet["ds"] >= "2022-09-01") & (df_prophet["ds"] <= "2022-09-30")]

# Entrenar modelo Prophet
modelo_prophet = Prophet(yearly_seasonality=True, daily_seasonality=True)
modelo_prophet.fit(train_prophet)

# Predicción para septiembre
future = modelo_prophet.make_future_dataframe(periods=30, freq="D")
forecast = modelo_prophet.predict(future)
forecast_sep = forecast[forecast["ds"].between("2022-09-01", "2022-09-30")]

# Métricas Prophet
y_true_prophet = test_prophet["y"].values
y_pred_prophet = forecast_sep["yhat"].values

rmse_prophet = np.sqrt(mean_squared_error(y_true_prophet, y_pred_prophet))
mae_prophet = mean_absolute_error(y_true_prophet, y_pred_prophet)
mape_prophet = np.mean(np.abs((y_true_prophet - y_pred_prophet) / y_true_prophet)) * 100

print("■ Prophet:")
print("RMSE:", round(rmse_prophet, 2))
print("MAE:", round(mae_prophet, 2))
print("MAPE:", round(mape_prophet, 2))
```

✓ 0.4s

Python

22:37:27 - cmdstanpy - INFO - Chain [1] start processing

22:37:27 - cmdstanpy - INFO - Chain [1] done processing

■ Prophet:

RMSE: 41220.57

MAE: 36803.8

MAPE: 927.72

```

from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense
from sklearn.preprocessing import MinMaxScaler

# Crear dataset reducido
df_lstm = df_sub.copy()
df_lstm["suavizado"] = df_lstm["monto"].rolling(window=7).mean()
df_lstm.dropna(inplace=True)

# Escalado
scaler = MinMaxScaler()
df_lstm["scaled"] = scaler.fit_transform(df_lstm[["suavizado"]])

# Crear secuencias (ventana de 30 días)
def crear_secuencias(data, ventana):
    X, y = [], []
    for i in range(ventana, len(data)):
        X.append(data[i - ventana:i])
        y.append(data[i])
    return np.array(X), np.array(y)

ventana = 30
serie = df_lstm["scaled"].values
X, y = crear_secuencias(serie, ventana)
X = X.reshape((X.shape[0], X.shape[1], 1))

# División en train/test manual para septiembre
fechas_lstm = df_lstm["fecha"].values[ventana:]
indices_septiembre = [i for i, f in enumerate(fechas_lstm) if pd.to_datetime(f) >= pd.to_datetime("2022-09-01")]

corte = indices_septiembre[0]
X_train, X_test = X[:corte], X[corte:]
y_train, y_test = y[:corte], y[corte:]

# Modelo LSTM
model = Sequential()
model.add(LSTM(50, activation='relu', input_shape=(ventana, 1)))
model.add(Dense(1))
model.compile(optimizer='adam', loss='mae')
model.fit(X_train, y_train, epochs=30, batch_size=32, verbose=0)

```

```

# Predicción
y_pred = model.predict(X_test)
y_pred_inv = scaler.inverse_transform(y_pred)
y_test_inv = scaler.inverse_transform(y_test.reshape(-1, 1))

# Métricas
rmse_lstm = np.sqrt(mean_squared_error(y_test_inv, y_pred_inv))
mae_lstm = mean_absolute_error(y_test_inv, y_pred_inv)
mape_lstm = np.mean(np.abs((y_test_inv - y_pred_inv) / y_test_inv)) * 100

print("■ LSTM:")
print("RMSE:", round(rmse_lstm, 2))
print("MAE:", round(mae_lstm, 2))
print("MAPE:", round(mape_lstm, 2))

```

✓ 7.6s

Python

```

c:\Users\Dumbledore\AppData\Local\Programs\Python\Python311\Lib\site-packages\keras\src\layers\rnn\rnn.py:199: UserWarning
super().__init__(**kwargs)
1/1 ----- 0s 222ms/step
■ LSTM:
RMSE: 6375.16
MAE: 5625.78
MAPE: 13.68

```

```

from prophet import Prophet

# Subconjunto de datos enero-septiembre 2022
df_sub = df[["fecha", "monto"]].copy()
df_sub = df_sub[(df_sub["fecha"] >= "2022-01-01") & (df_sub["fecha"] <= "2022-09-30")]

# Preparar datos para Prophet
df_prophet = df_sub.rename(columns={"fecha": "ds", "monto": "y"})
train_prophet = df_prophet[df_prophet["ds"] < "2022-09-01"]

# Entrenar modelo Prophet
modelo_prophet = Prophet(yearly_seasonality=True, daily_seasonality=True)
modelo_prophet.fit(train_prophet)

# Crear fechas futuras y predecir septiembre
future = modelo_prophet.make_future_dataframe( periods=30, freq="D")
forecast = modelo_prophet.predict(future)
forecast_sept = forecast[forecast["ds"].between("2022-09-01", "2022-09-30")]

```

```

import matplotlib.pyplot as plt
import pandas as pd

# 1. Filtrar solo las fechas de septiembre 2022
df_sept = df[(df["fecha"] >= "2022-09-01") & (df["fecha"] <= "2022-09-30")].copy()

# 2. Asegurar el orden correcto
df_sept = df_sept.sort_values("fecha").reset_index(drop=True)
forecast_sept = forecast_sept.reset_index(drop=True)

# 3. Crear dataframe para LSTM
df_sept_lstm = pd.DataFrame({
    "fecha": df_sept["fecha"].values,
    "real": df_sept["monto"].values,
    "lstm": y_pred_inv.flatten()
})

# 4. Graficar comparativo
plt.figure(figsize=(15, 6))
plt.plot(df_sept["fecha"], df_sept["monto"], label="Ventas reales", color="black")
plt.plot(forecast_sept["ds"], forecast_sept["yhat"], label="Predicción Prophet", color="royalblue")
plt.plot(df_sept_lstm["fecha"], df_sept_lstm["lstm"], label="Predicción LSTM", color="seagreen")
plt.title("Comparación de predicción de ventas ☐ Septiembre 2022")
plt.xlabel("Fecha")
plt.ylabel("Monto de venta (S/. sin IGV)")
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()

```

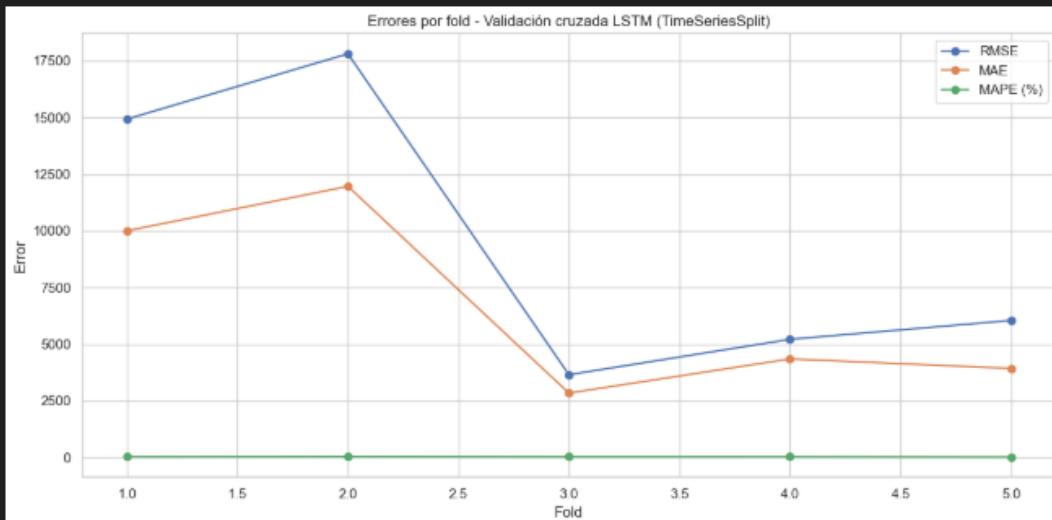
Anexo 6. Evaluación de la Hipótesis Específica H3

```
import matplotlib.pyplot as plt
import pandas as pd

# 1. Crear tabla con errores por fold
tabla_cv = pd.DataFrame({
    "Fold": [1, 2, 3, 4, 5],
    "RMSE": resultados_rmse,
    "MAE": resultados_mae,
    "MAPE": resultados_mape
}).round(2)

# 2. Graficar los errores por fold
plt.figure(figsize=(12, 6))
plt.plot(tabla_cv["Fold"], tabla_cv["RMSE"], marker="o", label="RMSE")
plt.plot(tabla_cv["Fold"], tabla_cv["MAE"], marker="o", label="MAE")
plt.plot(tabla_cv["Fold"], tabla_cv["MAPE"], marker="o", label="MAPE (%)")
plt.title("Errores por fold - Validación cruzada LSTM (TimeSeriesSplit)")
plt.xlabel("Fold")
plt.ylabel("Error")
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()

# 3. Mostrar tabla en consola
print(tabla_cv)
```



| | Fold | RMSE | MAE | MAPE |
|---|------|----------|----------|-------|
| 0 | 1 | 14933.91 | 10001.64 | 30.51 |
| 1 | 2 | 17817.09 | 11965.32 | 39.36 |
| 2 | 3 | 3650.62 | 2840.46 | 30.76 |
| 3 | 4 | 5212.82 | 4348.18 | 30.11 |
| 4 | 5 | 6045.08 | 3929.57 | 18.54 |



UNSCH

FACULTAD DE
INGENIERÍA
DE MINAS, GEOLOGÍA Y CIVIL

ACTA DE SUSTENTACIÓN DE TESIS N° 048-2025-FIMGC

PARA OPTAR EL TÍTULO PROFESIONAL DE INGENIERA DE SISTEMAS

En la Universidad Nacional de San Cristóbal de Huamanga de la ciudad de Ayacucho, en cumplimiento a la **RESOLUCIÓN DECANAL N° 363-2025-FIMGC-D**, a los **veintiocho días del mes de octubre de 2025**, siendo las **10:00 a.m.**, reunidos en el **Auditorio de la Escuela Profesional de Ingeniería de Minas**, bajo la presidencia del **Mg. Edith Felicitas GUEVARA MOROTE** y los miembros: **Mg. Celia Edith MARTÍNEZ CÓRDOVA** y **Mg. Elinar CARRILLO RIVEROS**, actuando como secretario docente el **Ing. Saúl Walter RETAMOZO FERNÁNDEZ**, para proceder a la sustentación de tesis para optar el **Título Profesional de Ingeniera de Sistemas**, del bachiller:

KEYLA ELYNETH SERNA BERROCAL

Quien presentó la tesis denominada:

Modelo machine learning para predecir las ventas de una empresa RETAIL, 2025

Los señores miembros del jurado, luego de expuesta la tesis y absueltas las preguntas, deliberaron y declararon:

Aprobado con Diecisiete (17)

Siendo las **12:00 p.m.** del día **28 de octubre de 2025**, culmina el acto de sustentación de tesis, y en conformidad con lo actuado, los miembros del jurado firman al pie del presente.

Mg. Edith Felicitas GUEVARA MOROTE
Presidente

Mg. Elinar CARRILLO RIVEROS
Miembro - Asesor

Mg. Celia Edith MARTÍNEZ CÓRDOVA
Miembro

Ing. Saúl Walter RETAMOZO FERNANDEZ
Secretario docente de la FIMGC



CONSTANCIA DE ORIGINALIDAD DE TRABAJO DE INVESTIGACIÓN

CONSTANCIA N° 028-2025-KPS-FIMGC/UNSCH

El que suscribe; responsable verificador de originalidad de trabajos de tesis de pregrado con el software Turnitin, en segunda instancia para las **Escuelas Profesionales** de la **Facultad de Ingeniería de Minas, Geología y Civil**; en cumplimiento a la **Resolución de Consejo Universitario N° 039-2021-UNSCH-CU**, Reglamento de Originalidad de Trabajos de Investigación de la Universidad Nacional San Cristóbal de Huamanga y **Resolución Decanal N° 697-2024-FIMGC-D**, deja constancia de originalidad de trabajo de investigación, que el/la Sr./Srta.

Nombres y Apellidos : Keyla Elyneth Serna Berrocal
Escuela Profesional : INGENIERÍA DE SISTEMAS
Título de la Tesis : Modelo machine learning para predecir las ventas de una empresa RETAIL, 2025
Evaluación de la Originalidad : **0%** Índice de Similitud
Identificador de la entrega : 2822359042

Por tanto, según los Artículos 12, 13 y 17 del Reglamento de Originalidad de Trabajos de Investigación, es **PROCEDENTE** otorgar la **Constancia de Originalidad** para los fines que crea conveniente.

En señal de conformidad y verificación se firma la presente constancia

Ayacucho, 28 de noviembre de 2025



Firmado digitalmente por:
PERALTA SOTOMAYOR Karel
FAU 20143660754 soft
Motivo: Soy el autor del documento
Fecha: 01/12/2025 09:06:29-0500

Modelo machine learning para predecir las ventas de una empresa RETAIL, 2025

por Keyla Elyneth SERNA BERROCAL

Fecha de entrega: 20-nov-2025 04:18p. m. (UTC-0500)

Identificador de la entrega: 2822359042

Nombre del archivo: MEMORANDO_N_701-2025-CERTIFICADO_DE_ORIGINALIDAD-KEYLA_ELNETH_SERNA_BERROCAL.pdf (3.18M)

Total de palabras: 20024

Total de caracteres: 119376

Modelo machine learning para predecir las ventas de una empresa RETAIL, 2025

INFORME DE ORIGINALIDAD

0%

INDICE DE SIMILITUD

0%

FUENTES DE INTERNET

0%

PUBLICACIONES

0%

TRABAJOS DEL ESTUDIANTE

FUENTES PRIMARIAS

Excluir citas

Activo

Excluir coincidencias < 30%

Excluir bibliografía

Activo